Faculty of Engineering and Science

Department of Electrical and Computer Engineering

Reconfigurable Wireless Sensor Network Design for Environmental Monitoring
in IoT Environment

Cho Zin Myint

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

September 2018

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

C Z. Myint.

Signature : ----------------

Date : 29th September 2018

Acknowledgements

I would like to express my sincere gratitude to my supervisors A/Prof. Dr. Lenin Gopal, Dr. Ling Huo Chong and A/Prof. Dr. M. V. Prasanna for their advice, guidance, support and patience on this project. Furthermore, I would like to thank Dr. Yan Lin Aung for sharing his knowledge on HDL coding and software programming.

I would also like to express the deepest appreciation to the chair A/Prof. Dr. Perumal Kumar, for his knowledge, constructive criticism, and time spent reviewing the thesis. In addition, a lot of thanks go to Prof. Dr. Clem Kuek for allowing me to do this project and their constant encouragement and support.

I would like to thank my parents who have always been very supportive in every aspect of my life, including this project.

I would also acknowledge my postgraduate friends who are always ready to help when needed.

Many thanks to the staff at Faculty of Engineering and Science and the laboratory.

Abstract

Since the water quality monitoring (WQM) is critical, a real-time continuous in-situ monitoring system for water quality detection has been increasingly popular among the researchers in recent years. The WQM system based on wireless sensor network (WSN) using different wireless technologies such as General Packet Radio Service (GPRS), Bluetooth, Wireless Local Area Network (WLAN) to transmit vital data and remotely monitor the quality of water has been developed due to the advantages of wireless technology, for example low power consumption, portability and low cost. For data collection in WSN systems in the Internet of Things (IoT) environment, a sensor interface device is critical for the applications of health care, environmental monitoring, surveillance, patrolling, security service, search and rescue missions, military operations, agricultural routines, and industrial control systems.

Despite many studies and researches and the tremendous capabilities of the technology, environmental monitoring application of WSN is still limited as signal types and the sampling rate of sensors are restricted by the type of devices. One of the current constraints of IoT is energy efficiency since the enormous numbers of sensors are deployed in IoT. In the proposed system, the wireless sensor node is designed for monitoring the data of five water parameters such as water level, pH value of water, water temperature, Carbon dioxide (CO_2) on the water surface and turbidity of water. The proposed WQM system comprises sensors, Field Programmable Gate Array (FPGA), Zigbee wireless communication unit and personal computer (PC). The FPGA is used as the central component of the proposed WQM system because of its unique reconfigurable real-time performance and synchronicity, and a very high speed integrated circuit hardware description language (VHDL) and C++ are programmed in it using Qsys tool and Quartus II software. Thus, the proposed design of the WQM application collected the real-time data in parallel with the fast rate from various sensor nodes. The standard of IEEE 1451.2 intelligent sensor interface specification is adopted for the proposed reconfigurable WSN system. In this research, the operation of the proposed reconfigurable WQM system based on WSN is examined through experiments and computer simulations.

No significant difference was found between the measurement of the designed device and laboratory measurement. Since the proposed system provides high execution speed and continual IP design and the system executes parallel processing, it offers to minimize about 33.6% of the operating cost and power consumption. The experimental results from the reconfigurable WSN design for WQM in IoT environment system report provide reliable outcomes which are low power consuming, strong communication ability, and display of real-time measurement accuracy. The model of WQM technology with the benefits of power efficiency, eco-friendly and user-friendly based on WSN in IoT technology is designed in this research to resolve the existing or emerging problems of WQM.

Author’s Note

Myint, C, Z., Gopal, L., Aung, Y, L., “WSN-based Water Quality Monitoring System in IoT Environment,” in *14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON 2017)*, 27–30 June 2017, Phuket, Thailand.

Myint, C, Z., Gopal, L., Aung, Y, L., “Reconfigurable smart water quality monitoring system in IoT environment,” in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, Wuhan, 2017, pp. 435–440.

Contents

List of Figures

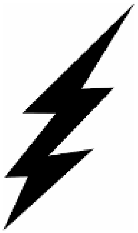
List of Tables

List of Abbreviation

AC	alternating current
ADC	analog-digital converter
CDMA	code division multiple access
CPLD	complex programmable logic device
DC	direct current
EEPROM	electrically erasable programmable read-only memory
FPGA	field programmable gate array
GSM	global system for mobile communications
GPRS	general packet radio service
HPS	hard processor system
IDE	integrated development environment
IoT	internet of things
IR	infrared
LCD	liquid crystal display
LDO	low drop out
LED	light emitting diode
MCU	microcontroller unit
NTU	nephelometric turbidity units
PC	personal computer
PCB	printed circuit board

pH	potential of hydrogen
PLL	phase-locked loop
ppm	parts per million
PWM	pulse width modulation
RAM	random access memory
RFID	radio frequency identification
RS232	recommended standard-232
SDRAM	synchronous dynamic random-access memory
SRAM	static random-access memory
SoC	system on chip
SOPC	system on programmable chip
UART	universal asynchronous receiver transmitter
USART	universal synchronous asynchronous receiver transmitter
USB	universal serial bus
VGA	video graphics array
VHDL	very high speed integrated circuit hardware description language
WSN	wireless sensor network

List of Symbol

○	sensor node
	wireless antenna
	wireless signal

Chapter 1

Introduction

In the internet of things (IoT) environment, wireless sensor network (WSN) has been increasingly important for wireless networks in a variety of applications especially in environmental monitoring. When sensor nodes and processors are reliable and cost-effective, wireless applications in environmental monitoring enable the emergence of power efficiency, eco-friendliness, user-friendliness, and wide-area coverage monitoring devices. The main goal of this thesis is to design and improve a field programmable gate array (FPGA) based WSN in an IoT environment. In this chapter, the necessary background of the WSN systems in IoT is presented.

1.1 Wireless Sensor Network

WSN is a network system of low-power wireless sensor nodes with a Central Processing Unit (CPU) and memory, and it provides high-resolution sensing of the environment [1]. Generally, a WSN is represented as a network of devices comprehended as nodes that collaboratively sense, monitor, and process, communicate, and control the correlation between humans or computers and the adjoining environment via wireless links [2]. A WSN comprises of efficiently communicated sensor nodes, processing the signal, embedded computing, and networking [3]. This system allows communication between monitoring devices, the monitoring individuals and the adjoining environment [4]. The sensors in WSN are logically linked to each other by self-organizing means.

The components of a WSN allow wireless connection inside the network. The sensors in any segment of the wireless network are connected to the computer or the application platform which is at one end of the network to exchange information. Sensors in WSN can be implemented as simple point elements or multipoint detection arrays. The conventional architecture of a WSN consists of small sensors including a memory, a microcontroller, a power supply, and a radio frequency (RF) transceiver and is capable of measuring or detecting the physical characteristics of the surrounding environment.

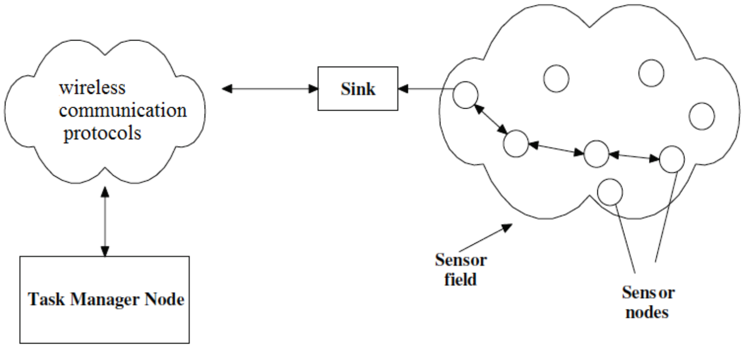


Fig. 1.1: Basic Concept of WSN

Generally, the sensor node consists of one or more sensors and is capable of on-node signal processing in managing information of the physical environment. Each sensor node is assembled with a sensor, a microcontroller, RF module, and a power supply, either a direct current (DC) power supply or a battery. The sensor provokes electrical signals based on the detected data of environmental or physical changes. The microcontroller manipulates and saves the detected data output. The transmitter of the RF module transmits the output data to the receiver of the monitoring computer, in which the data is received and monitored by the computer [5]. Fig.1.1 presents the concepts of WSNs where the assembled data from a sensor node is transmitted to the wireless network via a sink node.

An apparent data route is established between the application platforms and the physical world using gateways and sensors [6]. The WSN system enables operators to observe and operate linked tools from the control station across various wireless modules such as Bluetooth, Radio Frequency Identification (RFID), General Packet Radio Service (GPRS), Zigbee, WiFi, and 3G and 4G mobile technologies [7]. The monitoring team can detect the data across a wireless network which is created based on the above said wireless communication protocols. The WSN structure is shown in Fig.1.2. The WSN technologies have currently achieved universally recognized (IEEE 802.15.4) standard. In WSNs, the individual sensor nodes have restricted memory, processing power, and energy usage since the sensor nodes are generally miniature hardware platforms and are powered by a battery.

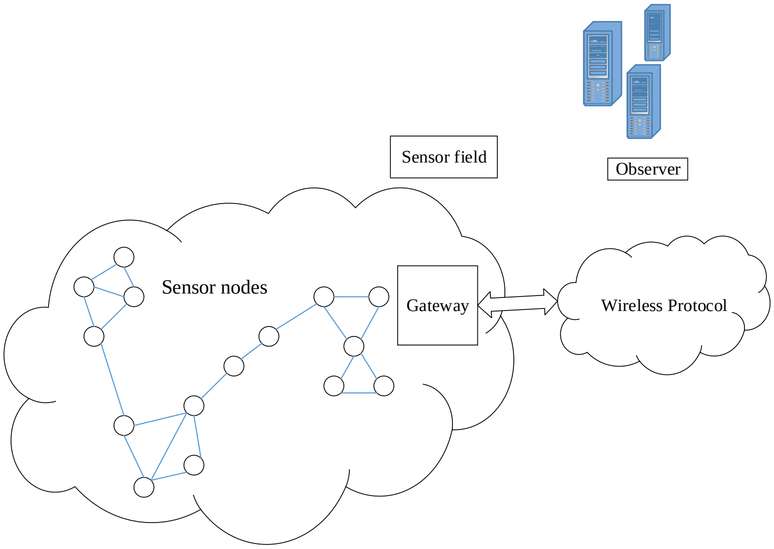


Fig. 1.2: WSN Structure

The advantages of using the technologies of WSN are rapid network installation, data acquisition, remote monitoring, low power consumption, high monitoring precision, wide coverage area, and low duty cycle. Thus, the energy efficient wireless communication systems play an important role as the possible and potential applications WSNs to the physical world is realistically unconstrained, ranging from monitoring surrounding and weather changes, physical security, localization, health care, logistics, positioning and tracking, and so on [8].

1.2 Applications of WSN

Although the WSN was originally used in military applications and large industrial functions, applications of WSN nowadays are used for its diverse benefits, where WSN is applicable from the small to immense industrial structures. The wireless technologies and WSN have been rapidly and progressively improving in facilitating individuals and their day-to-day chores in business [9]. The applications of WSN have been widened for the environmental monitoring systems, data acquisition, and automation of manufacturing processes and building control recently.

Nowadays, the technology of WSN is more advantageous for its lower cost for deployment, cost-effectiveness for both implementation and maintenance, while being able to maximize the performance times. The WSN is generally used for its diverse benefits such as environmental monitoring, surveying the development of city infrastructure, surveillance, telemedicine or remote health care, smart building and research in agriculture [5].

1.2.1 WSN in Surveying the Development of City Infrastructure

In recent years, monitoring roads, bridges and city buildings is one of the important missions because many transport infrastructures such as tunnels, bridges, and viaducts have collapsed because of natural catastrophes. In 2007, Saint Anthony Falls (I-35W) Bridge in Minneapolis in the United States collapsed and killed 13 people and injured 145 [10]. That bridge was re-built in 2008 using a sensor network system to monitor data on the subject of corrosion and structural behavior. A total of 350 state-of-the-art sensors were used to transform the conventional bridge into a smart bridge as well as fulfilling the long-term serviceability role [11].

WSN has played a significant role in monitoring the state of city infrastructure. In Greece, a sensor network was installed in the 2.9 km Charilaos Trikoupis Bridge which has six lanes with 100 sensors, such as strain gauges, accelerometers, weigh-in-motion devices, anemometers and temperature sensors [12]. The sensors sensed unusual vibrations in the cables gripping the bridge in 2004. Therefore, the engineers were able to install supplementary weight to moderate the cables on time. Furthermore, wireless monitoring systems in tunnels are implemented extensively around the world.

1.2.2 WSN in Environmental Monitoring

Since the WSN technology has been increasingly developed, real-time environmental conditions are distantly detected by using data collection on a real-time basis, transmission, and processing. The system consists of numerous sensor nodes and a control station. Each node comprises of a cluster of sensors and the data is detected by the applicable sensor and later transmitted to the main station via WSN protocol. The base station is generally a personal computer (PC) with Graphic User Interface (GUI) for clients to analyze the data of water quality [13].

The major purpose of environmental monitoring is to manage and detect various environmental conditions and weather changes or climate changes. Many works have been studied on architecture and development of environmental monitoring system based on sensor network development [14–17]. The wind speed and direction information, barometric pressure, air temperature, solar radiation, relative humidity, and rainfall are detected by the sensors and transmitted to the weather station. These measurements are used to estimate the weather and hostile natural circumstances. Therefore, the weather station provides weather forecast information to the residents. Long operation time is a critical need for environmental monitoring applications and a massive load of data is needed to be logged in the field due to the long transmission time over the network. In addition, the bandwidth and cost are also critical for the isolated network from other parts.

A WSN based environmental monitoring system applying diverse wireless communication protocols has drawn significant attention in recent years for which this increases the progress of small size multi-functional sensors which can interact in short range with low cost and power [18]. The applications of environmental monitoring include water quality monitoring (WQM) [19–21], forest fire surveillance [22], microclimates of glaciers [23–25] or rainforests [26, 27], greenhouse monitoring and control system [28–31], volcano monitoring [32-34], animal tracking [35, 36], earthquake monitoring [37, 38], and air quality monitoring [15, 39–41].

In recent years, the studies of WSN based environmental sensing and monitoring systems have attracted many researchers since the real-time data from distributed sensors can be collected and the systems can operate autonomously [42–45]. The WSN based environmental monitoring systems do not necessitate the involvement of an operator [46, 47]. Therefore the sensors can be positioned in the remote places where human intervention is lacking.

1.2.3 WSN in Healthcare Service

Since last decade, WSN in health care has been developing as a key area of research for healthcare practices such as pre-and-post hospital patient monitoring, medical treatment [48], and early disease warning systems [49]. When WSNs are designed for healthcare applications, they are referred to as wireless medical sensor networks (WMSNs). The WSN applications in health care are integrated into many types of wireless communication mode, and they are wearable, portable and implantable. The physiological data of patients are detected by the relevant sensors and sent to the remote healthcare professionals wirelessly without human intervention. Therefore, healthcare providers can monitor the patient’s vital physiological signs such as heart rate, body temperature, oxygen saturation in blood and blood pressure.

The WMSNs have improved significantly in the 21st century [50] due to its benefits of reducing the time of interaction needed between physicians and costs. The WSN applications in health care are assisting healthcare providers in enhancing efficiency and cost-effectiveness in long-term care accessibilities.

1.2.4 WSN in Agriculture

In the last decade, WSN applications in agriculture for monitoring, evaluating and managing agricultural practices have started emerging due to low cost, low data rate, and low power consumption. The WSN technology in agriculture involves a large scope of agricultural issues from regular human supervision through cultivation to farm harvest processing and pre-and-post production details of agricultural industry [51–53]. Sensor networks to improve agriculture quality, productivity and resource optimization have therefore been developed. The WSN in agriculture application allows data-logging and data-analysis to monitor and collect data such as humidity, air pressure, and soil moisture parameters in the agricultural domain.

One of the applications of WSN in agriculture is Precision Agriculture (PA) [54] which provides one of the most suitable scenarios for the deployment of WSNs. The advantages of PA are offering feedback in real-time basis on several diverse harvest and area variables. In addition, the PA allows accuracy in the harvest area size it detects in the supply proportions of fertilizer and water etc. The current WSN applications in agriculture are applied in vast harvest region observing, harvest yield enhancement, forest fire prevention, forest/vegetation monitoring, tracking animals and biomass studies [55]. The real-time data in the field is collected, extracted features from raw data, and transmitted to the base station.

1.2.5 WSN in Border Monitoring and Surveillance

The WSN technology is suitable for the applications in surveillance systems due to the high-bandwidth range or analytical wireless technology for even data traffic and outlying monitoring from a long distance [56]. The sensors of the surveillance system detect unusual events and report to the base station securely with the location information. Many recent studies have presented on border surveillance applications based on WSNs in which sensors are able to detect every moving object over a barrier [57–61]. The sensor nodes sense the relevant data in its surroundings, as the data is then computed and transmitted to the base station. The security video camera systems are required to detect the source of alarm sensed by the sensors.

1.2.6 WSN in Smart Building

The WSN in smart building applications is employed in survey lighting, heating, and ventilation [62]. The sensor nodes are linked through wireless networks and are controlled remotely. The sensors enable the evaluation of alarm circumstances and observe the action in the specific sectors of the building where it is positioned. The WSN in smart building applications allow technologies that improve energy effectiveness, user-friendliness and security of the buildings.

1.3 Internet of Things (IoT)

Although the term “Internet of Things” is relatively new, the hypotheses of connecting devices, computers and networks to observe and control devices have already been presented since the 1970s. In the 1990s, the development and progress in wireless technology-empowered “machine-to-machine” (M2M) innovations for the monitoring and controlling of industrial equipment. Nowadays, M2M is a vital implementation in technology that represents the connections and communications. IoT is a physical network which links all objects in order to interchange data and information among the data detecting devices such as sensors and computers coordinated with applicable protocols. The IoT has been evolving parallel to the development of WSNs. In other word, many objects or things are connected within a network in one structure or another.

The term IoT was created by Kevin Ashton in 1999 [63] and refers to distinctively distinguishable things and their communications with each other in an “internet-like” structure. Although IoT does not have any universal definition, the fundamental concept of IoT ubiquitously offers various subjects, such as sensors, radio frequency identification (RFID) tags and actuators to connect, interact and collaborate with each other to comprehend the functions of interaction, computation, and tasks [64]. The IoT is a physical network which consists of numerous connected devices such as sensors, microcontrollers, actuators, and computers. These devices are equipped with identifying, sensing, networking and processing capabilities to communicate with one another and with other devices and services over wireless communication [65].

The objects within the network are interconnected and can be controlled remotely in an IoT where the data can be exchanged and processed by all things per predefined designs within an IoT [66]. The technology of embedded electronics allows real/physical thing such as RFID, sensor, actuator, and computer to detect, compute, interact, and integrate flawlessly with the surrounding environment [67]. These things in an IoT system are re-useable, hence, the components of software and hardware can be reprocessed and upgraded effectively in the IoT environment. To collect the multiple sensors’ data, many wireless sensor nodes are essential to fulfill the conditions of environmental data collection in IoT. Therefore, the most accurate data from the sensor nodes can be collected [68].

The objectives of intelligent verifying, detecting, positioning, controlling things and tracking are accomplished by the applications of IoT [69]. IoT has become distinguished due to its substantial progress in technologies of implementing appliances such as embedded sensor nodes, Near Field Communication (NFC) devices and RFID tags and readers [70]. As the IoT standard is dynamically linked with the effective incorporation of RFID systems and WSNs, there have been several types of research on WSN in the IoT environment [71]. Zorzi et al. [72] stated that the main goal of IoT study is to incorporate WSN into a worldwide interrelated infrastructure. As the systems of RFID are commonly unassertive, the nodes counter to queries by a tag reader or an RFID reader. Therefore, the RFID system does not accommodate nodes to interact with each other separately to initiate communication.

The WSN nodes are required to enable the IoT to resolve this problem while preserving the self-dependency of energy [71]. The applications of IoT based on WSN allow the systems of information and communication to be imperceptibly implanted in the environment as the wireless network facilitates individuals to communicate distantly with the physical world [73]. The IoT is merely comprehensive utilisations of existing technologies while new communication modes are created. The IoT allows a diversity of gateway technologies, wired and wireless communication technologies, networking technologies and switching technologies. The summary of some common wireless protocols used in IoT applications is described in Table 1 [74].

1.3.1 Applications of IoT

Since the IoT allows the potential resolutions that significantly enhance health, energy efficiency, education, security and several additional phases of daily chores, IoT touches every facet of daily human lives nowadays. The IoT can be used in all industrial activities between companies, organizations, and individuals.

Table 1.1: Summary of some Common Wireless Communication Protocols in IoT [74]

Protocol	Coverage Range	Data Rates	Power Consumption
ZigBee	Short (10–20 m)	Low (20 Kbps–250 Kbps)	Low
Z-Wave	Short (30 m)	Low (40 Kbps–100 Kbps)	Ultra - low
INSTEON	Short (50 m)	Very Low (38.4 Kbps)	Low
Wavenis	Long (1 km)	Low (4.8 Kbps–100 Kbps)	Low
Wi-Fi	Medium (30–100 m)	High (typical 100–300 Mbps, up to 7 Gbps)	High
6LowPAN	N/A	N/A	Low
LoRaWAN	Very Long (15 km)	Very Low (0.3 Kbps–50 Kbps)	Low
NB-IoT	Very Long (10–15 km)	Medium (2 Mbps)	Ultra - low

1.3.1.1 IoT in Infrastructure Areas

Currently, the IoT has increasing usage in various infrastructure applications such as environmental monitoring, smart cities, smart homes, and building. The networks of the smart grid, sensors, monitors, cameras, and speaker accumulate data and the functional programs practice it to apply the diverse maintenances of the urban by using IoT technology [75]. The IoT can also be used to enhance a city's sustainable maintenance by managing the usage of energy and to upgrade the standard of living of the residents. The applications of IoT can also be used in air quality monitoring and control, waste management, traffic congestion, noise monitoring, automation, smart lighting, structure and composition of public buildings and smart parking [76].

1.3.1.2 IoT in Healthcare Service

One of the important applications of IoT is health care since IoT allows real-time monitoring of medical parameters and important physiological data such as heart rate, body temperature, blood glucose concentration, blood pressure, and blood oxygen concentration by using medical advanced sensor technologies and wireless communication [77]. The wearable sensors and devices are interconnected and can organize and perform the body sensor networks (BSNs) which allows healthcare providers to access constant remote monitoring of the medical parameters of home care patients or patients at a remote location. The biosensors in IoT collect the patients' physiological data in which these data are sent to distant healthcare providers. Recently, wearable biosensors have been developed to access physiological data of the patient, monitor daily activities of senior citizens at home, while providing health care for the elderly without interfering their daily activities [78].

1.3.1.3 IoT in Environmental Monitoring

Recently, environmental technology becomes a crucial field of sustainable growth globally; hence, environmental monitoring work has been increasingly popular among researchers. The studies on environmental monitoring show that WSN applications in IoT reduce cost while increasing its reliability, and maintaining longer operation time

[79]. The environmental monitoring system consists of a sensor acquisition interface device to detect a variety of WSN sensor data in IoT environments [80]. The IoT comprises the technology of integrating sensors and devices, the power of RFID tags, mobile technology and different intelligent technologies without human intervention [81]. Therefore, the implementation of low power consumption and low-cost sensors in IoT is a significant development of WSNs [82].

Currently, there are still certain constraints to IoT as enormous numbers of sensors that are deployed in IoT require a high cost of servicing including battery replacement. In addition, the sensors may profoundly differ from one another, thus, the requirements for determined scalability on any solution offered for the IoT. Since power efficiency is the major limitation in the architecture technique of IoT, low power sensor designs are demanded [83]. In the WSN system, when the data sensed by the appropriate sensors are computed, the resulted data is transmitted to the gateway or the base station server. According to Moore’s Law, analog circuits in wireless communication consume more power compared to digital circuits [84]. Therefore, transmitting circuits with low-power should also be designed for low-power sensor node in wireless communication. Any further extensive research is essential to explore the possible means of energy efficiency and scalability in the IoT.

Hence, WSN is a principal technology for IoT in which intelligent sensors are used to sense and monitor the data [85, 86]. To collect multiple sensors’ data, a lot of wireless sensor nodes are essential to fulfill the specifications of IoT environmental data collection, consequently, the most accurate data from the sensor nodes can be collected. Currently, there are many data collection interface appliances on the market. However, the existing interface devices are not discretely flexible to the altering environment of IoT due to the specialization of operation of those devices and restriction in physical properties of sensors used [68]. At present, either a microcontroller unit (MCU) or a Field Programmable Gate Array (FPGA) is commonly used as the primary controller in common data collection interface devices due to its low power consumption, low cost and ease of implementation.

However, the MCU cannot achieve parallel data collection in real-time due to the interruption in processing. Conversely, the FPGA has specific synchronicity, real-time performance and hardware logic control [87, 88]. Therefore, the FPGA enables a parallel system of multi-sensor data collection and improves system performance significantly in real-time basis [89]. The major challenges are the complexity of the WSN system and the diverse types of sensors in the IoT environment [90, 91]. Since the sensor data collection device is the crucial component of the WSN applications [86], the Institute of Electrical and Electronics Engineers (IEEE) Association has introduced the IEEE 1451 Smart Transducer Interface Module (STIM) standard to resolve the compatibility problem of the intelligent sensors [92]. The sensors observe network automatically by the STIM interface standard IEEE 1451 to improve the industrial WSN applications [93].

However, the sensors are high-priced and therefore, not favoured in the industrial WSN applications in the IoT environment. Even though there are some smart sensors available in the market, the flexibility with this standard remains restricted. Therefore, the sensor interface hardware based on the IEEE 1451 has been lately recommended to solve these problems with the capability of interfacing with different sensor topologies [94]. The important role in IoT is a low-power wireless network which is composed of sensor nodes and the IEEE 802.15.4 standard for energy-efficient wireless protocols utilized [95]. The architecture of IoT comprises of three layers: 1) the perception layer, 2) the network layer, and 3) the application layer [96].

Since the perception layer consists of sensors, RFID readers, M2M terminals, cameras and the data collection interface device which is important for the incorporation and collaboration of different environments and acquisition of sensor data [97], the design of the data collection interface is principally implemented to the perception layer of IoT [98] in this research. The network layer connects all things and the data can be shared by things with the connected things via the network layer. The data is sent to the control station or the decision-making unit. The application layer performs information exchanging and storage, management of data, and communication. The architecture of the IoT is depicted in Fig. 1.3. Although the architecture process of the reconfigurable intelligent sensor interface system to monitor the water quality was studied, further researches are required to evaluate the performance of the application.

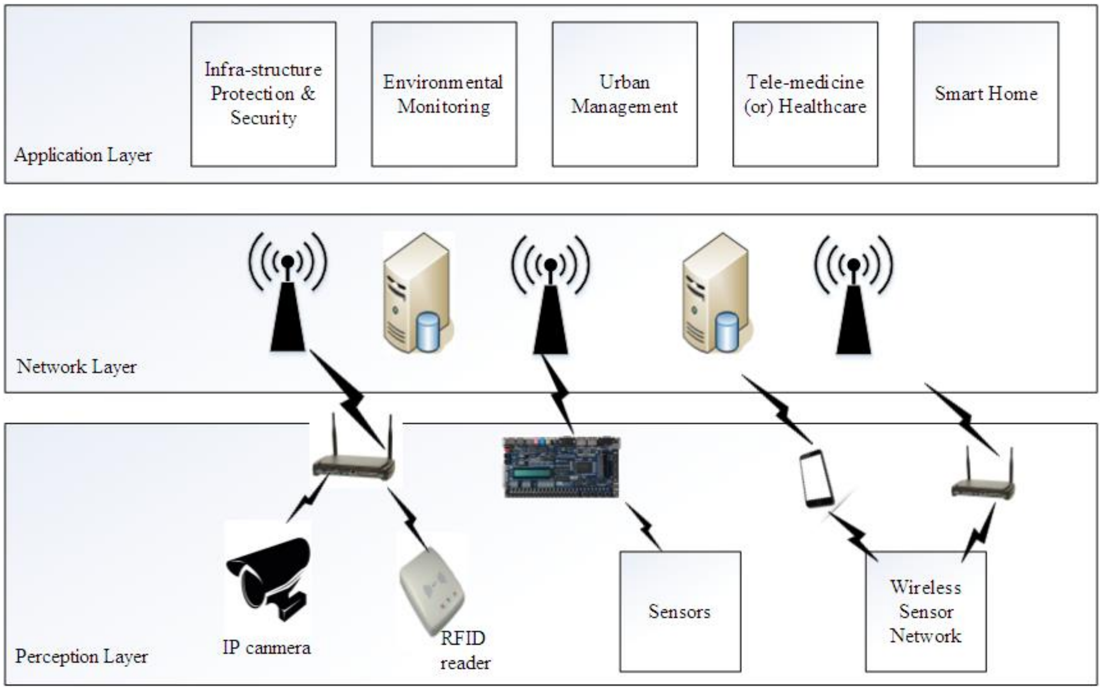


Fig. 1.3: Architecture of IoT

1.4 Reconfigurable System

In a reconfigurable system, programmable hardware is the core component and that can be temporarily customized for a specific program. A reconfigurable device comprises an array of computational elements such as logic blocks and multiple programmable configuration bits define the performance of the device. The logic blocks are connected by means of a set of programmable routing resources; as a result, customized digital circuits are mapped to the reconfigurable hardware by computing the logic functions of the circuit within the logic blocks. Subsequently, the blocks are connected together by means of configurable routing to achieve the required circuit [99]. The most common type of reconfigurable device is FPGA and its fine-grain reconfigurable architecture is based on single bit operation. The mapping functions of FPGA devices are challenging and FPGA devices involve a high volume of configuration data [100]. The static random-access memory (SRAM)-programmable bits of FPGA is connected to the configuration points in the FPGA, and programmable the SRAM bits configure the FPGA.

The main advantage of a reconfigurable processor is effective computations are performed and program-specific operations are achieved in the programmable hardware and the reconfigurable computation increases computational performance. As a result, energy consumption is achieved [99,100]. The reconfigurable processors outperform the processors on highly repetitive computing tasks with limited functional diversity [101]. Since the processing modules of wireless applications must establish simultaneously computational performance, ultralow-power consumption and a high degree of flexibility and adaptability, the ability to be reconfigured is required in the presence of multiple and evolving standards in dynamic conditions [100].

1.5 Motivation

Water is a necessity for the survival of all living things. However, pollution of water has become a primary cause of many diseases and deaths globally. The high mortality rate of 14,000 deaths daily around the world is due to the effects of water pollution [102]. Therefore, WQM serves high importance in environmental monitoring. The traditional method of water quality testing procedures involves collecting samples of the water manually and transporting the samples to a laboratory for analysis to examine the quality of water [103–105]. Therefore, the traditional method of water quality detection is laborious, costly, time and manpower consuming as it is dependent on human intervention where the water samples need to be transported from the water resource to the laboratory for analysis besides requiring the utilization of specialized tools and involvement of professionals.

As the laboratory-based analysis is a multifarious assignment that requires a number of operational procedures, the expenditure of the laboratory-based hardware system, the expenditure for construction of laboratories, successive building and system maintenance and is considered limitations [106]. Since the water conditions such as level of pH and carbon dioxide (CO_2) change over time, when a collection of water samples and examination of data could not be achieved within a short period of time, it is inconvenient to verify the real-time water conditions [104]. The traditional method of WQM systems has several sources of errors caused by both humans and the type of containers used for water sample collection.

Although the idea of water quality detection based on WSN was introduced in early 2000, it is increasingly attracting many researchers in recent years for the improvement of devices and communication technology since it is an autonomous solution to improve the environmental detection technology. In the last decade, the WSN based WQM has been increasingly popular and a number of researchers have studied, proposed and designed WSNs for WQM [107–110]. Many studies proved that WSN based WQM systems reduce the cost and time due to the training of staff for collecting water samples, transferring the water samples, for the laboratory examination and the data documentation [108], [111–113]. The advantages of WSN based WQM systems include:

1. Sensor nodes can be placed in a remote place.
2. Water data can be detected at the desired time.
3. Real-time analysis of water data can be transmitted to the monitoring person.
4. The cost is reduced.
5. Power consumption is reduced.
6. Labour is reduced.
7. Time is saved [103], [114–116].

The studies and implementation of WQM systems based on WSN have been proposed by several researchers all over the world in recent years [46], [107], [110], [112] [117–122]. In [46], the author proposed a “SmartCoast" multi-sensor system for water quality monitoring to measure water temperature, pH, conductivity, depth, and turbidity. The sensors are interfaced with the system in a “Plug and Play” fashion. In [107], the author designed a water quality measuring system and a prototype implementation of a water quality wireless sensor network. The author used triggers sleeping/wake-up modes to propose an energy consumption minimization strategy. The ALIX 2 embedded Linux board is used for wireless sensor gateway layer. In [120], the author deployed the sensors of the network on the sea surface to monitor the water characteristics such as temperature, pH, dissolved oxygen, etc.

In [112], the author developed a web-based wireless sensor network application for monitoring water pollution using Zigbee and WiMAX technologies. The CC2431EM is used as sensor nodes. The system allows users to remotely monitor the water quality from their offices. In [117], the author designed an autonomous WQM system using

Arduino Mega controller and sensors to measure water quality. In [118], the author designed a WQM system using water quality sensors, raspberry pi microcontroller, and Zigbee module. In [119], the author presented a smart sensor design to detect certain water contaminations using sensors to detect water parameters, a PIC microcontroller, and ZigBee wireless communication module. In [120], the author proposed the WQM system using ARM-based microcontroller, water parameter Sensors, ZigBee communication module, a PC and C programming language for software to collect the data of water level, PH, temperature, and carbon dioxide quantity.

In [121], the author proposed a WSN based on a real-time monitoring system for WQM to monitor the data of water parameters. In [122], the author designed a WQM system prototype based on WSN to measure the data of water parameters. Different researchers use different hardware such as sensors, microcontrollers, wireless protocols, gateway and monitoring devices, and software depending on the components utilized. Since the existing sensor interface devices are generally based on the relatively elaborated and dedicated electronic boards, the performance of the devices is limited. Further research should be executed to accomplish a broader space for development in the area of WSN in the IoT environment. The details of existing WQM systems will be discussed in Chapter 2. Since the environmental monitoring is critical and highly demanded, more environmental information is required. In this proposed research, the data of five water parameters such as temperature, water level, the concentration of CO_2 on the water surface, water properties (or) pH of the water and the turbidity of water in real time are detected by the five applicable sensors. The collected data of water quality is sent to the FPGA board for computing and processing. The computed data is then sent to the control station through the ZigBee wireless communication module. Therefore, the users can monitor real-time data of water quality from a remote place.

1.6 Objectives

The main aim of this research is to design and develop a field programmable gate array (FPGA) based wireless sensor network (WSN) in an internet of things (IoT) environment. Distinctively, the objectives of this research are to:

- design a reconfigurable water quality monitoring system (WQM) and develop very high speed integrated circuit hardware description language (VHDL) modules to detect water quality such as water pH, water level, the water temperature in degree Celcius, turbidity of water and concentration of CO_2 on the surface of the water in a real-time basis,
- investigate the performance of the proposed reconfigurable design on a system-on-chip (SoC) platform to enhance the quality of the developed system, and
- evaluate and validate the effectiveness of the proposed system using computer simulations and laboratory experiments.

1.7 Project System Diagram

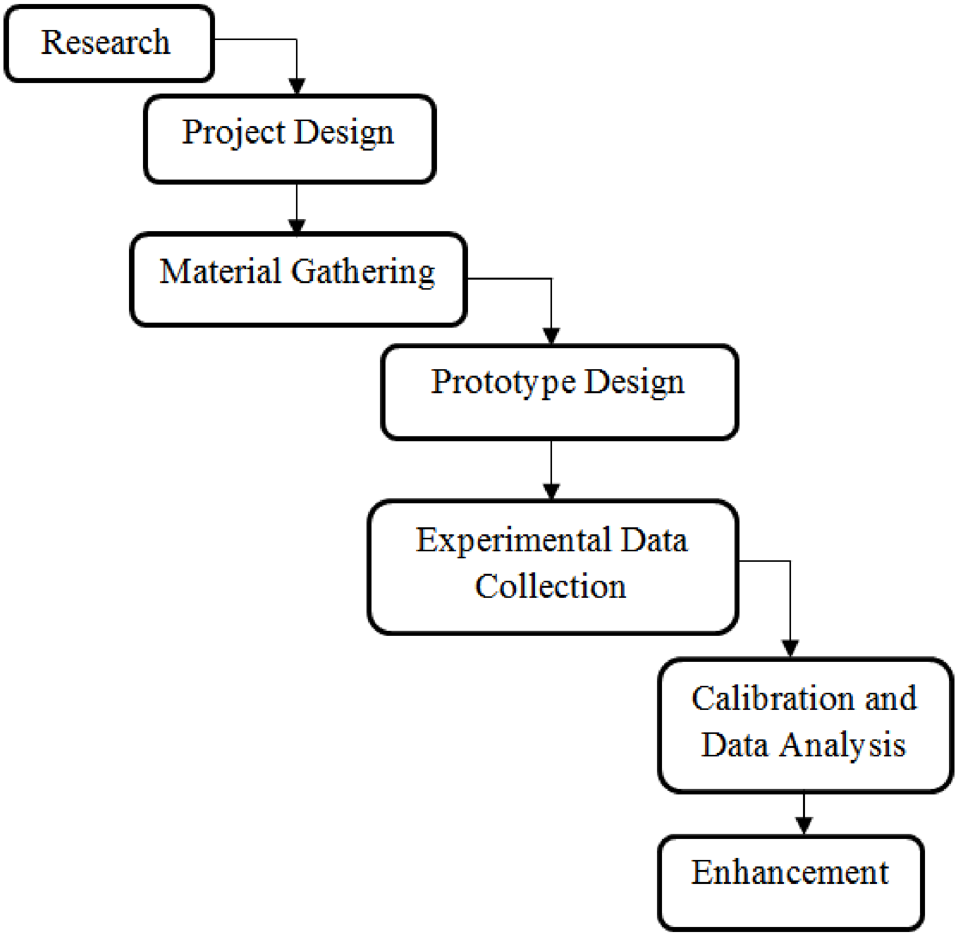


Fig. 1.4: Project System

Fig. 1.4 shows the project flow of the system in which seven activities are scheduled and carried out in order to complete this project.

1.8 Thesis Contribution and Outline

The contributions of the reconfigurable WSN design for WQM in the IoT environment includes: -

1. A prototype of a reconfigurable WSN system capable of detecting real-time data of water quality, wirelessly transmitting the computed data to a remote monitoring PC, saving the data on a console window of the utilized software.

2. Complete Qsys design details for the hardware system that facilitates reconfiguration of data processing algorithms, system hardware, and network topology.

3. Complete design details for the software system which includes the procedures for compiling, configuring and programming of an FPGA SoC based on the Nios-II processor.

The objective of this thesis is achieved by theoretical studies of WQM sensors, WSN and experimental analysis and deliberation of constraints of design solutions. The thesis is organized as follows:

Chapter 2 discusses the literature review of wireless WQM systems. The different wireless communication protocols are discussed while the advantages of a ZigBee network and the Zigbee based existing wireless WQM systems are also presented. The studies of WQM system in the IoT environment are also illustrated in this chapter.

In Chapter 3, the design and planning of the proposed reconfigurable WSN design for WQM in the IoT environment are discussed. The selection of the electronic components, hardware, and software of the proposed system are described in this chapter. Meanwhile, Chapter 4 covers the explanation of a system implementation of both hardware and software in detail. Chapter 5 demonstrates the experimental results with measurements of water parameters and Chapter 6 reviews the studies in a conclusion and includes the contribution of this thesis. Future work is proposed in this chapter.

Chapter 2: Literature Review

In this chapter, the different wireless protocols such as WiFi, Bluetooth, 3G/4G cellular technologies, Global System for Mobile Communications (GSM), GPRS, and Zigbee are discussed. The advantages of using Zigbee network in WSN applications are also discussed and the recent technology of Zigbee based WQM systems are presented. The review of WQM devices using different controllers such as PIC, MSP controller, Intel controller, Arduino controller, ARM processor, Atmel/ATMega controller, FPGA, and SoC boards are also discussed. The recent studies of Wireless WQM in the IoT environment are also presented and the limitations of existing systems of wireless WQM are also revealed.

Chapter 3: Designing the System Model

In this chapter, the selection of hardware and software such as FPGA SoC board and Nios- II softcore processor is discussed in detail. The comparison of hardware sensors and the selection of hardware sensors such as Ultrasonic sensor LV-Maxsonar EZ1, Temperature sensor DS18B20, Atlas Scientific pH kit, CO_2 sensor SKU: SEN0219, and Turbidity sensor SKU: SEN0189 over the other sensors for the advantages of selected sensors are discussed. The selection of Xbee Pro S1 wireless protocol over the Telegesis ETRX3 series modules and the selection of XTU software and Grafana software for the display dashboard are described.

Chapter 4: System Implementation

In this chapter, the implementation of the designed wireless WQM system in IoT environment is discussed starting from the Nios- II software development environment implementation in the initial section which is followed by the hardware design implementation in the Qsys tools of Quartus II. The implementation of hardware components on the FPGA board, the XBee RF module is presented. The WQM sensors are also installed and implemented. The flow chart of software analysis and the installation of software such as XCTU and Grafana is also simplified and presented. The total resource utilization of FPGA board is described in the last section of this chapter.

Chapter 5: Experimental Measurement

In Chapter 5, the operation of the designed device is displayed in the initial section. The measurement of the water parameters of the Lake is described in the latter section. Then, the collected data of water parameters are analyzed in the last section.

Chapter 2

Literature Review

In this chapter, a literature analysis survey is presented to describe the wireless communication protocols. The typical communication protocols in WSN and the architecture of the Zigbee wireless communication protocol are presented. The ZigBee based WQM systems are also presented. The WQM systems based on different microcontrollers and different wireless communication protocols are also described. The studies of WQM in IoT are also discussed in the latter section. The limitations of existing WQM systems are also presented.

2.1 Wireless Communication Protocols

In WSN, the wireless protocols offer wireless data transmission with relevant data rates between the sensor nodes and the control station or base station for a broad coverage area of applications. The well-known communication technologies are WiFi, Bluetooth, 2G/3G/4G, GSM/GPRS and ZigBee.

2.1.1 WiFi

WiFi is the well-known denomination of the wireless Ethernet 802.11b which is the standard for WLANs [123]. The WiFi is a physical/link layer interface and the layers above the physical and data link layers include TCP/IP [124]. The WiFi LANs operate using unlicensed spectrum in the frequency of the 2.4GHz band. Typically, the base station equipment of WLANs is possessed, accessed and operated by the end-user. The wireless link is a hundred meters from the end-user device to the base station which is then connected to the wireline LAN or to a wire line access line to a carrier's backbone network and then eventually to the Internet. The WiFi coverage that is supported by a base station is a hundred meters long, however, the coverage over a greater area can be achieved by using various control stations [123].

The minimum WiFi transmission range is 300m for outdoors and 100m for indoors. The WLANs supports data communication and real-time systems such as voice and video over IP networks [125]. The advantage of WiFi is its high bandwidth as the latest

WiFi technology 802.11n can achieve 300Mbps data transfer rate and about 100M to 150Mbps throughput [126]. Since the WiFi technology is devised for massive data transfer using high-speed throughput, it is used in the public, private organizations, universities and is embedded in all mobile devices such as smartphones, tablets, laptops, etc. The power consumption of Wi-Fi is estimated to be 116 mA at 1.8V when transmitted at 40 Mbps. Power efficiency (Power per bit) is estimated to be at 0.210/40,000,000 = 0.00525 μW/bit [127]. Although the power efficiency of WiFi is remarkable based on mathematical calculation and is excellent in large data transference, WiFi-based devices still need higher power for its peak current consumption.

2.1.2 Bluetooth

Bluetooth technology is commonly known as the IEEE 802.15.1 standard and it is constructed on a wireless radio system fabricated for transmitting data between devices located within the short range [128]. The Bluetooth system utilizes the omnidirectional radio waves which can penetrate walls [129], while Bluetooth radios use a spread-spectrum, full-duplex signal and frequency-hopping [130]. Bluetooth wireless communication systems work in the 2.4 GHz band with the original data rate of 1 Mbps up to the most recent 24 Mbps [131]. The specification of Bluetooth is a low-cost and low-power technology that offers a standardized platform for wireless connection between mobile devices and facilitates connections between devices. The architecture of Bluetooth comprises of link layer and application layer.

The Bluetooth wireless protocol can vary channels up to 1600 times per second to reduce the interference with other protocols within the same band. The transmission of Bluetooth does not stop when encountering the interference of another device; however, the transmission speed is reduced [85]. Two connectivity topologies namely the piconet and scatternet are defined in Bluetooth [132]. The communication between the devices over an ad-hoc network is known as piconet which is automatically and dynamically established as soon as the Bluetooth devices are activated in the radio proximity [133]. When the devices are connected to a network, one of the two devices take on the role as a “master” while the other devices perform as “slaves”. Each piconet is defined by a frequency-hopping channel based on the address of the master. However, the Bluetooth is a master-slave structured packet-based protocol in which one master can only interconnect with a maximum of seven slaves in a piconet [134].

A slave device can be set in standby mode to minimize power consumption. A scatternet is formed when the two or more piconets are connected. In a scatternet, the responsible devices perform the master role in one piconet and the slave's role in another piconet. A device in a scatternet can perform the role of slave in many piconets, but it can only perform the master role in only one of them [135]. The Bluetooth applications are mainly for indoor conditions and powered by a battery. The specification of Bluetooth is a range of not less than 10m and the physical layer can be designed to restrict power consumption to 1mW [136]. The Bluetooth devices within a range of 30.5m can critically communicate when walls are present between devices. The Bluetooth v4.0 is created for the transfer of small amounts of data, and its peak power consumption is lower than 15mA [137]. Since the Bluetooth technology is designed for short-range RF-based connectivity, it has restricted performance, therefore, the applicability of Bluetooth to WSN is limited and in most cases, thus it is not appropriate for the applications of WSN [138].

2.1.3 3G/4G Cellular Technologies

3G is the abbreviated form of "Third Generation" in which it is the third generation of mobile telecommunications technology proposing a download speed up to 3.1 Mbps. The 3G network allows users to perform data transmission thus allowing processes and applications such as file transmission, online TV viewing, voice and video calling, high definition videos viewing, games, internet surfing, etc. [139]. The 3G network includes two main components such as Radio Access Network (RAN) and Core Network (CN) [140]. Based on the different standards, the typical 3G data protocols are EDGE, EV-DO, and HSPA. The 3G RAN uses the existing GSM /GPRS RAN system which is linked to the packet switched network and the circuit switched network.

The 3G services provide mobile communication applications such as mobile internet access, wireless voice telephone, mobile TV and video calls, all in a mobile environment. The cost of 3G is high as it is based on the cellular infrastructure and the cost of upgrading base stations is very high. The power consumption of 3G is high and 3G users require close to base stations. The roaming and data/voice work has yet to be executed [139]. The 3G services provide mobile communication applications such as mobile internet access, wireless voice telephone, mobile TV and video calls, all in a mobile environment. The cost of 3G is high as it is based on the cellular infrastructure and the cost of upgrading base stations is very high. The power consumption of 3G is high and 3G users require close to base stations. The roaming and data/voice work has yet to be executed [139].

4G is the abbreviation for the fourth generation of cellular wireless standards, and the current 4G technologies are Wi-Max, HSPA+, and LTE, due to their use of the Multiple Input/Multiple Output (MIMO) standards [141]. 4G includes all the 3G facilities and it has a data transmission bandwidth of 200Mbps and frequency band of 2-8 GHz. Therefore, 4G technology allows users to collect and transmit data concurrently across multiple frequencies through their cell phones with a significant increase in speed as the aim of the 4G network is to increase data transmission speed dramatically. Currently, the applications of the 4G technology comprise of mobile web access, video conferencing, 3D television, cloud computing, gaming services, high-definition mobile TV and IP telephony. Besides, the cost of 4G is lower than that of 3G [139].

2.1.4 GSM

Global System for Mobile Communications (GSM) is a digital mobile telephone system that was started in Europe and other parts of the world. Today GSM networks operate on different frequency bands such as 850MHz, 900MHz, 1800MHz and 1900MHz and the GSM modem uses AT commands for sending and receiving SMS messages in sim cards [142]. The GSM network is classified into three comprehensive parts: the mobile station, base station subsystem and a network subsystem. The mobile station is transported by the subscriber, and the base station subsystem controls the radio link with the mobile station. The key part of the network system is the mobile service switching centre (MSC) which operates on the switching of calls between the mobile users and between mobile and fixed network users [143].

The mobile station and the control station subsystem interact across the Um interface which is commonly recognized as the air interface or radio link [144]. The users can transmit and receive data across the GSM network at a rate up to 9600 bps to users on

POTS (Plain Old Telephone Service), ISDN, Packet Switched Public Data Networks, and Circuit Switched Public Data Networks using a variety of access methods and protocols, such as X.25 or X.32. A modem is not required between the users and the network because GSM is a digital network. However, an audio modem is necessary for the GSM network to interwork with POTS [145].

2.1.5 GPRS

A packet radio principle is applied by General Packet Radio Service (GPRS) to transmit user data packets in a productive way between mobile stations and external packet data networks. The GPRS is constructed on the top of GSM for distributing data access feature to the users and providing a data rate of 115 Kbps. The GPRS transmits data using grouping exchange with high efficiency and delivers users connection in the form of mobile grouping IP or x.25. Since the GSM Base Station Subsystem (BSS) is adjusted to support the GPRS connectionless packet mode of operation, GPRS has the same bandwidth, frequency slot, wireless modulation, retransmission structure, data frame and frequency-hopping of TDMA as GSM, and it is considered as a service or feature of GSM.

The GPRS network includes two parts, namely wireless access and the main network. The wireless access performs the role as a communicator between mobile node and BSS while the main network enables the communication between BBS and router in a standard digital communication network. Since the GPRS technology is designed by developing the GSM standards bodies, the GPRS associates covering a packet-based air interface on the existing circuit switched GSM network [146]. Thus, in a system with defined functionality, interfaces and inter-network operation for roaming support are offered. Therefore, the user has a possibility to operate a packet-based data service with packet switching where GPRS radio resources are applied only when users are transmitting or receiving data.

