Universidade do Minho
Escola de Engenharia

Housam Aldakar

A Wireless Sensor Network for Environmental Monitoring System

December 2014

UMinho | 2014

Housam Aldakar A Wireless Sensor Network for Environmental Monitoring System

Universidade do Minho
Escola de Engenharia

Housam Aldakar

A Wireless Sensor Network for Environmental Monitoring System

Masters' Dissertation
Master in Industrial Electronics Engineering and Computers

Supervisor
Assist. Prof. Dr. Jorge Cabral

December 2014

DECLARAÇÃO

Nome : Housam Aldakar

Endereço electrónico: housam.al-dakar@live.com Telefone: 938416823

Número do Bilhete de Identidade: 959X03X57

Título dissertação:

A Wireless Sensore Network for Environmental Monitoring System

Orientador:

Prof. Jorge Miguel Nunes dos Santos Cabral

Ano de conclusão: 2015

Designação do Mestrado :

Master in Industrial Electronics and Computers

Universidade do Minho, ____/____/________

Assinatura: ____

Acknowledgements

I would like to thank my Masters' advisor, Professor Jorge Cabral for all his support during my master period. I am grateful for Dr. Reza Abrishambaf for helping me in everything I needed. The thanks also extend to all the ESRG team that helped anyway in this work. Also I would like to thank all my colleagues in the laboratory for the wonderful work environment.

I am grateful for The Global Platform for Syrian Students (APGES) represented by the former president of Portugal Dr. Jorge Sampaio with the assistance of Dr. Helena Barroco and all the platform team for believing in me and awarding me this scholarship.

I would also like to thank my parents Anis Aldakar and Hala Auki who always believed in me and supported my decisions, also I would like to thank my sister Yara for being the shoulder that I lean my head on. I will not forget my sister Dahlia that has always been supportive and helped with advice anytime.

I am grateful for my friend Yourkey Alber for helping me to have a new beginning.

Last but not least I would like to thank God because without his will I would never be here.

Finally I would like to thank all my friends for being there for me.

Abstract

Studying environmental changes posse some challenges on the level of the provided services to simplify the monitoring operation. And it is not possible to achieve a real quality of these services without an integrated information system that is automatically and self-developed that provides needed data for any service anywhere. Wireless sensors networks represent the latest embedded systems and wireless communication technologies that allows to gather information in a distributed way, and that in the field of monitoring and remote controlling in many applications, specially environmental, health, positioning, guiding and traffic organizing. This thesis aims to provide a general visualization of using wireless sensors network as an effective solution and a developed technology, more flexible and able to collect information and data automatically to provide services in the environmental field.

Keywords

WSN, Embedded System, Remote Control, Node, SoC, and Application.

Resumo

O estudo das mudanças climatéricas impõe alguns desafios ao nível dos serviços utilizados para facilitar as operações de monitorização. Ao mesmo tempo, não é possível alcançar a real qualidade destes serviços sem a necessidade de integração de um sistema automático e autodesenvolvido de informação que forneça os dados necessários para qualquer serviço em qualquer lugar. As redes de sensores sem fio representam os mais recentes sistemas embebidos e de tecnologias de comunicação sem fio que permite reunir informações de forma distribuída na área da monitorização e controle remoto, especialmente ao nível do meio ambiente, saúde, posicionamento, orientação e organizando o tráfego. Esta tese tem como objetivo proporcionar uma visualização geral de uso de redes de sensores sem fios como uma solução eficaz e uma tecnologia desenvolvida mais flexível e que capaz de coletar automaticamente informações e dados para prestação de serviços no domínio do ambiente.

Palavra chave

WSN, Embedded System, Remote Control, Node, SoC, and Application.

Table of Content

List of Figures

List of Tables

Table of shortcuts

ACID	Atomicity, Consistency, Isolation, Durability
ADC	Analog Digital Converter
ALU	Arithmetic Logic Unit
API	Active Pharmaceutical Ingredients
ASIP	Application Specific Instruction set Processor
ASP	Application Service Provider
ATM	Automated Teller Machine
CCD	Charged Coupled Device
CNC	Computer Numerical Control
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DAC	Digital Analog Converter
DBMS	Database Management System
DC	Digital Current
DDL	Data Definition Language
DMA	Direct Memory Access
DML	Data Manipulation Language
DQL	Data Query Language
DSP	Digital Signal Processor
ESB	Embedded Sensor Board
ESG	Electrocardiogram
FPGA	Field Programmable Gates Array
GIS	Graphical Information System
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer

HD	High Definition
IC	Integrated Circuit
IOAMR	Intelligent Operation Administration and Maintenance Router
IR	Instruction Register
IRS	Internal Revenue Service
IRS	Internal Revenue Service
ISA	Industrial Standard Architecture
ISM	Industrial, Scientific and Medical
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LCD	Liquid Cristal Display
LPRF	Low Power Radio Frequency
MMI	Multi Media Interface
MRFI	Minimal RF Interface
NRE	Non-Recurring Engineering
OLAP	Online Analytical Processing
OS	Operating System
PC[1]	Program Counter
PC[2]	Personal Computer
PDA	Personal Digital Assistance
PHY	Physical Layer
RAM	Random Access Memory
RF	Radio Frequency
RTC	Real Time Clock
SD	Secure Digital
SoC	System on Chip
SQL	Structured Query Language
SRD	Short Range Devices
TI	Texas Instruments

UoD	*Universe of Discourse*
UWB	*Ultra Wide Band*
WSN	*Wireless Sensors Network*

1 Introduction

Wireless sensors networks (WSN) are considered one of the most important technologies in our daily life, they have been a part of almost any type of monitoring system. A wireless sensors network is a network with self-configuration of a group of sensor nodes, which communicate with each other using radio frequency signals. These sensor nodes usually are deployed in quantity to monitor, sense and study the physical surrounding environment.

Wireless sensors networks enable to observe, study and sense what was previously unobservable, because of some obstacles that used to face the researchers such as distance, narrow places, toxic environments and even because the size of the equipment that had been used and other restrictions related to cables and wires.

Wireless sensors networks provide a bridge between the virtual and real physical worlds.

They also have a big range of potential applications in transportation, civil infrastructure, science, industry and last, but not least, security.

This technology is important for different type of people in the community, for instance, according to research studies [4] it is considered as a new technology that allows for further investigations to be made in that field, and as a governmental point of a view, there is a lot of application that could serve the society very well, in addition in the business field, it is considered as a growing new market, with a lot of opportunities to invest in and emerging economies, finally, in a networking point of view the wireless sensors networks are considered as the new members of the internet.

In another words, Wireless Sensors Networks (WSN) are considered one of the new technologies that recently brought the attention of science society, and lead to many scientific researches to study their applications and properties. This technology represents a scientific revolution in the field of networking, wireless communications and embedded systems, as the previous one that happened with wired networks during the evolution of internet during the eighties of the previous century. The appearance of wireless sensors networks lead to create many new applications related to monitoring, remote controlling and distributed computing in different fields such as environment, health monitoring, buildings and infrastructure safety check, security such as detecting intrusion to banned areas, traffic, farming and monitoring agricultural environment and so on.

From this point, this dissertation aims to create a general perception of using wireless sensors networks as an effective solution, developed technology, and more flexible of building a distributed information system that is able to provide a good service in environment monitoring field.

In this project the sensor nodes will be sensing climate changes in addition to luminosity and soil temperature, the data will be gathered wirelessly into a mini computer that will register this data into a database, user can check and use this data results to do some responds.

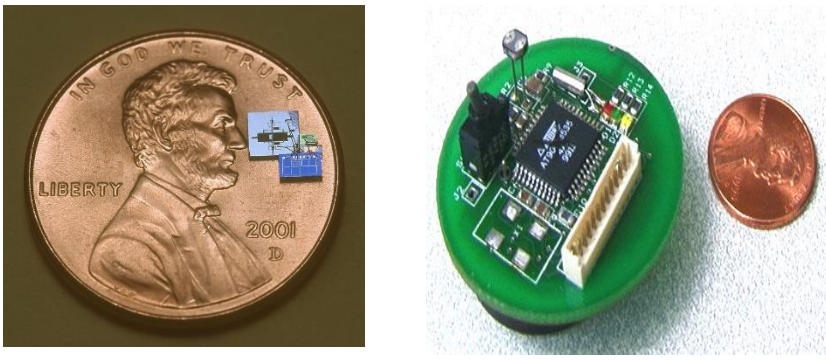
2 Wireless Sensor Network

2.1 Introduction

Information technology in less than a decade became a major part in our lives, it has affected the developed computers ability to perform complex processes in addition to the progress in each of the computer networks and embedded systems on many aspects of human life, currently high precision cameras connected to computers are used in many security applications, for example, it has been used to monitor important facilities, airports as well as roads traffic.

Many researchers and companies have been attracted to develop wireless sensor networks, so, several researches have the conviction that this kind of networking will become one of the technologies prevalent in the next few years, for example, companies spend millions of dollars to design smart sensors similar to dust particles. It is expected that these devices can shift the physical world to the digital one by gathering information related to the environment which these devices has been pdeployed in.

The previous sensor's generation suffered from the high cost and large size, and this is what attracted scientists to discuss the possibility of building and developing sensors that are smaller and less expensive, for example, the University of Berkeley produced a sensor with the name (weC) to measure the temperature and the intensity of light, Figure 2.1-A shows the extent of the small size of this device, however, in spite of its small size it has been provided with a radio frequency link (916.5 MHz) and has a range up to twenty meters, As Figure 2.1-B illustrates, another example is the so-called Golem Dust which has a volume of 11.7 mm^3, this device gathers its power from the sun and has the ability of bidirectional RF communication.



A **B**

Figure 2.1 - Sensors sizes, A. (Golem Dust) [39] B. (weC) [40]

Many designed experimental models illustrate the feasibility of deploying a large number of sensors on a small area, where they cooperate in forming a wireless network to control and monitor the under study phenomena, these devices are distributed either according to predefined scenario or they are deployed randomly, in the first type the distribution field is well known, and can locate the spread sensor devices in advance, and in the case of random distribution they are usually deployed by helicopters, and in both cases the sensors deployed in the study field transmit the data that been monitored to the main station as shown in Figure 2.2, which offer it to the user.

For the wireless networks to be capable of performing effectively, there are many challenges that must be addressed, including the fact that these devices suffer from limited energy and limited memory plus the medium computation ability, also the ability to spread, route, data integration and information processing in addition to the level of reliability represent some of the challenges that face the growth of its usage.

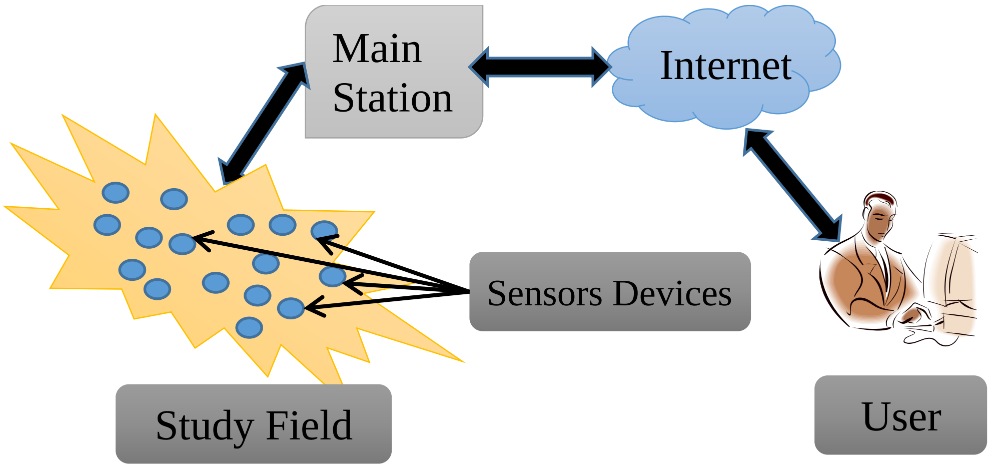


Figure 2.2 - Wireless Sensors Network Model [4]