A great number of GPRS users can possibly share the same bandwidth and be served from a single cell. Since the public communication network is extendible, the coverage of the public communication network of GPRS is immense. The GPRS is estimated to be used for e-mail, Web browsing, points of sale, traffic telemetric systems, and various vertical applications. Since the GPRS provides point-to-point and point-to-multipoint connections, a wide range of applications can be operated. The cost is based on usage, usually on a monthly basis. The GPRS is a formalized network connection which is powered by a telecommunications provider. Since the IP address is attached in the GPRS gateway node, the mobility of the data terminal to GPRS-based internal networks limited. The power consumption ranges from approximately 1uA during stop mode to 480mA during transmission on average.

2.1.6 ZigBee

ZigBee is a standard protocol based on IEEE 802.15.4 standard which is adopted by the ZigBee Alliance to achieve wireless communication with low power consumption and low data rate [147]. The ZigBee is designed to transmit data up to 250 kBps in the 2.4 GHz frequency band within 90m of indoor coverage and 3.2km of line-of-sight outdoor coverage. The ZigBee is used in different wireless applications such as commercial building automation, smart energy, telemedicine, home and hospital care, home automation, remote control for consumer electronics, telecommunication applications, and industrial process monitoring and control.

Based on an open global standard, hundreds of members from around the world have associated the Alliance and are collaborating to enable cost-effective and reliable networking of wireless devices. The ZigBee is created to fulfill the necessities of sensors and control devices for Low-Rate Wireless Personal Area Networks (LR-WPAN) and it is suitable for building an ad-hoc network between stationary or mobile devices for its features. These features include low power consumption, low cost, network flexibility, low data rate, and two-way wireless communications [148]. The ZigBee has been developed for its special advantages that enable safe and reliable data transmission with a transmission range of 100+ meters, low equipment costs, an accessible and flexible network configuration, and long-lasting batteries compared with other wireless technologies such as Bluetooth and Wi-Fi [149]. The architecture of the ZigBee is categorized into three divisions as follows:

1. IEEE 802.15.4e which comprises the MAC and PHY layers.

2. ZigBee Alliance which is composed of the Network (NWK) layer, the Application Support Sublayer (APS), security service management, and the ZigBee Device Object (ZDO).
3. ZigBee application section in which the users can apply the ZigBee application profiles and invent their own application [150].

The ZigBee specification is created on the top of the IEEE 802.15.4 standard to launch large-scale wireless network technology which mainly focuses on the development of two layers, namely the physical (PHY) layer and the media access control (MAC) layer [150]. The features of ZigBee such as low cost, minimal power consumption, low data rate, low complexity, and ease-to-implement are made of these layers. The specifications of the PHY layer are a wide range of operational low-power features, low-duty-cycle operations, strict power management, and low transmission overhead. As the PHY layer of the IEEE802.15.4 standard controls and communicates with the radio transceiver directly, the three operational frequency bands are operated by the PHY layer: 868 MHz in Europe, 915 MHz in the USA and 2.4 GHz in most other countries in the rest of the world. In addition, the PHY layer offers a number of features, such as receiver energy detection (RED), link quality indicator (LQI), and clear channel assessment (CCA) to support the operation of the MAC layer [151].

The MAC layer also defines different network topologies. The specification of MAC-layer is invented to promote the control and monitoring of industrial and home applications with low to medium data rates and average delay requirements. The network association and disassociation are managed by the MAC layer which also synchronizes admittance to the medium through two modes of operation namely beaconing and non-beaconing [152]. The beaconing mode controls and forwards data of the environment by the means of a consistently active device while the non-beaconing mode controls the use of unslotted, non-persistent CSMA-based MAC protocol. Fig. 2.1 shows the ZigBee stack architecture that consists of IEEE 802.15.4 and ZigBee reference model. The network layer integrates between the application layer and MAC Layer and it is important for network formation and routing to relay the signals to the targeted node. The functions of this layer are to connect and abandon a network, actual

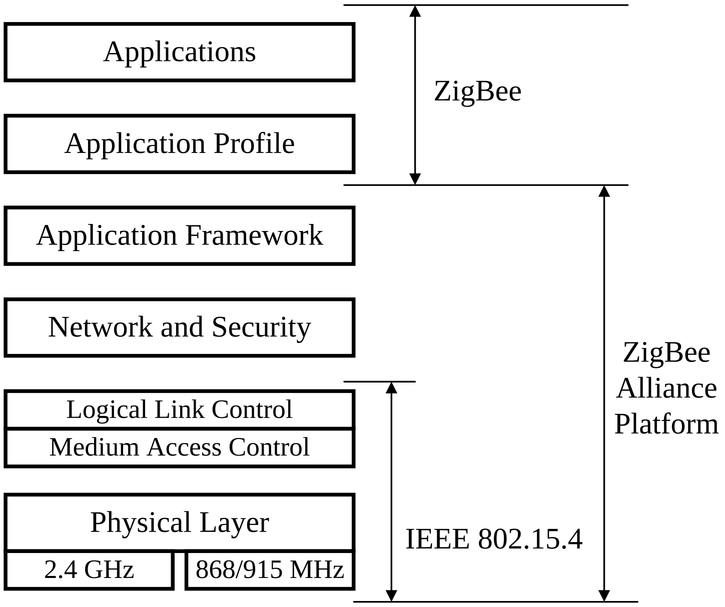


Fig. 2.1: Overview of ZigBee Stack Architecture

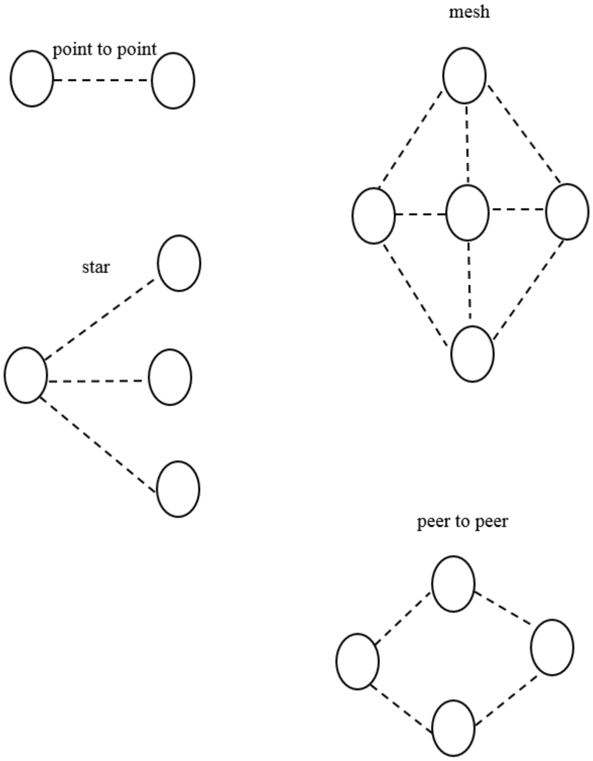


Fig. 2.2: Basic network topologies of ZigBee

routing, frame security, route discovery, one-hop neighbour discovery and information storage.

The network layer defines the basic topologies such as point to point, peer to peer, multipoint, star and mesh by developing the application of peer-to-peer topology. Fig. 2.2 shows the basic topologies of the ZigBee network. The network layer allows the maximization of the battery life of low power devices [153]. The application that supports the sub-layer retains tables for binding and forwards the messages between bound devices. The functions of ZDO are controlling and managing the application objects. The ZigBee specifications consist of ZigBee coordinator, ZigBee router, and ZigBee end device. A Zigbee coordinator performs the role of an access point which receives all the inward bound data packets from the router and end node. Conversely, a ZigBee end node transmits data packets to either a coordinator or a router sensor. A ZigBee router performs the roles of both transmitter and receiver by handling the data packets from/to adjacent routers and end nodes. The ZigBee router is an IEEE 802.15.4 full-featured ZigBee node. The router connects the existing networks and extends the ZigBee network by adopting new devices that transmit and receive information. The ZigBee router performs as a backbone of the network by applying the created routing protocols.

The ZigBee end device is frequently positioned at the end of the network to execute the sensing tasks with the help of a router or coordinator [153]. The ZigBee networks consist of a single coordinator device which is the PAN coordinator in accepting the new devices, forming the network, and assigning network address. The advantages of ZigBee are

- Low cost
- Low power consumption
- Easy to deploy
- Appropriate range of operation (30 – 100m)
- Excellent performance in environment with low signal-to-noise ratio
- Data transfer rate of 250kbps at 2.4GHz
- Security of data transfer
- Capabilities of implementation with any type of microcontroller

- Requirements of minimum of 64 kb of ROM and 2-32 kb of RAM.

The power consumption of a ZigBee device is 0.035706W when transferring 24 bytes of data. A ZigBee device uses up 0.035706 W when transferring 24 bytes of data [127]. The power consumption of ZigBee depends on their coverage area. Hence, sleep mode can be implemented in ZigBee device which will help to save most of the energy in the WSN.

2.2 ZigBee based WQM Systems

Bhatt and Patoliya [118] presented their study on "IoT Based Water Quality Monitoring System" to observe the water quality using a raspberry pi microcontroller and Zigbee module. The system consists of several sensors to measure pH, turbidity, conductivity, dissolved oxygen, and temperature. The aim of their work was to ensure the safe supply of drinking water the quality in real time and the system to be low cost, faster, more efficient, real-time and user-friendly. Cleote, Malekian, and Nair [119] proposed the Design of Smart Sensors for Real-Time Water Quality Monitoring using ZigBee to measure physiochemical parameters of water quality, such as flow, temperature, pH, conduction and the redox potential. The system consists of sensors to detect water parameters, a PIC microcontroller, and ZigBee receiver and transmitter modules for wireless communication. The system detects certain water contaminations.

Khaire [120] presented the Water Quality Data Transfer and Monitoring System in IoT Environment to collect the data of water level, PH, temperature, and carbon dioxide quantity. The system consists of an ARM-based microcontroller, water parameter Sensors, ZigBee communication module, a PC. A C programming language is used for software. The aim of the system is to design a cost-effective solution to interface palpatory to the sensor network and to achieve reliability and feasibility of the results. Curiel et al. [121] proposed a WSN based on a real-time monitoring system for WQM to monitor the data of water parameters such as pH and temperature using ZigBee protocol. The authors mentioned that the primary goal of their research work was to reduce the cost and time of the pH and temperature test of water quality.

2.3 Wireless WQM Systems using Microcontrollers

A number of researchers studied the wireless WQM systems using different microcontrollers such as PIC controller, MSP controller, Intel controller, Arduino controller, ARM processor, Atmel/ATmega controller, and FPGA board.

2.3.1 Wireless WQM System using PIC Controller

Karuppasamy et al. [154] presented Water Quality Monitoring and Control using Wireless Sensor Networks system using PIC microcontroller, water quality sensors, and ZigBee wireless communication module. The system detects water temperature, pH, DO and electrical conductivity in real-time and the detected water data is transmitted to relevant stakeholders through a web-based portal. The aim of the system is to monitor the water data and to control the water resources by providing relevant and timely information to stakeholders. Somasundaram and Edison [155] studied Monitoring Water Quality using RF Module using PIC microcontroller, pH sensor, temperature sensor, turbidity sensor, RF transceiver, UART and a PC with hyper terminal software. The system monitors the water available through the taps through various sensors. The water data is transmitted to a remote base station.

Devi et al. [156] proposed a real-time system for determination of drinking water quality using PIC microcontroller, an array of the sensor node, ZigBee transceiver, and GSM transmitters. The aim of the system is to determine the quality of water automatically using low cost and in-pipe sensors. Hasan and Khan [157] presented an analysis of GSM based automatic water quality control using PIC microcontroller, water quality sensors and GSM network. The automatic measurement and reporting system of water quality system has been developed in the system. The water data of pH, turbidity, and conductivity is detected and transmitted to the monitoring centre by GSM network. Somasundaram and Edison [158] designed RF-based WSN technology to monitor water quality using pH sensor, PIC microcontroller, turbidity sensor (LED-LDR assembly), temperature sensor, 2.4 GHz RF transceiver, 12V power supply, UART and a PC with hyper terminal software. In the proposed project, the water quality is monitored continuously, and the signals are transmitted to the control station. The users can easily observe the quality of the water with the help of GUI.

Karthikeyan et al. [159] studied the elevation of water monitoring scheme based on ZigBee and wireless antenna capabilities. The study implemented the design and execution of a prototype using WSN in which data is identified by means of dissimilar sensors at the node plane to compute various parameters such as turbidity, pH and oxygen quantity. The data from the results are transmitted via WSN to the control station. Pingle and Jadhav [160] designed a GSM based automatic water quality measurement and reporting system to examine the water parameters such as turbidity, pH level, temperature and DO. The PIC microcontroller is utilized as a core controller in the proposed study in which the qualities of water are measured in real-time. The relevant sensors, ADC, GSM module, and LCD are used as the electronic components and hardware of the proposed project.

Rajasekar [161] proposed a system to measure the water parameters using water quality sensors, PIC microcontroller, GSM network, and MPLAB IDE. The system measures water temperature, turbidity, conductivity and pH and the data is transmitted to the monitoring center by GSM. The aim of the system is to report abnormality in the water quality to the monitoring center. However, the WQM systems based on PIC microcontrollers cannot perform the parallel processing and the power consumption of the proposed WSN systems is very high.

2.3.2 Wireless WQM System using MSP Controller

Kumar et al. [162] proposed an efficient water quality and quantity monitoring system using MSP microcontroller. The system is implemented to monitor water quality while managing the quantity of drinking water using WSN that also monitors pH, temperature, conductivity, purity, salinity and water flow. Jinfeng and Shun [163] presented an aquaculture WQM system using ZigBee wireless module and MSP microcontroller. The system detects, transmits, displays and examines the data of water parameters such as pH value, temperature, water level and DO concentration. Duy et al. [164] presented an automated monitoring and control system for shrimp farms using sensors to measure pH, oxygen, and temperature of water. The system was implemented using ZigBee, GSM, Cloud, MSP430 microcontroller and LabVIEW technologies.

Kudva et al. [165] designed a real-time water balance monitoring system using an ultra-sound level sensor, a Texas Instrument MSP controller a sub-gigahertz radio to set up a hub and spoke system. The designed system monitors the water balance in a campus-scale water distribution system in real-time. Chen and Liu [166] proposed a smart monitoring system of aquaculture to measure the pond water quality parameters such as water level, temperature, PH and dissolved oxygen using water quality sensors, MSP microcontroller, and GPRS module. The system monitors the water parameters of the pond and the water data is transmitted through a GPRS communication module to a remote place. Xiao-Peng et al. [167] studied a sensor network to estimate the velocity of the water stream in rivers using the MMA72609 accelerator, MSP430F1612 microcontroller, and CC2420 IEEE 802.15.4 transmitter. The accelerators are utilized in the system to examine the bail swing angle.

2.3.3 Wireless WQM System using Intel Controller

Purohit and Gokhale [168] presented a real-time WQM system by applying various sensors for detecting water quality, Intel microcontroller, Analog to Digital Converter (ADC), GSM module and a liquid crystal display (LCD). The time and cost for development are increased for the complication of the circuit architecture as microcontrollers have more complex structures.

2.3.4 Wireless WQM System using Arduino Controller

Beri [169] proposed an autonomous real-time system to detect the physical and chemical parameters of water such as turbidity, temperature, and pH using the Arduino Atmega microcontroller and Zigbee wireless module. In the proposed design, the sensors detect the threshold level values and the Arduino Atmega microcontroller generates alarms and sends information of the abnormality of a particular quality parameter to the authenticated user through the ZigBee wireless module. The manipulated data through Atmega microcontroller is transmitted through Zigbee and displayed on the LCD. Faustine et al. [122] designed a WQM system prototype based on WSN to measure the data of water parameters including DO, pH, electrical conductivity (EC) values and temperature in a real-time basis. The system consists of water quality sensors, an Arduino microcontroller, and a network connection module.

Akila et al. [170] proposed a WQM system that monitors the pH level and temperature of the river water based on the Arduino board and sensors. In the proposed WQM system, the pH and temperature sensors are placed in the river. The analog output of all the sensors is converted into digital value, and subsequently, the data is sent to the Arduino board. The values are evaluated with the threshold value after conversion. In the case of inference value above the threshold value, the automated warning SMS alert is transmitted to the pollution control board via GSM. Rao et al. [117] designed an autonomous WQM system using Arduino Mega controller and sensors to measure temperature, light intensity, pH, EC, total dissolved solids (TDS), salinity (SAL), DO and oxidation-reduction potential (ORP). Curiel et al. [134] proposed a WSN for WQM to detect the water quality parameters such as temperature and pH through a real-time monitoring system using pH sensor, temperature sensor, Arduino Uni IDE that is based on ATmega chip and Zigbee protocol.

However, there are some disadvantages of Arduino technology including the relatively high cost of the chip. It is an 8-bit microcontroller and the top clock speed is limited to 20 MHz. There are not many internal resources regarding the Flash and Electrically Erasable Programmable Read-Only Memory (EEPROM) and Static Random-Access Memory (SRAM). Moreover, the overheating issue with the input voltage at 12V only after a few minutes of operation is also one of the concerns. Moreover, the Arduino libraries are not very efficient in certain parts hence the Central Processing Unit (CPU) and Random-Access Memory (RAM) cycles are wasted.

2.3.5 Wireless WQM System using ARM Processor

Kumar et al. [171] proposed a solar based WQM system in WSN using water quality sensors, ARM microcontroller, and ZigBee communication module. The system monitors the water parameters such as pH, Turbidity and oxygen level and the water data is transmitted to the base station through the ZigBee module. Barabde and Danve [172] developed a system for continuous detection of water quality at distant places using an ARM processor and GUI design in Matrix Laboratory (MATLAB). The architecture of the proposed sensor based WQM system comprises data monitoring nodes, a control station, and a remote station. The data of water parameters such as conductivity, turbidity, and pH are transmitted to the remote monitoring station. The received data at the remote station is presented in a visual format on a PC using MATLAB. When the received value is above the threshold value, the automated warning SMS alert is transmitted to the authorized person of the system.

Khetre [173] proposed the different data routing methods and WSN system for monitoring the water quality using ARM 7 microcontroller and ZigBee wireless module. Kaikade and Khandait [174] proposed a design for examining the water quality in real-time using ZigBee and ARM7 processor. The relevant sensors of the system detect the various water parameters such as turbidity, pH, temperature and DO and the detected data is operated by the processor and transmitted to the remote station through the wireless module. Patil and Khaire [175] designed a WSN for real-time observation and detection of water contamination using ARM7 controller and ZigBee wireless module. The proposed system detects the water parameters in water distribution systems such as pH, turbidity, EC, and temperature. Joshi [176] presented a WQM system using WSN technology and powered by a solar panel. The system consists of sensors, ARM microcontroller, and ZigBee wireless module. The system monitors water quality over different sites as a real-time application and the water quality data is transmitted to a remote site.

The disadvantage of using the ARM processor is that Windows programs on PC or laptop will need constant compilation [177]. Operating the Windows programs on an ARM processor is still relatively slow, and unsuitable for the vast array of tasks operated in a desktop or laptop since it is not binary.

2.3.6 Wireless WQM System using Atmel/ATmega Controller

Simbeye and Yang [178] studied a WSN monitoring and control system for aquaculture using Atmel microcontroller and ZigBee wireless module. In the system, the LabVIEW software platform is used for data analysis, computing, and display. Ramya et al. [179] designed a web-based system to observe the underground and fish pond water quality using the Atmel controller and Zigbee module. The proposed design monitors the pH scale level, temperature, DO, water level and TDS. The water parameters sensed by the corresponding sensors are sent to the microcontroller unit (MCU) and it is compared with the standard point and an alarm system is caused according to it. Seithi and Jadhav [180] designed an automatic water quality check system using Atmel microcontroller. The proposed system consists of multiple sensors, a microcontroller, ZigBee transceiver, and GSM module to measure the quality of water. Since the Atmel microcontroller is an ARM-based controller, the performance of the Windows program is relatively slow. However, the system needs to be more robust and scalable. Therefore, more network performance metrics are necessary to be studied and evaluated.

Anandrao and Kolhare [181] presented a wireless liquid monitoring system using ultrasonic sensor and ATmega328 microcontroller. The CC2500 module is used for the serial communication link to and from the microcontroller. The aim of the system is to monitor continuous and correct information about water level. Daigavane and Gaikwad [182] proposed a WQM system based on IoT using water quality sensors, Atmega328 microcontroller, and WiFi module. The system measures the temperature, PH, turbidity, flow of the water and the data is viewed on the internet using WiFi system. R. Kamble et al. [183] proposed an automatic WQM system using water quality sensors, ATmega328 microcontroller, and GPRS module. The data of water parameters such as pH, turbidity, and conductivity is measured and the data is displayed on a webpage using internet access. Al-Mamun et al. [184] designed a multi-sensor system for water level monitoring using an ATmega microcontroller, an ultrasonic distance sensor, LCD display, and RF module. The system measures the real-time water level. The system can be further improved according to the author.

2.3.7 Wireless WQM System using FPGA

The studies [185–189] show that the FPGA boards have been employed for the sensor systems development in diverse applications such as multimedia control, industrial control, environmental monitoring, safety and security, computer visualization, control, and signal processing, groundwater monitoring, water meter system and leakage detection, wake-up radio in WSN. The design of the FPGA-based water monitoring system promotes energy efficiency while parallel processing enables shorter processing time [190]. Hsia [188] designed an FPGA chip-based water meter system and leakage detection. A signal generator, a detection circuit, data encoder and a serial port for the transmission of data are comprehended by the FPGA chip. The system comprises a pressure sensor, an ADC, and an FPGA board.

Since energy consumption is the primary issue for the implementation of battery based WSN nodes that are used in monitoring applications, ultra-low power FPGA boards are more preferable to develop low power WSN nodes. Ahmad and Pawar [191] presented the hydro-monitor and quality analysis using FPGA and water quality sensors to measure the water parameters such as temperature, conductivity, Water flow, the water level in the tanks. The system is also designed for automated filling the water for homes or the societies without human intervention. Perera et al. [192] presented a single chip solution based on FPGA using 1-wire protocol for the prototype of smart sensor nodes. The design measures the pH value and temperature of water samples and authenticates the operation of the studied sensor node. The FPGAs can be easily replaced with a wired or wireless method by using a wireless module that has a UART interface in the proposed system.

Dai et al. [193] designed an environment online data collection system for Hongze Lake based on a GPRS using an FPGA chip. In the proposed design, a Nios-II microprocessor and its peripheral interface circuit are constructed by FPGA to detect the temperature, pH and dissolved oxygen of water. The water parameters are converted by the Nios-II microprocessor, and the data is transmitted to GPRS mobile data transmission module. Simultaneously, the data is displayed in numerical and waveform by remote data acquisition server. The system provides a general implementation method for the remote data transmission using the Nios-II processor as the main control chip of the system. Further studies are needed to continuously upgrade the system and improve the system’s performance.

Beena and Khaja [194] proposed a prototype of a WSN based automatic system to observe and control the quality of water such as temperature, pH, DO and turbidity from a remote place. The software of the system consists of the LabVIEW software and NI myRIO which has an FPGA, an inbuilt processor, Wi-Fi and web servicing capabilities. The measured data is transmitted to the farmer at a remote place through built-in Wi-Fi of NI-myRIO. The data is later displayed on the user’s computer screen together with audio output from speakers of the computer. The settings of the system can be changed by the user. When the variations of the detected parameters of the system occurred, the SMS alert is delivered to farmer's mobile at the base station via GSM modem.

2.3.8 Wireless WQM System in IoT Environment

Tiwari [195] designed a hardware/software based on a smart sensor interface device for WQM in IoT Environment using the Complex Programmable Logic Device (CPLD) as the core controller. The designed device detects the parameters of water such as light intensity, water temperature, CO_2, and turbidity. The CPLD programmable technology and the standard of IEEE1451.2 smart sensor identifications are combined in the designed device and the parallel and real-time with high speed on various diverse sensor data is detected. Chi et al. [68] demonstrated an IEEE 1451 standard based industrial WSN system to measure the quality. The author designed a reconfigurable smart sensor interface device using the combination of CPLD and the application of wireless communication in an IoT environment. Further study should be executed to accomplish a broader space for development in the area of WSN in the IoT environment. Since these interface devices are generally based on the relatively elaborated and dedicated electronic boards, the performance of the devices is limited.

As the WSN in IoT technology is used in environmental monitoring applications for time and cost-effectiveness, Vijayakumar and Ramya [196] presented a real-time WQM system in the IoT environment. The system includes numerous sensors to observe water data and the Raspberry PI B+ model as a primary controller. Because of its hardware and processor, the Raspberry Pi runs on Linux kernel, as it cannot run on Windows operating system.

2.4 Research Method

In the proposed WSN system, a reconfigurable smart sensor interface device that integrates data collection, data processing, and wired or wireless transmission is designed. The driver of chips on the interface device is also programmed inside the FPGA. In terms of data transmission, the proposed design achieved wireless communication through a ZigBee module. In the proposed research work, the parameters to determine and monitor the basic water conditions are the water level, water temperature, carbon dioxide on the water surface, the turbidity of water and water pH value. Therefore, the relevant sensors are chosen to collect relevant data. Next, the measured data is sent to the FPGA board which performs as a core brain of the water quality monitoring system. The VHDL and C are programmed in Altera Quartus II software for computing the data. The computed data is transmitted to the remote user through ZigBee wireless communication modules.

2.5 Summary

In this chapter, the literature that describes the wireless protocols used in WSN is presented. The most commonly used communication protocol in WSN has been discussed and the architecture of ZigBee wireless communication protocol has also been presented. The studies of WQM systems based on different microcontrollers and different wireless communication protocols have also been described. The limitations of existing WSN based on WQM systems in IoT have also been presented.

Chapter 3

Designing the System Model

The previous chapter discussed the review of WQM systems using different microcontrollers, different software, and different wireless protocols. It has been stated that further researches are essential to be executed to accomplish an extensive area for progress in the domain of WSN in the IoT environment. In this chapter, the challenges to meet the requirements of the WQM system are discussed. The design plan of the experimental set-up of the reconfigurable WSN design for environmental monitoring in an IoT environment that involves data assembling, data processing, and wireless transmission is also included in this chapter. The selection of hardware components such as sensors, electronic components and processor board, the selection of software, and the selection of wireless protocol to design the WQM system based on WSN in IoT is also illustrated. The discussions in this chapter are critical to creating a wireless WQM device and its working principle for implementation purposes. This includes the mathematical routines utilized which are to be transferred to the software program. Software programming will be presented in Chapter 4.

3.1 Challenges

For environmental monitoring applications based on WSN, consumption of energy is a high-priority issue for the implementation of a great number of energy-limited sensor nodes in an isolated environment. Thus, the choice of processor board, electronic components, and wireless communication protocol are important. All the sensor nodes are essential to be maintained small in size and maximize security while ensuring long-term use. The system must be user-friendly, easy to reconfigure, flexible and affordable. Therefore, the selection of software is also important.

3.2 Hardware Selection

Many researchers have studied the application of FPGAs in WSN systems and have compared the power consumption between a microprocessor and FPGA [197]. The system must be kept low in power, low cost, flexible to reconfigure and simple in use. As microcontroller-based WSN systems have more complex architecture and cannot perform parallel processing, therefore, the microcontroller based WSN systems are time-consuming and not cost effective. Lysecky and Vahid [198] presented the potential software speedups of 200 to 1000% and energy reduction by as much as 99% using FPGA board and soft-core processor. The authors discovered that operating time and power consumption can be reduced by implementing the combination of hardware and software solutions. Hinkelmann et al. [199, 200] stated that the FPGA allowed the integration of various types of sensors in addition to the application of different protocols of MAC. An FPGA could be used for wireless applications to develop the complex techniques of self-testing and evaluation of an assemblage of different sensors [201].

3.2.1 Selection of FPGA

Since a low-cost and a low-power single-chip of a fully integrated autonomous SoC-based wireless sensor node is essential to illuminate the recent problems of WQM systems, FPGA SoC is considered as the main hardware of the proposed project. The SoC comprises a Microprocessor without Interlocked Pipeline Stages (MIPS) based processor, a timer, a memory, a bus, a transceiver, and a battery. Although there are some FPGA manufacturers, the two biggest FPGA manufacturers are Altera/Intel and Xilinx. The other FPGA manufacturers are Quicklogic and Archronix, Lattice Semiconductor and Microsemi (formerly Actel) [202]. For the proposed project, Altera/Intel and Xilinx are reviewed and one of them will be selected. The power consumption, cost, features, the soft-core processor and availability and functionality of the system development software are the criteria in selecting FPGA.

3.2.1.1 Xilinx

Xilinx manufactures FPGAs, SoCs, and PLDs. For FPGA, the Xilinx manufactures the 5 families of FPGA: (1) Artix-7; (2) Kintex-7; (3) Virtex-7; (4) Spartan-6; (5) Spartan-7; (6) Virtex-6; (7) 7 Series FPGAs [203]. When considering the cost, the Spartan-6 family is preferred since the Xilinx FPGAs in the Spartan-6 family costs approximately half of the Cyclone FPGAs from Altera. The Spartan-6 devices comprise of up to 150,000 logic cells, 4.8Mb of embedded memory, 6 phase locked loops and 180 18×18-bit multipliers. An IDE which is known as Vivado in Xilinx is used for system development, in which the IP is offered. The Xilinx also provides a soft-core processor IP MicroBlaze for the Spartan-6 devices. The MicroBlaze is a Harvard architecture processor that utilizes 32-bit instructions. The MicroBlaze also includes Fast Simplex Link (FSL) that is a hardware acceleration feature which performs as a high-speed link between the MicroBlaze processor and hardware logic functions. The maximum performance specification of MicroBlaze is 209 DMIPS at a clock rate of 161MHz.

The embedded peripherals, optional memory management unit (MMU), cache size and pipeline length of the MicroBlaze processor are configurable. One of its attractive features is that the unused processor instructions can be erased from the final configuration to save FPGA area [204, 205].

3.2.1.2 Altera

Altera is the product brand name of Terasic/Intel which manufactures the implementation of specific integrated circuits (ASICs), programmable logic devices (PLDs), SoC and FPGAs. Three types of FPGA are produced; the Stratix Series which is known as High-End FPGAs, the Arria Series which is known as Mid-Range FPGAs and the Cyclone Series which has the lowest cost and the most powerful FPGAs [206]. Among these three types, Cyclone Series is preferred thus considered for the proposed project for its low cost. The Cyclone Series include Cyclone, Cyclone II, III, IV, and V where the Cyclone is preferred for SoC. Integrated development environments (IDE) which is a Quartus II software application for software development and downloads are available for Linux and Windows operating systems [207]. The tools of software and hardware for FPGA provided by the Altera/Intel Quartus Prime Pro Edition and the Xilinx are compared in Table 3.1 [208].

System on a Programmable Chip (SOPC) builder is included in Quartus II and it enables the specification of the components of an FPGA-based hardware system via a GUI. The SOPC builder is a powerful tool to create a complete system containing memory interfaces, peripheral devices and a processor [209]. Since the environmental monitoring applications require higher performance such as intensive data processing, the Altera FPGA is the most suitable choice for the hardware. Moreover, the Nios-II processor provides a migration path to an ASIC [210]. A 32-bit Reduced Instruction Set Computer (RISC) processor with a Harvard architecture Nios- II processor is supported by all the Altera/Intel FPGAs, SoCs, and ASICs. Unlike microcontrollers, the FPGA can be programmable depending on user’s applications since a fixed hardware structure is not available [211].

However, the operations can be executed “in a sequential manner” by using software and the interconnections are determined by the user in accordance with the purposed applications. Although both Altera and Xilinx have suitable hardware, features and development systems for the reconfigurable WQM system in IoT environment, the Cyclone V FPGA SoC with Nios-II soft-core processor was selected over Xilinx due to Nios-II processor’s migration path to an ASIC. Moreover, the features of Cyclone V FPGA SoC such as parallel processing, low data rate, reconfigurable significant computing resources as a multisensory data collection interface device with good compatibility and regulating interface standard are essential to be developed in the IoT environment.

3.2.2 Altera Cyclone V FPGA SoC 5CSEMA5F31C6

The Altera/Intel DE1-SoC 5CSEMA5F31C6 board is used to learn about digital logic, computer organization, and FPGAs. The remarkable reconfigurability paired with a high-performance and low-power processor system can be controlled. The FPGA and the hard processor system (HPS) are provided on the board. An ARM-based HPS comprising of peripherals, a processor and memory interfaces that are attached perfectly with the FPGA material which utilizes a high-bandwidth interconnect backbone are integrated into the DE1-SoC.

Table 3.1: Tools of software and hardware for FPGA [208]

Xilinx	Altera/Intel	Description
Vivado® HL Design Edition UltraScale® UltraScale+® 7 Series	Quartus® Prime Pro Edition Stratix® 10, Arria® 10 Cyclone® 10 GX	Optimized to support the advanced features in next generation FPGAs and SoCs.
ISE® Design Suite for: Spartan-6 Virtex® -6 CoolRunner™ Previous generations	Quartus® Prime Standard Edition for: Stratix® IV, V Arria® series Cyclone® IV, V, Cyclone® 10 LP MAX® series	Supports older generation device families. Note: Altera/Intel® recommends using newer version Quartus® Prime for new designs.
Vivado® HL WebPACK Edition	Quartus® Prime Lite Edition	Do not require license. Provide limited device and feature support.
SDAccel Environment Selected UltraScale® and 7 Series devices	FPGA SDK for OpenCL™ Stratix® 10, Arria® 10, Cyclone® V SoC	Development environment for OpenCL
SDSoC Environment	SoC FPGA Embedded Development Suite	Comprehensive tool suite for embedded software development on SoCs. You can code, build, debug, and optimize in single IDE.
Software Development Kit	Nios® II Embedded Design Suite (EDS)	Comprehensive development package to develop and debug code for SoCs.
System Generator for DSP	DSP Builder for FPGAs	DSP Design Tool
Vivado® High-Level Synthesis	Intel HLS Compiler	High-Level synthesis tool that takes in untimed software code as input and generates register-transfer level code for FPGAs.

The DE1-SoC development board provides FPGA hardware and HPS. The FPGA hardware includes Altera serial configuration device – EPCS128, USB-Blaster II onboard for programming JTAG Mode, 64MB SDRAM (16-bit data bus), 4 push-buttons, 10 slide switches, 10 Red user LEDs, six 7-segment displays, four 50MHz clock sources from the clock generator, 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks, VGA DAC (8-bit high-speed triple DACs) with VGA-out connector, TV decoder (NTSC/PAL/SECAM) and TV-in connector, PS/2 mouse/keyboard connector, IR receiver and IR emitter, two 40-pin expansion headers with diode protection, A/D converter, and 4-pin SPI interface with FPGA.

The HPS provides 800MHz Dual-core ARM Cortex-A9 MPCore processor, 1GB DDR3 SDRAM (32-bit data bus), 1 Gigabit Ethernet PHY with RJ45 connector, 2-port USB host, normal Type-A USB connector, micro SD card socket, accelerometer (I2C interface + interrupt), UART to USB, USB mini-B connector, warm reset button and cold reset button, one user button and one user LED, and LTC 2×7 expansion header. The FPGA device provides 85K programmable logic elements, 4,450Kbits embedded memory, 6 fractional phase lock loops (PLLs), and 2 hard memory controllers.

Meanwhile, memory devices such as 64MB (32M×16) SDRAM on FPGA, 1GB (2× 256M ×16) DDR3 SDRAM on HPS, Micro SD card socket on HPS, two port USB 2.0 Host, UART to USB (USB mini-B connector), 10/100/1000 Ethernet, PS/2 mouse/keyboard, IR emitter/receiver, and I2Cmultiplexer are available for communication. One LTC connector (one Serial Peripheral Interface (SPI) master, one I2C, and one GPIO interface), one 10- pin ADC input header and two 40-pin expansion headers are available for connectors. A 24-bit video graphics array (VGA) digital-to-analog converter (DAC) is used as the monitoring screen or the display of the DE1-SoC board. For video, 24 -bit CODEC, line -in, line -out, and microphone-in jacks for video input, TV decoder (NTSC/PAL/SECAM) and TV in the connector are provided respectively. Analog to Digital Converter (ADC) with 1 MSPS of fast throughput rate, 8 channels, and 12-bit resolution is also provided. The 2V direct current (DC) is compulsory for the power source of the board. The DE1-SoC board possesses various attributes, therefore, users can apply an extensive scope of prototypes and circuits, from simple circuits to numerous projects [211]. The top view, bottom view and block diagram of Cyclone V DE-1 SoC board are shown in Fig.3.1–3.3 [212].

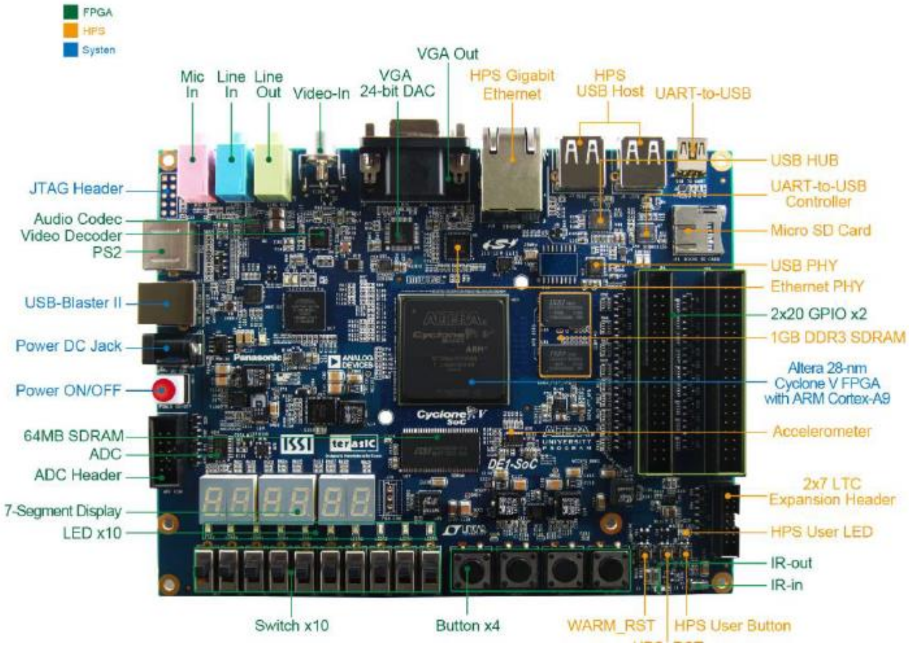


Fig. 3.1: Top View of CycloneV DE-1 SoC 5CSEMA5F31C6 Board [211].

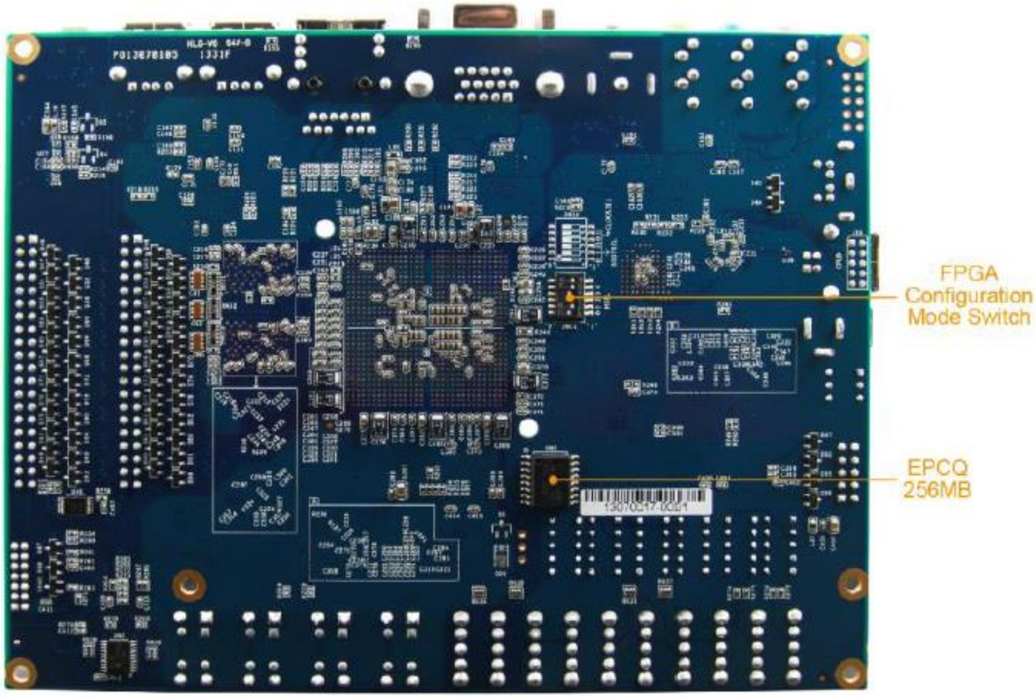


Fig. 3.2: Bottom View of CycloneV DE-1 SoC 5CSEMA5F31C6 Board [211].

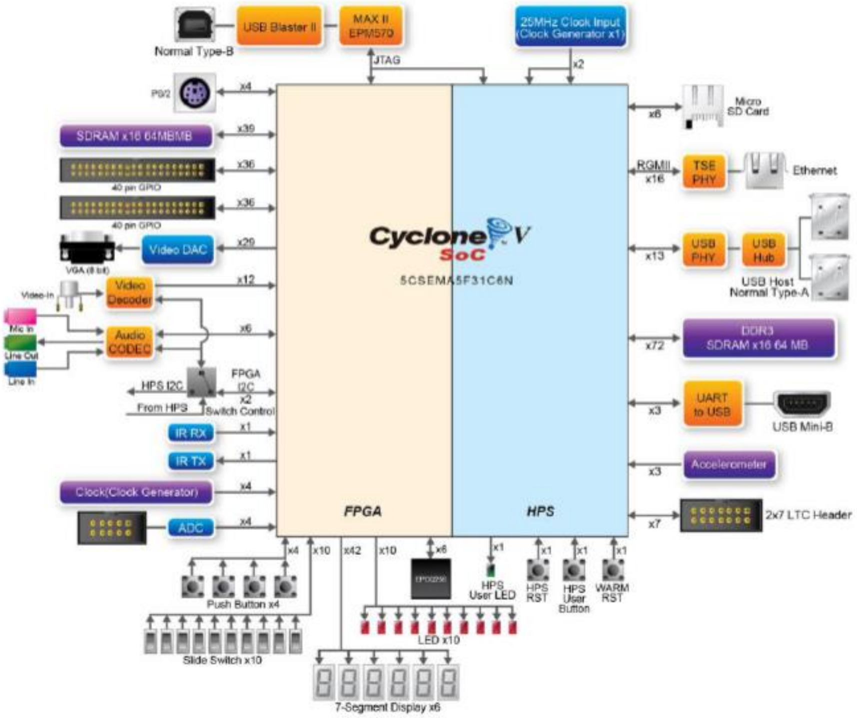


Fig. 3.3: Block diagram of DE1-SoC computer [211].

3.2.3 Nios-II Soft-core Processor

Nios-II processor has supported a 32-bit RISC processor with a Harvard architecture and it is supported by all Altera FPGAs, SoCs, and ASICs. Since the Nios-II processor is soft-core embedded core processor, it does not exist physically. The processor is virtually available on any Altera FPGA by using a SOPC of Altera FPGA Quartus II development tool. The Nios-II processor provides a flat register file, comprising of thirty-two 32-bit general purpose integer registers, and six 32-bit control registers. The SOPC builder provides three versions of the processor.

- (Nios II/f) represents the fast version designed for the high level of performance. The widest scope of configuration options is provided, and it can be used to optimize the processor for performance.

- (Nios II/s) is the standard version designed for the medium level of performance. This version requires fewer resources in an FPGA device as a trade-off for reduced performance.
- (Nios II/e) is the economy version that offers a low level of performance and the minimum amount of FPGA resources is required. The user-configurable features are limited.

Therefore, users can select the different types of processors, peripherals, memory and I/O interfaces for the processor system to meet the requirements of their own designs. Nios-II IDE which is the software development environment of Nios-II processor based on the GNU C/C++ compiler and Eclipse IDE. A Nios-II processor operates in three modes such as Supervisor mode, User mode, and Debug mode. The Supervisor mode enables the processor to carry out all instructions and complete all accessible functions. When the processor is reset, it enters the supervisor mode. The user mode is used to prohibit the performance of a few instructions that should be applied for system purposes only. Some processor features are not available in this mode. The debug mode is used by software debugging tools to apply characteristics such as watchpoints and breakpoints [210, 212– 213]. The Nios-II processor provides several advantages such as reducing the maintenance cost and components, and flexibility in choosing specific peripheral etc. [214].

3.3 Selection of Sensors

Since all WSN systems consist of a transmitter unit and receiver unit at the beginning of the overall design of a reconfigurable WQM system based on WSN, the transmitter unit of the system is planned to design. To detect the water quality such as water level, water pH, the water temperature in degree Celsius, the concentration of CO_2 on the surface of the water and turbidity of water, the relevant water parameter sensors must be used in the proposed WQM system. As power consumption is a critical issue in WSN systems, low power sensors should be considered in designing the proposed WQM system based on WSN. In addition, an effective and efficient system of WQM is an important implementation in addressing the issue of global water pollution, the costs and reliability of sensors are also important. To achieve the objectives of the proposed research work, the following sensors to detect the water parameters have been selected.

1. Ultrasonic sensor
2. Digital thermometer sensor
3. pH sensor
4. CO_2 sensor
5. Turbidity sensor

3.3.1 Ultrasonic Sensor

In the reconfigurable WSN design for the WQM system, the water level is one of the parameters to be monitored. The ultrasonic sensor is appropriate to detect uneven surface such as water since the sensor uses sound waves instead of light. Sonar or ultrasonic sensing is performed by transmitting a specific sound signal which will be partially reflected when it hits any objects, therefore the object or the device which transmits the signal will receive the reflected signal back as shown in Fig. 3.4. The sonar process applies the principle of propagation of acoustic energy at higher frequencies than normal hearing [215]. Since the duration between emitting and collecting sound waves is proportional to the distance of the object from the sensor, the distance from the transducer and the object is computed.

As sound travels at 343 ms^{-1} in the air [216], the distance between the emitting object and the receiving object can be estimated. There are two types of sonar such as active and passive sonar. The active sonar detects an object through the source object transmitting sonic signals into the water and then receives the reflection that is sent back to them [216]. The activity of an active sonar can be calculated using (3.1),

$$SNRAV = SL - 2\mathrm{TL} + \mathrm{TS} - (\mathrm{NL} - \mathrm{DI}) \tag{3.1}$$

where *SNRAV* is signal to noise ratio, *SL* is the source signal level intensity, *TL* is the

the target, *TS* is the sound reflectivity of the target equating how powerful the returning

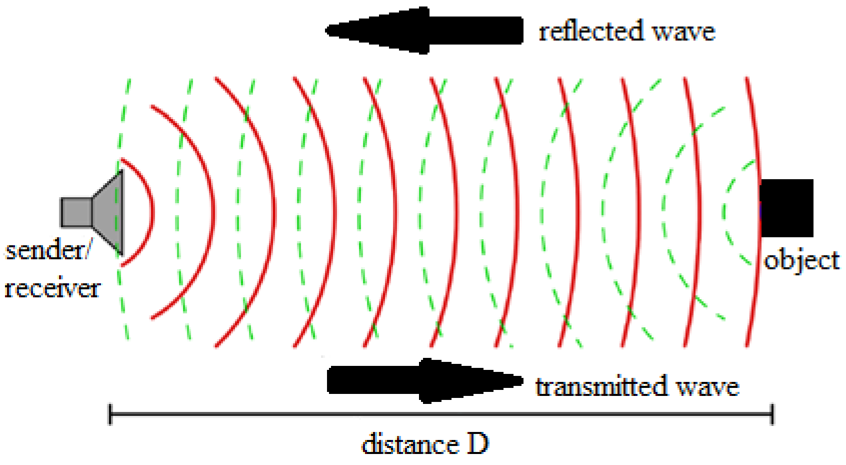


Fig. 3.4: Sonar Process Diagram

signal is, *NL* is the noise level being picked up by the sensors, *DI* is the directivity index which represents the amount of ambient background noise the sensors are rejecting. All the variables are measured in decibels. Passive sonar operates by detecting the sounds transmitted by other objects, such as the hum of engines in a submarine to find other underwater objects [217]. The mathematical equation of the passive sonar is shown in (3.2) in which the variables in decibels are the same variables as the active sonar equation [218].

$$SNRAV = SL - \mathrm{TL} - (\mathrm{NL} - \mathrm{DI}) \tag{3.2}$$

According to the method of the project, the active sonar should be utilized in the reconfigurable design of WSN in the WQM system.

3.3.1.1 Selection of Ultrasonic Sensor

Either one of the two sensors ultrasonic ranging from module HC - SR04 and ultrasonic LV-MaxSonar EZ1 sensor are considered to be selected for the reconfigurable WSN design for WQM in IoT environment. The ultrasonic ranging module HC - SR04 offers 0.02m - 4m non-contact measurement function. The ranging accuracy of HC - SR04 can achieve up to 0.003m. The HC - SR04 modules consist of ultrasonic transmitters, receive and control circuit. The operating voltage and current are 5V DC and 15mA respectively. It measures the object at 0.02m of minimum range and 4m of maximum range [219]. The LV-MaxSonar-EZ1 discovers objects from 0 – 6.45m and offers sonar range information from 0.152m out to 0.0645m with 0.025m resolution. It needs the power supply of 2.5V to 5.5V with 2mA typical current draw. The measurement readings can achieve up to every 50mS (20-Hz rate). The minimum sensing distance of the sensor is 0.152m. On the other hand, the LV-MaxSonar-EZ1 ranges and reports targets to the front sensor face. Large targets closer than 0.152m usually range from 0.152m [220]. For its benefits of low power consumption, easy to use, ability of detecting effective maximum range, balance between sensitivity and object rejection and flexibility of power sources, the ultrasonic LV-Maxsonar EZ1 is selected to design the proposed reconfigurable WSN for WQM in IoT environment to detect the distance from the water surface and the sensor. Table 3.2 shows the comparison between HC-SR04 [219] and LV-MAxSonar EZ1 [220].