This research aims to inform the reader about the of progress in the field of wireless sensor networks, it starts by review of the physical characteristics of the sensors, as there will be selection of some means of evaluating their performance, and then will show the challenges facing this kind of networks, and will file some different applications according to the attributes of network operation, in addition, will show some of the problems discussed in the field of sensor networks.

In the following pages of this chapter there are some explanations about the main components of the sensor model and ways to evaluate its performance, the various challenges facing the use of this kind of networks, the applications currently in use, and some existing problems and visions of the future work to further improve the performance of the networks.

2.2 What is Sensor Node?

A sensor node (in the context of WSN) is a device that contains a microprocessor, input transducers with monitoring capabilities and a radio frequency transceiver to enable wireless communication, the sensor node often suffers of the small size of the memory, also suffers from the limited energy supply.

Reviewing the components of the sensor, as in Figure 2.3 contains of the following units: a) sensing unit, b) data storage and processing unit, c) transceiver unit. The sensing unit consists of a sensor and a data conversion unit from analog to digital, the main task of this unit is to convert data sent or received to a formula that suits the nature of the data used in the storage and processing unit, at first strengthen the signal received from the sensor and then turning it into digital using the data conversion unit. The storage and processing unit is a chip that contains limited memory unit and data processor, and also there is the transceiver unit; this unit consists of a device for sending and receiving radio waves through the antenna installed on the sensor.

In addition to the units previously mentioned, there are three optional units, which are: a) GPS - depend on the application type, its main function is to determine the coordinates of the sensor in the monitored field compared to a fixed point, b) mobility unit which used to move the sensor from one place to another depending on the network requirements c) power generating unit where it refill energy supplies. More information in this field mentioned in reference number [1].

Based on the capabilities of the sensors they can be classified into four categories [2] [3], a specific device, general, high-bandwidth and other high-efficiency that work as a gateway for other devices, and each of these categories differentiate from the other in the amount of energy, the size of memory and the ability to calculate and communicate, usually these specifications are used to assess the efficiency of the hardware, more explaining about the ways of assessing the sensors in the following paragraphs.

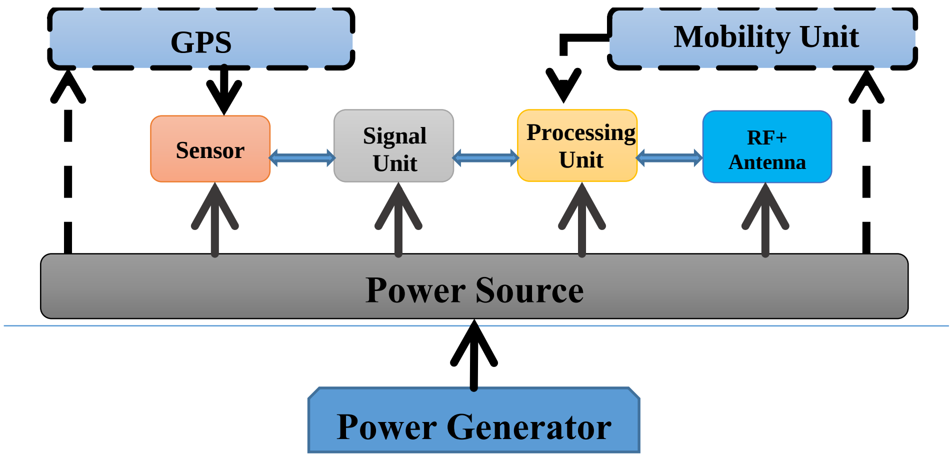


Figure 2.3 - Main units in a sensor node device [4]

2.2.1 Power

Modern applications in the areas of wireless sensors require on one hand devices with long lifespan, and on the other hand these devices have power restrictions, as there are several factors that affect power consumption [4] [5], for example, a) The number of input devices, b) The number of performed services, c) The transmit and receive duration, d) Surrounding environmental conditions such as temperature, e) The required reading accuracy, f) Used radio waves.

Usually sensor nodes are provided with two rechargeable AA type batteries, but with the use of hundreds of thousands of these devices in the monitoring field the recharging of the batteries is impractical, and therefore new strategies for power conservation must be found, as in integrating logical programming chips (FPGAs) with sensing device (ATMEL) [6], and can also make use of renewable energy such as solar energy or by vibration, which is one of the important ways to overcome the problem of energy.

2.2.2 Memory Size

The sensor nodes generally contain a small sized memory unit, this means that there are limits on the amount of gathered data that can be stored before running out of memory space. This means that this data must be fast processed and analyzed or sent to neighbors devices, the old types of sensors used the two types of volatile memory SRAM and SDRAM, while the new sensors contain the two types of memory together, but they are integrated with the same device chip in addition to the use of external flash memory, for example, the sensor Imote2 contains a built-in memory with a capacity of 256 KB and 32 MB of SRAM and 32 MB of SDRAM in addition to the 32 MB of flash memory, and although the flash memory technology requires a larger chip's space when compared to memory type of SRAM or SDRAM, they are the most efficient in energy consumption, but less efficient when writing repetition is required.

2.2.3 Ability of Data Processing

The processor in the sensor node plays an important role in analyzing and processing data observed by the device itself or received from other devices after completing the process of analyzing this data, the sensor node sends it in a message -could be encrypted- to neighboring devices, and this requires controlling the radio and dealing with message interpretation and storing, additionally the processor does another job which is data collection, and this collecting is a specific sensor responsibility to merge between local and received data, some of this collected data may be refused and the others might sent to neighboring devices, and one of these devices with high-efficiency processors device which was previously mentioned is Imote2, in this device the used processor is PXA271 Intel XScale, it also supports low frequencies - 13 MHz - and can work in low energy consumption mode - 8.5 Volt -, which is suitable to complex applications such as surveillance using digital cameras.

2.2.4 Connect ability

The radio unit is the most important components of the sensor node and it is also the most energy-consumption unit [7], estimate that 97% of the energy consumption related to the transmission and reception either with direct use of the radio unit or as a result of the processor unit waiting for the radio to complete the transmitting or receiving. The current radio technology operates on the basis of data transmission on short-wave, and this includes the standard technology such as Bluetooth, ZigBee and UWB (Ultra Wide Band), for example ZigBee technology enables a communication between 254 sensors at the same time using 2.4 MHz frequency. Some other nonstandard technology might be used for data transfer, but this may limit the ability of the sensors networks.

2.3 Challenges

The integration between the current sensing technologies with the operating systems, opens the way for the formation of (Ad-hoc) networks using low cost sensors, and these networks are able to monitor different factors, such as temperature, humidity, noise level, fixed or mobile objects, light, and others. It is expected from this network to monitor these factors and report them to a sensor node (acting as coordinator) with a powerful processor and large memory unit with the ability of wireless connection, but due to sensors prices, wireless sensor networks face a number of significant challenges, which are summarized as follows.

2.3.1 Deployment

The first stage of sensor nodes deployment process is considered the formation of the Wireless Sensors Networks, at this stage the deployment of these devices must be made with great care in order to meet the wanted targets (number of devices and range), but the nature of the field on which the sensor nodes will be deployed and the number of available devices might affect the deployment method. In case of possibility to access the fields and locating the sensors in advance, the process of deploying the sensors could happen manually or using a robot, on another hand some fields could be in inaccessible areas, or that there are a large number of sensors that should be deployed; so the best way in the deployment strategy might be through using airborne method even though the costs are higher when compared with the costs of traditional deploying.

In both methods, the responsibility to cover the field and the interdependence of the network and re-distribution is left to the sensor nodes, it is clear that with the process of deploying new challenges arise, and in addition to that, the sensors network is self-governing, and so when it is deployed it must discover its environment and neighbors to adapt itself according to that, and moreover, some other factors such as monitoring and multiple coverage [8] may increase the complexity of the deploying process.

2.3.2 Power

Despite the limited capabilities of the sensor nodes in terms of processing, storage and communication, it is expected for these restrictions to be resolved in the near future, but it seems that the restrictions in energy consumption imposes a harder challenge because of the slow progress in the batteries development, these sensors require frequent change of the batteries, which sometimes may be impossible, so energy efficient protocols are essential to improve the reliability of sensors networks.

2.3.3 Network Structure

If we compare the sensors networks with the traditional ones, the format and the structure of wireless sensor networks differs in a lot of things, for example there are many operating characteristics of the sensors that can only be known after the completion of organizing and deployment. Due to the limited capacity of these devices, many of the operations must be implemented in parallel such as monitoring and encryption and data transfer [3], because of all these differences, one definition for the structure of wireless sensor networks is a daunting task, and also the wireless sensor networks still rely on the techniques and protocols used in the structures of traditional networks.

2.3.4 The Reliability Level

There are many components that need to be considered when dealing with sensor networks, including the level of reliability, usually these networks suffer of transmission and reception errors due to collisions and congestion of data packets, and in addition to that, these devices are prone to failure due to either some damage or interference and control of these devices, in addition, there are a lot of details that relate to the form of the used messages, the messages may be formed as a single packet or set of packets that they are sent as several packets with each other or in the form of serial packets, these types of messages may require guarantee delivery or make a better effort in the delivery, the main station can send messages either for all sensors, to a specific set of them or to a specific area in the monitored field, and vice-versa. However the routing technology can be built on the basis of requesting a route when needed or use a predetermined route, all these factors have to be considered when dealing with reliable wireless sensor networks, and therefore, the development of a networks reliability protocol that takes all of these factors is another challenge.

2.3.5 Networks Monitoring and Routing

The challenges that face wireless sensor networks in terms of network monitoring and routing come from the physical and operational constraints of the sensors tools, on the other hand they come from the high demand for using the sensors by some applications, these two specifications - that make the conflicts with the wireless sensors network technology - makes it difficult to absorb routing protocols developed and used in ad-hoc networks, as well as the dynamism of wireless sensor networks is considered one of the challenges in the message routing, and dynamism in this context back to each of the networks types and the monitored factors where they are affected by the sensor's movement, while the monitored factors are affected by the monitored object, which may be moving, or the event that has been monitored - which can happen in any time - and this dynamism directly affect the routing protocols where it must maintain the devices connected and communicated.

Moreover, some applications could face a big challenge when it is intended to achieve several goals simultaneously while trying to send a message from source to destination, mainly to increase the networks age and provide additional security. For example, some applications may need to provide a balance between energy consumption and the amount of delay and the risk level in the transmission and coherence the information to promote data integrity and the missing messages and sensing accuracy and coverage ratio derived from networks [9].