Table 3.2: Comparison of features between HC-SR04 [219] and LV-MAxSonar EZ1 [220]

Specification	HC - SR04	LV-MaxSonar EZ1
Signal frequency	40kHz	42kHz
Factory calibrated unit	cm	Inch
Detection Range	2 – 400 cm	0 – 254 inches
Minimum operation distance	2 cm	5 inches
Output	Digital time base pulse corresponds to distance	Analog voltage, Digital Pulse Width, RS-232 Serial
Pulse width	uS / 58 cm	147uS/inch
Analog output voltage	NIL	9.8mV/inch
Operating voltage	+ 5VDC	2.5V – 5.5V
Quiescent current	< 2mA	2mA
Resolution	0.3 cm	1 mm
Dimension	45mm x 50 mm x 15mm	19.9mm x 22.1mmx 15.5mm

3.3.1.2 LV-MaxSonar-EZ1

The ultrasonic sensor functions by releasing a high-frequency sonic wave at systematic time period commencing from the front of the transducer. The sonic waves are reflected by an object and collected back in the transducer. The time between when the signal is transmitted and received is determined. As the sound wave is used by the ultrasonic sensor, it is more capable than a light wave in sensing irregular surface such as water surface. According to its data sheet [220], the LV-MaxSonar-EZ1 shown in Fig. 3.5 [220] detects objects from 0- 6.452m and offers sonar range information from 0.152m out to 6.452m with 0.0254m resolution. The time period between releasing and collecting sound waves is proportionate to the distance between the object and the sensor. The range of the sensor is measured from the front of the transducer. The targets closer than 0.152m from the sensor will typically range as 0.152m as the minimum distance detected by the sensor is 0.152m.

3.3.2 Digital Thermometer Sensor

In the project, the temperature thermometer sensor will be used to monitor the water temperature. There are a few numbers of the digital thermometer on the market. The selection of suitable thermometer for the project has been made after analyzing and comparing the different brand of thermometers as follows:

1. Analog temperature sensors TMP35/TMP36/TMP37
2. Precision temperature sensors LM135, LM235, LM335
3. The 1-Wire protocol temperature sensor

3.3.2.1 Precision Temperature Sensors LM135, LM235, LM335

LM135 series are easily calibrated precision Analog Output temperature sensors. They operate as a 2-terminal Zener, and the breakdown voltage is directly proportionate to the absolute temperature at 10mV/°K. The circuit operating current is 450µA to 5mA without characteristics alteration. The sensors are adjusted at +25°C, and a typical error is less than 1°C over a 100°C temperature range. These sensors provide a linear output [221].

Fig. 3.5: Ultrasonic Sensor LV-MaxSonar- EZ1 [220]

3.3.2.2 1-Wire Protocol Temperature Sensor

1-Wire protocol is a powerful technique to collaborate sensors and analog-to-digital converters (ADCs) to microprocessor [222]. All 1-Wire devices are incorporated together by a single wire that operates as a physical medium to interconnect between a host and several devices connected to the wire. Since the 1-Wire technology is constructed on a serial protocol which utilizes a single data line and one ground reference for communication, these devices can be powered up through the same wire. A master of 1-Wire controls the communication with one or more 1-Wire slave devices on the 1-Wire bus. The 1-Wire slave devices can be powered by a voltage range of 2.8V (minimum) to 5.25V (maximum) directly from the 1-Wire bus parasitic supply without external power supply.

The Dallas/Maxim Semiconductor 1-Wire bus is a simple signaling system that executes half-duplex bidirectional communications between a master controller and one or more slaves sharing a common data line for both power and data communication for slave devices. The slaves store charge on an internal capacitor when the line is in a high state and this charge is used for the device operation when the line is low during data transmission. A common 1-Wire master comprises of an open-drain I/O port pin with a pull-up resistor with a 3V–5V supply [223]. This effective communication scheme also enables adding of memory, authentication, and mixed-signal functions at any time. As the sensor is encapsulated in high-quality stainless steel tube and it is moisture proof to prevent from the rust, Dallas product 1-Wire digital thermometer DS18B20 [224] is selected to monitor the water temperature in the proposed reconfigurable WSN design for WQM in IoT environment.

The DS18B20 dispenses 9-bit to 12-bit Celsius temperature measurements. The DS18B20 involves only one port pin (and ground pin) for communication with a central microprocessor. A unique 64-bit serial code of the DS18B20 enables multiple DS18B20s to function on the same 1-Wire bus. The DS18B20 can be powered either straight from the data line or the 3.0V – 5.5V of the power supply. One of the benefits of the DS18B20 is the accuracy of ± 0.5°C from -10°C to +85°C. The temperature is converted to the 12-bit digital word in a maximum of 750 milliseconds [224]. The DS18B20 utilizes bus communication which is implemented by Maxim's exclusive 1-Wire bus protocol using one control signal. The DS18B20 is always a slave and the system is mentioned as a “single-drop” system when there is only one slave on the bus. Similarly, the system is referred to as “multi-drop” if there are multiple slaves on the bus. The system of the 1-Wire bus is divided into three sections as follows:

1. The hardware configuration: The 1-Wire bus of the DS18B20 requires a 5kΩ external pull-up resistor to be high in the idle state for the 1-Wire bus. The bus needs to be in the idle state if the transaction is to restart or if the suspension of the transaction is required for many reasons. Since the 1-Wire bus is in an inactive (high) state during the recovery period, the infinite recovery time can occur between bits so long. If the bus is held low for more than 480μs, all components on the bus will be reset.

2. The transaction sequences: It is essential to follow the transaction sequence such as Step 1: Initialization, Step 2: ROM Command (followed by any required data exchange), and Step 3: DS18B20 Function Command (followed by any required data exchange) when the DS18B20 is operated.

3. 1-Wire signaling (signal types and timing): The DS18B20 utilizes a precise 1-Wire communication protocol to ensure data integrity. This protocol describes several signal types such as reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All these signals are initiated by the bus master except for the presence pulse.

3.3.3 pH Sensor

In the proposed reconfigurable WSN design in the IoT environment for the WQM system, the pH sensor which can provide reliable accuracy of measurements should be selected to observe the pH value of water. The pH is a measurement of the hydrogen ion concentration [H^+], or the measurement of the acidity or alkalinity of a water solution. The acidic or alkaline properties of a water solution are controlled by the presence of a relative number of hydrogen ions (H^+) or hydroxyl ions (OH^-) as acidic water solution has a higher relative number of hydrogen ions, while an alkaline or a basic solution has a higher relative number of hydroxyl ions. The pH value of every water solution ranges from pH 0 to 14. At 25°C, the pH value 7 is the centre of the measurement scale or neutral since it is neither acidic nor basic properties.

The value below 7 shows acidic properties of water solution and the value above 7 demonstrates basic or alkaline properties of water solution. The mathematical formula of pH is the negative logarithm of the hydrogen ion concentration [225]. It can be stated mathematically as follows:

$$pH = -\log 10\ [H^+] \tag{3.4}$$

where [H^+] is hydrogen ion concentration in mol/L.

3.3.3.1 pH Measuring Technology

The pH is measured using a pH-sensitive electrode which is generally a glass electrode and a reference electrode. The pH electrode is a sensor covered by a glass tube which is sensitive to the hydrogen ions and examines the reaction between it and a sample solution, in which the pH value is identified. Since the pH-sensitive electrode alone cannot provide adequate information on pH, a second sensor electrode is essential [226]. The second sensor electrode is the reference electrode which provides the reference signals or potential for the pH sensor.

3.3.3.2 pH Electrode

The pH electrode is made of glass and it includes an inert glass tube with a pH sensitive glass tip which is sensitive to H^+ ions. The tip of the electrode is filled with an inner aqueous electrolyte solution which produces a potential (voltage) relative to the pH of the solution [227].

3.3.3.3 pH Reference Electrode

Generally, a reference electrode comprises of a silver wire coated with silver chloride in a solution of potassium chloride to resume a reproducible concentration of silver ions in the filled solution, and a reproducible potential (voltage) is resulted on the silver-silver chloride wire [227]. The reference electrode is designed to sustain a constant potential at any given temperature. It functions to finish the circuit of pH computed within the solution. A known reference potential for the pH electrode is yield by the reference electrode. The potential difference between the pH electrode and the reference electrode produces a millivolt signal proportionate to pH.

3.3.3.4 Selection of pH Sensor

There are different brands of pH sensors in the market. The Wedgewood Analytical's product Campbell Scientific CSIM11-L pH sensor [228] can be submerged in solution or inserted directly into tanks, pipeline and open channels to measure the pH of many ranges and decreases signal interference. The porous junctions of the pH probe are made of Teflon solutions. The pH probe includes an internal amplifier to boost the signal which is less susceptible to clogging. The probes can be mounted at any angle for its plunger-style. The probe can measure the full range of pH, with a 95% accuracy rating. It can be operated from 1°C to 80°C and is internally powered by two 3V Lithium batteries. Atlas Scientific pH sensor is a silver-silver chloride probe which can measure the full range of pH of any solution from pH 0 – 14. It can be operated from 1° to 99°C and has a response time of about 1 second and two decimal places of pH value is measured. The pH kit includes a pH probe, a BNC connector and an EZO pH circuit [229]. The user is allowed to design and configure without any additional electronic components to measure the accurate pH value. Since the accuracy of measurement is important, the Atlas Scientific pH kit is selected to design the proposed reconfigurable WQM system in IoT environment for its low operating DC voltage probe and its accurate two decimal places of pH readings. The pH sensor can be interfaced simply in breadboard circuits. The specifications of the Campbell Scientific CSIM11-L pH sensor [228] and Atlas Scientific pH sensor [229] are listed in Table 3.3.

3.3.3.5 Atlas Scientific pH Kit

Atlas Scientific pH kit consists of three major components such as EZO TM class embedded pH circuit, BNC shield, and pH probe. For collecting pH level, the pH probe is attached to the BNC shield, the BNC shield then conveys the pH probe detecting pH level to the embedded pH circuit. The detected pH level is transformed into binary by the embedded pH circuit, and the resulted pH data is transmitted to the processor. The Atlas Scientific pH probe measures the activity of hydrogen ions in water or solution. The point of the pH probe is a glass membrane which allows hydrogen ions from the water to be detected and to deactivate into the outer layer of the glass, whilst greater ions remain in the solution. A very small current is generated by the variance in the concentration of hydrogen ions outside and inside of the probe. The concentration of the measured hydrogen ions in the water is proportionate to the produced current [229].

Table 3.3: Specifications of Campbell Scientific pH sensor [228] and Atlas Scientific pH sensor [229]

Specification	**CSIM11-L**	**Atlas Scientific**
pH range	0 to 14	0 − 14
Operating Temperature Range	0° to 80°C	1 − 99 °C
Pressure Range	0 to 30 psig	0- 100 PSI
Accuracy	±0.1% (over full range)	±0.02
Response Time	95% of reading (in 10 s)	95% in 1s

3.3.4 CO_2 Sensor

In the atmosphere, the concentration of about 400 ppm of CO_2 is commonly present [230]. One ppm (parts per million) is equivalent to 1 milligram of something per liter of water (mg/l) or 1 milligram of something per kilogram soil (mg/kg) [231]. The pH of the water may rise above 8.3 during the daylight hours due to the process of photosynthesis which uses CO_2 in the surface layer of water [232]. Currently, the non-dispersive infrared (NDIR) CO_2 sensor is the usual type of sensor to observe the concentration of CO_2 [233]. The infrared (IR) light source is located at one end of a tube filled with air. An IR light shines through a tube towards an IR light detector which is hit by IR light. The IR light detector evaluates the amount of received IR light.

The IR light alone is absorbed by any gas molecules that is same in size as the wavelength of IR light when the IR light traverses the tube. Therefore, the other wavelengths of light pass through and hit the detector. The amount of IR radiation absorbed by CO_2 molecules is monitored. The variance between the intensity of light emitted by the IR lamp and the intensity of IR light collected by the detector is proportionate to the number of CO_2 molecules inside the tube [234].

3.3.4.1 Selection of CO_2 Sensor

To detect the CO_2 level on the water surface in the proposed wireless WQM system, the CO_2 sensor MG-811, and Analog Infrared CO_2 Sensor SKU: SEN0219 are considered.

3.3.4.2 CO_2 Sensor MG-811

The CO_2 sensor MG-811 [235] has high sensitivity to CO_2 as it can detect the concentration of CO_2 in the range of 350ppm –10000ppm (parts per million). Chemical reactions take place in the cell when the sensor is exposed to CO_2 gas, and then an electromotive force is produced. The output voltage of the sensor in clean air is in the range of 200mV– 600mV, the amplifier is used to amplify the output data to improve the accuracy of the measured data. The output voltage decreases when the CO_2 concentration increases. The response time of the MG-811 is less than 60 seconds. The output voltage of 0V – 3.3 V is corresponding to 0 – 10000 ppm. It has quick responding features after preheating as MG-811 CO_2 sensor module needs to be preheated for 24hours. After 6 hours operating the sensor, 1–2 hours of preheating is needed again. The power must be off for above 72 hours and preheating for 24 hours. The accuracy of MG-811 can be interfered by humidity [236].

3.3.4.3 Gravity: Analog Infrared CO_2 Sensor SKU: SEN0219

Gravity: Analog Infrared CO_2 Sensor SKU: SEN0219 is based on NDIR technology and it measures the CO_2 range of 0 – 5000ppm with analog output. The characteristics of SEN0219 are high sensitivity, waterproof and anti-corrosion, temperature compensation, low power consumption, stability, linear output, no poisoning, high cycle life, and anti-water vapour interference. The operating voltage is 4.5V – 5.5V DC, the average current is <60mA at 5V and the peak current is 150mA at 5V respectively. The accuracy of the CO_2 sensor is ± (50ppm + 3% reading) [237]. Gravity: Analog Infrared CO_2 Sensor SKU: SEN0219 shown in Fig. 3.6 [237] is selected to measure the concentration of CO_2 on the water surface since CO_2 sensor MG-811 is not preferred for its inconvenient preheating time. The comparison of specification between MG-811 [235] and SKU: SEN0219 [237] is presented in Table 3.4.

Table 3.4: Specifications comparison between MG-811 [235] and SKU: SEN0219 [237]

Specification	MG-811	SKU: SEN0219
Detection range	400 –10,000 PPM	0 – 5000ppm
Operating Voltage	5 V	4.5 – 5.5V DC
Operating Temperature	–20 – 50℃	0 – 50 ℃
Operating Current	200 ± 10mA	<60mA at 5V
Response Time	> 60 s	120s
Preheating Time	5 – 20 min	3 min
Output	Analog, Digital output 100 – 600mV	Analog (0.4 – 2V)

3.3.5 Turbidity Sensor

In this project, the turbidity sensor with effectiveness and accuracy should be selected to observe water quality by detecting the level of turbidity. The turbidity of the water is caused by suspended materials which absorb and scatter light by the inorganic particles and suspended organic (including algae). Turbidity is a unit of measurement quantifying the degree of water and is usually measured in Nephelometric Turbidity Units (NTU) [238].

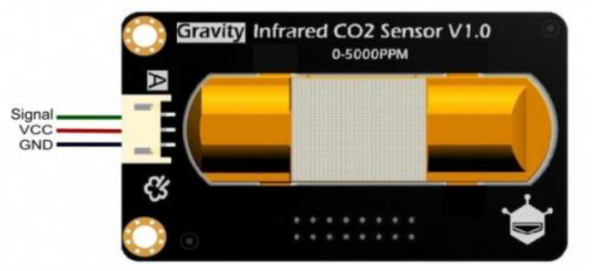


Fig. 3.6: Gravity: Analog Infrared CO_2 Sensor SKU: SEN0219 [237]

Turbidity is measured either in a laboratory or on the field using specialized optical equipment. A light is passed through a water sample, and the amount of light scattered is measured. The amount of scattering of light is proportionate to the turbidity [239].

3.3.5.1 Selection of Turbidity Sensor

The turbidity sensor, GLOBAL WATER WQ 770 [240], is a submergible turbidity meter in which the detecting sensor is attached in a rugged anodized cover and entirely enclosed to use on-site environmental monitoring. The sensor probe includes a light emitting diode (LED) as a light source and optical lenses that can focus light into a dark field sample region. The sensor transmits a focused light into the water, and the light is then reflected off in the water and is detected by the turbidity sensor’s photo-detector which is positioned at 90 degrees to the light beam since the light intensity sensed by the turbidity sensor is proportionate directly the turbidity of the water. From this relation, the turbidity of water can be calculated.

Turbidity sensor SKU: SEN0189 observes the suspended particles in water by calculating the rate of transmitting and scattering of the light since the amount of total suspended solids (TSS) in water depends on the light transmittance and scattering rate. When the TTS increases, the turbidity of water increases. The analog and digital signal output modes are available, and either mode is set depending on the MCU [241]. The potentiometer in the digital signal mode can be adjusted to control the threshold. The operating voltage and current of the turbidity sensor are 5V DC and 40mA (maximum) respectively. For the mapping from the output voltage to the NTU according to a different temperature, when the sensor is placed in pure water, the value of NTU is less than 0.5; the output should be “4.1±0.3V” when the temperature is 10°C – 50°C [238]. For the proposed reconfigurable WSN design for WQM in IoT environment, the WQ 770 is not a preferred selection in measuring the turbidity of water due to its high cost of USD 351.50 [240] while SKU: SEN0189 costs USD 13.78 [241].

3.4 Selection of Wireless Protocol

Despite the availability of many wireless applications for both industrial and personal usages that involve low data rates, reliability, security, low cost, less complexity, and longer battery life, the ZigBee technology has the performance characteristics that practically fulfill those requirements. The ZigBee standard is aimed to function in an unlicensed international frequency band. Since power consumption is a critical issue in WSNs, the Zigbee protocol is commonly used in existing WSN based WQM systems for communication protocol. It has been discussed in Chapter 2 that several studies present the benefits of ZigBee in WSN based WQM systems.

3.4.1 Selection of ZigBee Module

Two different brands of ZigBee module such as Telegesis ETRX3 series modules and XBee Pro module are considered.

3.4.1.1 Telegesis ETRX3 Series Modules

The Telegesis ETRX3 series modules [242] are constructed on the Ember ZigBee platform including the single chip EM351 or EM357 integrated with the ZigBee PRO compliant EmberZNet meshing stack. It is easy to use in many areas for various applications using a simple AT-style command interface and advanced hardware design. The configurable functionality of the Telegesis AT Command set commonly enables the ETRX3 series ZigBee modules to be utilized without a supplementary host microcontroller to save integration time and costs. The supply voltage VCC 0.3 to +3.6 V DC is required and the voltage on any Pad Vin -0.3 to VCC +0.3 V DC is required. Currently, the existing firmware in the market does not support the SPI and I2C buses, but it can be used with custom firmware.

3.4.1.2 XBee Module

XBee Pro S1 (IEEE 802.15.4 standard) is a registered brand name of Zigbee Standard and it is a product of the Digi International. The XBee 802.15.4 RF modules consist of Zigbee/Mesh topologies, and it can support both 2.4 GHz and 900 MHz frequency [243]. The XBee 802.15.4 (IEEE 802.15.4 standard) RF module comprises one XBee USB adapter, two Embedded-Antenna Module and one voltage adapter. The voltage adapter converts 5V to 3.3V. Since two XBee Series Embedded-Antenna Modules operate in universal asynchronous receiver/transmitter (UART) mode, XBee Series Embedded-Antenna Modules are the core units to design the wireless communication between the observing device and the FPGA board. This module proposes competence for long distance communication.

The digital input/output pins and additional pins of XBee can assist XBee standalone applications. The 10-bit pulse width modulation (PWM) of XBee modulates output and communicates to another XBee. The XBee Pro S1 modules are shown in Fig. 3.7 [243]. The XBee Pro series support both points to point and multi-point networks. The coverage of XBee Pro is outdoor line-of-sight up to 1600m for international variant. The configuration is not essential for out-of-box RF communications and the transmission power is 10mW (10dBm) [243]. Therefore, in the reconfigurable WSN design for WQM in IoT environment project, to transmit and acquire the data between the observing PC and FPGA board, two XBee Pro S1 (IEEE 802.15.4 standard) RF modules based on ZigBee wireless communication protocol are used for its benefits of ease of usage, lower power consumption, low cost and its ability to using with SPI and I2C buses.

3.5 Selection of Analog to Digital Converter

Since the turbidity sensor SKU: SEN0189 and CO_2 sensor SKU: SEN0219 are analog sensors, the analog output of those sensors is needed to convert to digital output. An analog to digital converter (ADC) with an effective and reliable feature should be selected for the proposed design. Among a large number of ADCs, a high-speed ADC

Fig. 3.7: XBee Pro S1 (IEEE 802.15.4 standard) RF Module [243]

AD7928 [244] with a fast conversion rate of 2.5µs per channel is selected. The analog voltage output is converted to 8 bits binary by 8 bits 8 channels AD7928 that are capable to handle maximum 8 analog voltage inputs in series. The signal interface of the turbidity sensor, the CO_2 sensor and the Serial Peripheral Interface (SPI) of ADC will be devised in the Nios-II processor in the Qsys Tool of Quartus II software.

3.6 Selection of Display Tools for Receiving Water Data

The data will be displayed on the monitoring PC when the wirelessly transmitted water parameter from the transmission unit is received. In the proposed reconfigurable WSN design for environmental monitoring in the IoT environment, both X-CTU software and Grafana software are selected to display the water parameters data.

3.6.1 X-CTU Software

X-CTU software [245] is manufactured by Digi Internationals and it can be used to configure the XBee. The XCTU supports configuration and communication for most Digi RF modules. Since the XBee Pro RF Module is one of the XCTU-compatible RF modules, the X-CTU software transmits data from PC to another side of XBee, and the monitored data transmitted from another side of XBee to PC Com port. The X-CTU transmits and receives the data in ASCII character and Hexadecimal.

3.6.2 Grafana Software

To visualize time series data of the parameters of water, Grafana [246] is an appropriate software for its easy installation on the monitoring PC operated in the Linux mode, dashboards, rapid and flexible client-side graphs with a multitude of options. The Grafana dashboards offer the following features [247]:

- Drag and drop panels, change row and panel width easily
- Search dashboards based on title or tags
- Templated dashboards
- Dashboard playlists
- Create / Update HTTP API
- Different panel types
- Supports panel plugins

For graphing, the Grafana allows the users as follows:

- Click and select a region to zoom
- Multiple Y-Axes, Render Bars, Lines, Points
- Thresholds, Logarithmic Scales
- Y-axis formats (bytes, milliseconds, etc)
- View or edit graph in fullscreen
- Full control over how each series should be drawn

3.7 Design Plan of Hardware and Software

After selecting hardware components, the draft design of WQM system is planned as shown in Fig. 3.8. The detail of the design is discussed in Chapter 4: System Implementation. The system software flow chart of the system is drafted as depicted in Fig. 3.9. The detail of the software system flow chart is discussed in the latter part of Chapter 4.

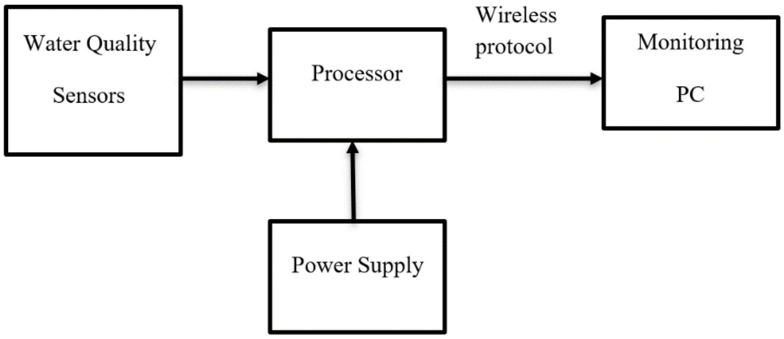


Fig. 3.8: Draft Design of WQM System in IoT environment

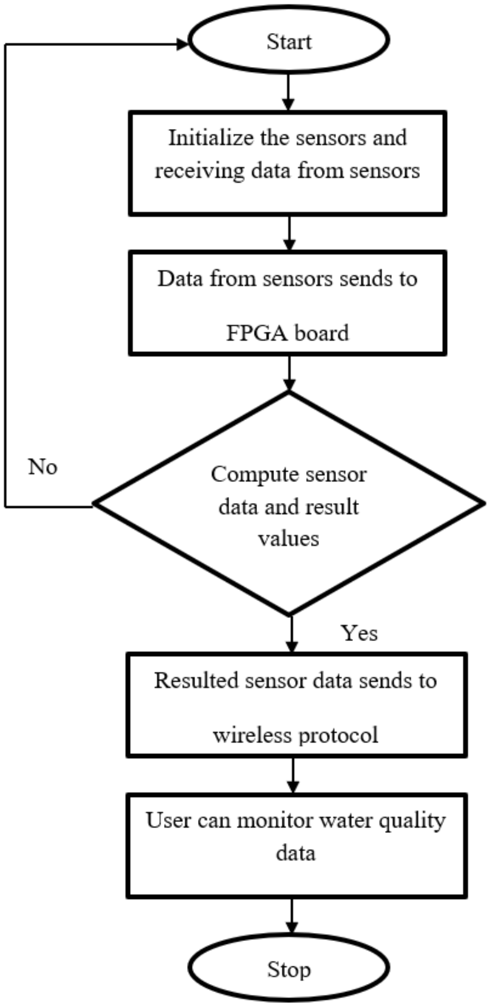

Start
Initialize the sensors and receiving data from sensors
Data from sensors sends to FPGA board
No
Compute sensor data and result values
Yes
Resulted sensor data sends to wireless protocol
User can monitor water quality data
Stop

Fig. 3.9: Draft Software Flow Chart WQM system in IoT environment

3.8 Summary

In this chapter, the challenges to meet the requirements of the WQM system have been discussed. The design plan of the experimental setup of the reconfigurable WSN design for environmental monitoring in the IoT environment has been presented. The selection of hardware and software components to implement the WSN based WQM system in IoT has also been discussed. To measure the water parameters, DS18B20 temperature sensor, Ultrasonic sensor LV-MaxSonar- EZ1, Atlas Scientific pH kit, Gravity: Analog Infrared CO_2 sensor SKU: SEN0219 and Turbidity sensor SKU: SEN0189 have been selected. For the processor, the Altera/Intel DE1-SoC 5CSEMA5F31C6 FPGA board has been selected and XBee Pro S1 has been selected for the wireless protocol. To display the real-time data of water parameters, the X-CTU software, and the Grafana dashboard software has been selected.

Chapter 4

System Implementation

The selections of hardware and software components for the reconfigurable WSN design for environmental monitoring in the IoT environment was discussed in Chapter 3. Based on the discussion of hardware and software components in Chapter 3, the reconfigurable WSN design for WQM in the IoT environment will be designed and built. The design in Qsys Tool of Nois- II processor will be presented in this chapter. The aspects of the implementations of both hardware and software will be studied in their specific sections. The results of the designed system will be analyzed in Chapter 5.

4.1 Design of Reconfigurable WSN System for WQM in IoT Environment

Initially, the block diagram of reconfigurable WSN system is designed as shown in Fig. 4.1. The system consists of two parts: the transmitter unit, which consists of the sensors, an FPGA board, and a wireless transmitter and the receiver module, which consists of only a wireless receiver and PC. At the beginning of the overall design of reconfigurable WSN system, the transmitter module of the system's design is planned. When the design of the transmitter unit of the reconfigurable WSN system for WQM in the IoT environment is considered, the following core components are planned and listed in the initial phase.

1. Sensors
2. FPGA Board
3. Wireless Transmitter Unit
4. Wireless Receiver Unit

Firstly, the components for the proposed design of reconfigurable WSN for WQM in IoT environment such as sensors, ADC, and XBee wireless communication modules are designed using Nios-II soft-core processor in Qsys Tools of Quartus II. Then, the complete design will be downloaded to the FPGA SoC board. The calculations of water

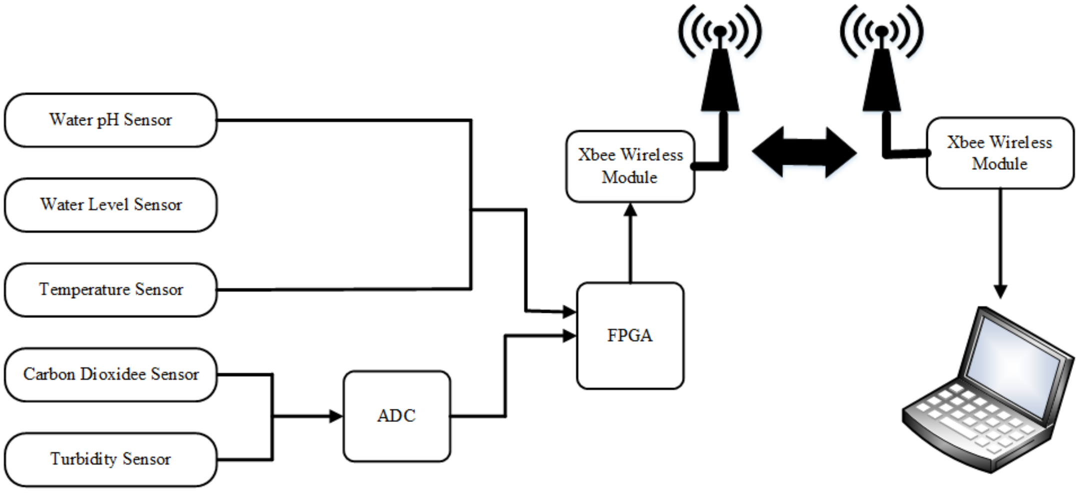


Fig. 4.1: Block Diagram of Reconfigurable WSN Design for WQM in IoT

parameters and commands of wireless transceiver modules are programmed in Nios-II Eclipse by applying C programming language. Then, the program will be compiled and downloaded to the FPGA board using Quartus II software.

4.2 Integrating of sockit_owm Component into Nios II Software Development Environment

At the beginning of the experimental design, the Digital Thermometer Sensor DS18B20 design is considered, and sockit_owm (1-wire master) module must be integrated into Nios II software development environment since DS18B20 is a 1-wire protocol-based sensor. To implement the physical layer of the protocol in hardware, the soft-core processor of Quartus II also needs to be integrated into Nios-II.

4.2.1 Parameters of sockit_owm

The default width of the data bus is 32bit. The base is accurately 7.5μs or 5.0μs, or in the range of 6.0 – 7.5μs for normal mode. For overdrive mode, the base is either 1.0μs or in the range of 0.5 – 0.66μs. The clock frequency of the system can also be sorted into the required base period. When the overdrive is empowered, both normal and overdrive mode periods are achieved. Exact time periods provide a better-optimized execution. The clock divider ratio parameters (register values) are calculated using the following formulae.

$$\text{CDR\_N} = f_{\text{clk}} \times 5.0\mu\text{s} - 1 \tag{4.1}$$

for example: $$\text{CDR\_N} = 1\text{MHz} \times 5.0\mu\text{s} - 1 = 5 - 1$$

$$\text{CDR\_O} = f_{\text{clk}} \times 1.0\mu\text{s} - 1 \tag{4.2}$$

for example: $$\text{CDR\_O} = 1\text{MHz} \times 1.0\mu\text{s} - 1 = 1 - 1$$

When the dividing factor is not a rounding integer, the controller timing is moderately off, and it supports solely a subdivision of 1-wire devices with its timing closer to the typical 30µs slot. If a round frequency is not feasible, range periods should be used. Base time periods BTP_N = “5.0” and BTP_O = “1.0” are optimized for one wire timing. The default timing restricts the range of available frequencies to multiples of 1MHz. If even this restriction is too strict, then timing BTP_N = “6.0” and BTP_O = “0.5” are used, where the actual periods can be in the range as follows-:

$$6.0\mu\text{s} <= \text{BTP\_N} <= 7.5\mu\text{s}$$

$$0.5\mu\text{s} <= \text{BTP\_O} <= 0.66\mu\text{s}$$

A third timing option is available for normal mode BTP_N = “7.5”, this option is optimized for logic size. The exact base time period is calculated using the formula as follows:

$$\text{BTP\_N} = \frac{(CDR\_N+1)}{f_{\text{clk}}} \tag{4.3}$$

The base time period and base frequency options are presented in Table 4.1 [224].

Table 4.1: Base Time Period and Base Frequency [224]

overdrive support (OVD_E)	disabled			Enabled		
base time period [µs] (normal mode)	7.5	5	6.0-7.5	7.5	5	6.0-7.5
base time period [µs] (overdrive mode)				1.0	1	0.5-0.66
base frequency (Mhz)	0.133	0.2	0.166-0.133			2-1.5

4.2.2 Module Ports

Module ports of sockit_owm are categorized into two ports such as 1-wire interface ports and CPU interface ports.

4.2.2.1 CPU bus Interface

The default bus interface is compatible with 32bit Avalon MM, and it is connected to Wishbone. Table 4.2 [224] shows that Wishbone is equivalent to the Avalon MM signals. All CPU related signals are actively high, and both read and write cycles are a single clock period long, the omitted Avalon MM signal bus_waitrequest is a constant "0". Read data is only valid when the first cycle of bus_ren is active in which this can change in the next cycle due to the interrupted status that has been cleared. When a reset (with presence detection) or a read/write bit 1-wire cycle is finished, this will be indicated through the interrupt signal. The interrupt is cleared after reading the control/status register.

4.2.2.2 1-wire Interface

An external open-drain driver is necessitated by the 1-wire module to enable the operation of a bidirectional pin. The schematic diagram of the 1-wire driver for the i-th line is described in Fig. 4.2 [226].

Table 4.2: Wishbone Equivalents of Avalon MM Signals [224]

bus signal	**Avalon MM signal**	**Wishbone signal**
bus_wen	avalon_write	cyc & stb & we
bus_ren	avalon_read	cyc & stb & ~we
bus_adr	avalon_address	Adr
bus_wdt	avalon_writedata	dat_w
bus_rdt	avalon_readdata	dat_r
bus_irq	avalon_interrupt	Irq

4.2.3 Interfacing Processor

The interfacing processor consists of 1.) Control status register, 2.) Enabling power, 3.) Clock divider ratio registers, and 4.) Driver access sequences (polling and interrupts).

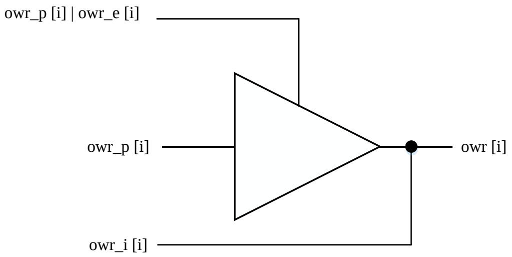


Fig. 4.2: Schematic Diagram of 1-wire Driver for the i-th Line [225]

4.2.3.1 Control Status Register

The sockit_own module is designed for a 32bit CPU through a single 32bit register if 1 to 16 owm 1-wire lines are required to design. The control status register is defined as the following C program.

```
#ifndef __SOCKIT_OWM_REGS_H__
#define __SOCKIT_OWM_REGS_H__
#include <io.h>
#define SOCKIT_OWM_CTL_REG              0
#define      IOADDR_SOCKIT_OWM_CTL (base)
IO_CALC_ADDRESS_NATIVE (base, SOCKIT_OWM_CTL_REG)
#define      IORD_SOCKIT_OWM_CTL      (base)                    IORD      (base,
SOCKIT_OWM_CTL_REG)
#define IOWR_SOCKIT_OWM_CTL (base, data)
  IOWR (base, SOCKIT_OWM_CTL_REG, data)
#define SOCKIT_OWM_CTL_DAT_MSK         (0x00000001) // data bit
#define SOCKIT_OWM_CTL_DAT_OFST        (0)
#define SOCKIT_OWM_CTL_RST_MSK         (0x00000002) // reset
#define SOCKIT_OWM_CTL_RST_OFST        (1)
#define SOCKIT_OWM_CTL_OVD_MSK         (0x00000004) // overdrive
#define SOCKIT_OWM_CTL_OVD_OFST        (2)
#define SOCKIT_OWM_CTL_CYC_MSK         (0x00000008) // cycle
#define SOCKIT_OWM_CTL_CYC_OFST        (3)
```

4.2.3.2 Enabling Power

More than one 1-wire line can be implemented in the sockit_owr module contingent upon the value of the OWM parameter. For the proposed reconfigurable WSN design for WQM in the IoT environment, only one 1-wire line is necessary. As only the single state counter is available in the module, parallel interaction with more than one device cannot be achieved. A particular synchronous reset or write/read bit cycle is offered. However, the time multiplexer can be implemented between lines on a bit transfer level. Hence, the completion of each cycle is essential, therefore, a new cycle on the same or different line can be provided. The following C codes are used for 1-wire line power enabling. All 1-wire lines can be powered independently.

```
#define SOCKIT_OWM_CTL_PWR_MSK      (0x00000010) // power (strong pull-up),
if there is a single 1-wire line
#define SOCKIT_OWM_CTL_PWR_OFST     (5)
#define SOCKIT_OWM_CTL_RSV_MSK      (0x00000020) // reserved
#define SOCKIT_OWM_CTL_RSV_OFST     (5)
#define SOCKIT_OWM_CTL_IRQ_MSK      (0x00000040) // irq status
#define SOCKIT_OWM_CTL_IRQ_OFST     (6)
#define SOCKIT_OWM_CTL_IEN_MSK      (0x00000080) // irq enable
#define SOCKIT_OWM_CTL_IEN_OFST     (7)
#define SOCKIT_OWM_CTL_SEL_MSK      (0x00000f00) // port select number
#define SOCKIT_OWM_CTL_SEL_OFST     (8)
#define SOCKIT_OWM_CTL_POWER_MSK (0xffff0000) // power (strong pull-up), if
there is more than one 1-wire line, 1-wire line status read multiplexer
#define SOCKIT_OWM_CTL_POWER_OFST (16)
```

4.2.3.3 Clock Divider Ratio Registers

When the clock divider ratio that registers cdr_n and cdr_o are executed (CDR_E=1), the reset value of the registers is determined by CDR_N and CDR_O parameters. The registers are written with new divider ratios while there is no 1-wire cycle in progress. The clock divider ratio registers are defined by C code as follows:

```
#define   SOCKIT_OWM_CDR_REG          1
#define   IOADDR_SOCKIT_OWM_CDR (base)
```

```
IO_CALC_ADDRESS_NATIVE (base, SOCKIT_OWM_CDR_REG)
#define    IORD_SOCKIT_OWM_CDR (base)
IORD (base, SOCKIT_OWM_CDR_REG)
#define IOWR_SOCKIT_OWM_CDR (base, data)
IOWR (base, SOCKIT_OWM_CDR_REG, data)
#define SOCKIT_OWM_CDR_N_MSK (0x0000ffff) // normal    mode
#define SOCKIT_OWM_CDR_N_OFST          (0)
#define SOCKIT_OWM_CDR_O_MSK            (0xffff0000) // overdrive mode
#define SOCKIT_OWM_CDR_O_OFST           (16)
#endif /* __SOCKIT_OWM_REGS_H__ */
```

4.2.3.4 Driver Access Sequences (Polling and Interrupts)

The 1-wire master performs the majority of the actions specified in the iButton specification as the physical layer. Polling or interrupts can be used. The reference implementation for Nios- II HAL and µC/OS-II is accessible as the source code. The C codes for polling and interrupts are as follows:

```
#include "sys/alt_dev.h"
#include "sys/alt_irq.h"
#include "sys/ioctl.h"
#include "sys/alt_errno.h"
#include "sockit_owm_regs.h"
#include "sockit_owm.h"
extern sockit_owm_state sockit_owm;
#ifndef SOCKIT_OWM_POLLING
#ifdef ALT_ENHANCED_INTERRUPT_API_PRESENT
static void sockit_owm_irq ();
#else
static void sockit_owm_irq (alt_u32 id);
#endif
void sockit_owm_init (alt_u32 irq)
{
  int error;
  // initialize semaphore for 1-wire cycle locking
  error = ALT_FLAG_CREATE (sockit_owm.irq, 0) ||
          ALT_SEM_CREATE (sockit_owm.cyc, 1);
  if (!error) {
   // enable interrupt
    sockit_owm.ien = 0x1;
    // register the interrupt handler
#ifdef ALT_ENHANCED_INTERRUPT_API_PRESENT
alt_ic_isr_register (0, irq, sockit_owm_irq, NULL, 0x0);
#else
    alt_irq_register (irq, NULL, sockit_owm_irq);
#endif
  }
}
#ifdef ALT_ENHANCED_INTERRUPT_API_PRESENT
static void sockit_owm_irq(void * state)
#else
static void sockit_owm_irq(void * state, alt_u32 id)
#endif
{
  // clear onewire interrupts
  IORD_SOCKIT_OWM_CTL (sockit_owm.base);
  // set the flag indicating a completed 1-wire cycle
  ALT_FLAG_POST (sockit_owm.irq, 0x1, OS_FLAG_SET);
}
#else
#endif
```

4.2.4 Nios-II HAL and µC/OS-II Drivers

The sockit_owm is integrated into the Nios II software development environment with a TCL script and C source and header files. The integration is achieved according to the instructions from Altera. The files for Nios II HAL and μC/OS-II integration are listed in Table 4.3. The software TCL script produces integration into the Nios II EDS, by defining:

- driver version and compatibility
- list of C sources and headers
- list of driver configuration options.

For the benefit of easy integration into an Altera FPGA design, the directory sockit_owm is applied as a portable component in FPGA devices.

Table 4.3: Files for Nios II HAL and μC/OS-II Integration

File	**description**
sockit_owm_hw.tcl	TCL script for integration into SOPC Builder
ockit_owm_sw.tcl	TCL script for integration into Nios II EDS
inc/sockit_owm_regs.h	specification of hardware registers and low-level access macros
HAL/inc/sockit_owm.h	initialization and interrupt handling code
HAL/src/sockit_owm.c	

Integration is completed based on the Altera specification with two TCL scripts such as sockit_owm_hw.tcl for SOPC Builder integration and sockit_owm_sw.tcl for Nios II EDS integration.

4.2.5 Nios II EDS

The C driver of sockit_owm is automatically incorporated into the BSP when the induction of SOPC Builder project comprises of the sockit_owm component which initiates the BSP.

4.3 Hardware Design Implementation in Qsys Tools of Quartus II

The Quartus II is a programmable logic device design software that creates and synthesizes the design of FPGA for the complete sensor controller and fabricates the VHDL of the interfaces. The compilation is executed, and the system is downloaded into the FPGA device. All designs of wireless WQM systems are performed in VHDL programming. Firstly, to create the project, in Quartus II 13.1 (64-bit) Web Edition is launched. From the “File” menu, “New Project Wizard” is selected. Then, the project directory, the name of the project, name of the top-level design entity, devices family and devices name for the project are determined. The project is named as DE2_Basic Computer and by clicking “Finish”, and a new project is created. After creating the new project, in the main Quartus II window, “Tool” is selected to launch the Qsys tool. The Qsys system is started and the “System Contents” tab is displayed.

The hardware characteristics of the Nios II system, such as a processor, and components to include in the system have to be defined in Qsys Tool. The following procedures are performed to design the project in Qsys system.

1. Specify the target FPGA and clock settings.
2. Add the Nios II core, on-chip memory, and other components.
3. Specify base addresses and interrupt request (IRQ) priorities.
4. Generate the Qsys system.

4.3.1 Specifying Target FPGA and Clock Settings

When the Qsys system is launched, the components are added to the system and the selected components are configured to fulfill the necessities of the design. The components are displayed via the window of the System Contents tab. Then, Project Settings is selected to do the necessary setting of the proposed project. The device family “Cyclone V” is selected and device “5CSEMA5F31C6” is selected according to the user’s FPGA SoC board. The accessible components are indicated at the left side of the window. The hardware system of the proposed project is created using the Qsys tool and runs under the control of a clock. From “View” menu, “Clock” is selected and the “clk_0” is set as 50-MHz as. The “clk_0” is the default clock input name for the Qsys system. The frequency specified for “clk_0” is essential to correspond to the oscillator that operates the FPGA.

4.3.2 Adding Components

The components are added to design the hardware of the wireless WQM system. The procedures of adding components are as follows:

1. Adding Nios-II processor
2. Adding On-Chip memory
3. Connecting the components
4. Adding parallel input-output I/O interface
5. Making necessary connections
6. Adding interval timer
7. Making connections
8. Adding system ID peripheral
9. Adding temperature sensor DS18B20
10. Adding JTAG UART
11. Adding MaxSonar UART
12. Adding high_res_timer
13. Adding SPI (3 wire serial)
14. Adding the SDRAM controller
15. Assigning interrupt numbers.

4.3.2.1 Adding Nios-II Processor

To specify the processor, from “Embedded Processors” on the left side of the Qsys window, Nios-II Processor is selected and click “Add”, which proceeds to the window. Then Nios II (e) is chosen for the economy version. The Nios- II processor appears as “nios2_qsys_0” in the “System Content”. The Qsys tool selects titles automatically for several components. The titles are not essentially explanatory enough to be simply connected with the target design, but those titles can be modified. To rename, right click on “nios2_qsys_0” and rename as “CPU” and then enter. The “nios2_qsys_0” is changed to “CPU”. The Nios- II processor has two inputs such as reset and interrupt. The processor begins to perform the instructions saved at memory addresses and is known as reset vector and interrupt vector when one of the inputs is initialized.

4.3.2.2 Adding On-Chip Memory

On-chip memory must be specified. From the category “Memories and Memory Controllers”, “On-Chip” is selected, and then “On-Chip Memory (RAM or ROM)” is selected and click “Add”. When the On-Chip Memory Configuration Wizard window appears, Data width is set to 32 bits and the Total memory size to 4K bytes (4096 bytes). Then, click “Finish”. In the “System Contents”, “onchip_memory2_0” appears, then it is renamed as “onchip_mem”.

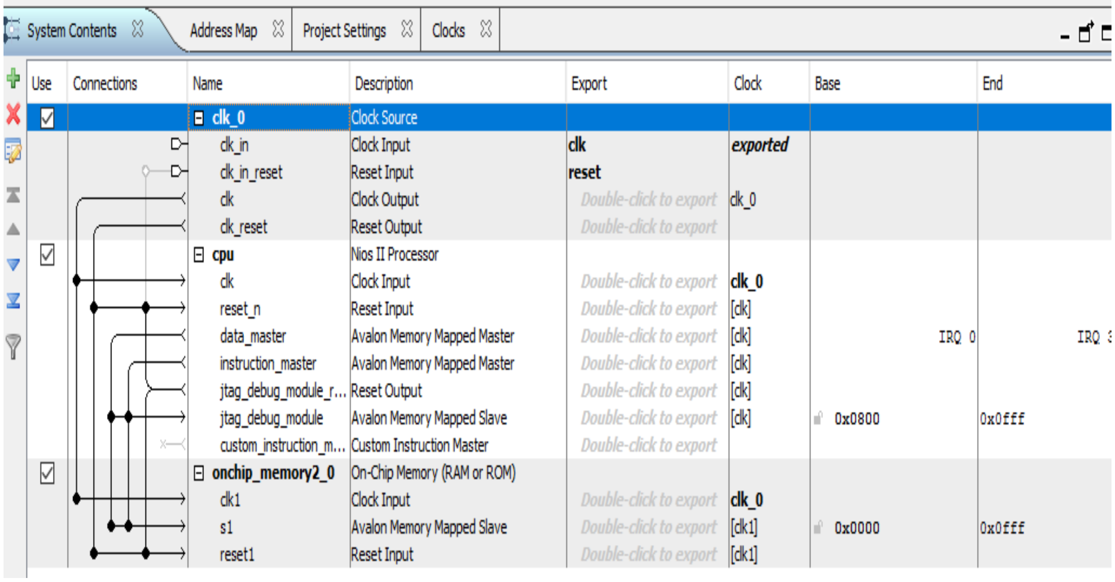
4.3.2.3 Connecting the Components

In the “System Contents” tab, the connections that are previously created are demonstrated by filled circles and the other accessible connections are indicated by empty circles. A connection is made by clicking on an empty circle and a connection is removed by clicking the filled circles. The following connections are made in the Connection area of the “System Contents” tab.

- Clock inputs of the processor and the memory to the clock output of the clock component
- Reset inputs of the processor and the memory to both the reset output of the clock component and the “jtag_debug_module_reset output”
- The s1 input of the memory to both the “data_master” and “instruction_master” outputs of the processor. The connections are established as shown in Fig. 4.3.

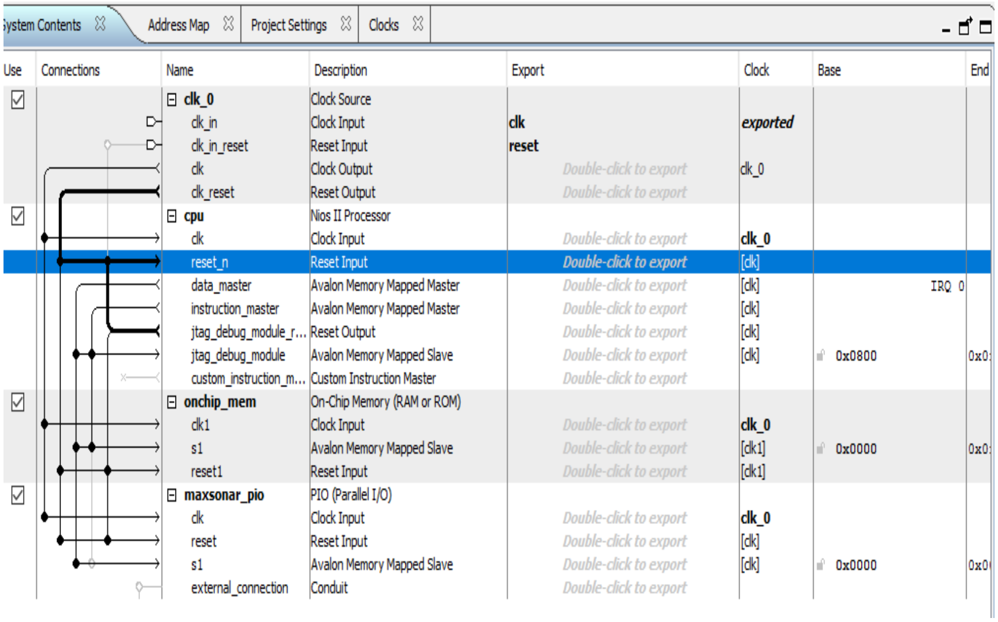
4.3.2.4 Adding parallel Input Output I/O Interface

Similarly, as specifying On-Chip Memory, the parallel input-output I/O interface is specified. Firstly, select “Peripherals”, and then select “Microcontroller Peripherals”, and then PIO is selected and click “Add” to reach the PIO Configuration Wizard again. In the PIO Configuration Wizard, the width of the port to be 8 bits is specified and the direction of the port to be “Output” is selected. Then, click “Finish” to proceed to the System Contents tab, pio_0 appears in the System Contents tab. Then, it is renamed from the “pio_0” to “maxsonar_pio”.