2.3.6 Programmability

One of the main challenges that needs to be addressed is the ability of the devices in wireless sensor networks to be programmed, and that because the current interface to sensors require users to participate in a lot of the programming details, where they must prepare and equip the communication between sensors and identify the gathering ways and choosing the databases in addition to some other functions, also the effort is little to absorb the sensors (queries). For example, (Yao and Gehrke) [10] suggested adding a new layer to translate queries which looks like (SQL) to and from the sensor, However, the language of inquiry currently being used it is only a simplified model and it is far from perfect, so it is very important that the interfaces allow the user to prepare the sensor in a simplified manner such as the availability and use of graphical interfaces or methods of voice interactive.

2.3.7 Network Security

The wireless sensor networks - as opposite as traditional networks - are usually deployed in open areas and exposed to environmental changes, so the sensors are susceptible to external direct attacks, and that the techniques of the current security and methods of data compression is considered inappropriate to be used, as a result of limitations in the calculation ability in the sensors, moreover, the structure of these devices and communication channels at its design was not considering the security objectives, and as a result, the user authentication, data confidentiality, the adoption of encryption key and the resistance to any kind of attacks on the communications imposed a new challenge.

2.3.8 Expandability and Mobility

Due to the limitation in the sensors communication they communicate using several leaps (Multi-hops), also they work with each other to transfer the allocated data to the main station, there will be also a number of messages flowing through the network, it all happens because of the amount of the local data plus data coming from neighboring devices, also covering the monitored field more than once considered essential to overcome any possible defect, the mobile sensors may affect the functioning of the network, where the process of tracking and positioning are required, so with all these factors, adding new sensors will pose the following question: Is it possible to maintain the same network communication efficiency with the introduction of a greater number of sensors considering overlapping and signals collision?

2.3.9 Data Collecting and Processing

The sensor nodes are more efficient in dealing with local data, compared with sending raw data to the main station for processing, and thus the large number of sensors may lead to a flood of messages to solve this problem some of the sensors are chosen to collect data, but the collecting algorithms requires to store the messages before processing, and this is a big challenge in sensors with weak processing capability and with limited storage space, some might suggest that the amount of data - sent from each device - could be small that it could be stored, and as an example, transmission of metadata (Meta-Data) or summary of the observed data can be enough to report any event from the sensor, note that selecting the collecting devices in a continuously changing structural wireless sensor network is another challenge.

2.4 Applications

Sensors networks can be used in different applications including environmental monitoring, border security, health care, military applications and disaster management these applications can be classified on the basis of sensor networks operation into three categories: Querying applications, Tasking applications or a combination of both. In the first category the main station may order a specific set of sensor nodes to monitor some events, while in the second category, the sensors send a report on the observed data when an event occurs. In the next sessions, some reviews about applications in each category.

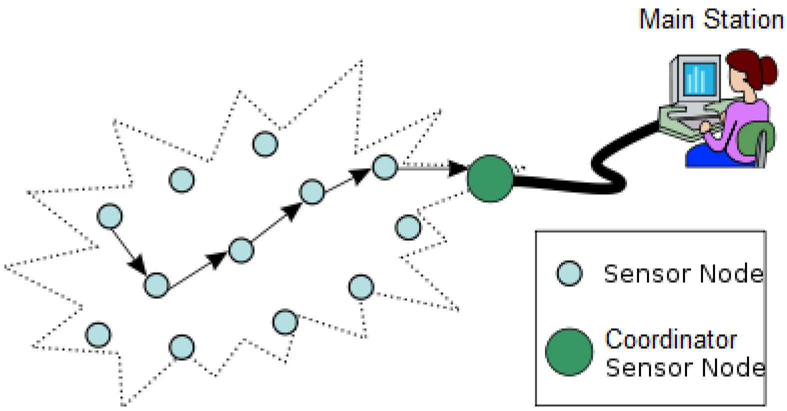


Figure 2.4 - Typical WSN with Sensor nodes, Coordinator and Main Station [41]

2.4.1 Querying Applications

It is used in applications in which they extract data from sensors via a language similar to SQL this method is effective considering the amount of energy consumed and that because the transmitted messages between sensors is the lowest number of messages, but this method requires using smart sensors that can store data for long periods of time and deal with various queries sent from the main control station, and these qualities may not be available for small-scale sensors, as in (Berkeley Motes) [11].

One example is reading the temperature and humidity by a defined sensor in the monitored field [12], and also smart elderly homes [13] - as shown in Figure 2.5 - are another example of these applications so they can be where some elderly request the sensor to know whether the main door is closed or not, or if the TV is off or not, or to ask to know if there was a lack in the food available in the refrigerator (fridge), and in addition to that, in some complex applications may requires more than one sensor to monitor an event in a particular geographic area.

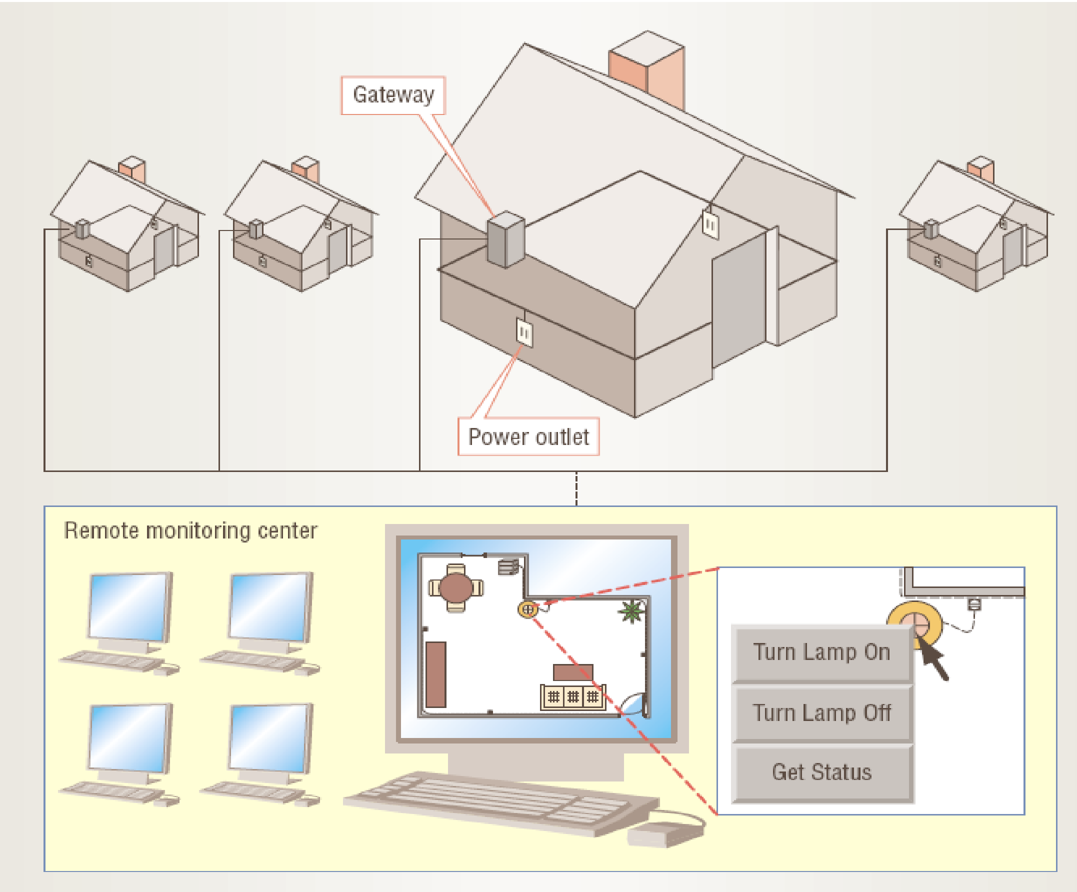


Figure 2.5 - Outline of the Smart house Technology (Gator Tech) [13]

In the agricultural environment field there has been used sensors networks to measure the soil properties, where a group of researchers at the Johns Hopkins University [14] built a model - shown in Figure 2.6 - that deployed a set of sensors in Baltimore forest, where these devices measured the soil moisture and temperature, and this experiment ensure the efficiency of this kind of networks in discovering the soil properties.

A **B**

Figure 2.6 - A. Baltimore Model, B. The Sensor used in the model [14]

In the applications of fires relief, Chen, Majidi, Doolin, Glaser and Sitar [10] made an experimental model to help extract information about a wildfire at a site, In this model, the researchers used a number of sensors called (FireBug), which is designed to measure the temperature, these devices send data to the control center, which is constantly moving around the site of the fire, as shown in Figure 2.7.

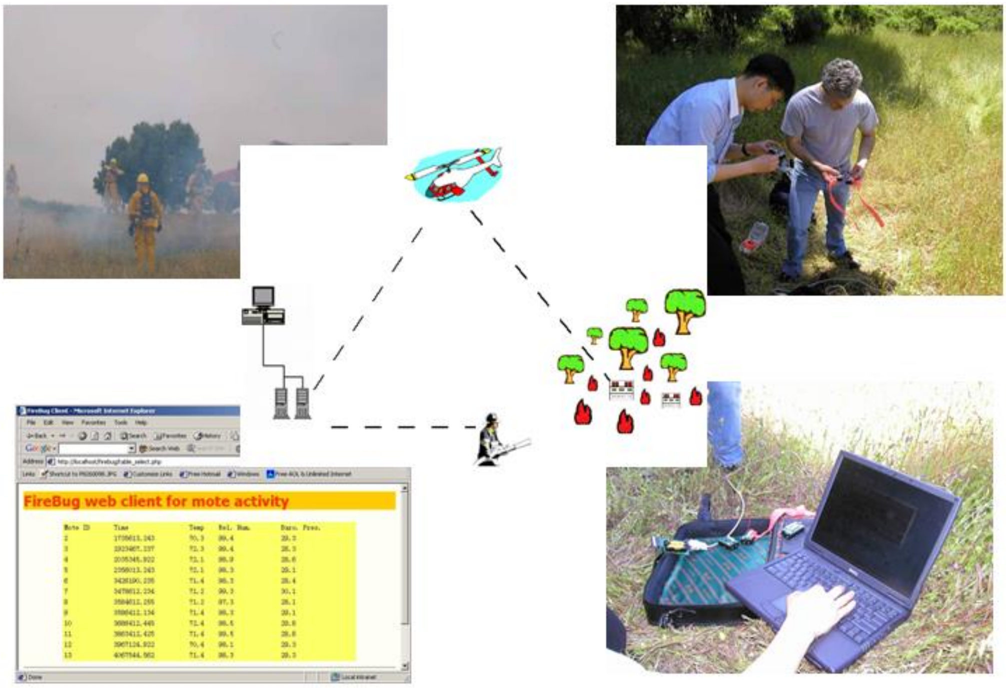


Figure 2.7 - FireBug network [10]

2.4.2 Tasking Applications

Sensors send report about collected data whenever getting an event, as an example, using sensors in the battlefield to detect the presence of danger [15], as well as the discovery of enemy movements and their directions, such applications may require continuous notification about the surrounding events, monitoring (Great Duck Island) is another example of such applications, which collaborates thirty two sensor to monitor the birds and their behavior on the island, this network has been able to answer the three questions identified by researchers [1], namely:

A) What is the nesting pattern in burrows during the (24 – 72) hours when one or both parents make shifts between comfort and fishing?
B) What are the changes that can be observed including burrows and environmental factors on the surface during the breeding season which is nearly seven months (April - October)?