Use	Connections	Name	Description	Export	Clock	Base	End	
☑		⊟ clk_0	Clock Source					
		clk_in	Clock Input	clk	exported			
		clk_in_reset	Reset Input	reset				
		clk	Clock Output	Double-click to export	clk_0			
		clk_reset	Reset Output	Double-click to export				
☑		⊟ cpu	Nios II Processor					
		clk	Clock Input	Double-click to export	clk_0			
		reset_n	Reset Input	Double-click to export	[clk]			
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]	IRQ 0		IRQ 3
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		jtag_debug_module_r...	Reset Output	Double-click to export	[clk]			
		jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0800	0x0fff	
		custom_instruction_m...	Custom Instruction Master	Double-click to export				
☑		⊟ onchip_memory2_0	On-Chip Memory (RAM or ROM)					
		clk1	Clock Input	Double-click to export	clk_0			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0000	0x0fff	
		reset1	Reset Input	Double-click to export	[clk1]			

Fig. 4.3: Established Connections



Use	Connections	Name	Description	Export	Clock	Base	End
☑		⊟ clk_0	Clock Source				
		clk_in	Clock Input	clk	exported		
		clk_in_reset	Reset Input	reset			
		clk	Clock Output	Double-click to export	clk_0		
		clk_reset	Reset Output	Double-click to export			
☑		⊟ cpu	Nios II Processor				
		clk	Clock Input	Double-click to export	clk_0		
		reset_n	Reset Input	Double-click to export	[clk]		
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]	IRQ 0	
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]		
		jtag_debug_module_r...	Reset Output	Double-click to export	[clk]		
		jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0800	0x0
		custom_instruction_m...	Custom Instruction Master	Double-click to export			
☑		⊟ onchip_mem	On-Chip Memory (RAM or ROM)				
		clk1	Clock Input	Double-click to export	clk_0		
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0000	0x0
		reset1	Reset Input	Double-click to export	[clk1]		
☑		⊟ maxsonar_pio	PIO (Parallel I/O)				
		clk	Clock Input	Double-click to export	clk_0		
		reset	Reset Input	Double-click to export	[clk]		
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000	0x0
		external_connection	Conduit	Double-click to export			

Fig. 4.4: Qsys System with Necessary Connections

4.3.2.5 Making Necessary Conncetions

The next step is to identify the compulsory connections for the two PIOs such as Clock input and Reset input as shown in Fig. 4.4.

- Clock input of the PIO to the clock output of the clock component.
- Reset input of the PIO to the reset output of the clock component and the “jtag_debug_module_reset output”.
- The “s1” input of the PIO the “data_master” output of the processor.

4.3.2.6 Adding Interval Timer

Commonly, a timer component is used by the control systems to empower accurate calculation of time, therefore, the Nios-II HAL requires a timer to offer a periodic system clock tick. The following steps are performed to add the timer:

1. On the “Component Library” tab, expand “Peripherals”, expand “Microcontroller Peripherals”, and then click “Interval Timer”.
2. Click “Add”. The Interval Timer parameter editor appears.
3. In the Presets list, select “Full-featured”.
4. Click “Finish”. Then, return to the Qsys “System Contents” tab, and the interval timer appears instantly in the system contents table.
5. In the “Name” column, right-click the interval timer and click “Rename”.
6. Type “sys_clk_timer” and press “Enter”. The Interval Timer is added and renamed as “sys_clk_timer”.

4.3.2.7 Making Connections

After adding the internal timer, the connections are done as follows:

- Connect the “clk” port of the “clk_0**”** clock source to the clk port of the interval timer.
- Connect the “clk_reset” port of the “clk_0**”** clock source to the reset port of the interval timer.
- Connect the “data_master” port of the Nios II processor to the “s1” port of the “interval timer”.

4.3.2.8 Adding System ID Peripheral

The system ID peripheral performs as a safeguard to prevent downloading software unintentionally that compiles for a different Nios II system. The Nios-II SBT for Eclipse prevents the downloading of programs compiling for a different system when the system ID peripheral is included. To add the system ID peripheral, the following procedures are performed:

1. On the “Component Library” tab, expand “Peripherals”, expand “Debug and Performance” and then click “System ID Peripheral”.

2. Click “Add”. The System ID Peripheral parameter editor is shown in the System Contents, it is renamed as “sysid”.

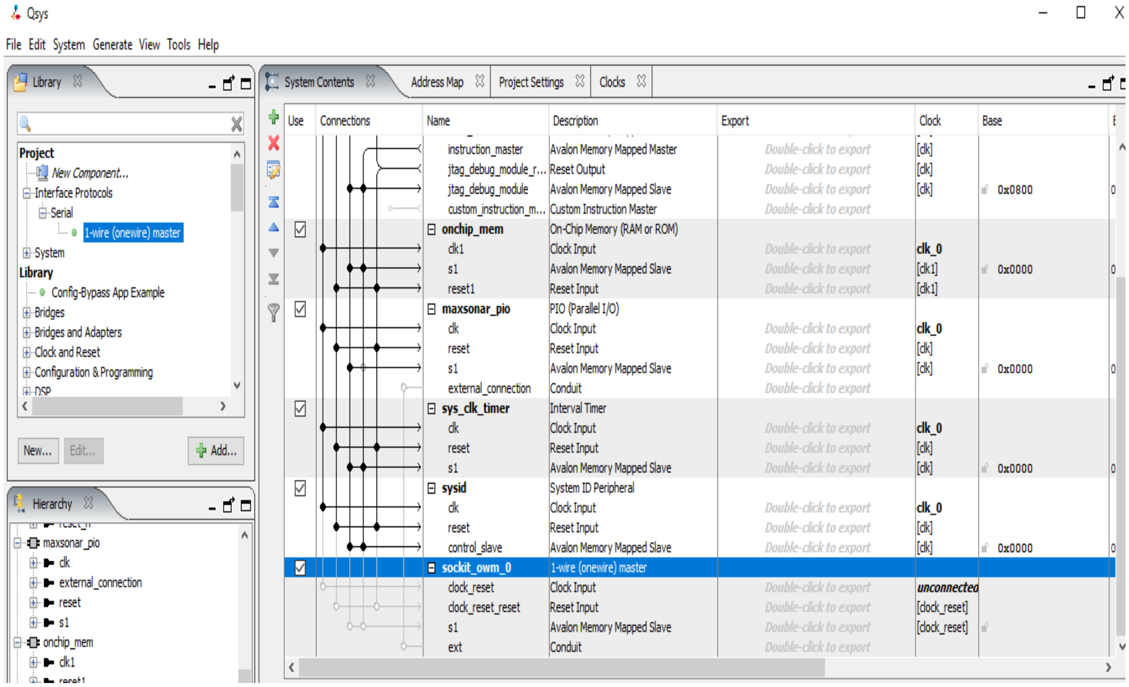
Specify the connections between the clk port of the “clk_0” clock source to the clk port of the System ID Peripheral, the “clk_reset” port of the “clk_0” clock source to the reset port of the System ID Peripheral.

4.3.2.9 Adding Temperature Sensor DS18B20

To add Temperature sensor DS18B20, “Interface Protocols” is selected, then “Serial” is selected, and select 1wire (one wire) master is selected and clicked. The “sockit_owm” editor appears, and by clicking “Finish”, it is included in the Qsys system by the name “sockit_owm_0” as shown in Fig. 4.5. It is renamed “ds10b20_owm”. Then specify the connections between the “data_master” port of the Nios II processor to the “s1” port of the “System ID Peripheral”.

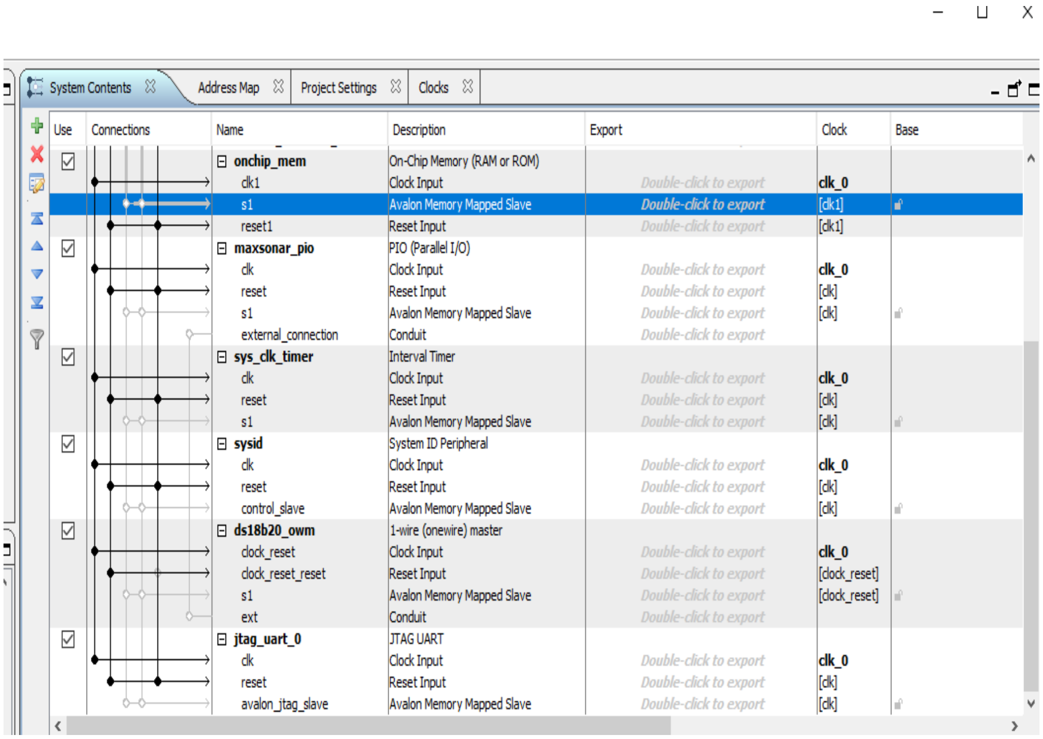
4.3.2.10 Adding JTAG UART

To build communication between the Nios- II system and the host computer, the connection must be made. This work can be achieved by instantiating the JTAG UART interface. First, select “Interface Protocols”, then select “Serial”, and select “JTAG UART” and click “Add” to achieve the JTAG UART Configuration Wizard. The default settings are not changed and then click “Finish” to revert to the “System Contents” tab. “JTAG UART” turns up in the system as shown in Fig. 4.6. Then, connections are specified. The “JTAG UART” is connected to the clock, reset and data-master ports, and the PIOs.



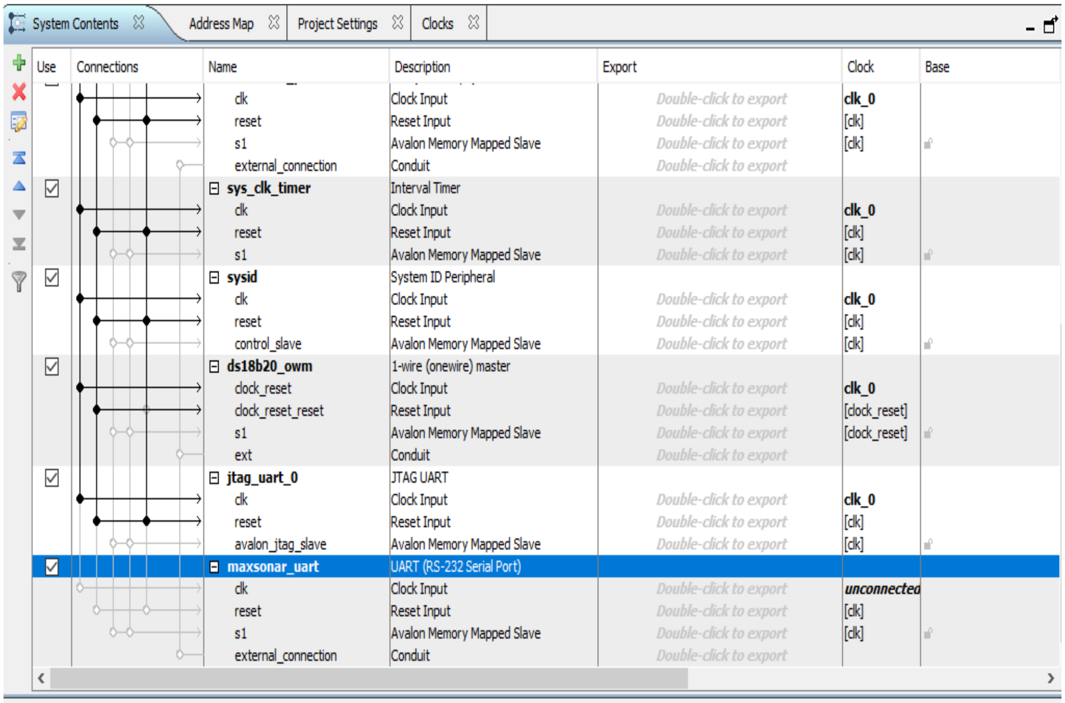
Use	Connections	Name	Description	Export	Clock	Base
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]	
		jtag_debug_module_r...	Reset Output	Double-click to export	[clk]	
		jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0800
		custom_instruction_m...	Custom Instruction Master	Double-click to export		
☑		⊟ **onchip_mem**	On-Chip Memory (RAM or ROM)			
		clk1	Clock Input	Double-click to export	**clk_0**	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0000
		reset1	Reset Input	Double-click to export	[clk1]	
☑		⊟ **maxsonar_pio**	PIO (Parallel I/O)			
		clk	Clock Input	Double-click to export	**clk_0**	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000
		external_connection	Conduit	Double-click to export		
☑		⊟ **sys_clk_timer**	Interval Timer			
		clk	Clock Input	Double-click to export	**clk_0**	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000
☑		⊟ **sysid**	System ID Peripheral			
		clk	Clock Input	Double-click to export	**clk_0**	
		reset	Reset Input	Double-click to export	[clk]	
		control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000
☑		⊟ **sockit_owm_0**	1-wire (onewire) master			
		clock_reset	Clock Input	Double-click to export	***unconnected***	
		clock_reset_reset	Reset Input	Double-click to export	[clock_reset]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	
		ext	Conduit	Double-click to export		

Fig. 4.5: Inclusion of sockit_owm in Qsys System



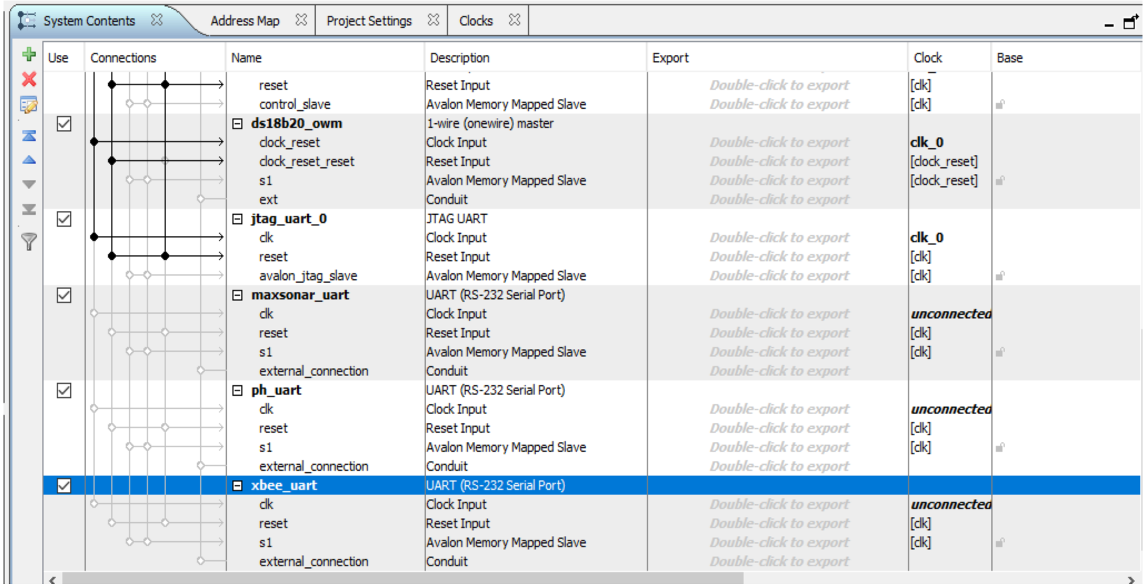
Use	Connections	Name	Description	Export	Clock	Base
☑		⊟ **onchip_mem**	On-Chip Memory (RAM or ROM)			
		clk1	Clock Input	Double-click to export	**clk_0**	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	
		reset1	Reset Input	Double-click to export	[clk1]	
☑		⊟ **maxsonar_pio**	PIO (Parallel I/O)			
		clk	Clock Input	Double-click to export	**clk_0**	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ **sys_clk_timer**	Interval Timer			
		clk	Clock Input	Double-click to export	**clk_0**	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **sysid**	System ID Peripheral			
		clk	Clock Input	Double-click to export	**clk_0**	
		reset	Reset Input	Double-click to export	[clk]	
		control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **ds18b20_owm**	1-wire (onewire) master			
		clock_reset	Clock Input	Double-click to export	**clk_0**	
		clock_reset_reset	Reset Input	Double-click to export	[clock_reset]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	
		ext	Conduit	Double-click to export		
☑		⊟ **jtag_uart_0**	JTAG UART			
		clk	Clock Input	Double-click to export	**clk_0**	
		reset	Reset Input	Double-click to export	[clk]	
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	

Fig. 4.6: Inclusion of JTAG UART in Qsys System



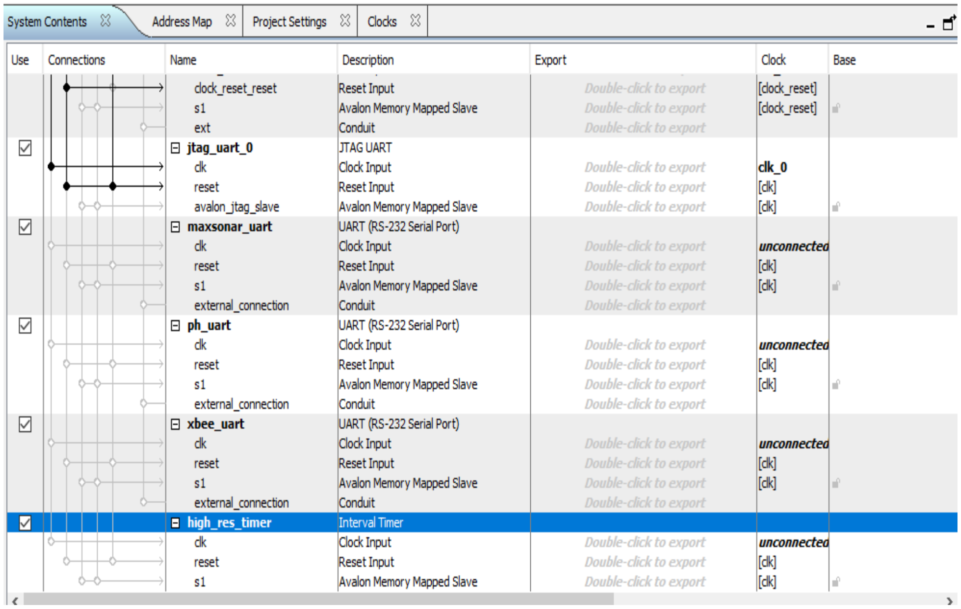
Use	Connections	Name	Description	Export	Clock	Base
		clk	Clock Input	Double-click to export	clk_0	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ **sys_clk_timer**	Interval Timer			
		clk	Clock Input	Double-click to export	clk_0	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **sysid**	System ID Peripheral			
		clk	Clock Input	Double-click to export	clk_0	
		reset	Reset Input	Double-click to export	[clk]	
		control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **ds18b20_owm**	1-wire (onewire) master			
		clock_reset	Clock Input	Double-click to export	clk_0	
		clock_reset_reset	Reset Input	Double-click to export	[clock_reset]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	
		ext	Conduit	Double-click to export		
☑		⊟ **jtag_uart_0**	JTAG UART			
		clk	Clock Input	Double-click to export	clk_0	
		reset	Reset Input	Double-click to export	[clk]	
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **maxsonar_uart**	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		

Fig. 4.7: Inclusion of maxsonar_uart in Qsys System



Use	Connections	Name	Description	Export	Clock	Base
		reset	Reset Input	Double-click to export	[clk]	
		control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **ds18b20_owm**	1-wire (onewire) master			
		clock_reset	Clock Input	Double-click to export	clk_0	
		clock_reset_reset	Reset Input	Double-click to export	[clock_reset]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	
		ext	Conduit	Double-click to export		
☑		⊟ **jtag_uart_0**	JTAG UART			
		clk	Clock Input	Double-click to export	clk_0	
		reset	Reset Input	Double-click to export	[clk]	
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **maxsonar_uart**	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ **ph_uart**	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ **xbee_uart**	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		

Fig. 4.8: Inclusion of ph_uart and xbee uart in Qsys System



Use	Connections	Name	Description	Export	Clock	Base
		clock_reset_reset	Reset Input	Double-click to export	[clock_reset]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	
		ext	Conduit	Double-click to export		
☑		⊟ jtag_uart_0	JTAG UART			
		clk	Clock Input	Double-click to export	clk_0	
		reset	Reset Input	Double-click to export	[clk]	
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ maxsonar_uart	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ ph_uart	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ xbee_uart	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ high_res_timer	Interval Timer			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	

Fig. 4.9: Inclusion of high-res-timer in Qsys System

4.3.2.11 Adding MaxSonar UART

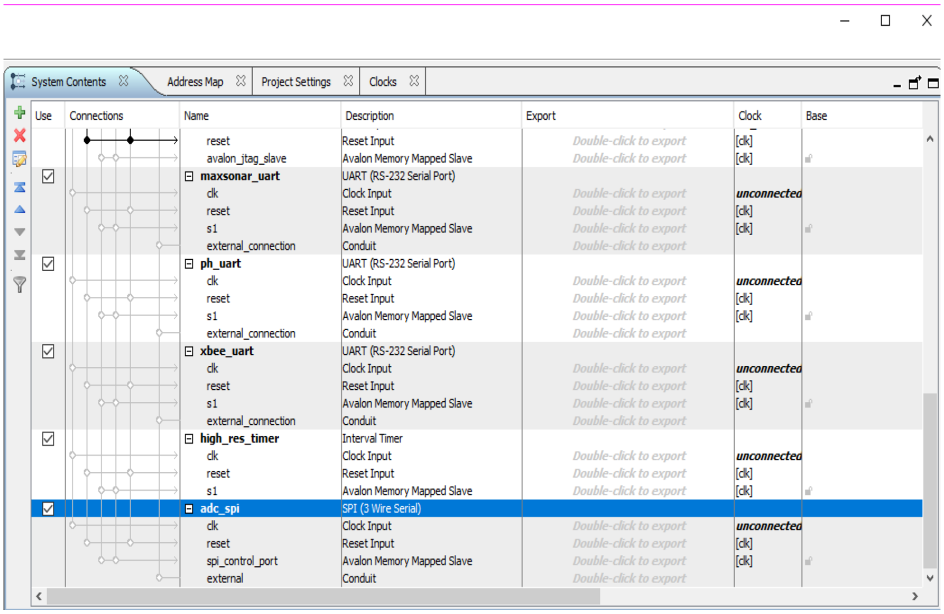
To add MaxSonar UART, “Interface Protocols” is selected, and “Serial” is selected, and “UART (RS-232 Serial Port)” is selected. The “UART (RS-232 Serial Port)” wizard appears. By clicking “Finish”, “uart_0” is shown in the “System Contents”. It is renamed as “maxsonar_uart” as shown in Fig. 4.7. Similarly, “ph-uart” and “xbee_uart” are added in the system as shown in Fig. 4.8.

4.3.2.12 Adding high_res_timer

On the “Component Library” tab, expand “Peripherals”, expand “Microcontroller Peripherals”, and then click “Interval Timer”. Click “Add” and the Interval Timer parameter editor shows up. In the “Presets” list, “Full-featured” is selected. By clicking “Finish”, the interval timer in the Qsys “System Contents” tab appears instantly in the system contents table. It is renamed as “high_res_timer” as shown in Fig. 4.9.

4.3.2.13 Adding SPI (3 Wire Serial)

Since the turbidity sensor and CO_2 sensor are analog sensors, the output of those sensors is needed to convert to digital output. Therefore, to add ADC, select “Interface Protocols”, then select “Serial”, and select “SPI (3 Wire Serial)”. By clicking “Finish” in



Use	Connections	Name	Description	Export	Clock	Base
		reset	Reset Input	Double-click to export	[clk]	
		avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **maxsonar_uart**	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ **ph_uart**	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ **xbee_uart**	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ **high_res_timer**	Interval Timer			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ **adc_spi**	SPI (3 Wire Serial)			
		clk	Clock Input	Double-click to export	*unconnected*	
		reset	Reset Input	Double-click to export	[clk]	
		spi_control_port	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external	Conduit	Double-click to export		

Fig. 4.10: Inclusion of adc_spi in System

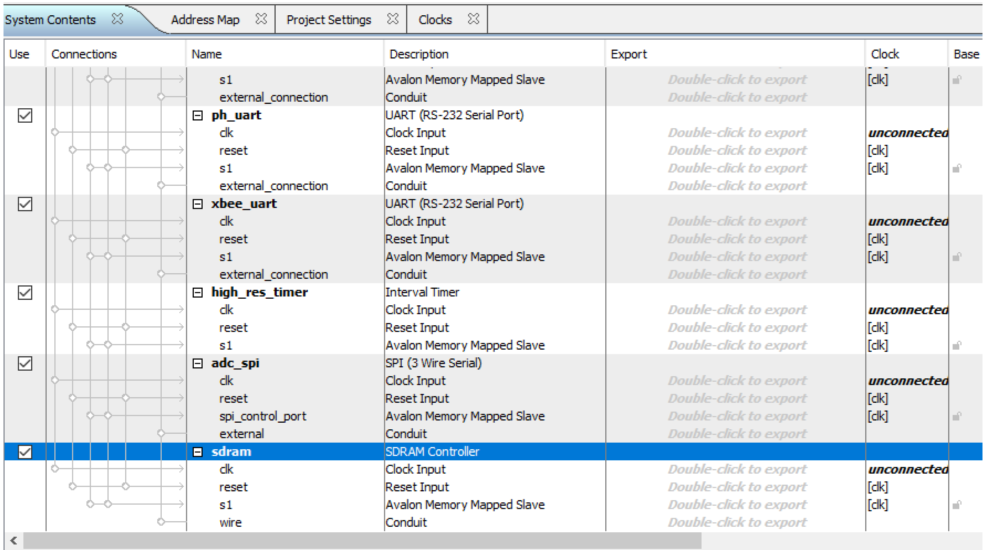
the SPI (3 Wire Serial) Wizard, the “spi_0” appears in the Qsys system. It is renamed as “adc_spi” Fig. 4.10.

4.3.2.14 Adding SDRAM Controller

To specify SDRAM controller, from the category “Memories and Memory Controllers”, “External Memory Interface” is selected, then from SDRAM Interfaces, “SDRAM Controller” is clicked. SDRAM Controller Wizard appears and click “Finish”. The “new_sdram_controller_0” appears in the Qsys system. It is renamed as “sdram” as shown in Fig. 4.11.

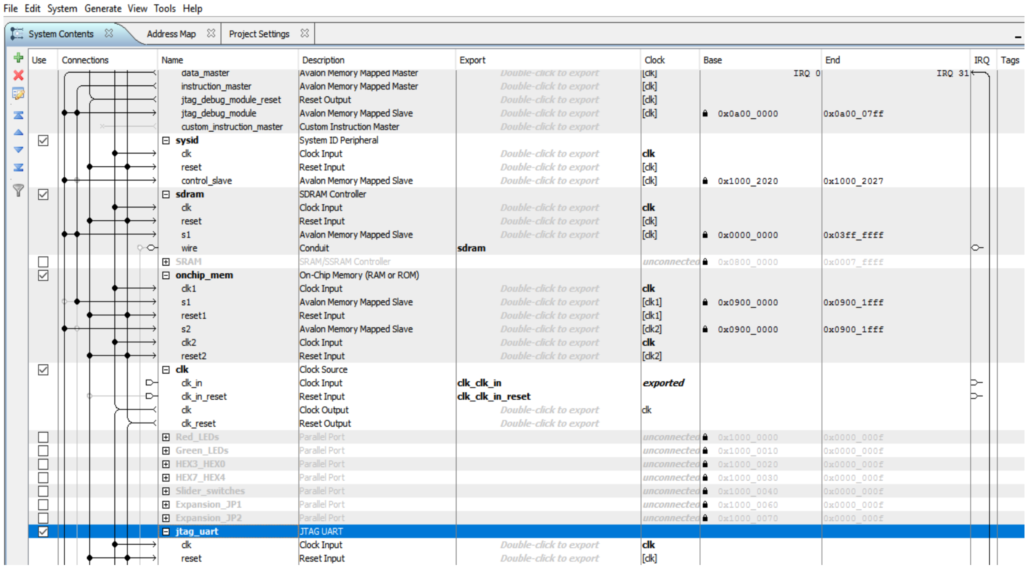
4.3.2.15 Assigning Interrupt Numbers

The Qsys tool additionally offers an “Assign Interrupt Numbers” command which links IRQ signals to process valid hardware results. Assigning an appropriate priority to the hardware interrupt is compulsory. Since the Nios II HAL understands low IRQ values as a higher priority, the timer component essentially comprises of the highest IRQ priority to uphold the accuracy of the system clock tick. In the “IRQ” column of the system, the Nios II processor is attached to the JTAG UART and interval timer. Base addresses and



Use	Connections	Name	Description	Export	Clock	Base
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ ph_uart	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ xbee_uart	UART (RS-232 Serial Port)			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external_connection	Conduit	Double-click to export		
☑		⊟ high_res_timer	Interval Timer			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
☑		⊟ adc_spi	SPI (3 Wire Serial)			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		spi_control_port	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		external	Conduit	Double-click to export		
☑		⊟ sdram	SDRAM Controller			
		clk	Clock Input	Double-click to export	unconnected	
		reset	Reset Input	Double-click to export	[clk]	
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	
		wire	Conduit	Double-click to export		

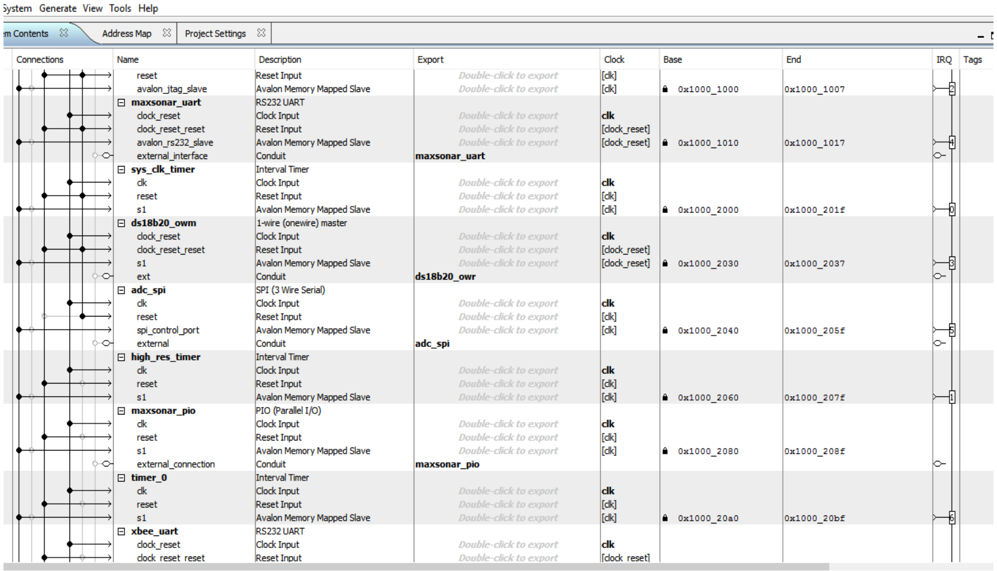
Fig. 4.11: Inclusion of sdram in Qsys System



Use	Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]	IRQ 0	IRQ 31		
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]				
		jtag_debug_module_reset	Reset Output	Double-click to export	[clk]				
		jtag_debug_module	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0a00_0000	0x0a00_07ff		
		custom_instruction_master	Custom Instruction Master	Double-click to export					
☑		⊟ sysid	System ID Peripheral						
		clk	Clock Input	Double-click to export	clk				
		reset	Reset Input	Double-click to export	[clk]				
		control_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x1000_2020	0x1000_2027		
☑		⊟ sdram	SDRAM Controller						
		clk	Clock Input	Double-click to export	clk				
		reset	Reset Input	Double-click to export	[clk]				
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0000_0000	0x03ff_ffff		
		wire	Conduit	sdram					
☐		⊞ SRAM	SRAM/SSRAM Controller		unconnected	0x0800_0000	0x0007_ffff		
☑		⊟ onchip_mem	On-Chip Memory (RAM or ROM)						
		clk1	Clock Input	Double-click to export	clk				
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk1]	0x0900_0000	0x0900_1fff		
		reset1	Reset Input	Double-click to export	[clk1]				
		s2	Avalon Memory Mapped Slave	Double-click to export	[clk2]	0x0900_0000	0x0900_1fff		
		clk2	Clock Input	Double-click to export	clk				
		reset2	Reset Input	Double-click to export	[clk2]				
☑		⊟ clk	Clock Source						
		clk_in	Clock Input	clk_clk_in	exported				
		clk_in_reset	Reset Input	clk_clk_in_reset					
		clk	Clock Output	Double-click to export	clk				
		clk_reset	Reset Output	Double-click to export					
☐		⊞ Red_LEDs	Parallel Port		unconnected	0x1000_0000	0x0000_000f		
☐		⊞ Green_LEDs	Parallel Port		unconnected	0x1000_0010	0x0000_000f		
☐		⊞ HEX3_HEX0	Parallel Port		unconnected	0x1000_0020	0x0000_000f		
☐		⊞ HEX7_HEX4	Parallel Port		unconnected	0x1000_0030	0x0000_000f		
☐		⊞ Slider_switches	Parallel Port		unconnected	0x1000_0040	0x0000_000f		
☐		⊞ Expansion_JP1	Parallel Port		unconnected	0x1000_0060	0x0000_000f		
☐		⊞ Expansion_JP2	Parallel Port		unconnected	0x1000_0070	0x0000_000f		
☑		⊟ jtag_uart	JTAG UART						
		clk	Clock Input	Double-click to export	clk				
		reset	Reset Input	Double-click to export	[clk]				

Fig. 4.12: Connecting Base Address and IRQ

IRQ are specified by connecting the IRQ line 31 from the JTAG UART to the Nios II processor by assigning the connection under the IRQ column, as presented in Fig. 4.12. The IRQ line 32 is connected to port 2, 4, 0, 3, 5, 1, 6, 7, 8 as shown in Fig. 4.13 and Fig. 4.14.



Connections	Name	Description	Export	Clock	Base	End	IRQ	Tags
	reset	Reset Input	Double-click to export	[clk]				
	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	[clk]	🔒 0x1000_1000	0x1000_1007	2	
	⊟ **maxsonar_uart**	RS232 UART						
	clock_reset	Clock Input	Double-click to export	**clk**				
	clock_reset_reset	Reset Input	Double-click to export	[clock_reset]				
	avalon_rs232_slave	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	🔒 0x1000_1010	0x1000_1017	4	
	external_interface	Conduit	**maxsonar_uart**					
	⊟ **sys_clk_timer**	Interval Timer						
	clk	Clock Input	Double-click to export	**clk**				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	🔒 0x1000_2000	0x1000_201f	0	
	⊟ **ds18b20_owm**	1-wire (onewire) master						
	clock_reset	Clock Input	Double-click to export	**clk**				
	clock_reset_reset	Reset Input	Double-click to export	[clock_reset]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	🔒 0x1000_2030	0x1000_2037	3	
	ext	Conduit	**ds18b20_owr**					
	⊟ **adc_spi**	SPI (3 Wire Serial)						
	clk	Clock Input	Double-click to export	**clk**				
	reset	Reset Input	Double-click to export	[clk]				
	spi_control_port	Avalon Memory Mapped Slave	Double-click to export	[clk]	🔒 0x1000_2040	0x1000_205f	5	
	external	Conduit	**adc_spi**					
	⊟ **high_res_timer**	Interval Timer						
	clk	Clock Input	Double-click to export	**clk**				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	🔒 0x1000_2060	0x1000_207f	1	
	⊟ **maxsonar_pio**	PIO (Parallel I/O)						
	clk	Clock Input	Double-click to export	**clk**				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	🔒 0x1000_2080	0x1000_208f		
	external_connection	Conduit	**maxsonar_pio**					
	⊟ **timer_0**	Interval Timer						
	clk	Clock Input	Double-click to export	**clk**				
	reset	Reset Input	Double-click to export	[clk]				
	s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	🔒 0x1000_20a0	0x1000_20bf	6	
	⊟ **xbee_uart**	RS232 UART						
	clock_reset	Clock Input	Double-click to export	**clk**				
	clock_reset_reset	Reset Input	Double-click to export	[clock_reset]				

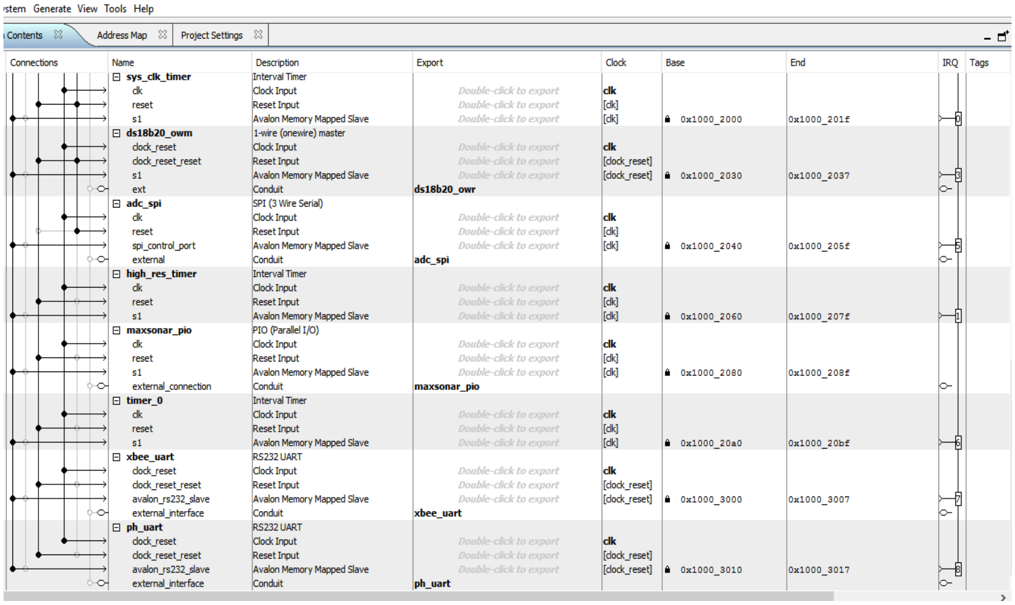
Fig. 4.13: Connection of IRQ Line 32 to port 2, 4, 0, 3, 5, 1, 6

4.3.3 Specifying Base Addresses and Interrupt Request (IRQ) Priorities

Since the accurate assignment of the base and end addresses of the several components in the implemented system has not been accomplished, the user can assign these addresses although the Qsys tool has assigned them automatically. On the System menu, "Assign Base Addresses" is clicked to create Qsys assign functional base addresses to each component in the system. In the Qsys window, by double-clicking on the base address for each component, the addresses can be assigned. The base and end addresses for "Avalon Mapped Slave" of each component are assigned, and the connections with IRQ are shown in Table 4.4. These addresses are locked by clicking on the adjacent lock symbol. The Qsys design of the proposed reconfigurable WSN for WQM in IoT environment is completed in Qsys tool, then, it is generated before it is compiled to Quartus II and downloaded to FPGA board.

4.3.4 Generating Qsys System

When specified components are implemented to the desired system, the specified system is saved and named as "nios_system". The file extension is ".qsys". Therefore, the file



Name	Description	Export	Clock	Base	End	IRQ
sys_clk_timer	Interval Timer					
clk	Clock Input	Double-click to export	clk			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x1000_2000	0x1000_201f	0
ds18b20_owm	1-wire (onewire) master					
clock_reset	Clock Input	Double-click to export	clk			
clock_reset_reset	Reset Input	Double-click to export	[clock_reset]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	0x1000_2030	0x1000_2037	3
ext	Conduit	ds18b20_owr				
adc_spi	SPI (3 Wire Serial)					
clk	Clock Input	Double-click to export	clk			
reset	Reset Input	Double-click to export	[clk]			
spi_control_port	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x1000_2040	0x1000_205f	5
external	Conduit	adc_spi				
high_res_timer	Interval Timer					
clk	Clock Input	Double-click to export	clk			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x1000_2060	0x1000_207f	1
maxsonar_pio	PIO (Parallel I/O)					
clk	Clock Input	Double-click to export	clk			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x1000_2080	0x1000_208f	
external_connection	Conduit	maxsonar_pio				
timer_0	Interval Timer					
clk	Clock Input	Double-click to export	clk			
reset	Reset Input	Double-click to export	[clk]			
s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x1000_20a0	0x1000_20bf	6
xbee_uart	RS232 UART					
clock_reset	Clock Input	Double-click to export	clk			
clock_reset_reset	Reset Input	Double-click to export	[clock_reset]			
avalon_rs232_slave	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	0x1000_3000	0x1000_3007	7
external_interface	Conduit	xbee_uart				
ph_uart	RS232 UART					
clock_reset	Clock Input	Double-click to export	clk			
clock_reset_reset	Reset Input	Double-click to export	[clock_reset]			
avalon_rs232_slave	Avalon Memory Mapped Slave	Double-click to export	[clock_reset]	0x1000_3010	0x1000_3017	8
external_interface	Conduit	ph_uart				

Fig. 4.14: Connection of IRQ Line 32 to port 7, 8

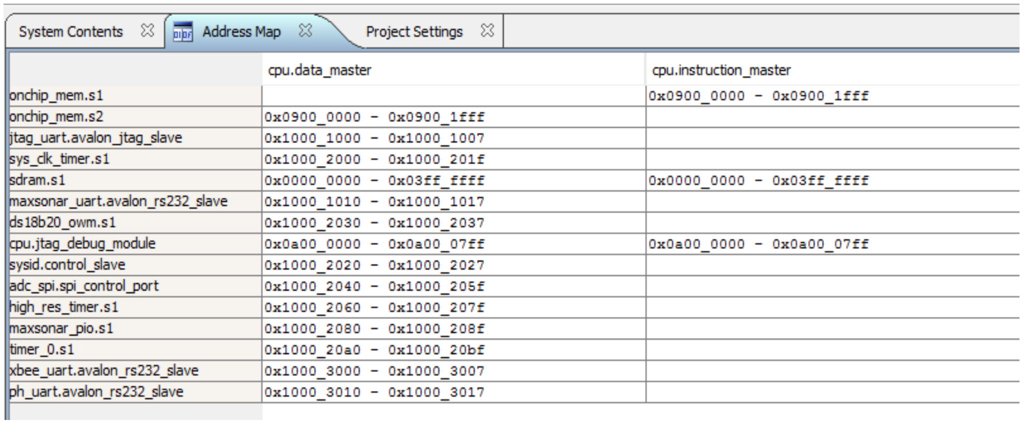
name is appeared as “nois_system.qsys”. Then, the “Generation” tab is selected, which leads to the window. Then, click “Generate” on the bottom of the window and when the generation process is effectively achieved, the generation process shows the message “Generate Completed”. The output file “nios_system. sopcinfo” is achieved. The output “. sopc” file will be compiled to Quartus II. Then, exit the Qsys tool to return to the main Quartus II window. The updates and modification to the designed system are done by returning to the Qsys tool and any component in the “System Contents” tab of the Qsys tool is picked, edited or deleted. When a new component is added, the system is regenerated essentially.

4.4 HDL Example

From the “Generate” tab, “HDL example” file can be viewed. When “HDL Example” is clicked, the window appears and “Verilog” is already selected in HDL Language where the Verilog module can be viewed. When “VHDL” is selected in “HDL Language” box, the file is changed to VHDL module. The HDL example is copied to declare an instance of the Qsys system to access the Nios II system in top-level Verilog or VHDL module and link the inputs and outputs of the PIO ports. It also connects the clock and reset inputs to the relevant pins on the FPGA device.

Table 4.4: Assigned Base and End Addresses and IRQ Connection

Name	Description	Clock	Base	End	IRQ
jtag_debug_module	Avalon Memory Mapped Slave	clk	0x0a00_0000	0x0a00_07ff	
control slave	Avalon Memory Mapped Slave	clk	0x1000_2020	0x1000_2027	
s1	Avalon Memory Mapped Slave	clk	0x0000_0000	0x03ff_ffff	
s1	Avalon Memory Mapped Slave	clk1	0x0900_0000	0x0900_1fff	
s2	Avalon Memory Mapped Slave	clk2	0x0900_0000	0x0900_1fff	
avalon_jtag_slave	Avalon Memory Mapped Slave	clk	0x1000_1000	0x1000_1007	2
avalon_rs232_slave	Avalon Memory Mapped Slave	clock-reset	0x1000_1010	0x1000_1017	4
s1	Avalon Memory Mapped Slave	clk	0x1000_2000	0x1000_201f	0
s1	Avalon Memory Mapped Slave	clock-reset	0x1000_2030	0x1000_2037	3
spi_control_port	Avalon Memory Mapped Slave	clk	0x1000_2040	0x1000_205f	5
s1	Avalon Memory Mapped Slave	clk	0x1000_2060	0x1000_207f	1
s1	Avalon Memory Mapped Slave	clk	0x1000_2080	0x1000_208f	
s1	Avalon Memory Mapped Slave	clk	0x1000_20a0	0x1000_20bf	6
avalon_rs232_slave	Avalon Memory Mapped Slave	clock-reset	0x1000_3000	0x1000_3007	7
avalon_rs232_slave	Avalon Memory Mapped Slave	clock-reset	0x1000_3010	0x1000_3017	8



	cpu.data_master	cpu.instruction_master
onchip_mem.s1		0x0900_0000 - 0x0900_1fff
onchip_mem.s2	0x0900_0000 - 0x0900_1fff	
jtag_uart.avalon_jtag_slave	0x1000_1000 - 0x1000_1007	
sys_clk_timer.s1	0x1000_2000 - 0x1000_201f	
sdram.s1	0x0000_0000 - 0x03ff_ffff	0x0000_0000 - 0x03ff_ffff
maxsonar_uart.avalon_rs232_slave	0x1000_1010 - 0x1000_1017	
ds18b20_owm.s1	0x1000_2030 - 0x1000_2037	
cpu.jtag_debug_module	0x0a00_0000 - 0x0a00_07ff	0x0a00_0000 - 0x0a00_07ff
sysid.control_slave	0x1000_2020 - 0x1000_2027	
adc_spi.spi_control_port	0x1000_2040 - 0x1000_205f	
high_res_timer.s1	0x1000_2060 - 0x1000_207f	
maxsonar_pio.s1	0x1000_2080 - 0x1000_208f	
timer_0.s1	0x1000_20a0 - 0x1000_20bf	
xbee_uart.avalon_rs232_slave	0x1000_3000 - 0x1000_3007	
ph_uart.avalon_rs232_slave	0x1000_3010 - 0x1000_3017	

Fig. 4.15: Address Map of Desired Qsys System

4.5 Address Map

The address map of the desired Qsys system of proposed reconfigurable WSN design for WQM in the IoT environment can be checked from the "Address Map" tab of the Qsys Tool as shown in Fig. 4.15. The complete Nios system design can be checked as a ".bsf" file in Quartus II as shown in Fig. 4.16. The pin assignments for the DE-SoC board are observed in the QSF file of the Quartus II project.

4.6 Schematic Diagrams of Processor and Components

The schematic diagrams of the processor and components can be viewed in the "Netlist Viewers" of the "Tools" menu. From the "Tools" menu, when "Netlist Viewers" is expanded, three viewers appear as "RTL viewer" such as "Technology Map Viewer (Post-Mapping)", "Technology Map Viewer (Post-Fitting)" and "State Machine Viewer". From the "RTL viewer", the complete schematic diagram of the hardware of the proposed reconfigurable WSN design for WQM in the IoT environment is observed as shown in Fig. 4.17. By clicking "Technology Map Viewer (Post-Mapping)", the schematic diagram of post mapping of the proposed system of reconfigurable WSN design for WQM in the IoT environment is observed as Fig. 4.18. The view of each component can be maximized for a clear view. The schematic diagram of the Nios-II processor and the port connections with the components is shown in Fig. 4.19.