C) What are the differences in small environments with the presence of large numbers of seagulls nesting or not?

(Demirbas et al.) and others talked about their experience in the deployment of sensors for (Internet-Sensor Integration for Habitat Monitoring (INSIGHT)) [16], where the deployed sensors send a report about temperature, humidity and photographs of active artificial radiation in addition reading interior voltages to the main station, and then this station analyze the data and build a database and make it available for browsing on internet.

In the volcanoes controlling program proposed by (G. Werner) and others [17] There are sixteen sensor in the network equipped with voice capabilities has been spread on the volcano (Volcán Reventador) north of Ecuador to keep an eye on different volcanic events, and in this program the researchers found that the sensor was able to register and report two hundred and twenty-nine earthquake in addition to monitoring the explosions and other acoustic events. As a final example of these applications the network of (CodeBlue) that respond to emergency situations [18] shown in Figure 2.8. This network consists of sensors and other wireless devices such as Personal Digital Assistants (PDA), and the main idea of this is to use sensors to monitor patients to alert doctors and nurses using the PDA for emergencies, and in addition to that the information monitored by sensors are saved in the patient's record for further analysis, and despite the fact that this model shows that sensor networks can be useful in the field of emergency response, however, some issues such as the level of reliability and security need careful study and deep analysis .

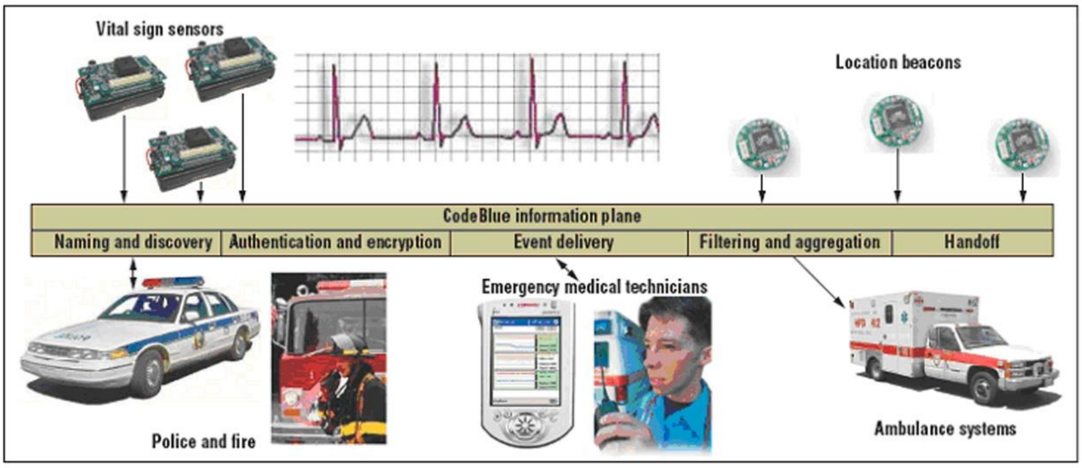


Figure 2.8 - CodeBlue [18]

2.4.3 Querying and Tasking Combination Applications

Merging between the two previous categories found a third category of applications that can make an inquiry from a sensor and receive data periodically based on events that are monitored, but this approach requires a great communication between sensors as well as between these devices and the main station, and in addition to that, some sensors with a high capability to handle inquiries are needed, and there is a new framework for sensor networks - proposed in [19] - to solve this problem by moving the high capable devices to the sensors network sides with a recommendation of integration among themselves and devices with low capabilities, where the last ones report the information which are monitored while the high capable devices collect and analyze these data and respond to incoming queries, one of the latest applications in this approach is the real-time classification and target locating - proposed in [20] -, where researchers have developed a network of sensors to classify and locate animals in populated environment, and analyzing the data observed by the network and then building a database available for queries, this network has managed to inventory and classify animals in populated environments with high qualification.

In the field of traffic monitoring (M. Karpinski and others.) [21] developed an application using the sensors to early warn drivers of congestion or accidents in the highways, and in addition to that it can use the information monitored by other surveillance devices to guide drivers to the best alternative route that can be taken, and another example of this kind of applications, it has been developed a sensors network to determine free spaces in a garage, inside car there is a tool to receive information periodically from sensors deployed in the garage, and in addition to that the current sensing technology may allow drivers to check the places reserved for them even before arriving.

In another example for these applications (FireNet) structure [22] to rescue from fires, the network assumes the existence of sensors with firefighters and their vehicles while the main vehicle is equipped with a laptop that works as a gateway, and the fighters are located using GPS devices, and based on this structure sensors can report the available information to the gates, in addition to receiving commands from the main gateway.

2.5 Open Issues

The effectiveness of sensors network stimulated many researchers to use them in different applications. In the previous section mentioned some of these applications, and in this section there will discussion about the issues that still need to be solved for the best use of these devices in daily life.

2.5.1 Security

Currently there are many ways of network security, such as the use of the secret key for encryption and the use of encrypted passwords - these methods have proven effective in the past few years - but its flaws that they need high computing ability, which is not available in the current sensors, and at the same time sensors networks may be used to transfer very important information, for example in medical emergency operations the networks are used to transfer medical records of the patients and their medical history, and likewise may use the network to transfer military information which needs to be highly classified, moreover the privacy and users authentication which are important issues in the field of network security, doctors and nurses at the emergency, for example, they need to access to the encrypted data of patients as fast as possible, a small number of research have been published in this field, on the other hand these researches are still in their early stages and needs more research.

2.5.2 Networks Maintenance and Data Routing

As an example highways monitoring using sensors networks [21], the network is responsible for the transmission of available information to many users and control unit in the roads, and this information affect directly our daily lives, and therefore in such applications must deploy sensors on the highways and this environment increases the probability of losing and disabled devices, therefore should be an efficient and fast way to address the imbalance caused by the loss or stop working of one of these devices while ensuring to not having overlap between the signals of the devices.