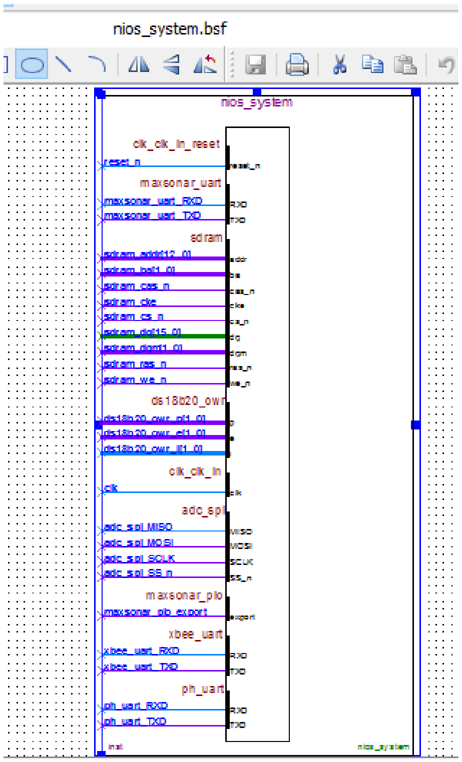


Fig. 4.16: Complete Nios system .bsf File

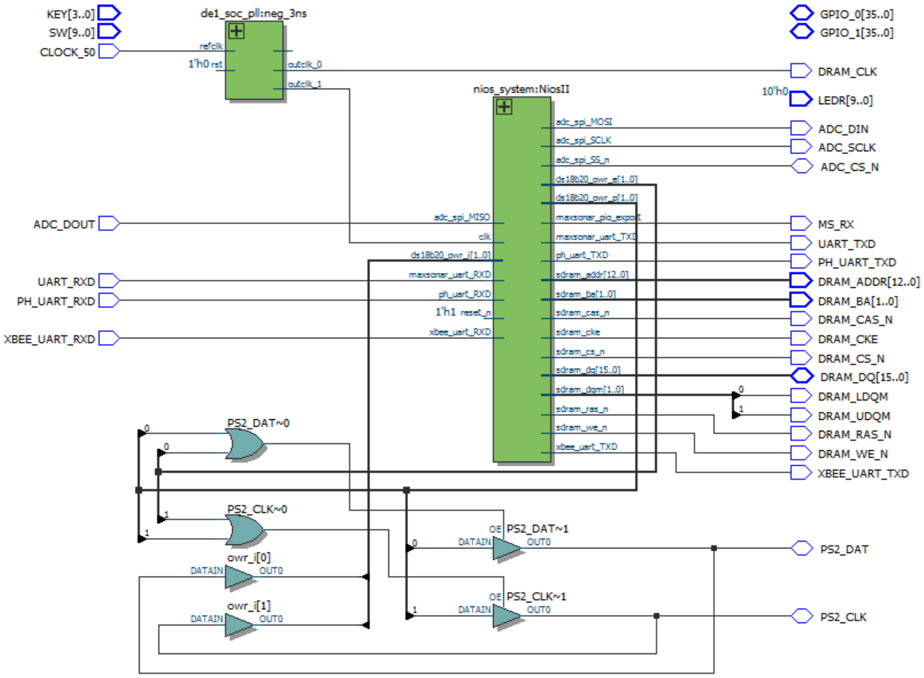


Fig. 4.17: Schematic diagrams WQM System

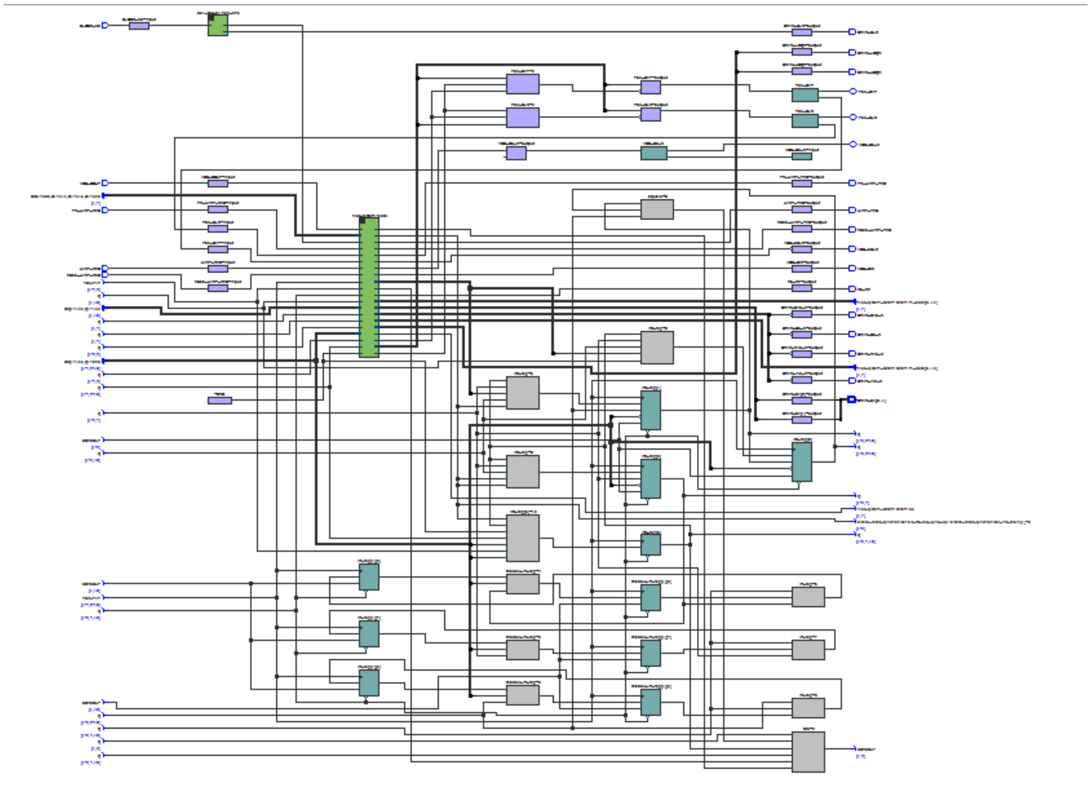


Fig. 4.18: Post Mapping Diagram of WQM System

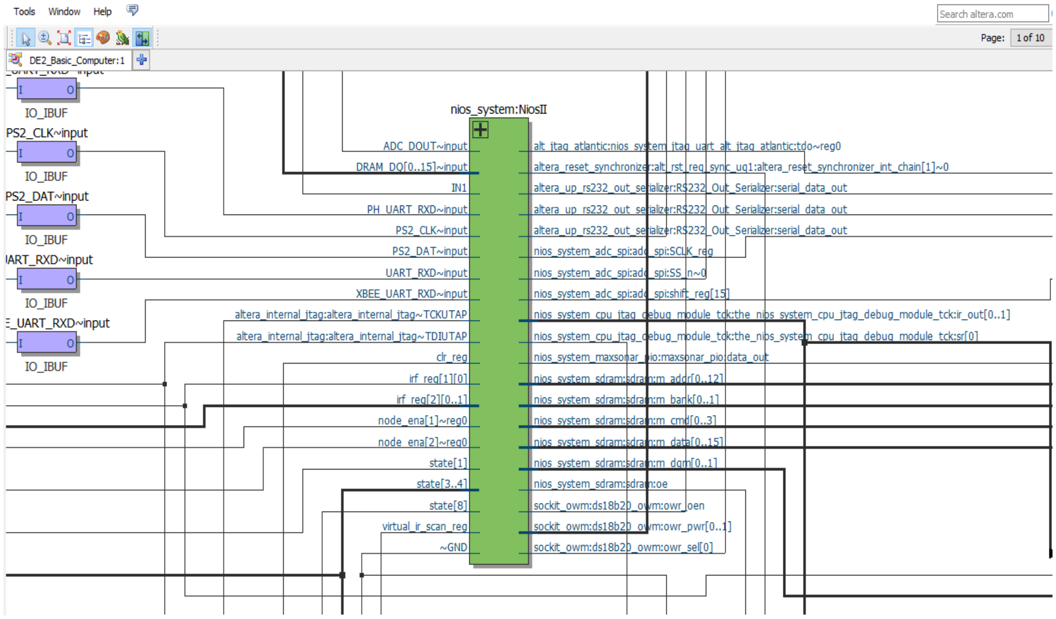

Tools
Window
Help
Search altera.com
Page: 1 of 10
DE2_Basic_Computer:1
IO_IBUF
PS2_CLK~input
PS2_DAT~input
IART_RXD~input
E_UART_RXD~input
nios_system:NiosII
ADC_DOUT~input
DRAM_DQ[0..15]~input
IN1
PH_UART_RXD~input
PS2_CLK~input
PS2_DAT~input
UART_RXD~input
XBEE_UART_RXD~input
altera_internal_jtag:altera_internal_jtag~TCKUTAP
altera_internal_jtag:altera_internal_jtag~TDIUTAP
clr_reg
irf_reg[1][0]
irf_reg[2][0..1]
node_ena[1]~reg0
node_ena[2]~reg0
state[1]
state[3..4]
state[8]
virtual_ir_scan_reg
~GND
alt_jtag_atlantic:nios_system_jtag_uart_alt_jtag_atlantic:tdo~reg0
altera_reset_synchronizer:alt_rst_req_sync_uq1:altera_reset_synchronizer_int_chain[1]~0
altera_up_rs232_out_serializer:RS232_Out_Serializer:serial_data_out
altera_up_rs232_out_serializer:RS232_Out_Serializer:serial_data_out
altera_up_rs232_out_serializer:RS232_Out_Serializer:serial_data_out
nios_system_adc_spi:adc_spi:SCLK_reg
nios_system_adc_spi:adc_spi:SS_n~0
nios_system_adc_spi:adc_spi:shift_reg[15]
nios_system_cpu_jtag_debug_module_tck:the_nios_system_cpu_jtag_debug_module_tck:ir_out[0..1]
nios_system_cpu_jtag_debug_module_tck:the_nios_system_cpu_jtag_debug_module_tck:sr[0]
nios_system_maxsonar_pio:maxsonar_pio:data_out
nios_system_sdram:sdram:m_addr[0..12]
nios_system_sdram:sdram:m_bank[0..1]
nios_system_sdram:sdram:m_cmd[0..3]
nios_system_sdram:sdram:m_data[0..15]
nios_system_sdram:sdram:m_dqm[0..1]
nios_system_sdram:sdram:oe
sockit_owm:ds18b20_owm:owr_oen
sockit_owm:ds18b20_owm:owr_pwr[0..1]
sockit_owm:ds18b20_owm:owr_sel[0]

Fig. 4.19: Schematic Diagram of Nios-II Processor and Port Connections

4.7 Integrating Qsys System into Quartus II Project

Firstly, Quartus II 13.1 (64 bit) Web Edition is launched and the project DE2_Basic_Computer.qpf file is opened. In completing the hardware design of the proposed reconfigurable WSN system for WQM in the IoT environment, the designed hardware circuit is compiled in FPGA by utilizing the Programmer Tool in the Quartus II software.

4.7.1 Building Project in Eclipse of Nios-II Software Build Tools

To build the project in the Eclipse, the following procedures must be done:

1. Creating a new Nios II Application and BSP from Template
2. Compiling the project
3. Running the software program on target hardware.

4.7.1.1 Creating a New Nios II Application and BSP from Template

From the menu bar, Tools is selected and expanded and “Nios-II Software Tools for Eclipse” is selected and clicked. The “Eclipse IDE” is opened and the Workspace Launcher dialog box appears, click “OK” to acquire the default workspace location as “DE1_SoC_DS18B20_v.1.8\ws”. Then, on the “File” menu, “New” is pointed, and then “Nios-II Application and BSP from Template” is selected and clicked. The Nios II Application and BSP from Template wizard appear. The target hardware information must be filled. Next to “SOPC Information File Name**”**, browse to the working directory and “nios_system. sopcinfo**”** file is selected as that file is the output file of Qsys design that is completed in Section 4.3.4. The CPU name is selected as CPU. In the Project name box, WQM_17 is typed for the project name since the WQM project is created on 17 September 2016. The project can be named anything according to the user, however, the underscore must be placed between the two words. For the “Project template”, “Hello World” is selected. By clicking “Finish”, the new projects “WQM_17”, C application project and “WQM_17_bsp”, a board support package that contains the details of the Nios II system hardware are created and displayed in the Project Explorer view, typically on the left side of the window.

4.7.1.2 Modifying C Program

To modify the C program as the desired project, the “WQM_17” is double-clicked, and “hello_world.c” is selected and double-clicked. Then, C codes are modified for the proposed reconfigurable WSN design for WQM in the IoT environment.

4.7.1.3 Calibrating pH Probe

The calibration of pH probe is written as follows:

```
// 13 - "Cal,mid,7.00"
char  cal_midpoint[]  = {'C', 'a', 'l', ',', 'm', 'i', 'd',      ',',      '7', '.', '0', '0', 0x0D};
// 13 - "Cal,low,4.00"
char  cal_lowpoint[]  = {'C', 'a', 'l', ',', 'l', '0', 'w',      ',',      '4', '.', '0', '0', 0x0D};
// 15 - "Cal,high,10.00"
char  cal_highpoint[] = {'C', 'a', 'l', ',', 'h', 'i', 'g', 'h', ',', '1', '0', '.', '0', '0', 0x0D};
int main(){
```

To get single pH reading, the C program is written as follows:

```
// get single PH reading
 // "R<CR>"
 for (i = 0; i < 2; i++){
send_char (get_single_ph[i], PH_UART_BASE);
            }
 // receive 11 bytes e.g. '14.000<CR>*OK<CR>'
do{
rx_cnt = alt_up_rs232_get_used_space_in_read_FIFO (ph_uart_dev);
} while (rx_cnt < 10);
for(i = 0; i < rx_cnt; i++){
ch = get_char(PH_UART_BASE);
if(i < 6){
ph_ch[i] = ch;
                        }
```

```
                    }
 ph_val = atof(ph_ch);
 sprintf (ph_pl_buf, "%f", ph_val);

#ifdef DEBUG_EN
printf ("PH        : %f\n", ph_val);
 #endif
```

4.7.1.4 Calculation of CO_2

The calculation of CO_2 is written as follows:

```
// for Infrared CO2 sensor
        co2_percent = ((adc_readings [2] * 1000) - 400) * (50.0 / 16.0);
        sprint (cdox_pl_buf, "%f", (float)(co2_percent));
```

4.7.1.5 Sending Assembled XBee Tx Frame and Monitoring Response Status

The C codes for sending XBee TX frame and monitoring the response status is written as follows:

```
alt_u8 Tx_Request (void) {
        /*
        alt_u8 retry = 1;
        alt_u8 retry_count = 0;
        while(retry){
                Send_API_Frame ();
                if(Wait_Response()){
                        if(retry_count == MAX_TX_RETRY_COUNT) {
                                PC_UartPutString (“Max Tx Retry Count Reached\n\r”);
                                while(1);
                                retry = 0;
                        }
```

```
            retry_count++;
        }else{
            // handle error-free response
            retry = 0;
        }
    }
    */
    return 0;
}
```

4.7.1.6 Data Transmission of XBee Tx Frame

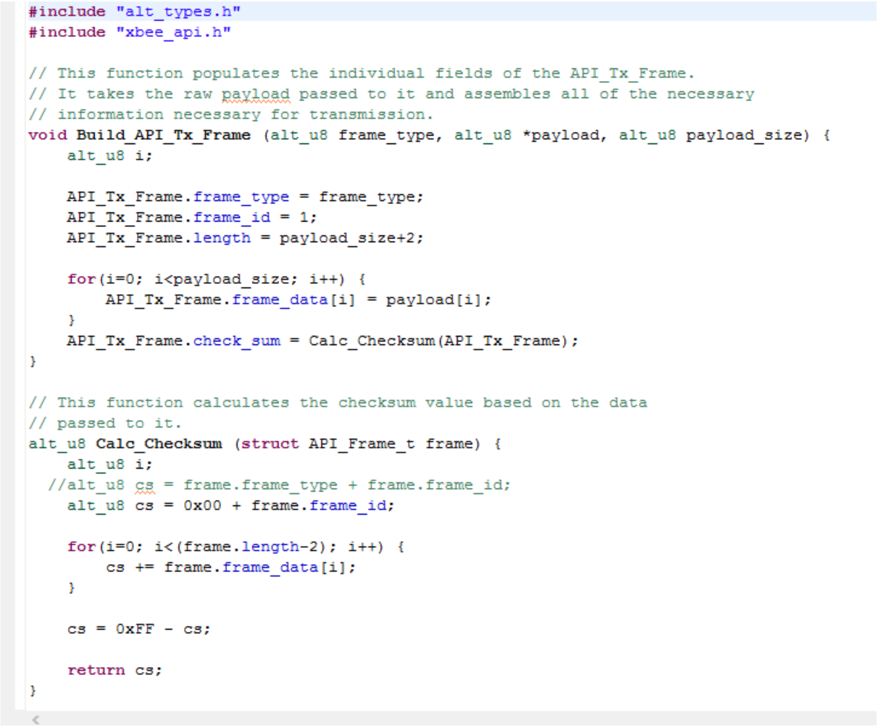
The C codes for the XBee TX frame in the UART to transmit the data is written as follows:

```
 void Send_API_Frame (void) {
  /*
XBee_SpiUartWriteTxData(0x7E);
    XBee_SpiUartWriteTxData(API_Tx_Frame.length >> 8);
XBee_SpiUartWriteTxData (API_Tx_Frame.length);
XBee_SpiUartWriteTxData (API_Tx_Frame.frame_type);
XBee_SpiUartWriteTxData (API_Tx_Frame.frame_id);
XBee_SpiUartPutArray (API_Tx_Frame.frame_data, (API_Tx_Frame.length-2));
    XBee_SpiUartWriteTxData (API_Tx_Frame.check_sum);
    */
}
```

A part of modified C codes for the proposed reconfigurable WSN design for WQM in IoT environment is shown in Fig. 4.20.

4.7.1.7 Compiling Project

After creating the project, it must be compiled to produce an executable software image. The following procedures are performed to compile the project.



```
#include "alt_types.h"
#include "xbee_api.h"

// This function populates the individual fields of the API_Tx_Frame.
// It takes the raw payload passed to it and assembles all of the necessary
// information necessary for transmission.
void Build_API_Tx_Frame (alt_u8 frame_type, alt_u8 *payload, alt_u8 payload_size) {
    alt_u8 i;

    API_Tx_Frame.frame_type = frame_type;
    API_Tx_Frame.frame_id = 1;
    API_Tx_Frame.length = payload_size+2;

    for(i=0; i<payload_size; i++) {
        API_Tx_Frame.frame_data[i] = payload[i];
    }
    API_Tx_Frame.check_sum = Calc_Checksum(API_Tx_Frame);
}

// This function calculates the checksum value based on the data
// passed to it.
alt_u8 Calc_Checksum (struct API_Frame_t frame) {
    alt_u8 i;
  //alt_u8 cs = frame.frame_type + frame.frame_id;
    alt_u8 cs = 0x00 + frame.frame_id;

    for(i=0; i<(frame.length-2); i++) {
        cs += frame.frame_data[i];
    }

    cs = 0xFF - cs;

    return cs;
}
```

Fig. 4.20: A Part of Modified C Codes

1. In the Project Explorer view, “WQM_17_bsp” is right-clicked and “Properties” is clicked.

2. The “Properties for WQM_17_bsp” dialog box emerges and the “Nios II BSP Properties” page is clicked.

3. The “Nios II BSP Properties” page appears, and it comprises of basic software build settings. Support C is checked and “OK” is clicked.

4. The BSP regenerates and the “Properties” dialog box closes.

5. “WQM_17_bsp” is right clicked, “Nios II” is selected and “BSP Editor” is selected and double-clicked.

6. In the “BSP Editor”, “timestamp_timer” is checked as “high_res_timer”.

7. For “bsp_cflags-optimozation”, -03 is typed.

8. “Generate” is clicked and then generate and exit.

9. “WQM_17_bsp” is right clicked and Generate BSP.

10. In the Project Explorer view of the Nios II SBT for Eclipse, the “WQM_17” project is right clicked and “Build Project” is clicked.

11. The “Build Project” dialog box appears, and the project is compiled by the Nios II SBT for Eclipse.

When the compilation fulfills, a “Build Finished” message shows up in the “Console” view as shown in Fig. 4.21. The output “.elf” file is achieved as “WQM_17.elf”.

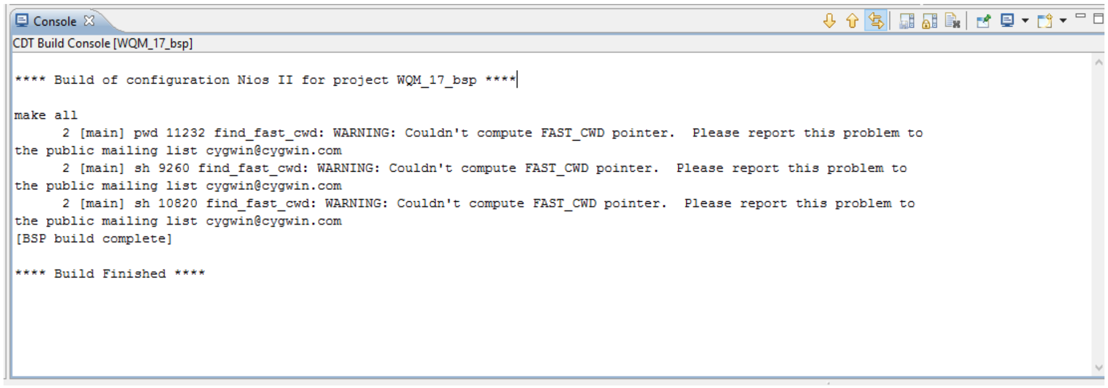
4.7.1.8 Running Program on the Targeted Hardware

The program must be downloaded to target hardware before running it. The following procedures are performed to download the software executable to the target board.

1. The “Programmer” is launched in Quartus II to program the FPGA from “Run” menu, “Run Configurations” is selected and double-clicked.
2. “Nios II Hardware” is selected and double-clicked.
3. In the “Run Configurations” dialog box, “Project name” and “ELF file name” are verified.
4. The “Target connection” is refreshed and the “Processor”, “Cable” and “Byte Stream Devices” are checked.
5. Then click “Run”. When the target hardware commences in running the program, the Nios II Console view displays character I/O output. The program must be run for 3 times until no errors are observed in the Console view.
6. Click the “Terminate” icon (the red square) on the toolbar of the Nios II Console view to abort the run session. When “Terminate” icon is clicked, the Nios II SBT for Eclipse detaches from the target hardware.

4.8 Implementing Hardware Components on FPGA Board

The hardware components of proposed reconfigurable WSN for WQM in the IoT environment has been already designed in Nios-II soft-core processor and downloaded to FPGA board. To perform the experimental measurements, the hardware components such as XBee RF module and sensors have to be implemented on the FPGA board.



```
**** Build of configuration Nios II for project WQM_17_bsp ****

make all
      2 [main] pwd 11232 find_fast_cwd: WARNING: Couldn't compute FAST_CWD pointer.  Please report this problem to
the public mailing list cygwin@cygwin.com
      2 [main] sh 9260 find_fast_cwd: WARNING: Couldn't compute FAST_CWD pointer.  Please report this problem to
the public mailing list cygwin@cygwin.com
      2 [main] sh 10820 find_fast_cwd: WARNING: Couldn't compute FAST_CWD pointer.  Please report this problem to
the public mailing list cygwin@cygwin.com
[BSP build complete]

**** Build Finished ****
```

Fig. 4.21: “Build Finished” Message in Console View

4.8.1 Implementing XBee RF Module

In the proposed reconfigurable WSN design for WQM in IoT environment, the XBee Pro S1 Development Kits is used since it contains RS-232 and USB interface boards which use two 20-pin receptacles to accept modules. The XBee Pro S1 RF Module is developed to base into a receptacle (socket) and therefore any soldering is not necessary when installing it to a board. The XBee Pro S1 Module is installed to an RS-232 interface board. The XBee-Pro module in Fig. 4.22 has an RF output power of 60 mW and two XBee-Pro modules are used in the proposed design for its benefit of having a longer range, where one module is the transmitter module to transmit data from the proposed design while the rest is receiver module as shown in Fig. 4.23 that is used to receive data on PC using a USB2UART port.

The transmitter module of XBee Pro S1 is connected to the I/O port of FPGA. Since power supply voltage of +3.3V is consumed instead of TTL level +5.0V, the XBee 5V to the 3.3V regulator is required to supply power the XBee Pro S1 wireless module. The voltage regulator converts from 5V to 3.3V under which the XBee Series Embedded-Antenna Modules is activated. The XBee USB adapter functions as a connector between the XBee modules and the monitoring device. The pin assignments for the XBee Pro RF module is presented in Table 4.5. Only 4 pins of XBee Pro S1 transmitter module are essential to be connected to the FPGA board as follows:

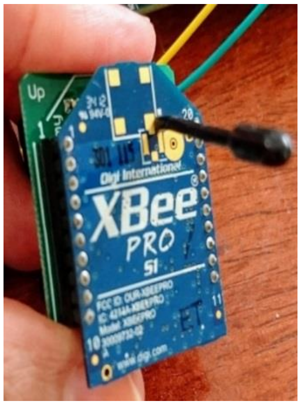


Fig. 4.22: XBee Pro S1 Wireless Transmitter Module

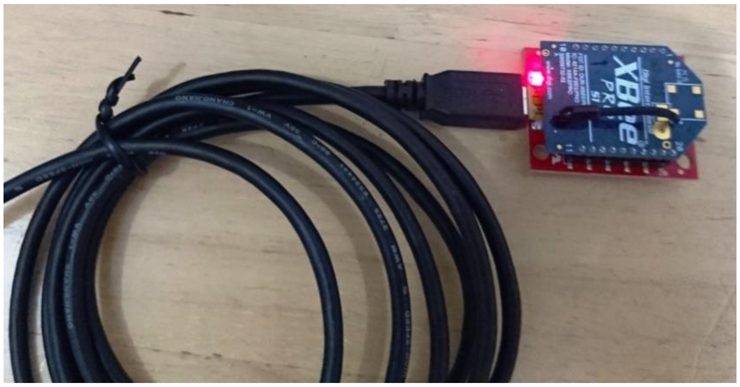


Fig. 4.23: XBee Pro S1 Receiver Module

1.	Pin 1	Vcc (+3.3 V)
2.	Pin 2	TX (Transmit)
3.	Pin 3	Rx (Receive)
4.	Pin 10	GND (Ground or 0V)

The pin wiring of XBee Pro S1 transmitter module is shown in Fig. 4.24. The pin 2 (Tx) of XBee module is connected to the pin 8 of GPIO 1 (JP 2) of FPGA, and the pin 3 (Rx) of XBee is connected to the pin 10 of GPIO 1 (JP 2) of FPGA. The power source from FPGA GPIO 0 (JP1) is connected to the pin 1 (Vcc) of XBee and the GND of FPGA is connected to the pin 10 (GND) of XBee respectively. Serial communications depending on the UARTs for both FPGA and XBee RF module

Table 4.5: Pin Assignments for XBee Pro Module

Pin	**Name**	**Direction**	**Description**
1	Vcc		Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]		Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND		Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

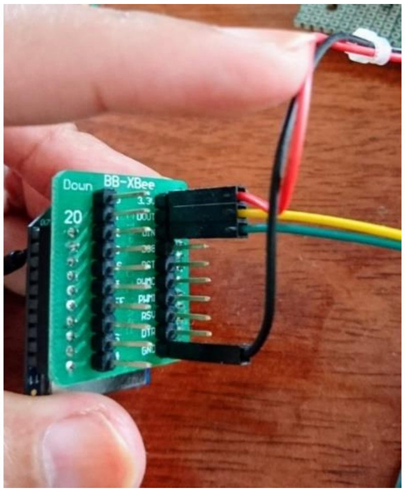


Fig. 4.24: Pin Connection of XBee Pro S1 Module to FPGA Board.

have to be configured with compatible baud rate. The baud rate of the XBee module can be fixed to any rate up to 250,000 bps (practically 115,200 bytes per second via software configuration). For the proposed reconfigurable WSN design, the default baud rate setting for the XBee-Pro module is 9600 bps. Both modules of XBee Pro S1 have activated in the Application Programming Interface (API) mode that enables better management of network properties. The XBee-Pro S1 RF Modules have an interface to FPGA through a logic level asynchronous serial port, hence, the XBee module can interconnect with any logic and voltage compatible to UART through the serial port.

The five sensors detect the water parameters and the data is sent to FPGA which is interfaced with XBee-Pro S1 RF Modules. The XBee Pro S1 is configured to support single point to point and NonBeacon communication system which operates within a Peer-to-Peer network topology. The peer to peer architectures provides fast synchronization times and fast cold starts time. Since the point to point network transmits the data from a single node to designated single node which is identified by a unique address within the network, hopping or routing data is not necessary for this protocol. The unicast transmission of this point to point network provides the most robust and fastest network topology.

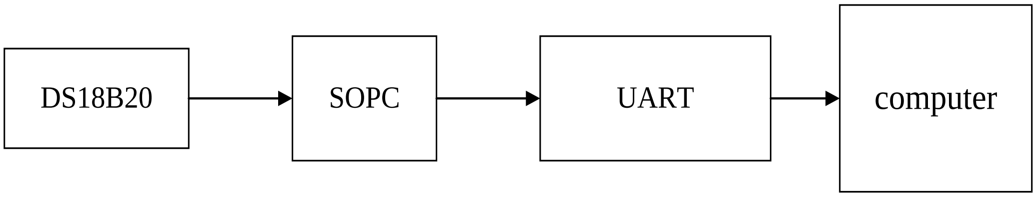


Fig. 4.25: Functional Block Diagram of Temperature Sensor

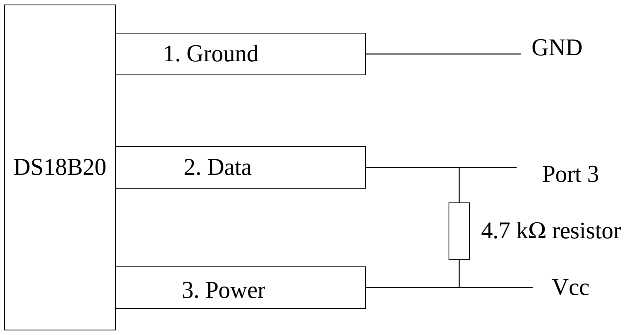


Fig. 4.26: Pin Connection of DS18B20 Sensor

4.8.2 Implementing DS18B20 Temperature Sensor

At the beginning of the experimental implementation, the Digital Thermometer Sensor DS18B20 is considered to be implemented based on the literature review. The block diagram shown in Fig. 4.25 has been developed. Since the DS18B20 is a 1-Wire Digital Thermometer, only one wire (and ground) is connected from the FPGA to a DS18B20. Hence, the bus communication uses one control signal and a 4.7 kΩ weak pull-up resistor is involved by the control line as shown in Fig. 4.26. The pin description of DS18B20 is shown in Table 4.6 [225]. The devices are identified and addressed by the microprocessor on the bus using a unique 64-bit code. Therefore, only one microprocessor is required to command many other DS18B20s allocated over a big area.

The operating temperature of the DS18B20 is in the range of -55°C to +125°C and is precise to ±0.5°Cover the range of -10°C to +85°C. The block diagram of the DS18B20 is presented in Fig. 4.27 [225]. The master must provide a command “Convert T” to initiate the temperature measurement and analog to digital conversion, while the

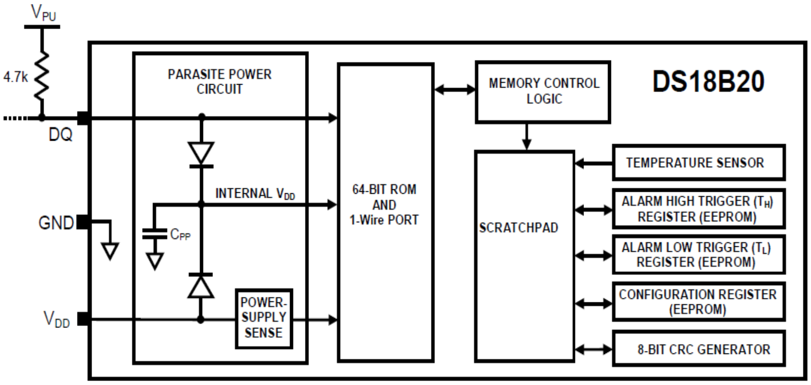
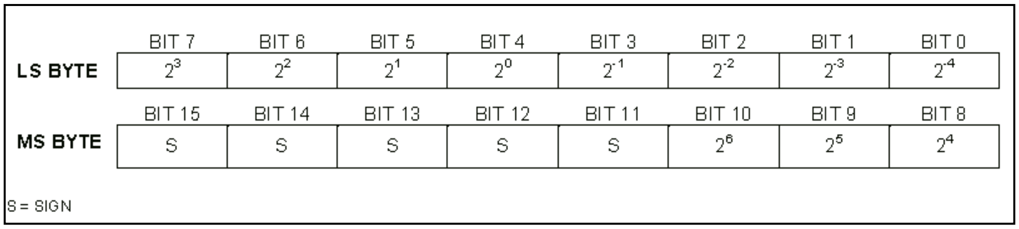


Fig. 4.27: Block Diagram of DS18B20 [225]



	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2^6	2^5	2^4

S = SIGN

Fig. 4.28: Temperature register format [225]

temperature output data is stored in the temperature register. The configuration register of DS18B20 can be set in a resolution of the conversion from temperature to digital 9-12 bits that are compatible with the increments of 0.5 °C, 0.25 °C, 0.125 °C and 0.0625 °C as presented in Table 4.7. The temperature register saves the temperature as 16- bit number as shown in Fig. 4.28 [225]. The temperature sensor DS18B20 is attached to the configurable Nios-II soft processor system on the Cyclone V FPGA of Terasic DE-1

Table 4.6: Pin Description of DS18B20 [225]

Name	**Function**
N.C	No Connection
V_{DD}	Optional $V_{DD.}$ V_{DD} must be grounded for operation in parasite power mode.
DQ	Data Input. Output. Open-drain 1- Wire interface pin. Also provides power to the device when used in parasite mode
GND	Ground

SoC board. The system consists of configurable Nios-II soft processor, SDRAM memory, 1-wire controller for interfacing with the sensor, interval timer, UART and other peripherals.

Table 4.7: Details Corresponding to Bits Resolution in Configuration [225]

Bits Resolution	**Increment (° C)**	**Undefined Resolution Bit**
9 bits	0.5	bit 0, bit 1 and bit 2
10 bits	0.25	bit 0 and bit 1
11 bits	0.125	bit 0
12 bits	0.0625	none

The complete system is executed in VHDL and makes use of the IP cores available in Altera Qsys and open source 1-wire controller. The DS18B20 is connected to FPGA board via the PS/2 connector port which is the female of a PS/2 connector using 6-pin Mini-DIN PS/2 connector cable which is the male of a PS/2 connector. The PS/2 connector provides a 5V power supply. The connector has pins for ground, +5V, serial data, and serial clock. The Data and Clock pins of the PS/2 connector are utilized to connect the two 1-wire lines of DS18B20. The Data and Clock lines are both open-collector with 2kΩ pull-up resistor which is needed to apply the 1-wire bus and 120Ω series resistors to prevent damage to the hardware and software of Vcc. The Pin-out of the PS/2 connector which is connected to the FPGA board is shown in Table 4.8 [225].

Table 4.8: Pin-out of PS/2 Connector on FPGA DE-1 SoC Board [225]

pin	**name**	**description**
1	OWR [0]	1-wire line 0 with a 2kΩ pull-up to 5V and 120Ω serial resistor to the FPGA pin
2	NC	not connected
3	GND	Ground
4	VCC	+5V power supply
5	OWR[0]	1-wire line 1 with a 2kΩ pull-up to 5V and 120Ω serial resistor to the FPGA pin
6	NC	not connected

4.8.2.1 UART Circuit Design with DS18B20

The RS232 sensor interface is derived using the UART interface. The transducer controller performs the commands to activate the sensor. The data is allocated serially in the 1-wire digital thermometer DS18B20 through a single data line with a ground reference. The master device is linked to a slave device. The operations of 1-wire digital thermometer are as follows:

(1) reset sequence with reset pulse and presence pulse;

(2) write “1”: send a “1” bit value to 1-wire slaves;

(3) write “0”: send a “0” bit value to 1-wire slaves;

(4) read bit: read a bit from 1-wire slaves.

4.8.2.2 Power Source for DS18B20

The DS18B20 can be operated in parasite power mode (power from the data line) which is without external power supply since the power is supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The DS18B20 can also be powered by an external supply on VDD. In the project, the parasite power mode is used as the application of the remote temperature sensing system. The VDD pin is connected to the ground when the DS18B20 is functioned in parasite power mode. The power from the 1-Wire bus via the DQ pin is stolen by the parasite-power control circuit when the bus is high. The DS18B20 is supplied by the power from the stolen charge while the bus is high, and some of the charges are saved on the parasite power capacitor (CPP) to supply the power when the bus is low.

4.8.2.3 SOPC System with Temperature Sensor DS18B20

The system structural diagram of the 1-Wire temperature sensor is depicted in Fig. 4.29. The XBee communication module and 1-wire digital thermometer DS18B20 are attached to DE 1-SoC FPGA system via the UART, one-wire, and RS232 interfaces respectively. The temperature sensor DS18B20 interacts over a 1-wire bus which requires only one data line (and ground) for communication with a microcontroller. After programming the configuration bit stream on FPGA, the software for Nios-II processor was downloaded and execution is started. The temperature of the remote sensing device is sampled in the one-second interval and is shown on the Nios-II console

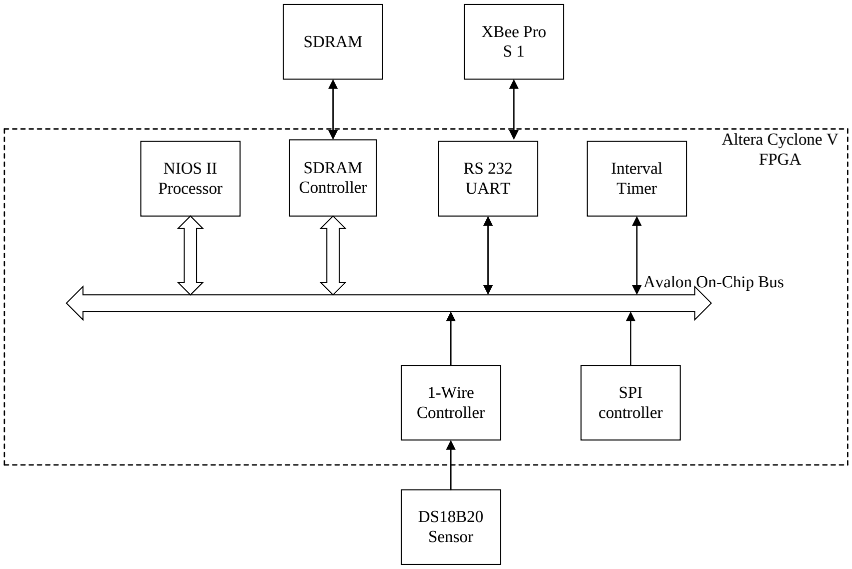


Fig. 4.29: Structure Diagram of the SOPC System with DS18B20 Sensor

Fig. 4.30: Experimental Set-up of DS18B20 Sensor

output. The 1-wire sensor requires an external open drain driver to drive a bidirectional pin. Fig. 4.30 shows the experimental set-up of the DS18B20 sensor for real-time monitoring of water temperature.

4.8.3 Ultrasonic Sensor LV-MaxSonar-EZ1

After designing the temperature sensor, the Maxbotix LV-MaxSonar-EZ sensor is designed for measuring the distance between the water surface and a sensor which is placed at the lake bank. The sensor has the capability of computing the distance and execution the compulsory calculations automatically, returning the distance in inches without applying an external processor.

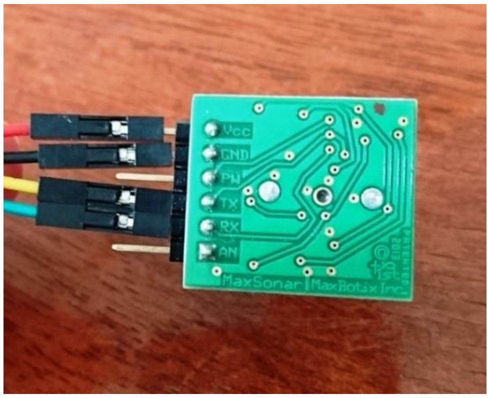


Fig. 4.31: Pin Out of LV-MaxSonar-EZ1

The MaxSonar-EZ1 operates from 2.2V to 5.5V, with a 2mA current draw. The LV-MaxSonar-EZ sensor comprises of 3 separate outputs to update the range data instantaneously: Analog Voltage, Pulse Width, and RS232 Serial. The pin out description of LV-MaxSonar-EZ is presented in Table 4.9 [220]. The pins connected to the FPGA are shown in Fig. 4.31.

4.8.3.1 SOPC System with LV-MaxSonar-EZ1

The structural diagram of the SOPC system after implementing the ultrasonic sensor is shown in Fig. 4.32. The ultrasonic sensor LV-MaxSonar-EZ1 is integrated with RS232 and the default state is the UART mode acting as the transmission (TX) line. The default baud rate is 9600, 8 bits, no parity, no flow control and one stop bit. The LV-MaxSonar-EZ1ultrasonic sensor accumulates the range data by three individual modes such as analog voltage mode, pulse width mode, and RS232 serial mode. These three modes are accordant with the pin out of the ultrasonic sensor. The Tx of the sensor is connected to GPIO 1 D1 and Rx is connected to GPIO 1 D5. The power source of 5V from FPGA GPIO 1 is networked with Vcc pin of the LV-MaxSonar-EZ1 sensor. The ND pin of LV-MaxSonar-EZ1 is attached to the FPGA GPIO 1 ground pin. The experimental set-up of the LV-MaxSonar-EZ1 sensor is shown in Fig.4.33. The third calibration point is set at pH 10, and the pH probe is placed in pH 10 solution and the command “Cal, high, 10.00<CR>” is sent to the embedded pH EZO circuit.

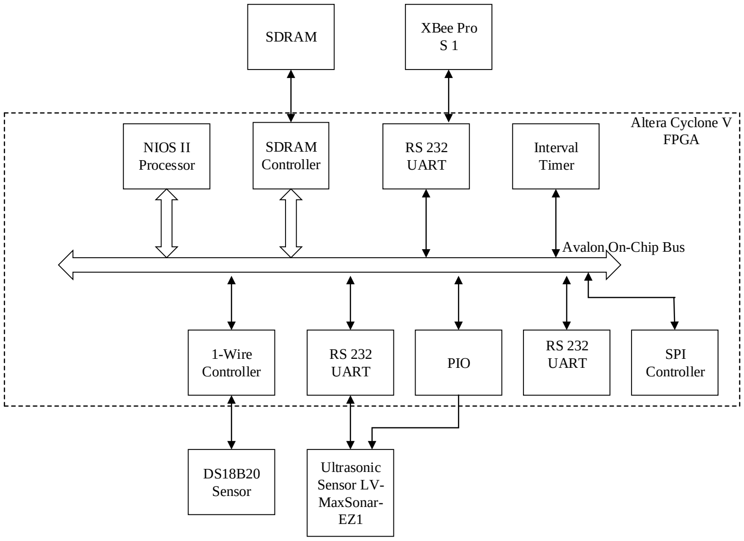


Fig. 4.32: Structure Diagram of SOPC System after Implementing Ultrasonic Sensor

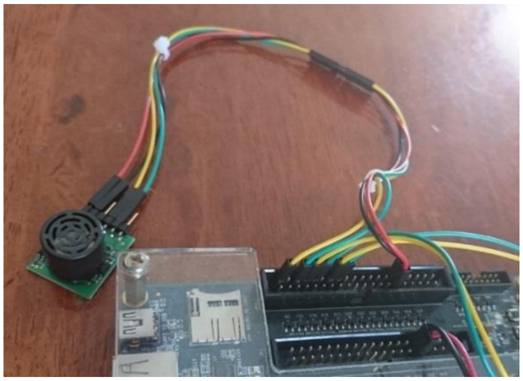


Fig. 4.33: Experimental Set-up of LV-MaxSonar-EZ Sensor

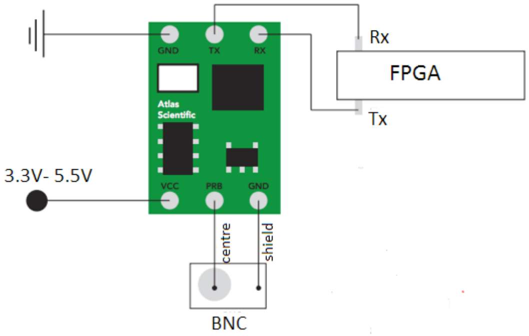


Fig. 4.34: Wiring Diagram of pH Sensor Circuit

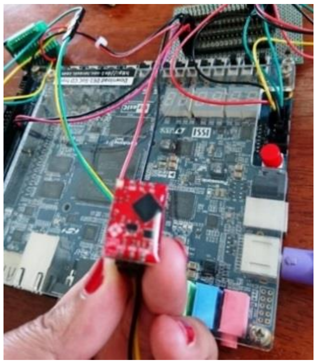


Fig. 4.35: Embedded pH Circuit and BNC shield

Table 4.9: Pin-out Description of LV-MaxSonar-EZ1 [220]

Pin 1-BW	Leave open or hold low for serial output on the TX output. When BW pin is held high the TX output sends a pulse (instead of serial data), suitable for low noise chaining.
Pin 2-PW	This pin outputs a pulse width representation of range. The distance can be calculated using the scale factor of 147uS per inch.
Pin 3-AN	Outputs analog voltage with a scaling factor of (Vcc/512) per inch. A supply of 5V yields ~9.8mV/in. and 3.3V yields ~6.4mV/in. The output is buffered and corresponds to the most recent range data.
Pin 4-RX	This pin is internally pulled high. The LV-MaxSonar-EZ will continually measure range and output if RX data is left unconnected or held high. If held low the sensor will stop ranging. Bring high for 20uS or more to command a range reading.
Pin 5-TX	When the *BW is open or held low, the TX output delivers asynchronous serial with an RS232 format, except voltages are 0-Vcc. The output is an ASCII capital “R”, followed by three ASCII character digits representing the range in inches up to a maximum of 255, followed by a carriage return (ASCII 13). The baud rate is 9600, 8 bits, no parity, with one stop bit. Although the voltage of 0-Vcc is outside the RS232 standard, most RS232 devices have sufficient margin to read 0-Vcc serial data. If standard voltage level RS232 is desired, invert, and connect an RS232 converter such as a MAX232. When BW pin is held high the TX output sends a single pulse, suitable for low noise chaining. (no serial data)
Pin 6-+5V	Vcc – Operates on 2.5V - 5.5V. Recommended current capability of 3mA for 5V, and 2mA for 3V.
Pin 7-GND	Return for the DC power supply. GND (& Vcc) must be ripple and noise free for best operation.

4.8.4 Atlas Scientific pH Sensor

The Atlas Scientific pH sensor includes an embedded pH EZO circuit, pH probe, and BNC shield. The wiring diagram of the pH sensor circuit is shown in Fig. 4.34. The GND pin of pH circuit is connected to the FPGA at pin 30 GPIO 0 (JP1) ground pin, the Vcc pin of the pH circuit is connected to the power source of 5 V from pin 29 GPIO 0 (JP 1) via power array as shown in Fig. 4.35. The Rx pin of pH circuit is connected to pin 4 of FPGA GPIO 1 (JP2) and the Tx pin is connected to pin 16 of FPGA GPIO 1 (JP2) pin. The embedded pH circuit is operated in either two modes which are the UART mode or the I2C mode. In the wireless WQM system, the UART mode is utilized since it is the default mode with a baud rate of 9600 bps, 8 data bits, 1 stop bit, no parity and no flow control. The output data of the embedded pH circuit is encoded in the ASCII characters and followed by a carriage return “<CR>”.

The total number of 3 calibrations of the pH probe is essential before utilizing the pH sensor. According to its datasheet [229], the first calibration must be set at the Mid-point where the pH value is 7.00. The default temperature compensation of EZO™ pH circuit is set at 25° C. The pH probe is placed in the pH7 solution for one hour until the readings are stabilized. Then, the calibration is done at the midpoint value using the command “Cal, mid, 7.00<CR>”. The second calibration is chosen with a pH value of 4, the pH probe is positioned in the pH 4 solution (pH 4) and the command “Cal, low, 4.00<CR>” is conveyed to the pH EZO circuit. The pH data is converted into binary by the embedded pH circuit and the resulted data is transmitted to the FPGA board. The calibration commands are as follows:

```
// 13 - "Cal,mid,7.00"

char  cal_midpoint[]  = {'C', 'a', 'l', ',', 'm', 'i', 'd',      ',',      '7', '.', '0', '0', 0x0D};

// 13 - "Cal,low,4.00"

char  cal_lowpoint[]  = {'C', 'a', 'l', ',', 'l', '0', 'w',      ',',      '4', '.', '0', '0', 0x0D};

// 15 - "Cal,high,10.00"

char   cal_highpoint[] = {'C', 'a', 'l', ',', 'h', 'i', 'g', 'h', ',', '1', '0', '.', '0', '0', 0x0D};
```

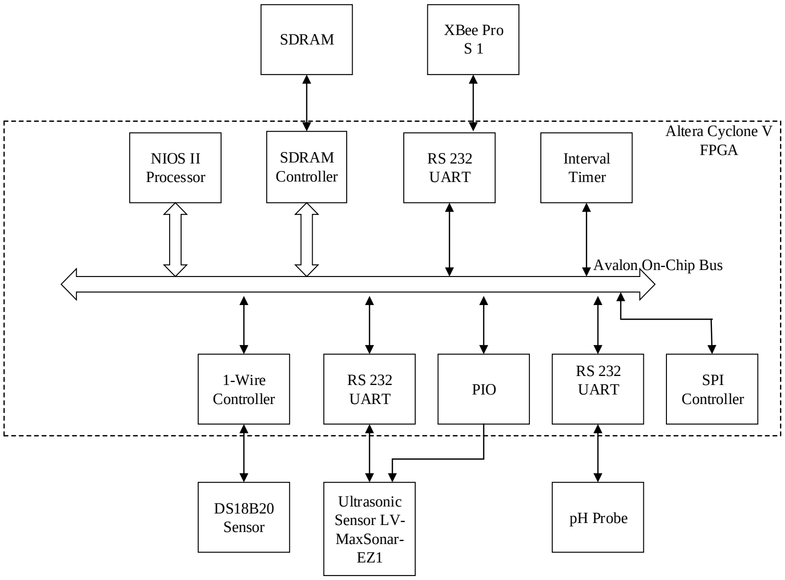


Fig. 4.36: Structure Diagram of SOPC System after Implementing pH Sensor

The structure diagram of the SOPC system after implementing a pH sensor is shown in Fig. 4.36. The XBee communication module, 1-wire digital thermometer DS18B20, and Atlas pH sensor kit are associated with DE 1-SoC FPGA board via the UART, one-wire, and RS232 interfaces respectively. An Atlas pH sensor has a hardware/firmware which is linked to the transducer interface module via the RS232 interface. The received data is identified by the command files and it enables the receiving data buffer to save the next reserving bytes of the message.

4.8.5 Gravity: Analog Infrared CO_2 Sensor SKU: SEN0219

As the CO_2 Sensor SKU: SEN0219 is an analog sensor, a high-speed ADC AD7928 with a rapid conversion rate of 2.5µs per channel is used for converting the analog output to digital. The analog voltage output is converted to 8 bits binary by 8 bits 8 channel ADC. The ADC is built-in and implemented on the FPGA board. The pin header of ADC consists of ten pins such as a +5V pin, (+), a 0V pin (-), and eight analog channels (0-7). The pin out for the DE1-SoC's ADC pin header is shown in Fig. 4.37. The ADC is connected to the FPGA via four signal lines as follows:

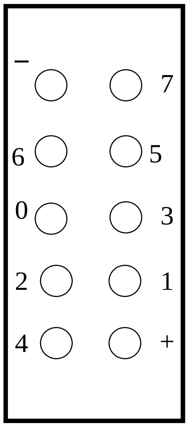


Fig. 4.37: Pin Out Diagram of DE1- SoC ADC Pin Header

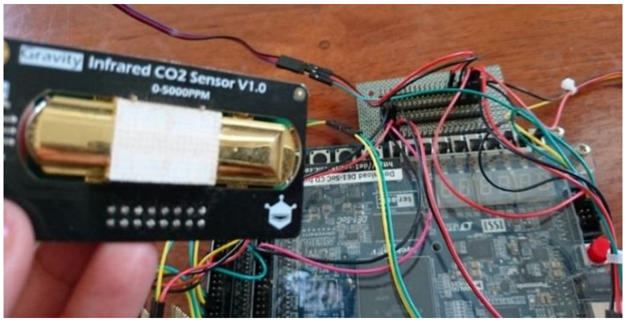


Fig. 4.38: Wiring of CO_2 Sensor SKU: SEN0219 to FPGA Board

1. chip select signal from the FPGA
2. serial clock signal from the FPGA
3. serial data signal from the FPGA
4. serial data signal from the ADC.

The ADC driver provides a sample of data from the ADC in every 67 cycles of the 50MHz clock. The CO_2 sensor consists of three pins such as Vcc (4.5 − 5.5V), GND, and signal pin which provides the analog output (0.4 − 2V). The power Vcc pin of CO_2 sensor is attached to the power source of FPGA at the pin 29 of GPIO 0 (JP1) and the GND pin of CO_2 sensor is connected to the GND of FPGA at the pin 30 of GPIO 0 (JP1) respectively via power array as shown in Fig. 4.38. The signal pin is connected to pin 2 of ADC header of FPGA board. The signal interface of the CO_2 sensor and the SPI ADC are implemented in Qsys and controlled by DE1- SoC board which accumulates the 8-channel analog signals and 24-channel digital signals circularly.

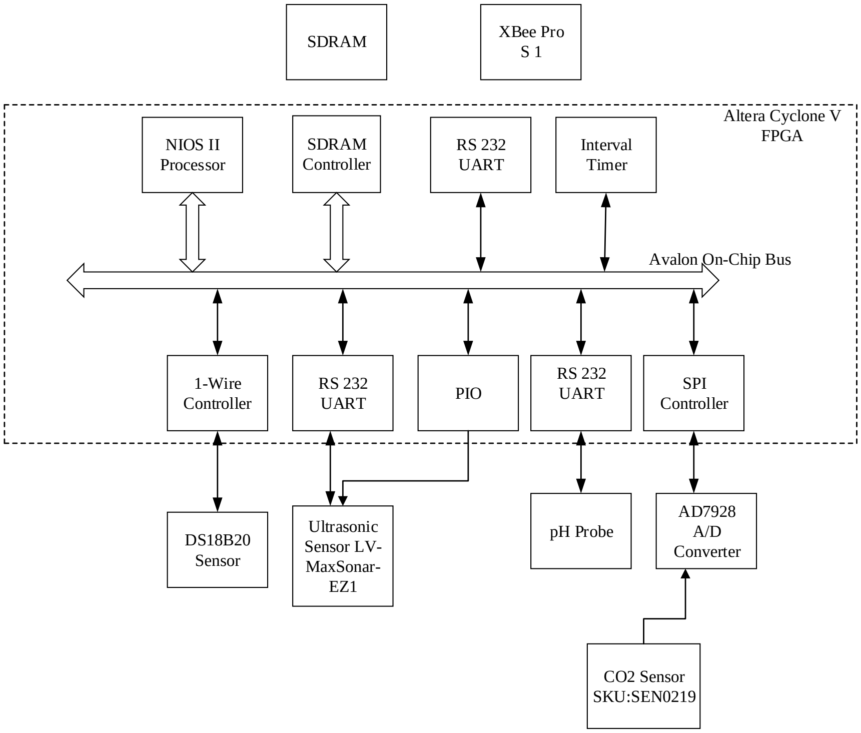


Fig. 4.39: Structure Diagram of SOPC System after Implementing CO_2 Sensor

The collected data is saved in the integrated SRAM on the interface device. The collected data of CO_2 concentration is sent to monitoring PC through XBee Pro S1 RF module. The structural diagram of the SOPC system after implementing CO_2 sensor is shown in Fig. 4.39.

4.8.6 Turbidity Sensor SKU: SEN0189

Although the turbidity sensor SKU: SEN0189 offers both analog and digital signal output modes, in this wireless WQM system, the analog output mode is utilized for its analog output. The turbidity sensor consists of three pins such as Power, GND, and signal as shown in Fig. 4.40. In the proposed design of the WQM system, the analog output (0V − 4.5V) is used. The ADC AD7928 is used for converting the analog output to digital voltage output. The power (Vcc) pin of turbidity sensor is connected to the power source of FPGA at the pin 29 of GPIO 0 (JP1) and the GND pin of turbidity sensor is connected to the GND of FPGA at the pin 30 of GPIO 0 (JP1) respectively via power array.

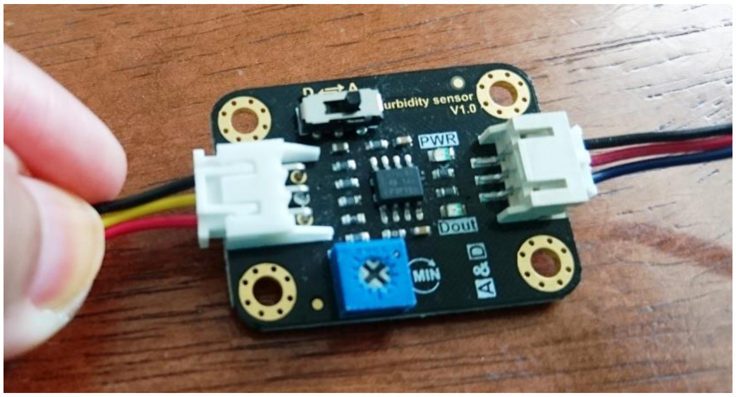


Fig. 4.40: Turbidity Sensor SKU: SEN0189 Circuit

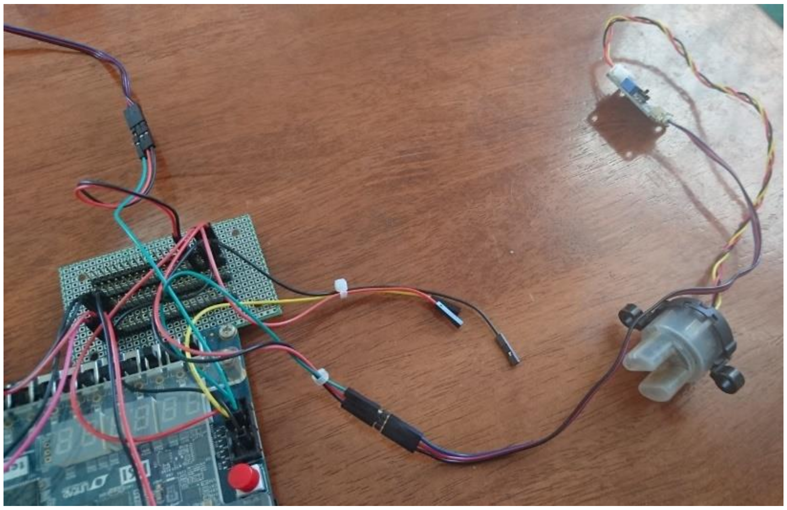


Fig. 4.41: Wiring of Turbidity Sensor

The signal pin out of turbidity sensor is connected to pin 4 of the ADC header of FPGA board. The wiring connection of the turbidity sensor to FPGA is shown in Fig. 4.41. The signal interface of the turbidity sensor and the ADC are devised with Qsys tool and controlled by the DE-1 SoC design board which collects the 8-channel analog signals and 24-channel digital signals circularly. The collected data is saved in the integrated SRAM on the interface device. In terms of data transmission, the collected data of the turbidity level is transmitted to the monitoring PC through wireless communication via the XBee Pro S1 RF module. When the turbidity sensor is installed, all water parameter sensors of reconfigurable WSN design in the IoT environment for WQM are implemented.

4.9 System Architecture of Reconfigurable WSN Design for WQM in IoT Environment

The Atlas pH sensor kit, 1-wire digital thermometer DS18B20, the XBee communication module and the ultrasonic sensor are associated with DE 1-SoC FPGA system via one-wire, UART and RS232 interfaces, respectively. The analog output of the CO_2 sensor and Turbidity sensor are digitized by the AD7928 ADC and are connected to the FPGA system by the SPI controller. In the Qsys Tool, the schematic diagrams of the CO_2 sensor and Turbidity sensor are not displayed in the system contents for their analog functions. Alternatively, the "adc-spi" represents the Qsys designs of the CO_2 sensor and Turbidity sensor in Qsys Tool. The ultrasonic sensor and pH sensor are integrated with RS232 and the default state is TX line UART mode. The default baud rate is 9600, 8 bits, one stop bit, has no flow control and no parity. The temperature sensor of the DS18B20 interconnects over a 1-wire bus, therefore only one data line (and ground) is required for the communication with the microprocessor. In the embedded system, the SPI bus assists the microprocessor to communicate with the off-chip sensors, memory, and control devices. The SPI is fabricated for linking the on-chip processors and peripherals together into a SOPC. The SOPC system is commonly composed of the components, the Nios-II processor and some peripherals. The SPI transmits the inbound data through the UART interface to the processor when the gateways collect the transmitted data from sensor nodes.

An interface protocol, Avalon bus is fabricated to link on-chip processors and peripherals together into the SOPC. The port connections between the master and slave components and the timing are defined by the Avalon bus. Via the Avalon, the Nios-II processor is linked to its embedded peripherals such as on-chip random access memory (RAM), UART, PIO, JTAG, SPI, UART/RS232 serial port, synchronous dynamic random-access memory (SDRAM) controller and Timer. The multiple slave devices such as SPI, UART, and general-purpose input/output (GPIO) and custom logic are connected to the Avalon on-chip bus.

The Nios-II processor is a general purpose configurable soft-core processor and it consists of a 32-bit CPU and a combination of peripherals and memory on a single chip. The SDRAM synchronizes itself with the timing of the CPU. When the requested data is ready, the exact clock cycle is identified by the memory controller consequently. The

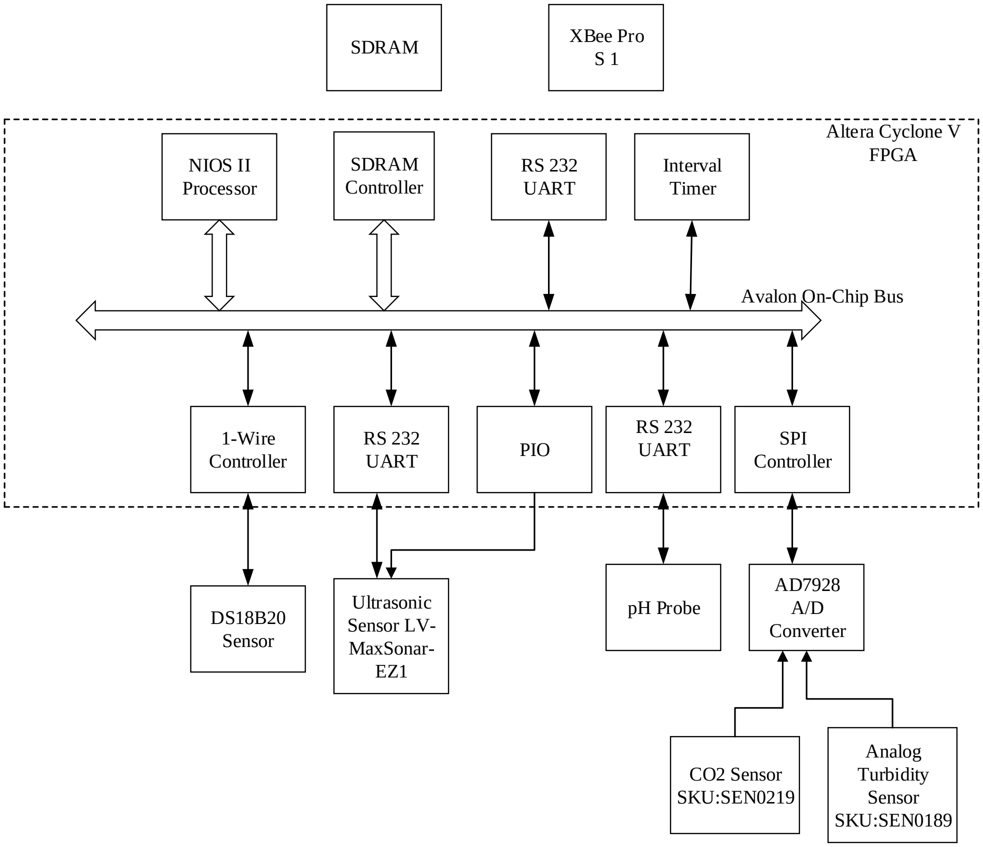


Fig. 4.42: System Architecture of Reconfigurable WQM System

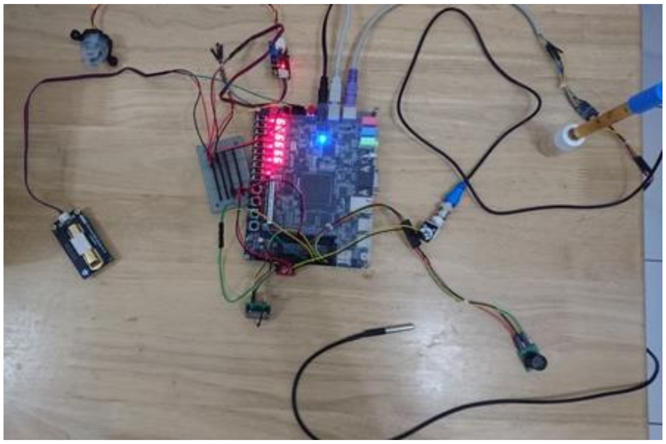


Fig. 4.43: Experimental Set-up of Transmitter Unit of WQM System

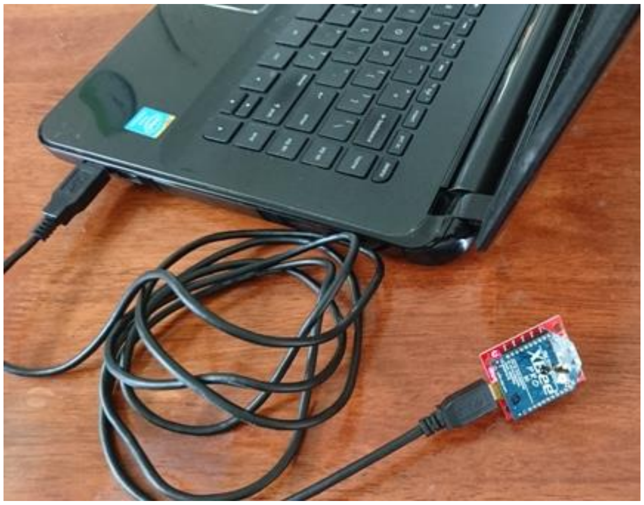


Fig. 4.44: Experimental Set-up of Receiver Unit of WQM System

Nios-II processor is connected to the Zigbee hardware for wireless transmission by means of the UART. The accumulated data is saved in the integrated SRAM on the interface device. In terms of wireless communication, the computed data is transmitted to monitoring PC through XBee Pro S1 RF module. The complete system is assigned in VHDL and accesses the IP cores available in Altera Qsys. The system architectural diagram of the proposed reconfigurable WSN design for WQM in the IoT environment is shown in Fig. 4.42. The experimental set-up of the transmitter unit of the reconfigurable WSN design is shown in Fig. 4.43 where WQM in IoT environment is powered 12 V, either by battery or direct line power supply, and that of the receiver unit is presented in Fig. 4.44.

4.10 Software Implementation

In section 4.2, it has been discussed that the Altera Quartus II Software is the major FPGA development tool. The software program of the transmitter unit of the proposed reconfigurable WSN design for WQM in IoT environment is composed of the C codes running over the embedded Nios-II processor within the FPGA processor and VHDL codes. Python codes are programmed on both the LiClipse and the Grafana to exhibit the wirelessly collected data of water parameters on PC. To view the LiClipse and the Grafanam, the Linux mode is used as the operating system of monitoring PC.

4.10.1 Software Program for Receiver Module

When the wireless water quality data is transmitted to the monitoring PC, the Pythons codes in the LiClipse are used to exhibit the collected data of water temperature, the distance between the sensor and water surface, the turbidity of water, CO_2 on the water surface and water pH. A part of Python codes to display the resulted data of water parameters are as follows:

```
fp = open('WQM.csv', 'w')
fp.write('Temperature (C)' + ', ' + 'Distance (mm)' + ', ' + 'Turbidity' + ', ' + 'CO2 (ppm)'
+ ', ' + 'pH' + '\n');
fp.close()
def dump(data): #define callback function
        if 'source_addr' in data:
     #addr     = data['source_addr']
     #rssi_val = int(data['rssi'].encode('hex'),16)
     #addr_int = int(addr.encode('hex'), 16)
     rf_data = data['rf_data']
     #fid     = data['options']
     # now = datetime.datetime.now().strftime('%Y-%m-%d_%H:%M:%S')
     #now = datetime.datetime.now()
       sample_list = rf_data.split(';')
        if len(sample_list) == 5:
        temp =     float(sample_list[0])
        dist =    (float(sample_list[1]) * 25.4)
        turb =     float(sample_list[2])
        co2  = int(float(sample_list[3]))
        ph   =     float(sample_list[4])
        print 'T: ' + str(temp) + ', D: ' + str(dist) + ', U: ' + str(turb)+ ', C: ' + str(co2) + ',
P: ' + str(ph)
       fp = open('WQM.csv', 'a')
        fp.write(str(temp) + ', ' + str(dist) + ', ' + str(turb)+ ', ' + str(co2) + ', ' + str(ph) +
'\n');
        fp.close()
```

```
point = [
    {
        "measurement": "water_quality",
        "tags": {
            "host"  : "Node-A",
            "region": "Sarawak"
        },
        "fields": {
        "Temperature"    : temp,
        "Water Level"    : dist,
        "Turbidity"      : turb,
        "Carbon Dioxide" : co2,
        "pH Level"       : ph,
        }
    }
```

where *T* is the temperature, *D* is the distance between the sensor and water surface, *U* is the turbidity of water, *C* is the CO_2 on the surface of water and *P* is the pH value of water. The water parameters are displayed both in the Liclipse and in the Grafana which is installed in PC to observe time series data. The software program of receiver module runs in Linux mode of the operating system.

4.11 Flow Chart of Software Analysis

When the reconfigurable WSN design in the IoT environment for the WQM system is powered on, the data from each sensor node is detected one after another using time multiplexing. Then, the data collected from all 5 sensors is converted to 8 bits binary. The detected data is saved in the integrated SRAM on the interface device. The data is sent to the XBee transmitter module. The flow chart of the wireless WQM system is illustrated in Fig. 4.45.

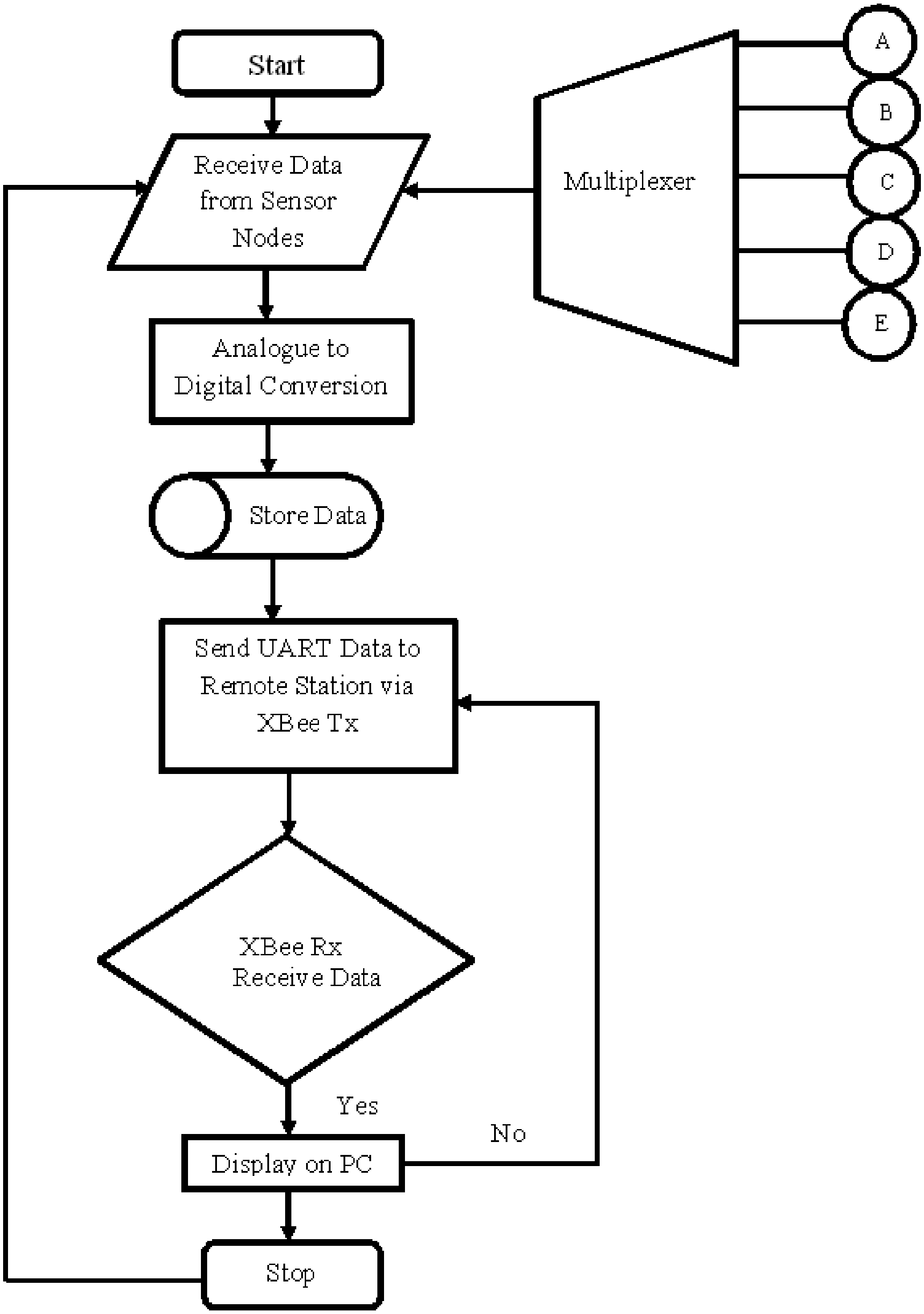

Start
Receive Data from Sensor Nodes
Multiplexer
A
B
C
D
E
Analogue to Digital Conversion
Store Data
Send UART Data to Remote Station via XBee Tx
XBee Rx Receive Data
Yes
No
Display on PC
Stop

Fig. 4.45: Flow Chart of Software of WQM System

4.12 XCTU Installation

X-CTU [245] software is downloaded from www.digi.com/xctu and installed. The configuration of X-CTU is done referring to the User Guide. The Digi's X-CTU Software is installed on PC by double-clicking the “setup_X-CTU.exe” file. The RF module is mounted to the sensor interface FPGA board, then, the module assembly is connected to the PC. The X-CTU software is launched on the PC and the “PC Settings” tab is selected. The baud and parity settings of the Com Port matching those of the RF module are verified.

4.13 Grafana Installation

To install Grafana [246], the Ubuntu Server 16.04 LTS 64bit and InfluxDB are installed initially referring to https://docs.influxdata.com/influxdb/v0.12/introduction/installation. Then, Grafana Beta Version is installed and the user profile is created using user ID and password at http://localhost:3000/. A dashboard is created referring to the Grafana User Manual.

4.14 Total Resource Utilisation of FPGA Board

The implementation parameters of FPGA system of the proposed reconfigurable WSN design in the IoT environment for WQM can be checked in “DE1_SoC_SDRAM_RTL_Test.fit.summary” file. The total resource utilization to design the FPGA board of the reconfigurable WSN system in the IoT environment for WQM is shown in Table 4.10.

Table 4.10: Total Resource Utilisation of FPGA Board

Parameter	**Used resources**
Logic utilization (in ALMs)	1,724 / 32,070 (5%)
Total registers	2662
Total pins	149 / 457 (33 %)
Total block memory bits	83,072 / 4,065,280 (2 %)
Total PLLs	1 / 6 (17 %)
Max Clock Frequency	58.75 MHz

4.15 Summary

In this chapter, the design of the hardware and software implementation of a reconfigurable WSN system for WQM in the IoT environment has been discussed in detail. In the first section, the hardware design implementation in Qsys Tools of Quartus II has been discussed. The compiling of Qsys design to Quartus II has been presented and the building of a project in Quartus II using C program has also been described. The latter part of the chapter discusses the hardware implementation of the components to the FPGA board. The software programs of the receiver unit of the proposed reconfigurable WSN design for WQM in the IoT environment have been discussed and the flow chart of the software system has been also presented. The total resources utilization of FPGA board is presented in the final part of this chapter.

Chapter 5

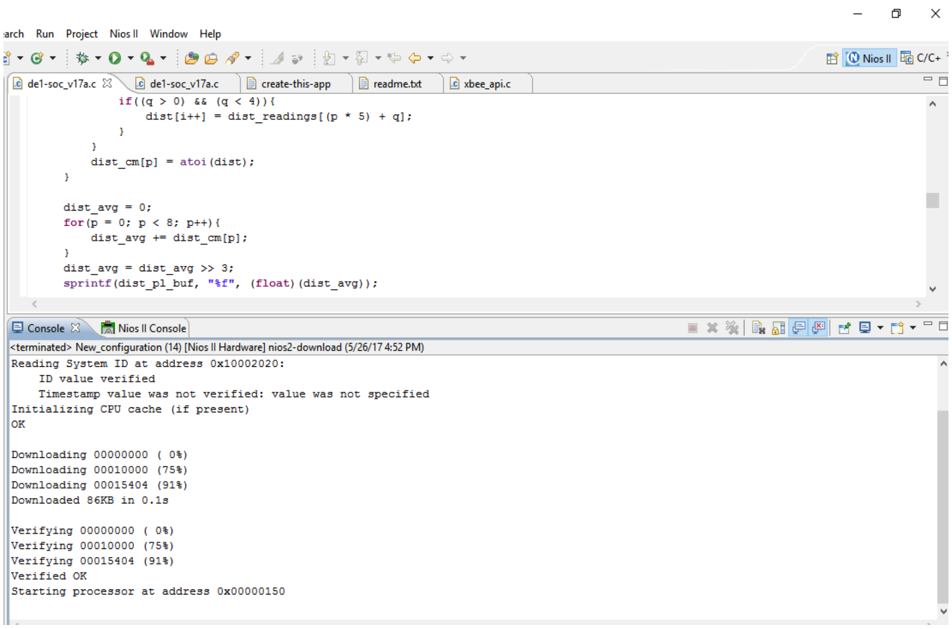
Experimental Measurement

In this chapter, the operating procedures of the designed device to measure the water quality is presented. The measurements and results of the reconfigurable WSN design for WQM in the IoT environment are reported. The data of water parameters which are monitored two times a day for fifteen (15) days in the morning and in the evening are presented. The discussion on the results of the designed device is presented with the analysis.

5.1 Operation of the Designed Device

After implementing the hardware and software components, the designed device is tested to measure the water parameters of tap water before actually measuring water parameters of lake water. The operating procedures for measuring the water parameters are as follows:

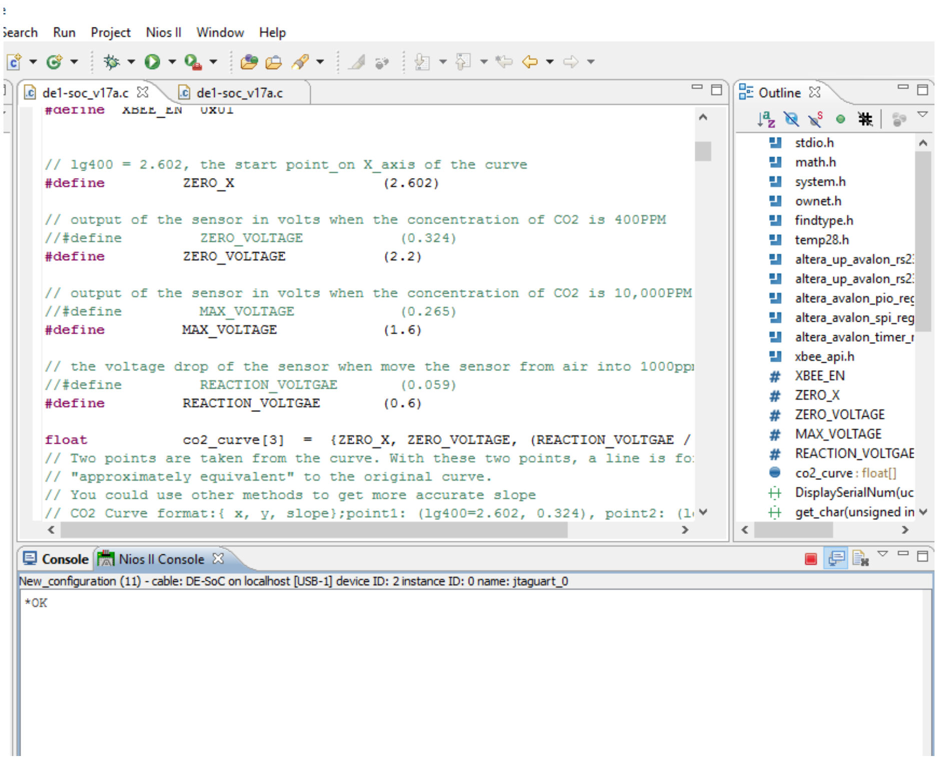
1. Connect the FPGA board and XBee receiver module to PC via USB cable, and both FPGA and PC are switched on.
2. Launch the Quartus II 13.1 (64-bit) Web Edition software and the project "DE2_Basic_Computer.qpf" is opened.
3. Launch the Quartus II "Programmer".
4. Launch Nios-II Software Build Tools for Eclipse from "Tools".
5. Select the workspace as "DE1_SoC_DS18B20_v.1.8\ws".
6. Select the output existing project files "WQM_17" and "WQM_17.bsp" and run configuration.
7. In "Run configurations" wizard, select the project name as "WQM_17" and check the project ".elf" file name.
8. Check the processor, cables and byte stream devices and refresh target connection.
9. Run the program. The program will be verified as shown in Fig. 5.1. Run for three times until no error message is observed in Nios II Console as shown in Fig. 5.2.
10. Check the XBee transmitting messages in Nios II Console as shown in Fig. 5.3.



```
            if((q > 0) && (q < 4)){
                dist[i++] = dist_readings[(p * 5) + q];
            }
        }
        dist_cm[p] = atoi(dist);
    }

    dist_avg = 0;
    for(p = 0; p < 8; p++){
        dist_avg += dist_cm[p];
    }
    dist_avg = dist_avg >> 3;
    sprintf(dist_pl_buf, "%f", (float)(dist_avg));
```

```
<terminated> New_configuration (14) [Nios II Hardware] nios2-download (5/26/17 4:52 PM)
Reading System ID at address 0x10002020:
    ID value verified
    Timestamp value was not verified: value was not specified
Initializing CPU cache (if present)
OK

Downloading 00000000 ( 0%)
Downloading 00010000 (75%)
Downloading 00015404 (91%)
Downloaded 86KB in 0.1s

Verifying 00000000 ( 0%)
Verifying 00010000 (75%)
Verifying 00015404 (91%)
Verified OK
Starting processor at address 0x00000150
```

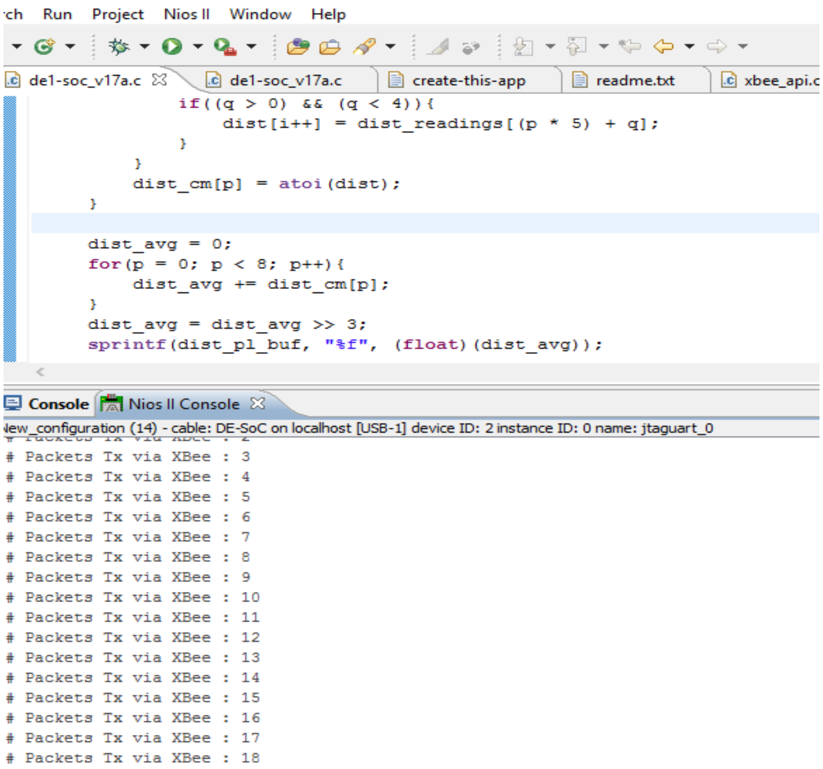
Fig. 5.1: Program Verification



```
#define  XBEE_EN   0x01


// lg400 = 2.602, the start point_on X_axis of the curve
#define         ZERO_X                   (2.602)

// output of the sensor in volts when the concentration of CO2 is 400PPM
//#define         ZERO_VOLTAGE             (0.324)
#define         ZERO_VOLTAGE             (2.2)

// output of the sensor in volts when the concentration of CO2 is 10,000PPM
//#define         MAX_VOLTAGE              (0.265)
#define         MAX_VOLTAGE              (1.6)

// the voltage drop of the sensor when move the sensor from air into 1000ppı
//#define         REACTION_VOLTGAE         (0.059)
#define         REACTION_VOLTGAE         (0.6)

float           co2_curve[3]  =  {ZERO_X, ZERO_VOLTAGE, (REACTION_VOLTGAE /
// Two points are taken from the curve. With these two points, a line is fo:
// "approximately equivalent" to the original curve.
// You could use other methods to get more accurate slope
// CO2 Curve format:{ x, y, slope};point1: (lg400=2.602, 0.324), point2: (l
```

```
New_configuration (11) - cable: DE-SoC on localhost [USB-1] device ID: 2 instance ID: 0 name: jtaguart_0
*OK
```

Fig. 5.2: Checking Messages in Nios-II Console



```
                if((q > 0) && (q < 4)){
                    dist[i++] = dist_readings[(p * 5) + q];
                }
            }
            dist_cm[p] = atoi(dist);
        }

        dist_avg = 0;
        for(p = 0; p < 8; p++){
            dist_avg += dist_cm[p];
        }
        dist_avg = dist_avg >> 3;
        sprintf(dist_pl_buf, "%f", (float)(dist_avg));
```

```
New_configuration (14) - cable: DE-SoC on localhost [USB-1] device ID: 2 instance ID: 0 name: jtaguart_0
# Packets Tx via XBee : 2
# Packets Tx via XBee : 3
# Packets Tx via XBee : 4
# Packets Tx via XBee : 5
# Packets Tx via XBee : 6
# Packets Tx via XBee : 7
# Packets Tx via XBee : 8
# Packets Tx via XBee : 9
# Packets Tx via XBee : 10
# Packets Tx via XBee : 11
# Packets Tx via XBee : 12
# Packets Tx via XBee : 13
# Packets Tx via XBee : 14
# Packets Tx via XBee : 15
# Packets Tx via XBee : 16
# Packets Tx via XBee : 17
# Packets Tx via XBee : 18
```

Fig. 5.3: XBee Transmitting Messages in Nios II Console

Since the FPGA designed based on the SRAM when the FPGA is switched off, the SRAM data is erased. When the FPGA has been switched on again, the configuration is needed again. As the FPGA needs to configure again after switching off, it cannot automatically operate as soon as the switch is turned on. The advantage of using FPGA is the reconfigurable function. Hence, the user can update the Qsys design anytime and the IP is reusable. The pH sensor probe is error-prone depending on the position of the probe in the solution; therefore, the sensing area of the probe needs to be correctly immersed in the solution. The pH of tap water is initially measured by the designed device. The sensing area of the pH probe is immersed in tap water without touching the walls of the container as shown in Fig. 5.4. The pH value of tap water is 7.00 as the tap water in Miri City is suitable for drinking as shown in Fig. 5.5.

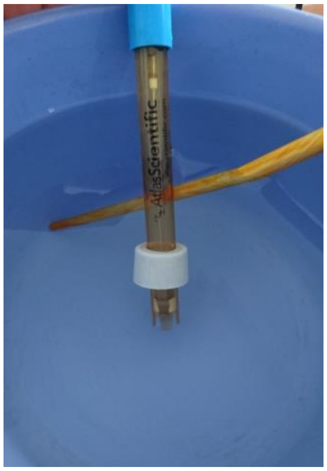
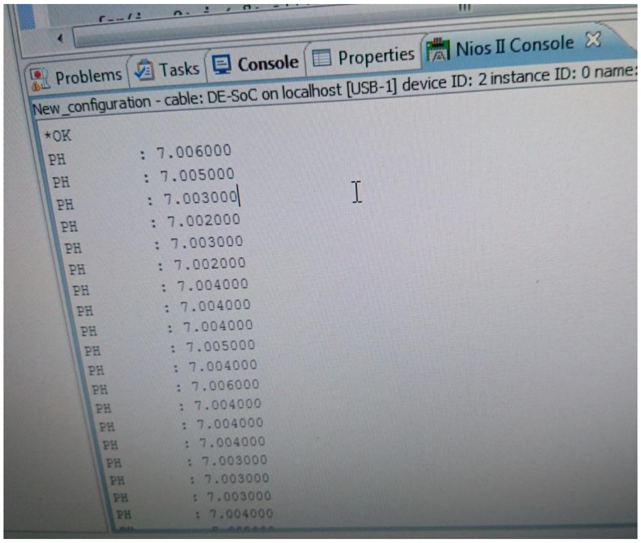


Fig. 5.4: Sensing Area of pH probe immersed in Water



New_configuration - cable: DE-SoC on localhost [USB-1] device ID: 2 instance ID: 0 name:

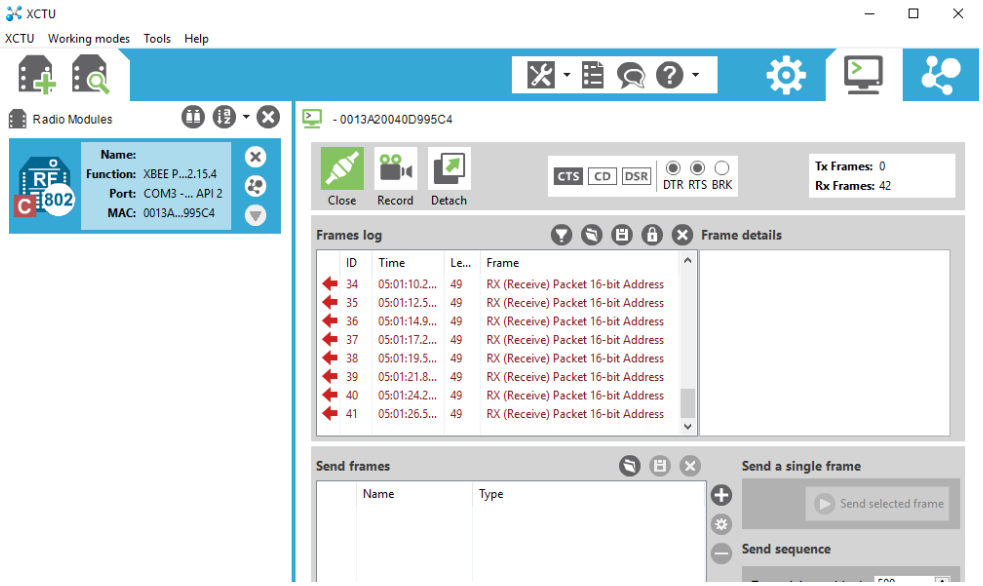
```
*OK
PH        : 7.006000
PH        : 7.005000
PH        : 7.003000
PH        : 7.002000
PH        : 7.003000
PH        : 7.002000
PH        : 7.004000
PH        : 7.004000
PH        : 7.004000
PH        : 7.005000
PH        : 7.004000
PH        : 7.006000
PH        : 7.004000
PH        : 7.004000
PH        : 7.004000
PH        : 7.003000
PH        : 7.003000
PH        : 7.003000
PH        : 7.004000
```

Fig. 5.5: Result of pH Value of Tap Water

5.1.1 Displaying Receiving Data of Water Parameters in X-CTU

To display the real-time data of water parameters in X-CTU, the following procedures must be done.

1. Launch the X-CTU to view the wireless water parameters in X-CTU.
2. Add the Radio Module port “COM 3” which is the XBee receiver module USB port.



- 0013A20040D995C4

Name:
Function: XBEE P...2.15.4
Port: COM3 -... API 2
MAC: 0013A...995C4

Tx Frames: 0
Rx Frames: 42

ID	Time	Le...	Frame
34	05:01:10.2...	49	RX (Receive) Packet 16-bit Address
35	05:01:12.5...	49	RX (Receive) Packet 16-bit Address
36	05:01:14.9...	49	RX (Receive) Packet 16-bit Address
37	05:01:17.2...	49	RX (Receive) Packet 16-bit Address
38	05:01:19.5...	49	RX (Receive) Packet 16-bit Address
39	05:01:21.8...	49	RX (Receive) Packet 16-bit Address
40	05:01:24.2...	49	RX (Receive) Packet 16-bit Address
41	05:01:26.5...	49	RX (Receive) Packet 16-bit Address

Name	Type

Fig. 5.6: Checking Receiving Data

3. When COM 3 is added, check the "transmit interval box" as 500 ms. Double-click the frame icon.
4. The received data appears in the frame. Click on one serial data, the Hexa value of water parameters appears as shown in Fig. 5.6.
5. Scroll down to see the ASCII value of water parameters.

5.1.2 Displaying Receiving Data of Water Parameters in LiClipse

For displaying the real-time data of water parameters in LiClipse, the following procedures must be done.

1. In the Linux mode of the monitoring PC, open terminals for sudo influx.
2. Launch the LiClipse, and run Python codes, the water parameters will be displayed, and the samples are refreshed every 5s.

5.1.3 Displaying Receiving Data of Water Parameters in Grafana

To view time series data in Grafana, browse http://localhost:3000/ using Mozilla browser and launch "WQM v1.0" dashboard.

Table 5.1: Water Quality of Tap Water at Room Temperature

Temperature **(° C)**	**Ultrasonic Sensor Level** **(m)**	**pH**	**CO_2** **(ppm)**	**Turbidity** **(V)**
30	1.6	7.00	700	4.2 (≈0.5 NTU)

5.2 Power Supply

The proposed system of reconfigurable WSN design for WQM in the IoT environment is powered by either direct line of electricity or the rechargeable Lithium Polymer battery of 6000 mAh. The power supply of direct current (DC) 12V/2A is needed to power the sensor interface FPGA SoC board. When the FPGA board is powered by the rechargeable Lithium Polymer battery 6000 mAh, the proposed reconfigurable smart WQM system can be operated for a maximum of 6 hours continuously. The lithium-polymer battery is compact and portable with a weight of 0.875 kg. The battery provides a high discharge rate with 35C (210A) and high reliability. The nominal voltage of the battery is 22.2V. [248].

5.3 Testing Designed Device

The experimental set-up of the designed device is shown in Fig. 5.7. The designed device is tested by measuring water data of pipe water at room temperature. The ultrasonic sensor is placed on the dining table and it is tested by placing high and low places. The distance between the ultrasonic sensor and the floor is measured. The test results are presented in Table 5.1. The temperature of water at room temperature is measured at 30˚C, the altitude of the ultrasonic sensor from the floor is measured at 1.6m, the pH of tap water is measured at 7, the CO_2 is measured at 700 ppm and the turbidity sensor measures the 4.2 V (≈0.5 NTU). Fig. 5.8 shows the changes in water level result due to the altitude of the ultrasonic sensor when the sensor is placed higher and lower.

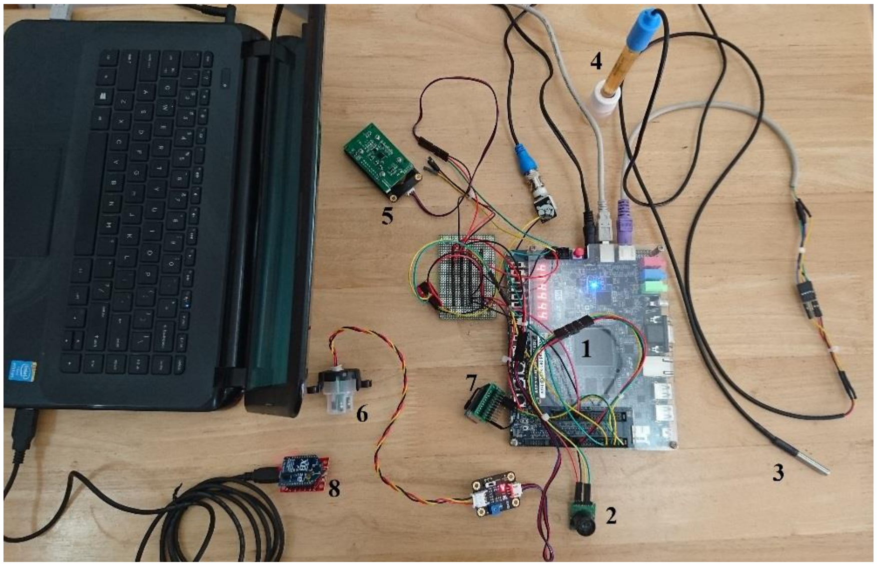


Fig. 5.7: Experimental Set-up of Designed WQM System:
1. FPGA board, 2.Ultrasonic sensor, 3.Digital Thermometer sensor, 4.pH sensor, 5. CO_2 sensor, 6. Turbidity sensor, 7. XBee RF transmitter module, 8. XBee RF receiver module.

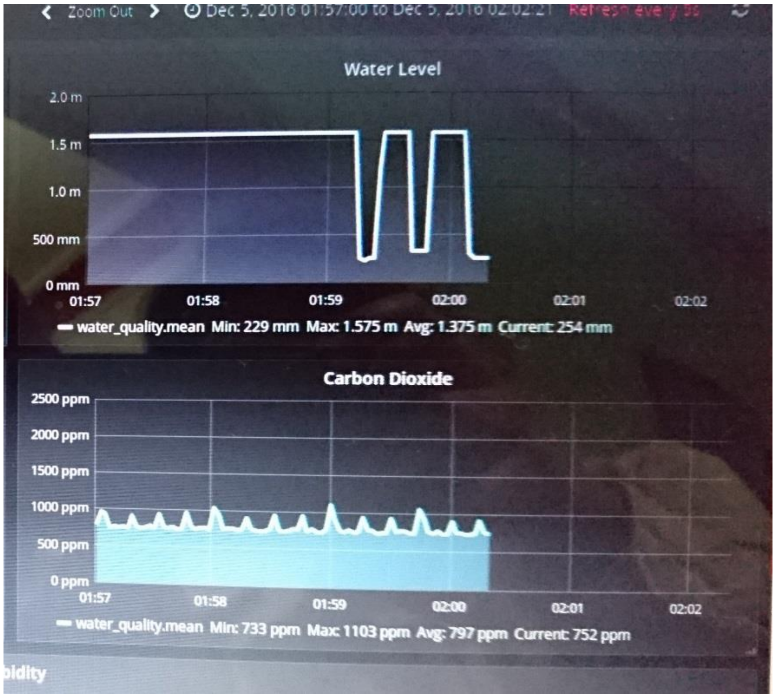


Fig. 5.8: Testing WQM System in Room Temperature

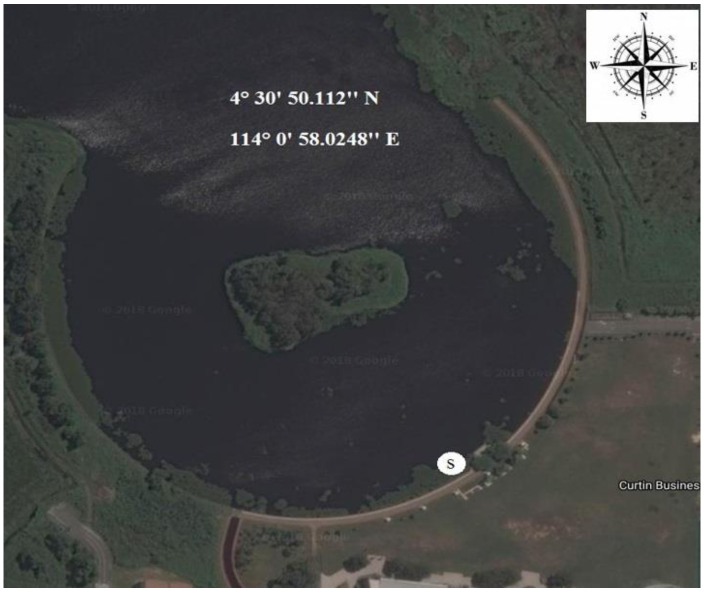


Fig. 5.9: Study Site of WQM System

5.4 Measuring Water Parameters of the Lake

The quality of lake surface water was measured from Curtin Lake which is located in Curtin University, Sarawak campus, Miri, East Malaysia in April 2017. In Fig. 5.9, the study site is marked as “S” (latitudes 4° 30' 50.112" N and longitudes 114° 0' 58.0248" E) is located in northern Sarawak, in the Borneo Island. The study site is located 18 m above the sea level according to the GPS Status & Toolbox [246]. In Miri, there are a few industrial areas such as palm oil, timber processing plants, offshore oil platforms, and shipbuilding industries. The climate of the study area is typically tropical; annual southwest (April to October) and northeast (October to February) monsoons. Recently, the highest recorded temperature in April has been 38°C, with the lowest recorded temperature 22°C [250]. The ash from burning bushes near the study site is one of the causes of the presence of suspended particles in water bodies. The study site is located near a large body of water. The sensor interfaced DE-1 SoC FPGA board (Fig. 5.10) is placed at the bank of the lake to measure the physical parameters of water at 5:30 pm.

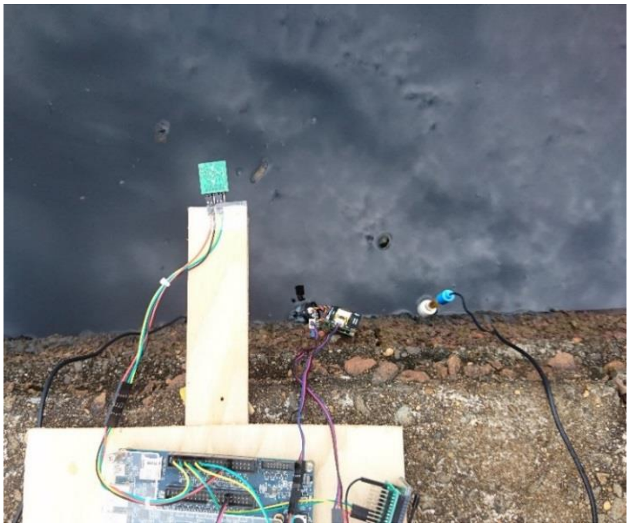


Fig. 5.10. Sensor Interface DE-1 SoC FPGA Board

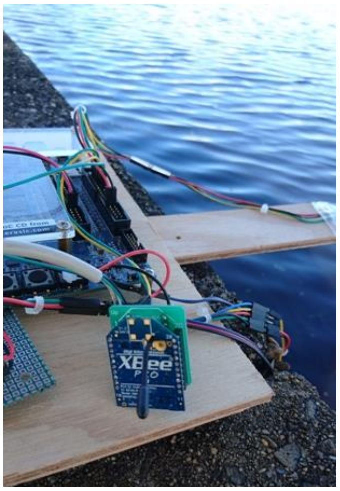
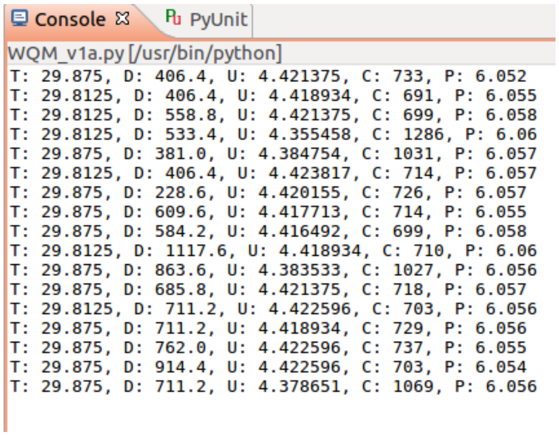


Fig. 5.11: XBee Pro S1 Transmitter

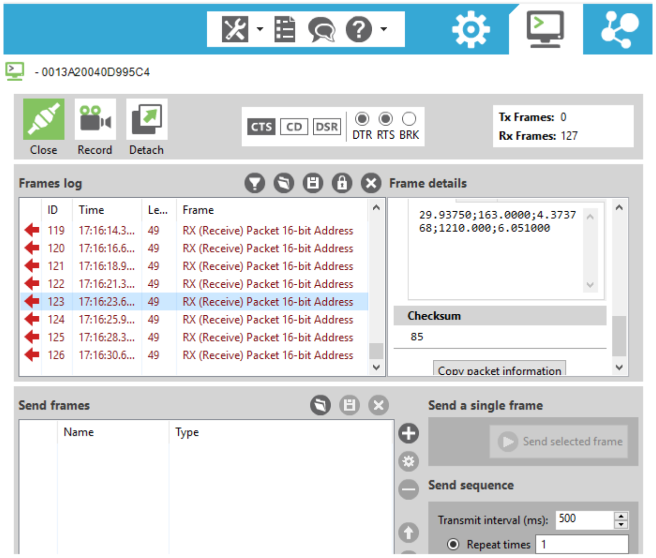
Fig. 5.12: XBee Pro S1 Receiver



WQM_v1a.py [/usr/bin/python]

```
T: 29.875, D: 406.4, U: 4.421375, C: 733, P: 6.052
T: 29.8125, D: 406.4, U: 4.418934, C: 691, P: 6.055
T: 29.8125, D: 558.8, U: 4.421375, C: 699, P: 6.058
T: 29.8125, D: 533.4, U: 4.355458, C: 1286, P: 6.06
T: 29.875, D: 381.0, U: 4.384754, C: 1031, P: 6.057
T: 29.8125, D: 406.4, U: 4.423817, C: 714, P: 6.057
T: 29.875, D: 228.6, U: 4.420155, C: 726, P: 6.057
T: 29.875, D: 609.6, U: 4.417713, C: 714, P: 6.055
T: 29.875, D: 584.2, U: 4.416492, C: 699, P: 6.058
T: 29.8125, D: 1117.6, U: 4.418934, C: 710, P: 6.06
T: 29.875, D: 863.6, U: 4.383533, C: 1027, P: 6.056
T: 29.875, D: 685.8, U: 4.421375, C: 718, P: 6.057
T: 29.8125, D: 711.2, U: 4.422596, C: 703, P: 6.056
T: 29.875, D: 711.2, U: 4.418934, C: 729, P: 6.056
T: 29.875, D: 762.0, U: 4.422596, C: 737, P: 6.055
T: 29.875, D: 914.4, U: 4.422596, C: 703, P: 6.054
T: 29.875, D: 711.2, U: 4.378651, C: 1069, P: 6.056
```

Fig. 5.13: Measurement of the designed device at the study site



- 0013A20040D995C4

Tx Frames: 0
Rx Frames: 127

ID	Time	Le...	Frame
119	17:16:14.3...	49	RX (Receive) Packet 16-bit Address
120	17:16:16.6...	49	RX (Receive) Packet 16-bit Address
121	17:16:18.9...	49	RX (Receive) Packet 16-bit Address
122	17:16:21.3...	49	RX (Receive) Packet 16-bit Address
123	17:16:23.6...	49	RX (Receive) Packet 16-bit Address
124	17:16:25.9...	49	RX (Receive) Packet 16-bit Address
125	17:16:28.3...	49	RX (Receive) Packet 16-bit Address
126	17:16:30.6...	49	RX (Receive) Packet 16-bit Address

```
29.93750;163.0000;4.3737
68;1210.000;6.051000
```

Checksum
85

Name	Type

Transmit interval (ms): 500
● Repeat times 1

Fig. 5.14: Measurement of Water Parameters at Study Site

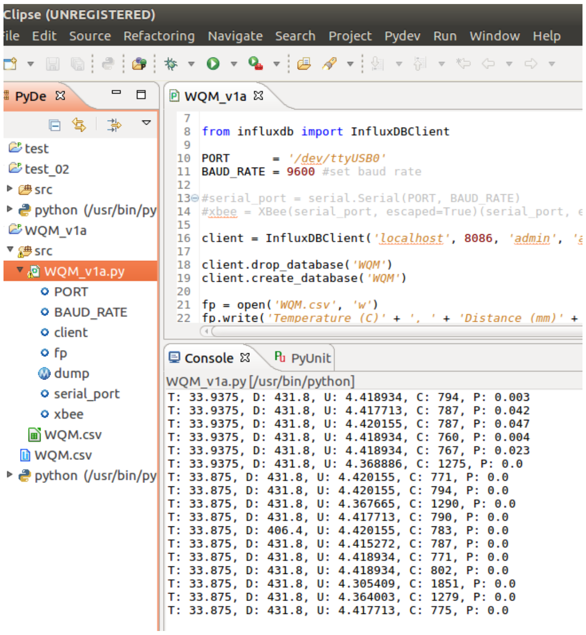
The pH sensor probe, turbidity sensor, and temperature sensor are submerged in the lake water and CO_2 sensor is positioned 5cm above the water surface. The ultrasonic sensor is placed at the same level as the lake bank because the measurement between the water level and the sensor will be measured. The XBee Pro transmitter (Fig. 5.11) which is interfaced with the FPGA board and XBee Pro receiver (Fig. 5.12) which is connected to the monitoring PC is shown. The XBee Pro receiver is placed 86 m far away from the XBee Pro transmitter unit. Before the actual data collection, the designed device is tested at the study site and water parameters are measured according to the operating procedures which have been discussed in Section 5.1. When the sensor panel is powered on, the sensors are initiated to examine the relevant parameter of water data. Then, the detected water parameters are sent wirelessly to the observing PC which is placed 86 m away from the sensor interfaced FPGA panel. The measurements of water parameters are presented in LiClipse and X-CTU. In LiClipse, as shown in Fig. 5.13, the temperature of water “T” is shown in degree Celcius, the distance between the sensor placed at the same level with lake bank and the water surface “D” is shown in millimeter, the turbidity of water “U” is shown in output voltage, the CO_2 above the water surface “C” is shown in ppm, and the pH value “P” is shown in 2 decimal values. However, in the X-CTU, the water level is measured in inches in order to observe the different distance measurement as shown in Fig. 5.14.

5.5 Errors due to Mal-position of pH probe

The pH value of water shows 0.00 in LiClipse as shown in Fig. 5.15 and the pH value shows 2.65 in X-CTU as shown in Fig. 5.16. The mal-position of the pH probe in the water causes incorrect measurements of pH. When the sensing area of the pH probe is not fully immersed in the water, the reading of pH sensor achieves incorrect readings of pH probe.

5.6 Results Collected for Fifteen (15) Consecutive Days

The water quality of the lake is collected at the same site twice a day for fifteen (15) consecutive days (12 April – 26 April). Depending on the weather condition, the water data is measured around 7:00 – 7:30 am in the morning and at 5:00-5:00 pm in the evening during this period of time. On a rainy day, the water data is collected after the rain stopped. The data is being monitored continuously and displayed in real time since the default of the system is set in continuous mode and the data is refreshed every 5s. Hence, the time interval between each sample of water quality is 5s. The water quality of the lake is collected at the same site for two times a day on fifteen (15) consecutive days. The total of 2537 measurements is made in fifteen (15) days. As the console window of

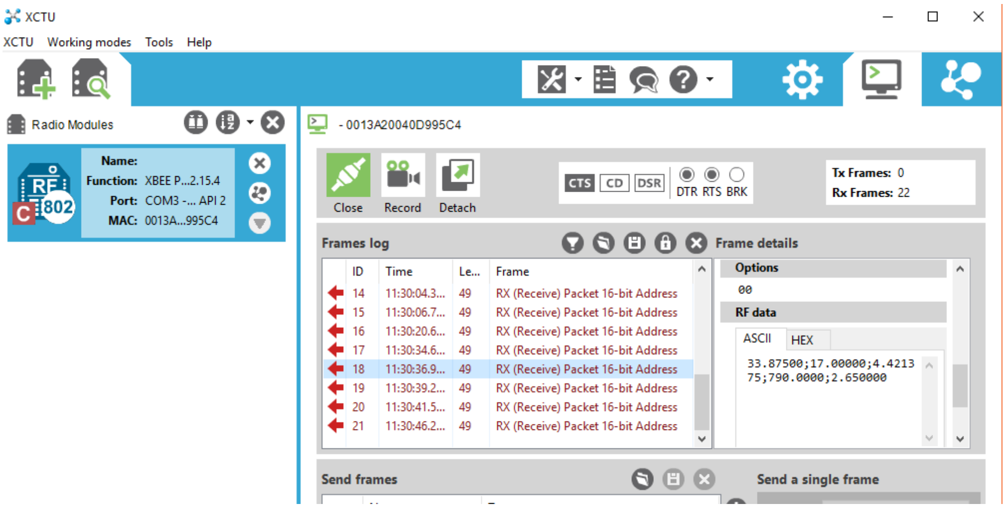


```
from influxdb import InfluxDBClient

PORT      = '/dev/ttyUSB0'
BAUD_RATE = 9600 #set baud rate

#serial_port = serial.Serial(PORT, BAUD_RATE)
#xbee = XBee(serial_port, escaped=True)(serial_port, e

client = InfluxDBClient('localhost', 8086, 'admin', 'a

client.drop_database('WQM')
client.create_database('WQM')

fp = open('WQM.csv', 'w')
fp.write('Temperature (C)' + ', ' + 'Distance (mm)' +
```

WQM_v1a.py [/usr/bin/python]

```
T: 33.9375, D: 431.8, U: 4.418934, C: 794, P: 0.003
T: 33.9375, D: 431.8, U: 4.417713, C: 787, P: 0.042
T: 33.9375, D: 431.8, U: 4.420155, C: 787, P: 0.047
T: 33.9375, D: 431.8, U: 4.418934, C: 760, P: 0.004
T: 33.9375, D: 431.8, U: 4.418934, C: 767, P: 0.023
T: 33.9375, D: 431.8, U: 4.368886, C: 1275, P: 0.0
T: 33.875, D: 431.8, U: 4.420155, C: 771, P: 0.0
T: 33.875, D: 431.8, U: 4.420155, C: 794, P: 0.0
T: 33.875, D: 431.8, U: 4.367665, C: 1290, P: 0.0
T: 33.875, D: 431.8, U: 4.417713, C: 790, P: 0.0
T: 33.875, D: 406.4, U: 4.420155, C: 783, P: 0.0
T: 33.875, D: 431.8, U: 4.415272, C: 787, P: 0.0
T: 33.875, D: 431.8, U: 4.418934, C: 771, P: 0.0
T: 33.875, D: 431.8, U: 4.418934, C: 802, P: 0.0
T: 33.875, D: 431.8, U: 4.305409, C: 1851, P: 0.0
T: 33.875, D: 431.8, U: 4.364003, C: 1279, P: 0.0
T: 33.875, D: 431.8, U: 4.417713, C: 775, P: 0.0
```

Fig. 5.15: Displaying Error Value of pH in LiClipse



Name:
Function: XBEE P...2.15.4
Port: COM3 -... API 2
MAC: 0013A...995C4

- 0013A20040D995C4

Tx Frames: 0
Rx Frames: 22

ID	Time	Le...	Frame
14	11:30:04.3...	49	RX (Receive) Packet 16-bit Address
15	11:30:06.7...	49	RX (Receive) Packet 16-bit Address
16	11:30:20.6...	49	RX (Receive) Packet 16-bit Address
17	11:30:34.6...	49	RX (Receive) Packet 16-bit Address
18	11:30:36.9...	49	RX (Receive) Packet 16-bit Address
19	11:30:39.2...	49	RX (Receive) Packet 16-bit Address
20	11:30:41.5...	49	RX (Receive) Packet 16-bit Address
21	11:30:46.2...	49	RX (Receive) Packet 16-bit Address

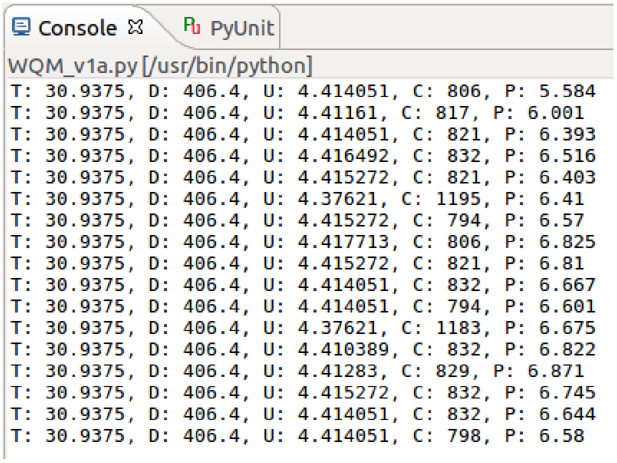
Options
00
RF data
ASCII HEX

```
33.87500;17.00000;4.4213
75;790.0000;2.650000
```

Fig. 5.16: Displaying Error Value of pH in X-CTU

LiClipse shows the seventeen (17) samples of water data at a time, seventeen (17) samples of measured water quality data of the lake are shown in Fig. 5.17.

The temperature of water shows 30.9375° Celsius, the distance between the ultrasonic sensor and water level is 431.8 mm for all seventeen (17) samples measured at



```
WQM_v1a.py [/usr/bin/python]
T: 30.9375, D: 406.4, U: 4.414051, C: 806, P: 5.584
T: 30.9375, D: 406.4, U: 4.41161, C: 817, P: 6.001
T: 30.9375, D: 406.4, U: 4.414051, C: 821, P: 6.393
T: 30.9375, D: 406.4, U: 4.416492, C: 832, P: 6.516
T: 30.9375, D: 406.4, U: 4.415272, C: 821, P: 6.403
T: 30.9375, D: 406.4, U: 4.37621, C: 1195, P: 6.41
T: 30.9375, D: 406.4, U: 4.415272, C: 794, P: 6.57
T: 30.9375, D: 406.4, U: 4.417713, C: 806, P: 6.825
T: 30.9375, D: 406.4, U: 4.415272, C: 821, P: 6.81
T: 30.9375, D: 406.4, U: 4.414051, C: 832, P: 6.667
T: 30.9375, D: 406.4, U: 4.414051, C: 794, P: 6.601
T: 30.9375, D: 406.4, U: 4.37621, C: 1183, P: 6.675
T: 30.9375, D: 406.4, U: 4.410389, C: 832, P: 6.822
T: 30.9375, D: 406.4, U: 4.41283, C: 829, P: 6.871
T: 30.9375, D: 406.4, U: 4.415272, C: 832, P: 6.745
T: 30.9375, D: 406.4, U: 4.414051, C: 832, P: 6.644
T: 30.9375, D: 406.4, U: 4.414051, C: 798, P: 6.58
```

Fig. 5.17: Seventeen (17) Samples of Water Quality Data

a time. The water level and temperature of water do not show changes in a short time. The Carbon dioxide, pH and the turbidity of water are measured at different times. The output of the turbidity sensor is shown in voltage. The output voltage of the turbidity sensor is inversely proportional to the NTU, the increase in voltage gives the decrease in NTU. Therefore, the maximum voltage gives the minimum level of turbidity. The water quality of the lake is measured in the morning on the first day of data collection.

5.7 Overall Analysis of Water Data Collected for Fifteen days

The overall descriptive analysis of the received water data collected in the morning and in the evening is discussed in the following sections.

5.7.1 Overall Analysis of Water Data Collected in Morning Time

The mean value of water data of each parameter collected in the morning for fifteen (15) days is taken for analysis. The overall analysis of water data is presented in Table 5.2.

5.7.2 Overall Analysis of Water Data Collected in Evening Time

The mean value of the water data of each parameter collected in the evening for fifteen (15) days is taken for analysis. The overall analysis of water data is presented in Table 5.3.

Table 5.2: Overall Descriptive Analysis of Water Data Collected in Morning for Fifteen (15) days

Parameters	**Range**	**Maximum**	**Minimum**	**Mean**	**Standard Deviation**
Temperature (˚C)	1.718	30.961	29.243	30.245	0.575
Distance between water surface and sensor (m)	0.267	0.559	0.292	0.402	0.059
Turbidity (V) (NTU)	1.698	4.407 (<0.5 NTU)	2.709 (>500 NTU)	3.772 (<500 NTU, >50 NTU)	0.694
CO_2 (ppm)	256.483	898.565	642.082	793.171	98.646
pH	1.222	6.785	5.563	6.174	0.46

Table 5.3: Overall Descriptive Analysis of Water Data Collected in Evening for Fifteen (15) days

Parameters	**Range**	**Maximum**	**Minimum**	**Mean**	**Standard Deviation**
Temperature (˚C)	3.576	33.661	30.085	31.597	1.102
Distance between water surface and sensor (m)	0.356	0.61	0.254	0.414	130.762
Turbidity (V) (NTU)	1.527	4.428 (<0.5 NTU)	2.901 (>500 NTU)	3.6294 (<500 NTU, >50 NTU)	0.742
CO_2 (ppm)	253.118	898.518	645.4	750.52	87.5
pH	0.961	6.223	5.262	5.743	0.37

5.8 Discussion on the Data of fifteen (15) days

From the collected water data for fifteen (15) days, the pH of lake water was found to be slightly acidic. It was observed that the water turbidity increased, and pH value decreased after the rain. The temperature of lake water in the morning ranges from 29.24˚C – 30.96˚C. In the evening, the temperature of lake water was in the range of 30.08˚C and 33.66˚C. The temperature is approximate to the environmental air temperature of the study area. The lowest temperature 29.1875˚C is measured at 7:30 am on 15 April, and the highest temperature 33.8125˚C is measured at 5:00 pm on 25 April. The hydrogen ions pH value of the lake water samples collected in the morning ranges 5.56 – 6.78. For the evening collected data, the pH value was in the range of 5.25 and 6.22. The lowest pH value 5.241 is measured at 5:00 pm on 21 April and the highest pH value of 7.02 is measured at 7:30 am on 26 April.

The CO_2 on the surface of the water is observed in the range of 642.08 ppm and 898.5 ppm in the morning, and 645.4 ppm and 898.5 ppm in the evening. The lowest CO_2 value 529 ppm is measured at 5:30 pm on 20 April, and the highest CO_2 value 1656 ppm is measured at 5:45 pm on 14 April and at 5:30 pm on 16 April. The turbidity of the lake water on the morning was found in the range of less than 0.5 NTU to greater than 500 NTU. The mean value of turbidity was in the range of less than 500 NTU and greater than 50 NTU. It can be concluded that the turbidity of the water is between 50 NTU and 500 NTU. The distance from the water surface to the lake bank was observed in the range of 0.292 m and 0.559 m in the morning, and 0.254 m and 0.61 m in the evening. The lowest distance 0.254 m and the highest distance 0.635 m from the water surface to the lake bank is measured at 7:30 am on 25 April.

5.9 Analysis of the Relations of Water Temperature, CO_2 and pH

Since water data is not significantly changed in the morning time or the evening time, the random two hundred (200) cases of water data collected for fifteen (15) days are selected to analyse in Statistical Package for the Social Science (SPSS) Version 20 for the correlation of temperature, CO_2, and pH. As the temperature and pH of water are not significantly different in one hour, the temperature is divided by ten (10) and the pH is divided by six (6).

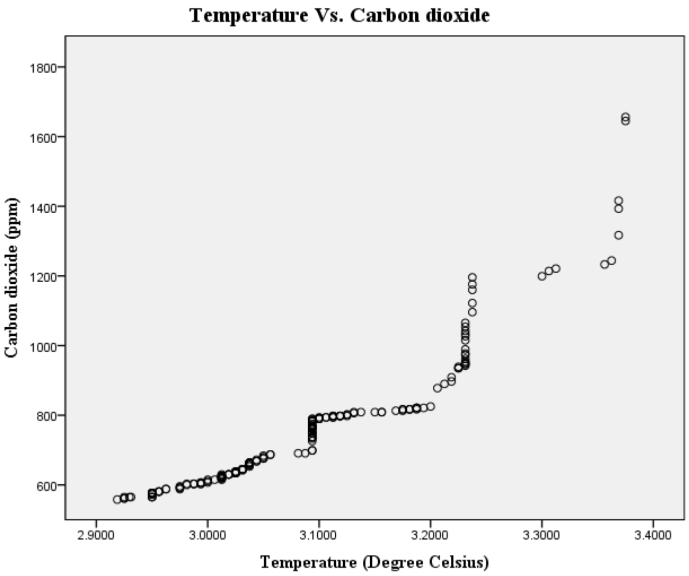


Fig. 5.18: Two Hundred Cases of Temperature and CO_2

Table 5.4: Correlations between Temperature and CO_2

Correlations

		T2	C
T2	Pearson Correlation	1	.942**
	Sig. (2-tailed)		.000
	N	200	200
C	Pearson Correlation	.942**	1
	Sig. (2-tailed)	.000	
	N	200	200

**. Correlation is significant at the 0.01 level (2-tailed).

5.9.1 Temperature Vs CO_2

Temperature of the water is important to be monitored frequently because it effects water chemistry. The chemical reactions of water commonly increase when temperature increases [251]. Fig. 5.18 presents the Temperature Vs CO_2 graph. The correlations between the temperature and CO_2 are depicted in Table 5.4. The correlation is positive

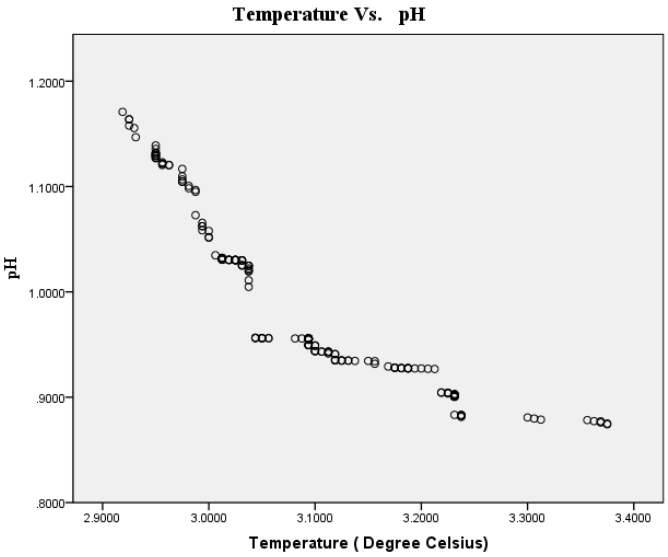


Fig. 5.19: Two Hundred Cases of Temperature and pH

Table 5.5: The correlations between temperature and pH

Correlations

		T2	pH
T2	Pearson Correlation	1	-.910**
	Sig. (2-tailed)		.000
	N	200	200
pH	Pearson Correlation	-.910**	1
	Sig. (2-tailed)	.000	
	N	200	200

**. Correlation is significant at the 0.01 level (2-tailed).

and significant at the 0.01 level (two-tailed). The temperature affects the DO in the water, therefore warmer water has less DO and higher CO_2 than the cooler water [251].

5.9.2 Temperature Vs pH

Fig. 5.19 shows the Temperature Vs pH graph. The correlations between the temperature and pH are depicted in Table 5.5. The Pearson correlation between the temperature and pH is negative and significant at the 0.01 level (two-tailed).

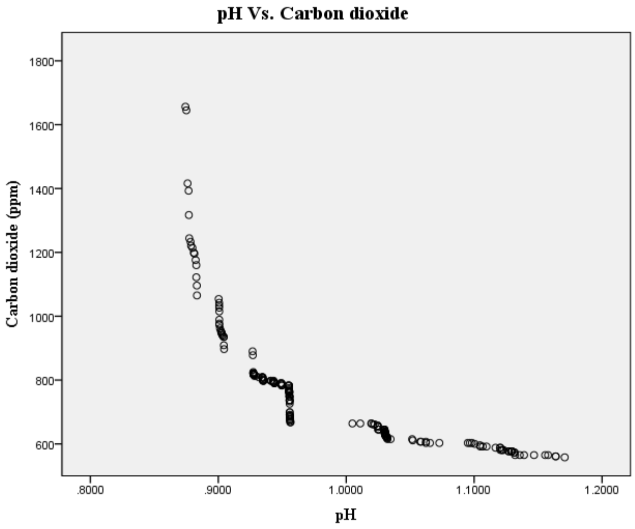


Fig. 5.20: Two Hundred Cases of CO_2 and pH

Table 5.6: Correlations between CO_2 and pH

Correlations

		C	pH
C	Pearson Correlation	1	-.794**
	Sig. (2-tailed)		.000
	N	200	200
pH	Pearson Correlation	-.794**	1
	Sig. (2-tailed)	.000	
	N	200	200

**. Correlation is significant at the 0.01 level (2-tailed).

The concentration of chemical substances in the water depends on the temperature. During day time, the rate of photosynthetic activity affects the concentration of CO_2 which causes the increase in pH. The pH of water decreases when temperature increases [251].

5.9.3 CO_2 Vs pH

Fig. 5.20 shows the CO_2 Vs pH graph. The correlations between the CO_2 and pH are depicted in Table 5.6. The Pearson correlation between the CO_2 and pH is negative and significant at the 0.01 level (two-tailed). The studies show that CO_2, pH, and temperature are directly related to each other since the pH is determined by the bicarbonate-carbonate level and free CO_2 [252, 253]. The temperature of water influences water chemistry like dissolved oxygen, solubility, pH, conductivity, etc. [254]. In general, the oxygen concentration of water decreases when the temperature of water increases [255]. Due to the exposure to air and biological activities, fluctuations in pH can be found during the day. The increase in photosynthesis activity causes a higher pH with the decreasing CO_2 level during the day [256].

5.10 Verification of the Measurements of the Design Device

The distance between the lake bank and water surface was measured using a measuring tape to validate the ultrasonic sensor results as shown in Fig. 5.21. No significant difference was found between the measurement of the ultrasonic sensor and that of measuring tape. The mean value of the resulted data of pH and water temperature collected by the proposed reconfigurable WSN design is compared with the laboratory measurements in Table 5.7. The temperature difference between the laboratory measurement and the designed device was found 0.15. The difference for pH is found at 0.4. From the research paper of a team of researchers who studied the water quality of this lake [257], it was found that the laboratory measurement of pH value of the lake water was in the range of 5.92–6.82. The laboratory measurement of the water temperature was found in the range of 26˚C – 33˚C. The measurement of the designed device's pH data was in the range of 5.56–6.78 in the morning and 5.26–6.22 in the evening. According to the studies of the relation of water temperature, pH, and CO_2, it can be concluded that the resulted data of reconfigurable WSN design for WQM is reliable.

Table 5.7: Comparison between Laboratory and WQM System Measurements

Parameter	**Laboratory**	**Designed device**
Temperature (˚C) Mean Value	30.77	30.92
pH Mean Value	6.3	5.91

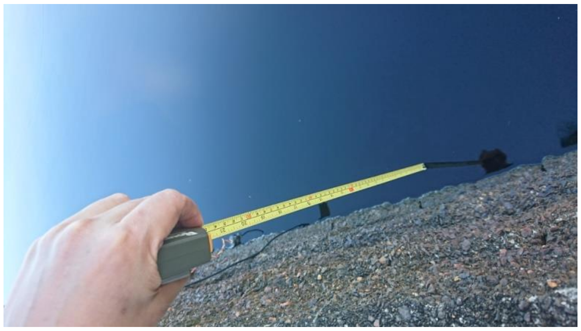
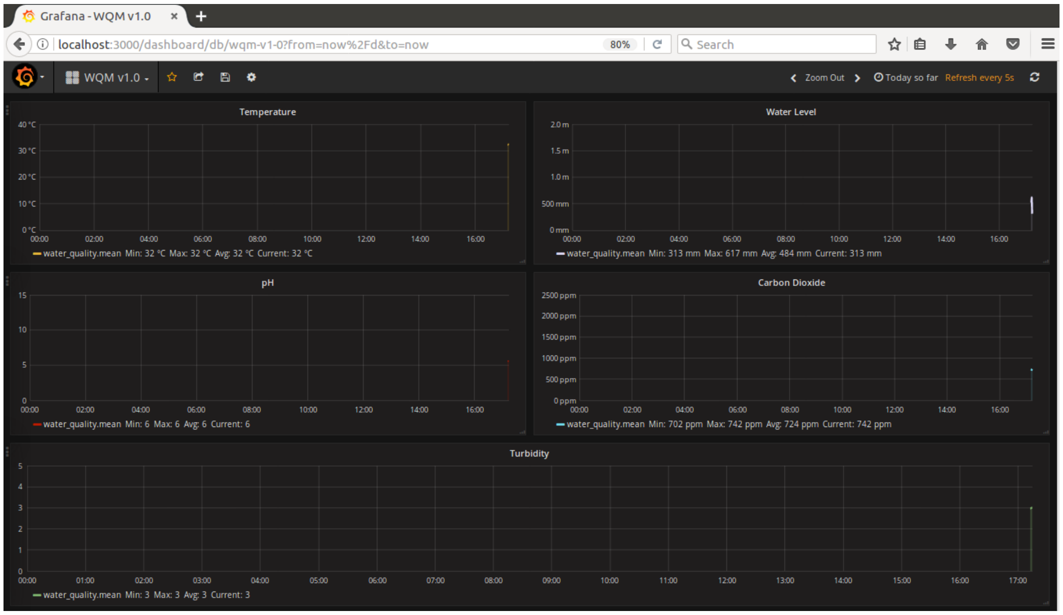


Fig. 5.21: Result Validation of WQM System using Measuring Tape



localhost:3000/dashboard/db/wqm-v1-0?from=now%2Fd&to=now

water_quality.mean Min: 32 °C Max: 32 °C Avg: 32 °C Current: 32 °C

water_quality.mean Min: 313 mm Max: 617 mm Avg: 484 mm Current: 313 mm

water_quality.mean Min: 6 Max: 6 Avg: 6 Current: 6

water_quality.mean Min: 702 ppm Max: 742 ppm Avg: 724 ppm Current: 742 ppm

water_quality.mean Min: 3 Max: 3 Avg: 3 Current: 3

Fig. 5.22: Five Parameters of Water on Grafana Dashboard

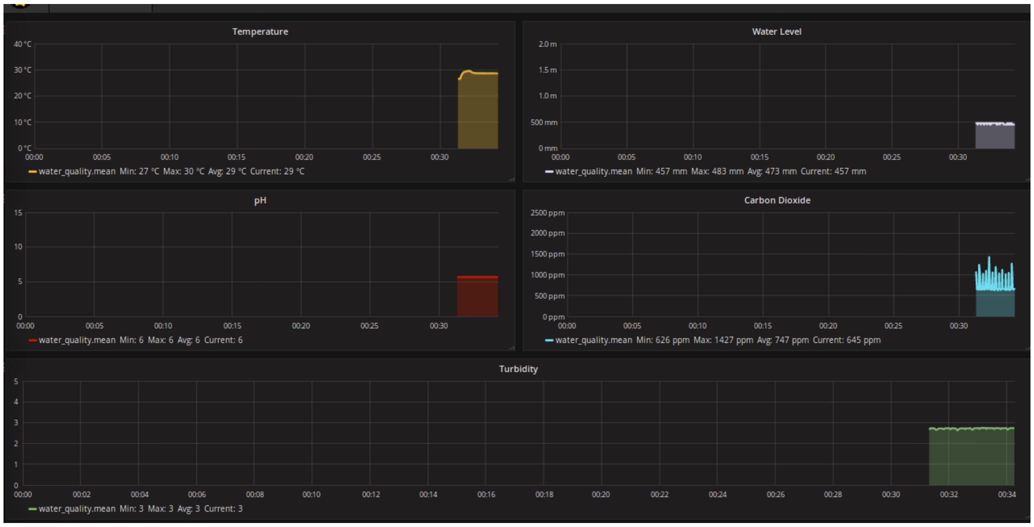


Fig. 5.23: Five Parameters of Water for Six (6) Hours

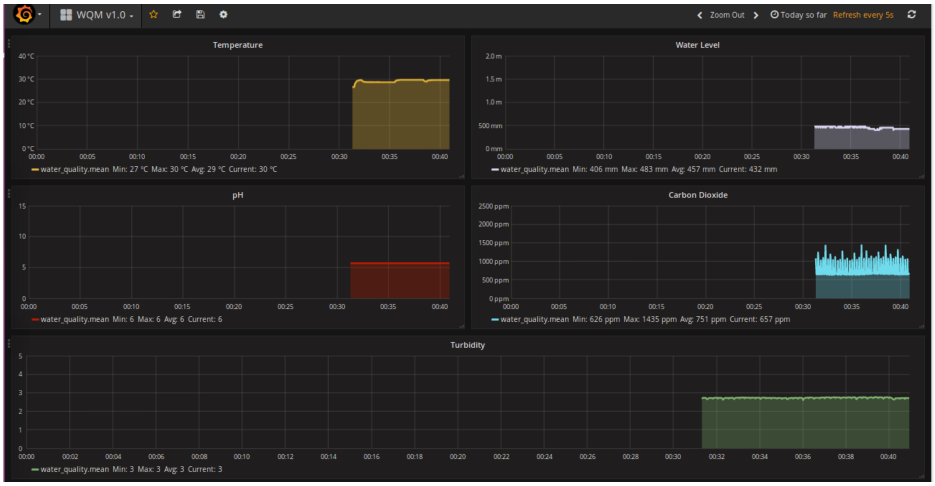


Fig. 5.24: Five Parameters of Water for Seven (7) Hours

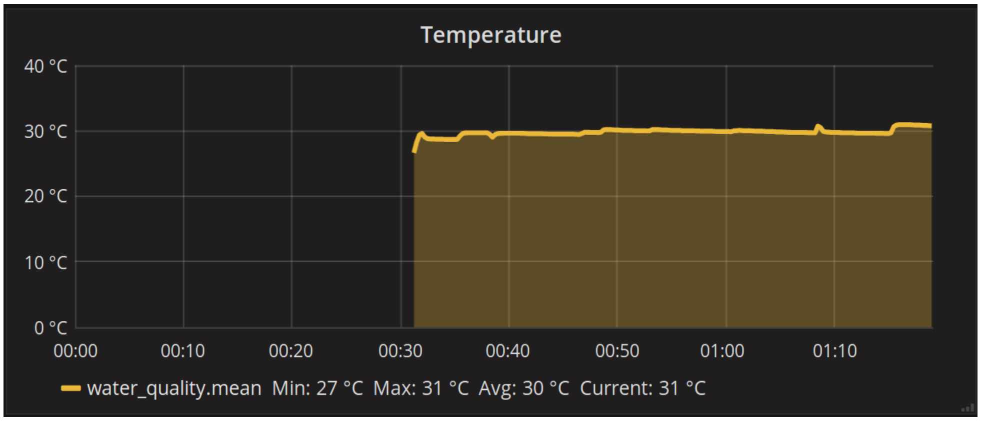


Fig. 5.25: Water Temperature Data for Eight (8) Hours

5.11 Time Series Data of Collected Water Quality

The wirelessly collected data of water parameters are presented on the Grafana. Fig. 5.22 shows the initially collected data of water parameters at 9:15 am. The time series data of water parameters during six (6) hours period is shown in Fig. 5.23 and for seven (7) hours is shown in Fig. 5.24 respectively. The data is detected constantly and presented in real time since the default of the system is set in continuous mode. The interval of the detecting time is selected for 8 hours and the data is regenerated every 5 s. The data of water quality was collected for eight (8) hours. The pH data, the temperature of water, the distance between the water surface and the sensor, CO_2 on the surface of the water, turbidity of water collected for 8 hours is shown in Fig. 5.25 –5.29.

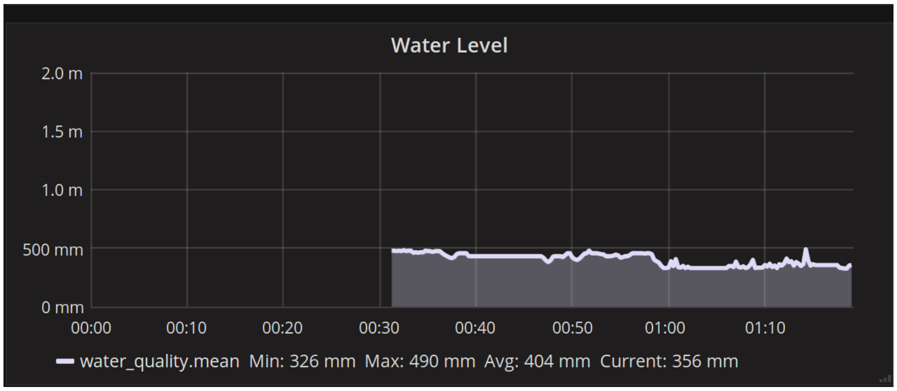


Fig. 5.26: Ultrasonic Sensor Data for Eight (8) Hours

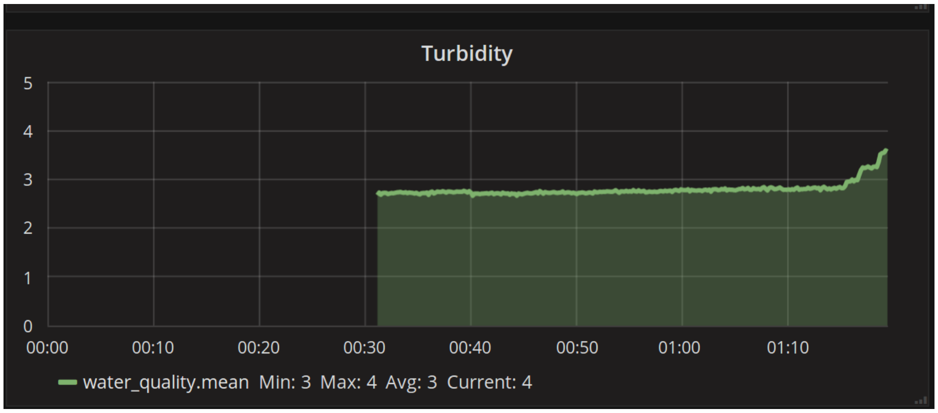


Fig. 5.27: Output Voltage Data of Turbidity Sensor for Eight (8) Hours

5.12 Limitations of the Designed Device

As the pH sensor is error-prone and sensitive for the position and placement of the sensing area of the pH probe, the user needs to be careful with the handling of pH probe. The pH sensor should be calibrated so often to achieve a correct result of pH value. The output of the turbidity sensor offers in voltage, the output voltage is needed to convert to the unit of turbidity NTU. Although the data of XBee Pro S1 mentions that the coverage distance is about 100 m, the proposed design of WQM achieved the maximum distance of 88 m. Since the Lake is situated on the lower land of the campus, the XBee transmitter unit is located lower than the buildings. The XBee receiver could not receive the transmitted data of the transmitting unit when the receiver is placed on the higher land for the position of XBee antenna. Therefore, the receiver unit must be placed at the same ground level as the transmitter unit.

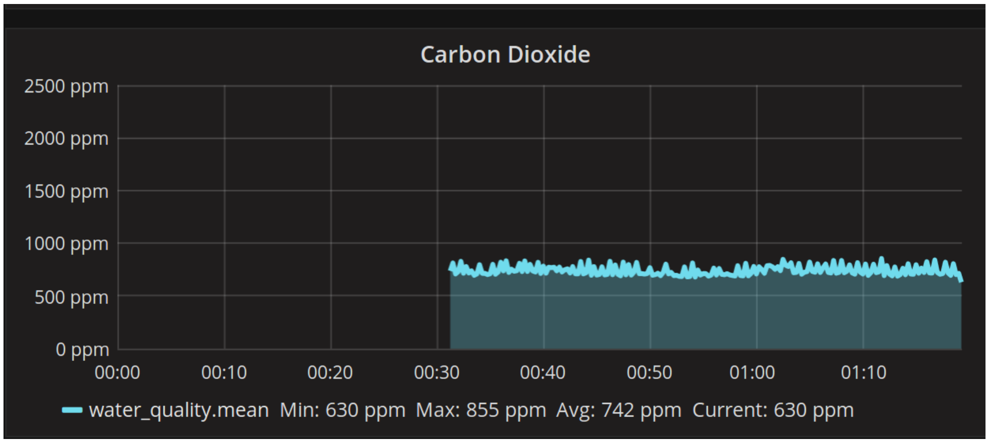


Fig. 5.28: CO_2 Data for Eight (8) Hours

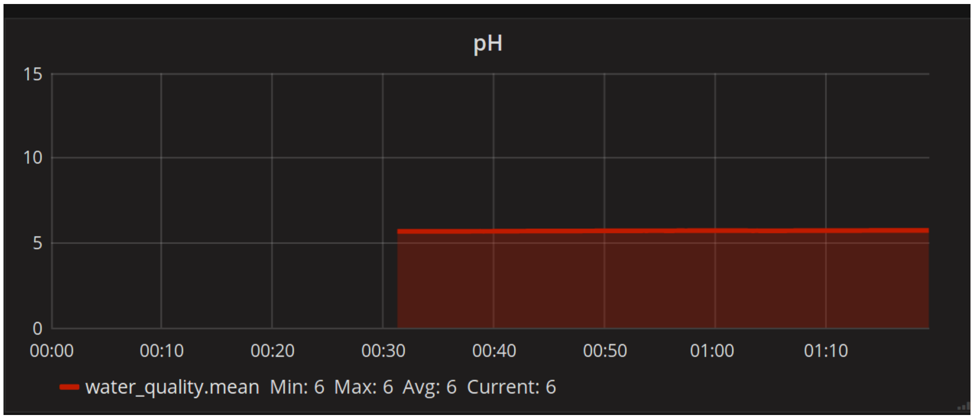


Fig. 5.29: pH Data for Eight (8) Hours

5.13 Summary

The measurements of water quality have been performed. The correct measurements of the water quality have validated that the designed device could provide accurate readings. Hence, the verifications of the concepts have been completed by the results obtained. There are several limitations of the designed device of reconfigurable WSN design for WQM in IoT. Accurate measurements can be obtained in ideal situations.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The main goal of this research is to develop VHDL modules and to design and implement a reconfigurable WSN system for WQM in IoT environment which has beneficial features such as small in size, low-power consumption, portable weight and low-cost. Additionally, it contains the environmental detecting sensors, along with data processing communication capabilities. This phase has been developed and discussed.

The advantages and benefits of the designed reconfigurable WSN system for environmental monitoring have been included. The detailed design and the detailed block diagram of the reconfigurable WSN system for environmental monitoring in the IoT environment have been explained. The design of the hardware system in Qsys Tool has been described. The software design has also been presented. The completed design of the implementation of both hardware and software has been detailed. The features of the water parameter sensors have been discussed.

The proposed system of reconfigurable WSN design for WQM in the IoT environment works on both battery and power supply. The proposed reconfigurable WQM system is powered by either direct line of electricity or the rechargeable Lithium Polymer battery of 6000 mAh. The maximum of 6 hours of operating time is achieved when the system is powered by the Lithium Polymer battery. The designed device is placed at the lake bank to detect the water quality of the lake and the monitoring PC is placed at 86m away from the designed sensor interfaced device to receive the wirelessly transmitted data of the designed device. The operation of the reconfigurable WSN design for WQM in the IoT environment has been completed by the measurement procedures. The measurements of water quality in the morning and evening for 15 consecutive days are highlighted. Depending on the weather condition and environmental changes, the slight changes in water quality data have been obtained. The readings of the reconfigurable WSN design for WQM in the IoT environment are recorded. The values under the different conditions such as after the rain, in the sun, in the cloudy weather are also recorded.

It is found that the values of pH, CO_2, and turbidity of water are slightly different depending on the environmental conditions. The temperature of the water is depending on the temperature of the day. There are some limitations of the designed reconfigurable device although the readings are reliable. The mal-position of the pH sensor probe can cause inaccurate readings. The accurate readings of the pH sensor can be obtained in the correct position of the pH sensing probe. After verifying the real-time data of water parameters, it can be concluded that the system has achieved and accomplished the practicability and reliability in the application for real monitoring purposes.

The time interval of water data detection is modified contingent to the requirement. The proposed system requires high execution speed and continual IP design and the system executes parallel processing. Hence, it offers an opportunity to minimize the operating cost and power consumption, and it exceeds the performance of the common microcontrollers-based WSN design by utilizing the FPGA SoC board. One of the advantages of using FPGA is its reconfigurable function. Hence, the user can update the Qsys design anytime and the IP is reusable. The experimental results from the reconfigurable WSN design for WQM in IoT environment system report offer reliable outcomes which are low power consumption, strong communication ability, and display of real-time measurement accuracy. The designed reconfigurable WQM system reduces the costs and time of detecting water quality of a lake as a part of the environmental survey.

6.2 Further Work

The present study demonstrates the design and implementation of a real-time monitoring system for detecting the data of five water parameters. The present design of the device gives the wireless water parameter measurements on a PC monitor. The WSN network will be developed in the future comprising of more number of nodes to extend the coverage range. Additionally, to improve water level monitoring, the alarm system can be installed. The reconfigurable design of WQM based on WSN system can be additionally upgraded by the amalgamation of additional sensors for a broader measuring area. This study can be exploited in numerous application areas of remote monitoring. For achieving more scalable and robust architecture, the detecting program can be connected to the web server through a passive IP address in one port of the receiving computer. Since both hardware and software systems are reconfigurable, new devices can be added without having to make any further changes to the network.

Bibliography

[1] M. Welsh, D. Malan, Duncan, T. Fulford-Jones and S. Moulton, ‘‘Wireless Sensor Networks for Emergency Medical Care,’’ in *GE Global Research Conference*, Harvard University and Boston University School of Medicine, Boston, MA, Mar. 8, 2004.

[2] R. Verdone, D. Dardari, G. Mazzini and A. Conti, “Wireless Sensor and Actuator Networks,” *Elsevier*, London, UK, 2008.

[3] W. R. Heinzelman et al., ‘‘Adaptive Protocols for Information Dissemination in Wireless Sensor Networks,’’ in *Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking* (MobiCom’99), Seattle, WA, Aug. 1999.

[4] A. Broring et al., “New Generation Sensor Web Enablement,” *Sensors*, 11, 2011, pp. 2652–2699.

[5] I. F. Akyildiz, W. Su, Y.Sankarasubramaniam and Y. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks*, Vol.38, pp. 393–422, 2002.

[6] J. Cecílio and F. Pedro, “Wireless Sensors in Heterogeneous Networked Systems: Configuration and Operation Middleware,” *Springer International Publishing*, 2014.

[7] H. Sharma and S. Sharma, “A Review of Sensor Networks: Technologies and Applications,” in *Engineering and Computational Sciences (RAECS)*, 2014 Recent Advances in, Chandigarh, pp.1–4, 2014.

[8] C. Buratti, A. Conti, D. Dardari and R. Verdone, “An Overview on Wireless Sensor Networks Technology and Evolution”, *Sensors* 2009, Vol. 9, pp.6869-6896, 2009; ISSN 1424-8220, www.mdpi.com/journal/sensors.

[9] S. Li, L. Xu, X. Wang and J. Wang, “Integration of Hybrid Wireless Networks in Cloud Services Oriented Enterprise Information Systems,” *Enterp. Inf. Syst.*, Vol. 6, No. 2, pp. 165–187, 2012.

[10] J. M. Lopez-Higuera, L. R. Cobo, A. Q. Incera and A. Cobo, “Fiber Optic Sensors in Structural Health Monitoring,” in *Journal of Lightwave Technology*, Vol. 29, No. 4, pp. 587–608, Feb.15, 2011.

[11] T. J. Thorp, “A critical analysis of the Saint Anthony Falls (I-35W) Bridge, Minneapolis,” Proceedings *of Bridge Engineering 2 Conference* 2009 Apr. 2009, University of Bath, Bath, UK.

[12] L. Sheppard, “Innovative Infrastructure: Smart Bridges”, Nov.10, 2010. [Online]. Available: http://buildipedia.com/aec-pros/public-infrastructure/innovative-infrastructure-smart-bridges?print=1&tmpl=component. Accessed: 12 Apr. 2017.

[13] C. Jian, Q. Suxiang, H. Hongsheng and Y. Gongbiao, “Wireless Monitoring and Assessment System of Water Quality Based on GPRS,” *2007 8th International Conference on Electronic Measurement and Instruments*, Xi'an, 2007, pp. 2124–2127.

[14] J. A. Lopez, F. Soto, P. Sanchez, A. Iborra, J. Suardiaz and J. A. Vera, “Development of a sensor Node for Precision Hortculture,” *Sensor* 2009, 9, pp. 3240–3255.

[15] N. Kularatna and B. H. Sudantha, “An Environmental Air Pollution Monitoring System Based on the IEEE 1451 Standard for Low Cost Requirements,” in *IEEE Sensors Journal*, Vol. 8, No. 4, pp. 415–422, Apr. 2008.

[16] S. Choi, N. Kim, H. Cha and R. Ha, “Micro Sensor Node for Air Pollutant Monitoring: Hardware and Software Issues,” *Sensors* 2009, Vol. 9, pp. 7970–7987.

[17] J. Hayes, S. Beirne, K. T. Lau and D. Diamond, “Evaluation of a low cost Wireless Chemical Sensor Network for Environmental Monitoring” in *IEEE Sensors* 2008 Conference, pp. 530–533.

[18] M. M. Islam, M. M. Hassan, G. W. Lee and E. N. Huh, “A survey on virtualization of wireless sensor networks,” *Sensors*, Feb. 15, 2012, Vol. 12, No.2, pp. 2175–2207.

[19] T. L. Dinh, W. Hu, P. Sikka, P. Corke, L. Overs and S. Brosnan, “Design and Deployment of a Remote Robust Sensor Network: Experiences from an Outdoor Water Quality Monitoring Network,” in *32nd IEEE Conference on Local Computer Networks (LCN 2007),* Dublin, 2007, pp. 799–806.

[20] D. V. Chapman (Ed.), “Water quality assessments: a guide to the use of biota, sediments, and water in environmental monitoring,” 1996.

[21] R. V. O'Neill, C. T. Hunsaker, K. B. Jones, K. H. Riitters, J. D. Wickham, P. M. Schwartz, I. A. Goodman, B. L. Jackson and W. S. Baillargeon, “Monitoring environmental quality at the landscape scale,” *Bio Science*, Sep. 1, 1997, Vol. 47, No. 8, pp. 513–519.

[22] B. Son, Y. S. Her and J. G. Kim, “A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains,” *International Journal of Computer Science and Network Security (IJCSNS),* Vol. 6, No. 9, September 2006, pp. 124–130.

[23] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach and M. Parlange, “SensorScope: Out-of-the-Box Environmental Monitoring,” in *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, St. Louis, MO, 2008, pp. 332–343.

[24] K. Martinez, R. Ong and J. Hart, “Glacsweb: a sensor network for hostile environments,” in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.*, 2004, pp. 81–87.

[25] M. Conti and S. Giordano, “Multihop Ad Hoc Networking: The Reality,” in *IEEE Communications Magazine*, Vol. 45, No. 4, pp. 88–95, Apr. 2007.

[26] M. E. Carew, V. J. Pettigrove, L. Metzeling and A. A. Hoffmann, “Environmental monitoring using next generation sequencing: rapid identification of macroinvertebrate bioindicator species,” Frontiers *in zoology*, Aug. 7, 2013, Vol.10, No.1, p.45.

[27] C. M. S. Figueiredo, E. F. Nakamura, A. D. Ribas, T. R. B. de Souza and R. S. Barreto, “Assessing the communication performance of wireless sensor networks in rainforests,” *2009 2nd IFIP Wireless Days (WD),* Paris, 2009, pp. 1–6.

[28] H. Liu, Z. Meng and S. Cui, “A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses,” in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, 2007, pp. 2344–2347.

[29] Q. Zhang, X. L. Yang, Y. M. Zhou, L. R. Wang and X. S. Guo, “A wireless solution for greenhouse monitoring and control system based on ZigBee technology,” *Journal of Zhejiang University-Science A.* 1 October 2007, Vol. 8, No. 10, pp.1584–1587.

[30] M. Mancuso and F. Bustaffa, “A wireless sensors network for monitoring environmental variables in a tomato greenhouse,” in *2006 IEEE International Workshop on Factory Communication Systems*, Torino, Italy, 2006, pp. 107–110.

[31] Y. Zhou, X. Yang, X. Guo, M. Zhou and L. Wang, “A Design of Greenhouse Monitoring & Control System Based on ZigBee Wireless Sensor Network,” in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, 2007, pp. 2563–2567.

[32] W. Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi and R. LaHusen, “Air-dropped sensor network for real-time high-fidelity volcano monitoring,” in *Proceedings of the 7th international conference on Mobile systems, applications, and services 2009* Jun 22.pp. 305–318. ACM.

[33] R. Huang, W. Z. Song, M. Xu, N. Peterson, B. Shirazi and R. LaHusen, “Real-World Sensor Network for Long-Term Volcano Monitoring: Design and Findings,” in *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, No. 2, pp. 321–329, February 2012.

[34] M. T. Lazarescu, “Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications,” in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 3, No. 1, pp. 45–54, March 2013.

[35] J. W. Santo Domingo, D. G. Bambic, T. A. Edge and S. Wuertz, “Quo vadis source tracking? Towards a strategic framework for environmental monitoring of fecal pollution,” *Water Research,* Aug 31, 2007, Vol. 41, No. 16, pp. 3539–3552.

[36] I. Amundson and X. D. Koutsoukos, “A survey on localization for mobile wireless sensor networks,” In *Mobile Entity Localization and Tracking in GPS-Less Environments*. Lecture Notes in Computer Science, 2009, Vol. 5801. *Springer,* pp. 235–254.

[37] D. Estrin, A. Sayeed and M. Srivastava, “Wireless sensor networks”, in *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (MobiCom’02)*, 2002. Vol. 255. ACM, Atlanta, GA.

[38] I. F. Akyildiz and E. P. Stuntebeck, “Wireless underground sensor networks: Research challenges,” *Ad Hoc Networks, 2006,* Vol. 4, No. 6, pp. 669–686.

[39] C. C Conrad and K. G. Hilchey, “A review of citizen science and community-based environmental monitoring: issues and opportunities,” *Environmental monitoring and assessment*, May 1, 2011, Vol. 176, No. 1, pp.273–291.

[40] I. N. Athanasiadis and P. A. Mitkas, “An agent-based intelligent environmental monitoring system,” *Management of Environmental Quality, An International Journal*, June 1, 2004, Vol. 15, No. 3, pp.238–249.

[41] S. Bhattacharya, S. Sridevi and R. Pitchiah, “Indoor air quality monitoring using wireless sensor network,” in *2012 Sixth International Conference on Sensing Technology (ICST),* Kolkata, 2012, pp. 422–427.

[42] D. Estrin, L. Girod, G. Pottie and M. Srivastava, “Instrumenting the world with wireless sensor networks,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221),* Salt Lake City, UT, 2001, pp. 2033–2036, Vol.4.

[43] S. G. S. Yadav, A. Chitra and C. L. Deepika, “Reviewing the process of data fusion in wireless sensor network: A brief survey,” International *Journal of Wireless and Mobile Computing, 2015,* Vol. 8, No. 2, pp.130–140.

[44] W. Y. Chung and J. H. Yoo, “Remote water quality monitoring in wide area,” *Sensors and Actuators B: Chemical,* 2015, pp. 51–57.

[45] L.Selavo et al., “Luster: wireless sensor network for environmental research,” in *Proceedings of the 5th international conference on Embedded networked sensor systems*, 2007, pp. 103–116.

[46] B. O'Flynn *et al*., “SmartCoast: A Wireless Sensor Network for Water Quality Monitoring,” in *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, Dublin, 2007, pp. 815–816.

[47] S. Zhang and L. Zhang, “Water pollution monitoring system based on Zigbee wireless sensor network,” in *2011 International Conference on Electronics, Communications and Control (ICECC)*, Zhejiang, 2011, pp. 1775–1779.

[48] B. P. L. Lo, S. Thiemjarus, R. King and G-Z. Yang, “Body sensor network – a wireless sensor platform for pervasive healthcare monitoring,” *Archit Des*, 2005, Vol. 13, No. 2–3, pp. 77–80.

[49] T. Gao, D. Greenspan, M. Welsh, R. R. Juang and A. Alm, “Vital Signs Monitoring and Patient Tracking Over a Wireless Network,” *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, Shanghai, 2005, pp. 102–105.

[50] P. Kumar et al., “E-SAP: Efficient-Strong Authentication Protocol for Healthcare Applications Using Wireless Medical Sensor Networks,” *Sensors 2012*, Vol. 12, pp. 1625 – 1647.

[51] J. Burrell, T. Brooke and R. Beckwith, “Vineyard computing: sensor networks in agricultural production,” *IEEE Pervasive Computing,* Vol. 3, No.1, pp. 38–45, Jan-Mar 2004.

[52] K. Mayer, K. Taylor and K. Ellis, “Cattle health monitoring using wireless sensor networks,” in Second *IASTED International Conference on Communication and Computer Networks*, Cambridge, Massachusetts, USA, Nov. 2004.

[53] W. Zhang, G. Kantor and S. Singh, “Integrated wireless sensor/actuator networks in an agricultural application,” in *Second ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, p. 317, Baltimore, Maryland, USA, Nov. 2004.

[54] D. D. Chaudhary, S. P. Nayse, L. M. Waghmare, “Application of Wireless Sensor Networks for Greenhouse Parameter Control in Precision Agriculture,” *International Journal of Wireless & Mobile Networks (IJWMN)*, Vol. 3, No. 1, February 2011.

[55] A. Sivasankari, and S. Gandhimathi, , “Wireless Sensor based Crop Monitoring System for Agriculture using Wi-Fi Network,” *Dissertation, International Journal of Computer Science and Information Technology Research*, ISSN 2348-1196 (print) ,ISSN 2348-120X (online) Vol. 2, Issue 3, pp. 293–303, July–Sept. 2014, Available: www.researchpublish.com.

[56] G. S Sankar and S. Angadi, “Wireless Sensor Network for Border Monitoring,” The *International Journal Of Engineering And Science (IJES)*, Vol. 2, No.4, pp.65–68, 2013.

[57] A. Saipulla, B. Liu and J. Wang, “Barrier coverage with airdropped wireless sensors,” *MILCOM 2008 - 2008 IEEE Military Communications Conference*, San Diego, CA, 2008, pp. 1–7.

[58] A. Saipulla, C. Westphal, B. Liu and J. Wang, “Barrier Coverage of Line-Based Deployed Wireless Sensor Networks,” *IEEE INFOCOM 2009*, Rio de Janeiro, 2009, pp. 127–135.

[59] K. F. Ssu, W. T. Wang, F. K. Wu, T. T. Wu, “K-Barrier Coverage With a Directional Sensing Model,” *International Journal on Smart Sensing and Intelligent Systems*, March 2009, Vol. 2, No. 1.

[60] M. Chitnis, Y. Liang, J. Y. Zheng, P. Pagano and G. Lipari, “Wireless Line Sensor Network for Distributed Visual Surveillance,” in *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, 2009, Tenerife, Canary Islands, Spain.

[61] G. Yang and D. Qiao, “Barrier Information Coverage with Wireless Sensors,” *IEEE INFOCOM 2009*, Rio de Janeiro, 2009, pp. 918–926.

[62] H. Grindvoll, O. Vermesan, T. Crosbie, R. Bahr, N. Dawood and G. M. Revel, “ A Wireless Sensor Network for Intelligent Building Energy Management based on Multi Communication Standards–A Case Study,” Journal of Information Technology in Construction - ISSN 1874-4753, ITcon, 2012, Vol. 17, Amor, pp. 43–62.

[63] M. Luis, C. Matos, S. Tomic and P. Graca, “Technological Innovation for the Internet of Things,” in 4*th IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2013*, Costa de Caparica, Portugal, April 15-17, 2013, Proceedings, Vol. 394 of IFIP Advances in Information and Communication Technology, Springer, 2013, p.4.

[64] L. Atzori, A. Iera and G. Morabito, “The Internet of Things: A Survey,” *Computer Networks*, Oct. 2010, Vol. 54, No. 15, pp. 2787–805.

[65] P. Sethi, and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” *Journal of Electrical and Computer Engineering*, vol. 2017 (2017), Article ID 9324035, 25 pages, https://doi.org/10.1155/2017/9324035

[66] S. Li, D. X. Li and S. Zhao, “The internet of things: a survey”, *Information Systems Frontiers* 17, 2015, No. 2, pp. 243-259.

[67] M. Roberto, M. Biru and D. Rotondi, “Towards a definition of the Internet of Things (IoT),” IEEE *Internet Initiative* 1, 2015.

[68] Q. Chi, H. Yan, C. Zhang, Z. Pang and L. D. Xu, “A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment,” in *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1417-1425, May 2014.

[69] J. A. Stankovic, “Research Directions for the Internet of Things,” in *IEEE Internet of Things Journal*, Vol. 1, No. 1, pp. 3–9, Feb. 2014.

[70] C. Zhu, V. C. M. Leung, L. Shu and E. C. H. Ngai, “Green Internet of Things for Smart World,” in *IEEE Access*, Vol. 3, No., pp. 2151–2162, 2015.

[71] A. Luigi, A. Iera and G. Morabito, “The internet of things: A survey,” *Computer networks,* Vol. 54, No. 15, 2010, pp. 2787–2805.

[72] M. Zorzi, A. Gluhak, S. Lange and A. Bassi, “From today’s INTRAnet of things to a future INTERnet of things: a wireless- and mobility-related view,” in *IEEE Wireless Communications,* Vol. 17, No. 6, pp. 44–51, December 2010.

[73] S. Chen, H. Xu, D. Liu, B. Hu and H. Wang, “A Vision of IoT: Applications, Challenges, and Opportunities with China Perspective,” in *IEEE Internet of Things Journal*, Vol. 1, No. 4, pp. 349–359, August 2014.

[74] Y. Chen and D. Han “Water quality monitoring in smart city: A pilot project,” Elsevier, Automation in Construction, Vol. 89, pp. 307–316, 2018.

[75] B. Eleonora, “The Internet of Things vision: Key features, applications and open issues”, *Computer Communications,* December 1, 2014, Vol. 54, No.1, pp. 1–31.

[76] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, “Internet of Things for Smart Cities,” in *IEEE Internet of Things Journal,* Vol. 1, No. 1, pp. 22-32, Feb. 2014.

[77] H. Cai, L. D. Xu, B. Xu, C. Xie, S. Qin and L. Jiang, “IoT-Based Configurable Information Service Platform for Product Lifecycle Management,” in *IEEE Transactions on Industrial Informatics,* Vol. 10, No. 2, pp. 1558–1567, May 2014.

[78] D. Miorandi, S. Sicari, F. De Pellegrini and I. Chlamtac, “Internet of things: vision, applications and research challenges,” *Ad hoc Networks,* Sept. 30, 2012, Vol. 10, No. 7, pp. 1497–1516.

[79] M. T. Lazarescu, “Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications,” in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems,* Vol. 3, No. 1, pp. 45–54, March 2013.

[80] F. Salvadori *et al*., “Monitoring in Industrial Systems Using Wireless Sensor Network with Dynamic Power Management,” in *IEEE Transactions on Instrumentation and Measurement,* Vol. 58, No. 9, pp. 3104–3111, Sept. 2009.

[81] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprise,” *Business Horizons*, August 31, 2015, Vol. 58, No. 4, pp. 431–440.

[82] L. Mainetti, L. Patrono and A. Vilei, “Evolution of wireless sensor networks towards the Internet of Things: A survey,” *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*, Split, 2011, pp. 1–6.

[83] P. Sarwesh, N. S. Shet and K. Chandrasekaran, “Energy-Efficient Network Architecture for IoT Applications,” in *Beyond the Internet of Things*, Springer International Publishing, 2017, pp. 119–144.

[84] A. Apsel, X. Wang and R. Dokania, “Design of ultra-low power impulse radios,” *Analog Circuits and Signal Processing, Springer Science & Business Media*, 29 October 2013, Vol.124, p.1.

[85] K. Sohraby, D. Minoli and T. Znati, “Wireless sensor networks: technology, protocols, and applications,” John Wiley & Sons, April 6, 2007. pp.77,78, 110,111, 173.

[86] S. Li, L. D. Xu and X. Wang, “Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things,” in *IEEE Transactions on Industrial Informatics*, Vol. 9, No. 4, pp. 2177–2186, November 2013.

[87] Y. Chen and V. Dinavahi, “Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems,” in *IET Generation, Transmission & Distribution*, Vol. 7, No. 5, pp. 451–463, May 2013.

[88] A. Myaing and V. Dinavahi, “FPGA-Based Real-Time Emulation of Power Electronic Systems with Detailed Representation of Device Characteristics,” in *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 1, pp. 358–368, January 2011.

[89] R. Dafali, J. P. Diguet and J. C. Creput, “Self-Adaptive Network-on-Chip Interface,” in *IEEE Embedded Systems Letters*, Vol. 5, No. 4, pp. 73–76, Dec. 2013.

[90] Z. Pang, K. Yu, J. Åkerberg and M. Gidlund, “An RTOS-based architecture for industrial wireless sensor network stacks with multi-processor support,” in *2013 IEEE International Conference on Industrial Technology (ICIT)*, Cape Town, 2013, pp. 1216–1221.

[91] W. Viriyasitavat, L. D. Xu and W. Viriyasitavat, “A New Approach for Compliance Checking in Service Workflows,” in *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1452–1460, May 2014.

[92] Z. Hanzalek and P. Jurcik, “Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks with Time-Bounded Data Flows: Application to IEEE 802.15.4/ZigBee,” in *IEEE Transactions on Industrial Informatics*, Vol. 6, No. 3, pp. 438–450, Aug. 2010.

[93] K. C. Lee, M. H. Kim, S. Lee and H. H. Lee, “IEEE-1451-based Smart Module for In-vehicle Networking Systems of Intelligent Vehicles,” in *IEEE Transactions on Industrial Electronics*, Dec. 2004, Vol. 51, No. 6, pp. 1150–1158.

[94] V. Mattoli, A. Mondini, B. Mazzolai, G. Ferri and P. Dario, “A Universal Intelligent System-on-chip based Sensor Interface,” *Sensors*, 2010, Vol. 10, No. 8, pp. 7716–7747.

[95] N. Fourty, A. ven de Bossche and T. Val, “An advanced study of energy consumption in an IEEE 802.15.4 based network: everything but the truth on 802.15.4 node lifetime,” *Comp. Commun.*, 2012, Vol. 35, No, 14, pp. 1759–1767.

[96] L. Li, H. Xiaoguang, C. Ke and H. Ketai, “The applications of WiFi-based Wireless Sensor Network in Internet of Things and Smart Grid,” in *2011 6th IEEE Conference on Industrial Electronics and Applications*, Beijing, 2011, pp. 789–793.

[97] Z. Bi, L. D. Xu and C. Wang, “Internet of Things for Enterprise Systems of Modern Manufacturing,” in *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1537-1546, May 2014.

[98] S. Fang *et al*., “An Integrated Approach to Snowmelt Flood Forecasting in Water Resource Management,” in *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 1, pp. 548-558, Feb. 2014.

[99] K. Compton and H. Scott, “Reconfigurable Computing: A Survey of Systems and Software,” ACM Computing Surveys, Vol. 34, No. 2, June 2002, pp. 171–210.

[100] A. O. El-Rayis, T. Arslan and K. Benkrid, “Review of Reconfigurable Architectures for the Next Generation of Mobile Device Telecommunications Systems, Consumer Electronics Times (CET), October 2014, Vol. 3, Issue. 4, pp. 256–277.

[101] R. D.Wittig and P. Chow, “One Chip: an FPGA processor with reconfigurable logic,” Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, pp. 126–135, Apr. 1996.

[102] R. N Bhargava, V. Rajaram, K. Olson and L. Tiede, “Ecology and Environment,” The Energy and Resources Institute (TERI), 2016, p.76.

[103] T. G Sanders, “Design of Networks for Monitoring Water Quality,” *Water Resources Publication*, 1983, Littleton, CO.

[104] R. O. Strobl and P. D. Robillard, “Network design for water quality monitoring of surface freshwaters: A review”, *Journal of Environmental Management, 2008,* Vol. 87, No. 4, pp. 639–648.

[105] J. Bhardwaj, K. K. Gupta and R. Gupta, “A review of emerging trends on water quality measurement sensors,” *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, Mumbai, 2015, pp. 1–6.

[106] United Nations. 2005. United Nations Environment Programme: Operational Guide for Data Submission. [Online].
Accessed: Nov.12, 2016. Available: http://www.unep.org/gemswater/Portals/24154/pdfs/op_guide_for_data_2005.pdf.

[107] M. Zennaro et al., “On the Design of a Water Quality Wireless Sensor Network (WQWSN): An Application to Water Quality Monitoring in Malawi,” in *2009 International Conference on Parallel Processing Workshops*, Vienna, 2009, pp. 330–336.

[108] Z. Wang, Q. Wang and X. Hao, “The Design of the Remote Water Quality Monitoring System Based on WSN,” in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, Beijing, 2009, pp. 1–4.

[109] S. c. Yang and Y. Pan, “The Application of the Wireless Sensor Network (WSN) in the Monitoring of Fushun Reach River in China,” in *2010 Second International Conference on Computer and Network Technology*, Bangkok, 2010, pp. 331–333.

[110] A. Alkandari, M. Alnasheet, Y. Alabduljader and S. M. Moein, “Water monitoring system using Wireless Sensor Network (WSN): Case study of Kuwait beaches,” in *2012 Second International Conference on Digital Information Processing and Communications (ICDIPC)*, Klaipeda City, 2012, pp. 10–15.

[111] J. Wang, X. Ren, Y. Shen and S. Y. Liu, “A Remote Wireless Sensor Networks for Water Quality Monitoring,” in *2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering*, Macao, 2010, pp. 7–12.

[112] S. Silva, N. N. Hoang, V. Tiporlini and K. Alameh, “Web based water quality monitoring with sensor network: Employing ZigBee and WiMax technologies,” in *8th International Conference on High-capacity Optical Networks and Emerging Technologies*, Riyadh, 2011, pp. 138–142.

[113] F. Adamo, F. Attivissimo, C. G. C. Carducci and A. M. L. Lanzolla, “A Smart Sensor Network for Sea Water Quality Monitoring,” in *IEEE Sensors Journal*, Vol. 15, No. 5, pp. 2514–2522, May 2015.

[114] J. Bartram and R. Ballance, “Water Quality Monitoring: A Practical Guide to the Design and Implementation of Freshwater Quality Studies and Monitoring Programmes,” *1996, E*&FN Spon, London, UK.

[115] R. J. Wagner, Jr. R. W. Boulger, C. J. Oblinger and B. A. Smith, “Guidelines and Standard Procedures for Continuous Water-Quality Monitors: Station Operation, Record Computation, and Data Reporting,” *Technical Report 2006*, U.S. Department of the Interior, Washington, DC.

[116] WHO (Ed.). 2011. *Guidelines for Drinking-Water Quality* (4th ed.). WHO Press, Geneva, Switzerland.

[117] A. S. Rao, S. Marshall, J. Gubbi, M. Palaniswami, R. Sinnott and V. Pettigrovet, “Design of Low-cost Autonomous Water Quality Monitoring System,” in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, 2013, pp. 14–19.

[118] J. Bhatt, J. Patoliya, “IoT based Water Quality Monitoring System,” *Proceedings of 49th IRF International Conference*, 21 February 2016, Pune, India, ISBN: 978-93-85973-46-8.

[119] N. A. Cleote, R. Malekian and L. Nair, "Design of Smart Sensors for Real-time Water Quality Monitoring", Vol. 13, No. 9, September 2014 IEEE, pp. 1–16.

[120] S. Khaire, "Water Quality Data Transfer and Monitoring System in IoT Environment," International Journal of Innovations & Advancement in Computer Science (IJIACS), ISSN 2347 – 8616, Vol. 6, Issue 11, November 2017.

[121] C. E. H. Curiel, V. H. B. Baltazar and J. H. P. Ram´ırez, "Wireless sensor networks for water quality monitoring: Prototype design," *International Journal of Environmental, Chemical, Ecological, Geological, and Geophysical Engineering,* 2016, Vol. 10, No. 2, pp. 158–163.

[122] A. Faustine, A. N. Mvuma, H. J. Mongi1, M. C. Gabriel, A. J. Tengel and S. B. Kucel, "Wireless Sensor Networks for Water Quality Monitoring and Control within Lake Victoria Basin: Prototype Development," *Wireless Sensor Network*, Vol. 6, pp. 281-290. 2014. DOI: http://dx.doi.org/10.4236/wsn.2014.612027

[123] W. Lehr and L. W. McKnight, "Wireless internet access: 3G vs. WiFi?" Telecommunications Policy, Jul 31, 2003, Vol. 27, No.5, pp. 351–70.

[124] C. H. Vishnuvardhan, P. N. Reddy and V. Ramesh, "Interference Avoiding for Mobile Ad Hoc Networks," International Journal of Innovative Research and Development, May 31, 2013, Vol. 2, No.5. ISSN 2278–0211.

[125] O, A., Chukwudi, "Deploying Voice Over Wireless Land Area Networks for Enterprises," (Doctoral dissertation, Blekinge Institute of Technology).

[126] Z. Qin, J. Zhang, L. Shi, L. Wang, L. Shu and Y. Guo, "Agile Networking in Smart Grids," IEIE Transactions on Smart Processing & Computing, Aug. 2012, Vol. 1, No. 1, pp. 34–49.

[127] [Online] http://www.hotenda.com/media/articles/Comparing-Lowpower-Wireless-Technologies.html, Accessed: 12 November, 2016.

[128] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche and C. Loge, "The smart home concept: our immediate future," In *E-Learning in Industrial Electronics, 2006 1ST IEEE International Conference*. 18 December, 2006. pp. 23–28. IEEE.

[129] Q. H Mahmoud, "Wireless application programming with J2ME and Bluetooth," Sun micro system resource for developer, February 2003.

[130] J. Haartsen, M. Naghshineh, J. Inouye, O. J. Joeressen and W. Allen, "Bluetooth: Vision, goals, and architecture," ACM SIGMOBILE Mobile Computing and Communications Review, October 1998, Vol.1, No. 2, Issue. 4 pp. 38-45.

[131] J. Rodriguez, (Ed.), **"**Fundamentals of 5G Mobile Networks," *John Wiley & Sons*, 2015, p.138

[132] Y. Wang, W-Z. Song, and X-Y. Li, "Scatternet formation and self-routing in Bluetooth networks," *Wireless LANs and Bluetooth*, 2005, Vol.4, p. 241.

[133] A. G. Ambal, T. Harikrishnan, H. Jacob, H. Babu and T. George, "Bluetooth Aided Safety Band for Women," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering an ISO 3297: 2007 Certified Organization Vol. 5, Special Issue 3, March 2016,* National Conference on Recent Advances in Electrical & Electronics Engineering (NCREEE'16).

[134] H. Sreenivas and H. Ali, "An evolutionary Bluetooth scatternet formation protocol," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the*, 2004, pp. 8.

[135] G. Tan, A. Miu, J. Guttag and H. Balakrishnan, "Forming scatternets from bluetooth personal area networks," [Online]. Massachusetts Institute of Techonology, Accessed: 12 Apr.2016. Available: http://lcs. mit. edu/, Tech. Rep. MIT-LCS-TR-826. October 2001.

[136] K. Preetha, "A novel solution to the short range bluetooth communication", *CoRR, 2011* (2011).
Available: http://dblp.uni-trier.de/db/journals/corr/corr1111.html#abs-1111-2097

[137] R. Nilsson and B. Saltzstein, "Bluetooth low energy technology makes new medical applications possible," Medical Electronic Design, 2011 Nov, Vol. 15, No.2, Issue.5.

[138] L'Ruiz-Garcia, L. Lunadei, P. Barreiro and I. Robla, "A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends," *sensors*, 2009 Jun 16, Vol. 9, No. 6, pp. 4728–4750.

[139] G. Kumar, S. Bhola and M. Batra, "Comparative Study of 3G/4G Network Service," Compusoft, 2014, Oct 1, Vol. 3, No. 10, p.1211.

[140] A. I. Gardezi, "Security in wireless cellular networks", Washington University in St. Louis, St. Louis. 2006.
https://www.cse.wustl.edu/~jain//cse574-06/ftp/cellular_security/index.html

[141] [Online] http://techchannel.radioshack.com/difference-between-3g-4g-2437.html. Accessed:14 Nov. 2016.

[142] A. Kulkarni and A. Patange, "A review on ZigBee, GSM and WSN based home security by using embedded controlled sensor network," *International Journal of Embedded systems and Applications (IJESA)* Vol.6, No.3/4, Dec. 2016.

[143] M. J. Mohammad and Mumtazul-Imam, "Traffic management in GSM networks," in *2008 Second International Conference on Electrical Engineering*, Lahore, 2008, pp. 1-6.

[144] A. K Singh, "Science & Technology For UPSC," *McGraw-Hill Education Pvt. Ltd.*, 2007, p.288

[145] G. Balaji, E. Kanniga and G. Brindha, "GSM based analysis of network using mobile communications," *Middle-East Journal of Scientific Research*, 2014, Vol. 20, No. 3, pp. 369-76.

[146] B. Ghribi and L. Logrippo, “Understanding GPRS: the GSM packet radio service”, *Computer Networks,* Nov. 2000, Vol. 30, No. 34, (5), pp. 763–79.

[147] P. Samundiswary and R. Surender, “Performance comparison of GTS mechanism enabled IEEE 802.15.4 based Wireless Sensor Networks using LAR and DYMO protocol,” in *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE),* Hosur, 2014, pp. 254–258.

[148] S. M., Sargun, and S.B Rana, “Wireless Personal Area Networks architecture and protocols for multimedia applications,” *Topology,* Vol. 4, No. 1, 2015.

[149] B. Mihajlov and M. Bogdanoski, “Overview and Analysis of the Performances of ZigBeebased Wireless Sensor Networks,” *International Journal of Computer Applications,* Vol. 29, No. 12, pp. 28–35, September 2011.

[150] S. H. Yang, “Wireless Sensor Network: Principal, Design and Applications,” *Signals and Communication Technology, Springer Science & Business Media*, 2013, pp. 11–34.

[151] P. Shiny, I. T. Banu, B. Thenmozhi, U.T. Sasikala and M. R. Prabhu, “Intelligent Traffic Control System for Congestion Control Using Image Processing, Ambulance Clearance, and Stolen Vehicle Detection”, *International Journal of Advanced Research Trends in Engineering and Technology (IJARTET),* Vol. 3, Special Issue 19, Apr. 2016.

[152] P. Kaushik, N. K. R. Patel and J. Singhai, “Energy Efficient Clear Channel Assessment for LR-WPAN”, *IJCSI International Journal of Computer Science Issues,* Vol. 8, Issue 3, No. 2, May 2011.

[153] A. Khayyat and A. Safwat, “The Synchronized Peer-to-Peer Framework and Distributed Contention-Free Medium Access for Multihop Wireless Sensor Networks”, *Journal of Sensors,* Vol. 2008 (2008), Article ID 728415, 28 pages

[154] K. Karuppasamy et al., “Water Quality Monitoring and Control using Wireless Sensor Networks,” International Research Journal of Engineering and Technology (IRJET), Vol. 03, Issue 03, March 2016. Available at www.irjet.net.

[155] P. K. Somasundaram and D. J. Edison, “Monitoring Water Quality using RF Module,” International Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol. 2, Issue 7, July 2013. Available at: www.ijaiem.org.

[156] B. Menaka Devi *et al.*, “Real Time System for Determination of Drinking Water Quality,” International Journal of Computer Science and Mobile Computing, Vol.3 Issue.9, September- 2014, pg. 732-740.

[157] M. T. Hasan and S. Khan, “GSM Based Automatic Water Quality Control Analysis,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), Vol. 5, Issue 6, June 2016.

[158] P. K. Somasundaram and D. J. Ediosn, “Monitoring Water Quality using RF Module,” *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, Vol. 2, Issue 7, Jul. 2013, pp:220–224.

[159] V. Karthikeyan, S. Geethanjali, M. Mekala and T. Deepika, “Water Eminence Scrutinizing Scheme Based on Zigbee and Wireless Antenna Expertise- A Study,” *International Journal of Scientific Research in Computer Science Applications and Management Studies*, Vol.3, Issue. 1, Jan. 2014.

[160] N. S Pingle and S. B. Jadhav, “Automatic Measurement and Reporting System of Water Quality Based on GSM,” *International Journal of Electrical and Electronics Research*, ISSN 2348-6988 [online] Vol. 4, Issue 2, pp. 20-27, Apr.– Jun. 2016, Available: www.researchpublish.com

[161] K. Rajasekar, “Measurement and Analysis of Water Quality using GSM,” International Journal of Advanced Technology in Engineering and Science, Vol.03, Issue 03, March 2015. Available at: www.ijates.com.

[162] N. N. Kumar, K. Vinayswethi and V. N. Kumar, "Design and Implementation of an Efficient Water Quality and Quantity Monitoring System," *International Journal of Software and Hardware Research in Engineering*, Vol. 1, Issue. 4, December 2013.

[163] L. Jinfeng and C. Shun, "A ZigBee-based Aquiculture Water Quality Monitoring System," International *Journal of u- and e- Service, Science and Technology,* Vol.8, No. 10, pp.367–376, 2015.

[164] N. T. K. Duy et al., "Automated monitoring and control system for shrimp farms based on embedded system and wireless sensor network," In Electrical, Computer and Communication Technologies (ICECCT), 2015 IEEE International Conference, March 2015, pp. 1–5.

[165] Kudva et al., "Towards a Real-Time Campus-Scale Water Balance Monitoring System," 2015 28th International Conference on VLSI Design and 2015 14th International Conference on Embedded Systems, 2015.

[166] H. L. Chen and X.Q. Liu, "A Novel Kind of Smart Monitoring System of Aquaculture," 2nd International Conference on Electrical, Computer Engineering and Electronics (ICECEE 2015).

[167] H. Xiao-peng and H. Xiao-liang, "Hardware Design of River Flow Velocity Monitoring System," *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, Shangshai, 2011, pp. 304–307.

[168] A. Purohit and U. Gokhale, "Real Time Water Quality Measurement System based on GSM," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* Vol. 9, No. 3, pp. 63–67, May – June. 2014.

[169] N. N. Beri, "Wireless Sensor Network Based System Design for Chemical Parameter Monitoring in Water," *International Journal of Electronics, Communication & Soft Computing Science and Engineering*, Vol. 3, No. 6.

[170] U. Akila, R. Elackiaselvi, R. Maheshwari, K. Shunmugavalli, T. Prathiba, “Industrial Sewage Water Quality Monitoring System”, *International Journal of Engineering Research and General Science*, Vol. 3, No. 2, pp. 1285–1292, March–April 2015.

[171] R. K. Kumar et al., “Solar based Advanced Water Quality Monitoring System using Wireless Sensor Network,” International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 3, March 2014.

[172] M. N. Barabde and S. R. Danve, “Continuous Water Quality Monitoring System for Water Resources at Remote Places”, *International Journal of Engineering Research and General Science*, Vol. 3, No. 3, part-2, May–June 2015.

[173] A. C. Khetre, “Automatic Monitoring & Reporting of Water Quality by using WSN Technology and Different Routing Methods,” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET*), Vol. 2, No. 12, December 2013.

[174] A. A. Kaikade and A. P. Khandait, “Water Parameters Monitoring in Real Time Using Sensor Network and ZigBee,” *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol. 3, Issue. 5, May 2015.

[175] T. R. Patil and R. M. Khaire, “Wireless Sensor Network for Real Time Monitoring and Detection of Water Contamination,” *International Journal of scientific research and management (IJSRM)*, vol. 3, Issue. 6, pp. 3031–3303, 2015.

[176] A. A. Joshi., “Water Quality Monitoring System Using ZigBee and Solar Power Supply,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), Vol. 4, Issue 10, October 2015.

[177] [Online] Grabham, D., "ARM Vs Intel: The Next Processor War Begins", Jan. 2011. Available: http://www.techradar.com/news/phone-and-communications/mobile-phones/arm-vs-intel-the-next-processor-war-begins-920875, Accessed: 29 December 2015.

[178] D. S. Simbeye and S. F. Yang, "Water Quality Monitoring and Control for Aquaculture Based on Wireless Sensor Networks," *Journal of Networks*, Vol. 9, No.4, pp: 840-849, April 2014.

[179] V. Ramya, V. Balaji and T. Akilan, "A Web based Observance and Alerting for Underground and Fish Pond Water Quality," *International Journal of Computer Applications* (0975 – 8887), Vol. 110, No. 5, January 2015.

[180] S. Sethi, and S. Jadhav, "Automation for Checking Water Quality Using Zigbee and GSM," *International Journal of Advanced Computing And Electronics Technology* (*IJACET*), Vol. 3, No. 1, 2016.

[181] D. A. Anandrao and N. R. Kolhare, " Digital Liquid Level Meter using Atmega328 Microcontroller," International Journal of Engineering Sciences & Research Technology (IJESRT), Vol. 6, Issue. 1, January 2017.

[182] V. V. Daigavane and M. A. Gaikwad, "Water Quality Monitoring System based on IoT," Advances in Wireless and Mobile Communications, Vol. 10, No. 5, 2017, pp. 1107-1116. Available at: http://www.ripublication.com.

[183] R. Kamble et al., "Automatic Water Quality Monitoring System using Arduino," International Journal of Recent Innovation in Engineering and Research (IJRIER), Vol. 02 Issue. 02, February 2017, pp. 87–90.

[184] M. Al-Mamun, N. Ahmed, N. U Ahamed, S. A. Rahman, B. Ahmad and K. Sundaraj, "Use of wireless sensor and microcontroller to develop water-level monitoring system," *Indian Journal of Science and Technology*, 2014 Sep 15, Vol. 7, No. 9. pp. 1325–1330.

[185] A. D. L. Piedra, A. Braeken and A. Touhafi, “Sensor Systems based on FPGAs and their Applications,” *A survey, Sensors*, 2012, Vol. 12, No. 9, pp. 12235–12264.

[186] G. J. Garcia, C. A. Jara, J. Pomares, A. Alabdo, L. M Poggi and F. Torres, “A Survey on FPGA based Sensor Systems: Towards Intelligent and Reconfigurable Low Power Sensors for Computer Vision,” *Control and Signal Processing, Sensors*, Vol. 14, No. 4, pp. 6247–6278, 2014.

[187] S. Anumalla, B. Ramamurthy, D. C. Gosselin and M. Burbach, “Ground water monitoring using smart sensors,” in *2005 IEEE International Conference on Electro Information Technology*, Lincoln, NE, 2005, pp. 6.

[188] U. R. Suryawanshi and M.S.Chavan, “Reconfigurable Low Power Smart Water Quality Monitoring Using FPGA,” International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), Vol. 7, Issue 6, June 2018. Available at: www.ijirset.com.

[189] V. Rosello, J. Portilla and T. Riesgo, “Ultra low power FPGA-based architecture for Wake-up Radio in Wireless Sensor Networks,” in *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, Melbourne, VIC, 2011, pp. 3826–3831.

[190] F. Karray, M. Wassim, M. Abid, D. Houssaini, A. M. Obeid, S. M. Qasim, M. S. Saleh, “Architecture of Wireless Sensor Nodes for Water Monitoring Applications: From Microcontroller-based System to SoC Solutions,” in *20th IMEKO TC4 International Symposium*, 15–17 Sept. 2014.

[191] A. Ahmad and S. Pawar, “Hydro-Monitor and Quality Analysis Using FPGA: An Overview”, *International Journal for Modern Trends in Science and Technology*, Vol. 03, Issue 06, June2017, pp. 110–115.

[192] M. D. R. Perera, R. G. N. Meegama and M. K. Jayananda, “FPGA Based Single Chip Solution with 1-Wire Protocol for the Design of Smart Sensor Nodes,” *Hindawi Publishing Corporation Journal of Sensors*, Vol. 2014, Article ID 125874, 11 pages, http://dx.doi.org/10.1155/2014/125874.

[193] J. Dai, L. H. Fu and L.F. Wang, “The Design of Environment Online Data Acquisition System of Hongze Lake Based on GPRS,” *in 4th International Conference on Computer, Mechatronics, Control and Electronic Engineering (ICCMCEE 2015).*

[194] V. Beena and Khaja, “Water Quality Measurement and Control from Remote Station for Pisiculture Using NI myRIO,” *International Journal of Innovative Research in Electronics and Communications (IJIREC)*, Vol.2, Issue. 4, June 2015, pp. 16–21. *ISSN 2349-4042 (Print) & ISSN 2349-4050 (Online).*

[195] G. Tiwari, “Hardware/Software Based a Smart Sensor Interface Device for Water Quality Monitoring in IoT Environment”, *International Journal of Technology and Science*, Vol. 3, Issue. 1, pp. 5–9, 2014.

[196] N. Vijayakumar and R. Ramya, “The real-time monitoring of water quality in IoT environment,” in *2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]*, Nagercoil, 2015, pp. 1–4.

[197] J. R. Frigo, E. Y. Raby, S. M. Brennan, C. Wolinski, C. Wagner, F. Charot, E. Rosten and V. K. Kulathumani, “Energy efficient sensor node implementations,” in *18th annual ACM/SIGDA international symposium on Field Programmable Gate Arrays, FPGA '10*, (New York, NY, USA), pp. 37–40, ACM, 2010.

[198] R. Lysecky and F. Vahid, “A study of the speedups and competitiveness of FPGA soft processor cores using dynamic hardware/software partitioning,” in *Proceedings of Design, Automation and Test in Europe, 2005,* Vol. 1, pp. 18–23.

[199] H. Hinkelmann, A. Reinhardt, S. Varyani and M. Glesner, “A Reconfigurable Prototyping Platform for Smart Sensor Networks,” in *2008 4th Southern Conference on Programmable Logic*, San Carlos de Bariloche, 2008, pp. 125–130.

[200] H. Hinkelmann, P. Zipf and M. Glesner, “A Domain-Specific Dynamically Reconfigurable Hardware Platform for Wireless Sensor Networks,” in *2007 International Conference on Field-Programmable Technology*, Kitakyushu, 2007, pp. 313–316.

[201] IEEE Standard for a Smart Transducer Interface for Sensors and Actuators Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. IEEE Std 1451.5-2007, IEEE: New York, NY, USA, pp. C1–C236, 2007.

[202] [Online] Available: http://www.fpgadeveloper.com/2011/07/list-and-comparison-of-fpga-companies.html. Accessed: 15 December 2015.

[203] [Online] Available: https://www.xilinx.com/products/silicon-devices/fpga.html. Accessed: 13 December 2015.

[204] Spartan-6 Family Overview data sheet. pdf. [Online] Available: www.xilinx.com. Accessed: 13 December 2015.

[205] MicroBlaze Processor Reference Guide pdf. [Online] Available: www.xilinx.com. Accessed: 13 December 2015.

[206] [Online] Available: https://www.altera.com/products/fpga/overview.html. Accessed: 14 December 2015.

[207] Altera Quartus II Handbook. [Online] Available at http://www.altera.com. Accessed: 28 December 2015.

[208] Intel FPGA Design Flow for Xilinx Users [Online] Available: https://www.intel.com. Accessed on 11 April 2018.

[209] Introduction to the Altera SOPC Builder Using VHDL Designs pdf. [Online] Available: http://www.altera.com/. Accessed: 28 December 2015.

[210] Altera Nios II Processor Reference Handbook pdf. [Online] Available: http://www.altera.com/. Accessed: 28 December 2015.

[211] [Online] Available: http://fpgacenter.com/fpga/fpga_or_cpu.php. Accessed: 4 June 2016.

[212] DE1-SoC User Manual pdf. [Online] Available at www.terasic.com. Accessed: 22 December 2015.

[213] Altera, Developing Nios II Software, Embedded Design Handbook pdf. [Online] Available: www.altera.com. Accessed: 26 December 2015.

[214] S. Bhoyar and D. V. Padole, "NIOS II Based Embedded Web Server Development for Networking Applications," in *IJCAT International Journal of Computing and Technology*, Vol.1, Issue. 3, April 2014.

[215] B. Siciliano, O. Khatib (Eds), **"**Springer Handbook of Robotics", *Springer Science & Business Media*, 2008, p.491.

[216] R. A Serway and J. W. Jewett, "Physics for Scientists and Engineers", *Chapters 1-39, 8th Edition, Cengage Learning*, 2012, p. 492.

[217] W. C. Knight, R. G. Pridham and K. S. M, "Digital Signal Processing for Sonar," *Proceedings of the IEEE,* November 1981, vol. 69, pp. 1451-1503.

[218] Ultrasonic Ranging Module HC - SR04 datasheet pdf. [Online] Available: www.Elecfreaks.com. Accessed: 30 December 2015.

[219] "Ultrasonic principle - where high performance sounds good," Microsonic. [Online] Available: http://www.microsonic.de/en/Interesting-facts.htm. Accessed: 12 November 2015.

[220] LV-MaxSonar®-EZ™ Series datasheet pdf. [Online] Available: www.maxbotix.com. Accessed: 29 December 2015.

[221] LM135-LM235-LM335 Precision temperature sensors datasheet pdf. [Online] Available: www.st.com. Accessed: 27 December 2015.

[222] S. Tumanski, "Principles of Electrical Measurement," *Series in Sensors*, *CRC Press*, 2006, p.399.

[223] I. Jeras, "sockit_owm 1-wire (onewire) master," Copyright (CC BY-SA 3.0) 2010–2011.

[224] [Online] DS18B20 Programmable Resolution 1-Wire Digital Thermometer, Available at www.maximintegrated.com. Accessed: 29 Dec. 2015.

[225] J. R. Gray, "Environmental Instrumentation and Analysis Handbook," (Eds): R. D Down and J, H. Lehi, *John Wiley & Sons*, 2005, pp.460–462.

[226] R. Gill, “Modern Analytical Geochemistry: An Introduction to Quantitative Chemical Analysis Techniques for Earth,” *Environmental and Materials Scientists*, Longman Geochemistry Series, Routledge, 2014, p.79–80.

[227] T. L. M. Bartelt, “Instrumentation and Process Control,” *Cengage Learning*, 2006, p.153.

[228] [Online] Campbell Scientific. Available: https://www.campbellsci.com/csim11. Accessed: 2 Jan. 2016.

[229] Atlas Scientific Environmental Robotics pH Probe, pH probe datasheet pdf. [Online] Available: www. Atlas-Scientific.com. Accessed: 2 Jan. 2016.

[230] J, D., Mauseth, “Botany”, Jones & Bartlett Publishers, 2016, p.309.

[231] M.Staden, F Musco., “Local Governments and Climate Change: Sustainable Energy Planning and Implementation in Small and Medium Sized Communities”, *Advances on Global Change Research*, vol.39, *Springer Science & Business Media*, 2010, p.425.

[232] Schwarts, M., (Ed.), Encyclopedia of Coastal Science, Springer Science & Business Media, 2006, p.36.

[233] L, B., Mendes, et al. “NDIR gas sensor for spatial monitoring of carbon dioxide concentrations in naturally ventilated livestock buildings,” *Sensors*, 2015, Vol. 15, No.5, pp. 11239–11257.

[234] J. Hodgkinson, et al. “Non-dispersive infra-red (NDIR) measurement of carbon dioxide at 4.2 μm in a compact and optically efficient sensor,” *Sensors and Actuators B: Chemical* 186, 2013, pp. 580–588.

[235] MG-811 data sheet pdf. [Online] Available: www.alldatasheets.com. Accessed: 22 Dec. 2015.

[236] W. Zheng, X Zhang., X Qiao., H Yan., W Wu, “The Design of Smart Wireless Carbon Dioxide Measuring Instrument Used in Greenhouse,” (Eds): D. Li, Y. Liu and Y. Chen, In *Computer and Computing Technologies in Agriculture IV. CCTA 2010*, IFIP Advances in Information and Communication Technology, vol 346. Springer, Berlin, Heidelberg, 2011.

[237] Gravity: Analog Infrared CO_2 Sensor for Arduino SKU: SEN0219. [Online] Available: www.dfrobot.com. Accessed: 25 Jul. 2016.

[238] F. R. Spellman and J. E. Drinan, “The Drinking Water Handbook,” 2nd Edition, CRC Press, 2012, p. 147.

[239] N. Komninos (Ed.), “Sensor Applications, Experimentation, and Logistics,” *First International Conference, SENSAPPEAL 2009*, Athens, Greece, September 25, 2009, Revised Selected Papers, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, 2010, Vol.29, p.45.

[240] Global Water Instrumentation, Inc. Turbidity Meter: WQ770 User Manual pdf. [Online] Available: www.globalw.com. Accessed: 21 December 2015.

[241] [Online] Turbidity Sensor SKU: SEN0189. Available: https://www.dfrobot.com/wiki/index.php/Turbidity_sensor_SKU:_SEN0189 . Accessed: 26 Aug. 2016.

[242] The Telegesis, ETRX35x-LRS Product Manual (Rev 1.08) ZigBee Modules Product Manual pdf. [Online] Available: www.mouser.com. Accessed: 22 December 2015.

[243] XBee®/XBee-PRO® RF Modules data sheet. pdf. [Online] Available: www.digi.con. Accessed: 22 December 2015.

[244] Analog Devices, 8-Channel, 1 MSPS, 8-/10-/12-Bit ADCs with Sequencer in 20-Lead TSSOP, AD7908/AD7918/AD7928 data sheet. pdf, [Online] Available: www.analog.com. Accessed: 23 July 2016.

[245] XCTU Configuration and Test Utility Software, User Guide pdf. [Online] Available: www.digi.com/xctu. Accessed: 20 December 2015.

[246] [Online] Grafana https://grafana.com/. Accessed: 24 December 2015.

[247] [Online] http://staging.grafana.org/features/. Accessed: 24 December 2015.

[248] [Online] Gens ace 6000mAh 22.2V 35C 6S1P Lipo Battery. Available: http://www.gensace.de/gens-ace-6000mah-22-2v-35c-6s1p-lipo-battery.html

[249] GPS Status & Toolbox 7.6.163 for Android, Available at https://downloads.tomsguide.com/GPS-Status-Toolbar,0301-36219.html

[250] April Climate History for Miri, Available at http://www.myweather2.com/City-Town/Malaysia/Miri/climate-profile.aspx?month=4

[251] V. S. Kale., “Consequence of Temperature, ph, Turbidity and Dissolved Oxygen Water Quality Parameters. *Organization (WHO)*, Vol.*3, Issue*. 8, 2016.

[252] Mathew. P. M., “Limnology and productivity of Govindgarh Lake,” Rewa (M.P.). *J. India. Fish.Soc*. (India), 1975, Vol. 7, pp. 16–24.

[253] Michael. P., “Ecological methods for field and laboratory investigations,” Tata McGraw-Hill, c1984, New Delhi; London, pp.397–400.

[254] Ramchandra. T. V. and Solanki. M., “Ecological Assessement of Lentic Water Bodies of Bangalore, “Indian Institute of Science, Bangalore Envis Technical Report. (2007), p. 25.

[255] Awasthi. U. and Tiwari. S., “Seasonal trends in abiotic factors of a lentic habitat (Govindgharh Lake) Rewa,” M.P. Indian Eco. Env. And Cons., 2004, vol.10, Issue 2, pp. 165–170.

[256] Satpathy. K.K., et al., “A comparison of water quality of two fresh water sources at Kalpakkam,” In *Proceedings of DAE-BRNS national symposium on Limnology*, 2007.

[257] A. A. Kumar. Et. al. “Water quality monitoring: A comparative case study of municipal and Curtin Sarawak's lake samples”, IOP Conf. Series: Materials Science and Engineering 121, 2016.

“Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.”