2.5.3 Mobility

The internal structure of the wireless sensors network as a result of the continuous movement of the devices, for example in (FireNet) previously mentioned, the firefighters who hold these devices are continuously moving, and therefore the data observed from these devices need to be sent on real time, therefore designing an efficient and fast protocol to send this data - considering the changing of the internal structure of the network and the environment - still needs more research.

2.5.4 Performance Standards

The field of wireless sensors networks is still in its initial stages, and the structural and protocols used in this area is still in a state of constant evolution and this is what requires the existence of metrics that can help to preview and select the amount of efficiency, and the development of these standards can take advantage of the available research in the field of temporary wireless networking - that has evolved in the past few years - which has defined standards of efficiency when studying these networks.

2.6 SoC

The SoC are Initials of the phrase (System on Chip), which means integrating computer components which can be found on the motherboard or other electronic system on one circuit chip. This chip must has the processor unit and the input and output interfaces in addition to modern communication that works with various types of signals such as Digital Signal, Analogue Signal and mixed. SoCs are very common in the mobile electronics market because of their low power consumption. A typical application is in the area of embedded systems.

Theses specifications describe Microcontrollers as well, in fact there is a fundamental and essential difference that controllers often include what it needs of memory RAM and Flash inside and often these memories sizes do not go beyond 1MByte, which means insufficiency in case we wanted a reliable system to run Windows or Linux, and here comes the role of the SoC that needs to be supported with external memory to work like Flash memory to save programs and RAM to run the program, and ROM which contains the basic boot.

The SoC generally contains:

1. Processor or controller or digital signal processor core DSP Core, and could contain more than one processor core.
2. Memory segments of various kinds, "and these sizes are small and do not compensate for the need of additional external memories".
3. Clock pulses generators including oscillators which provide the main frequency and the PLL circuits responsible for the doubling of the frequency.

4. Peripheral circuits such as counters and timers and the real time clock circuits (RTC).
5. Chip interfaces that usually include industrial standards like USB, FireWire, Ethernet, USART, SPI, and other.
6. Analog interfaces like ADC (Analogue Digital Converter) and DAC (Digital Analogue Converter).
7. Voltage regulators and energy consumption organizers.

Currently many things are added such as PCI-Express, graphical processors GPU, etc., and the SoC contain protocols for connecting between these inner components as AMBA made by ARM.

2.6.1 SoCs Field of Use

In fact, SoC can be used in any place that needs a computer and even it is able to reach places Computer can't reach as most important the Mobile devices and the Smartphones, perhaps the current revolution carried out by the Android system and Apple's Attempts to overcome it with their system IOS increase the demand for these chips, all the Smartphone currently rely on the SoC as the heart beating inside them, and with the rapid development that we are witnessing now SoC designing companies are working hard to push it to its limits in terms of energy efficiency and access to their best performance.

2.7 CC2530 SoC

The CC2530 [23] is a true system-on-chip (SoC) solution for IEEE 802.15.4, Zigbee and RF4CE applications. It enables robust network nodes to be built with very low total bill-of-material costs. The CC2530 combines the excellent performance of a leading RF transceiver with an industry-standard enhanced 8051 MCU, in-system programmable flash memory, 8-KB RAM, and many other powerful features. The CC2530 comes in four different flash versions: CC2530F32/64/128/256, with 32/64/128/256 KB of flash memory, respectively. The CC2530 has various operating modes, making it highly suited for systems where ultralow power consumption is required.

Short transition times between operating modes further ensure low energy consumption.

2.8 Contiki OS

Contiki is known as an open source operating system used for wireless sensor networks and the Internet of things. Contiki provides a low-power networking for the resource constrained systems along with a development and good simulation environment that makes research, development, and deployment easy. Examples of where Contiki can be used include street lighting systems, sound monitoring for smart cities, radiation monitoring systems, and alarm systems [24].

Contiki contains the low-power wireless Rime communication stack, the uIP TCP/IPv4 stack, and the IPv6 Ready certified uIPv6 TCP/IPv6 stack [25], completed with 802.15.4 6LoWPAN header compression and fragmentation.

Despite providing multitasking and a built-in TCP/IP stack, Contiki needs only about 10 KB of RAM and 30 KB of ROM. A full system, completed with a graphical user interface, needs about 30 kilobytes of RAM.

As the number of different sensor device platforms increases, it is desirable to have a common software infrastructure that is portable across hardware platforms.

The currently available sensor platforms carry completely different sets of sensors and communication devices.

Due to the application specific nature of sensor networks, it is not expected that this will change in the future. The single unifying characteristic of today's platforms is the CPU architecture which uses a memory model without segmentation or memory protection mechanisms. Program code is stored in reprogrammable ROM and data in RAM. Contiki was designed so that the only abstraction provided by the base system is CPU multiplexing and support for loadable programs and services. As a consequence of the application specific nature of sensor networks, we believe that other abstractions are better implemented as libraries or services and provide mechanisms for dynamic service management [26].

2.9 System Overview

A running Contiki system consists of the kernel, libraries, the program loader, and a set of processes. A process may be either an application program or a service. A service implements functionality used by more than one application process. All processes, both application programs and services, can be dynamically replaced at run-time.

Communication between processes always goes through the kernel. The kernel does not provide a hardware abstraction layer, but let device drivers and applications communicate directly with the hardware.

A process is defined by an event handler function and an optional poll handler function. The process state is held in the process' private memory and the kernel only keeps a pointer to the process state. On the ESB platform [26], the process state consists of 23 bytes. All processes share the same address space and do not run in different protection domains. Inter-process communication is done by posting events.

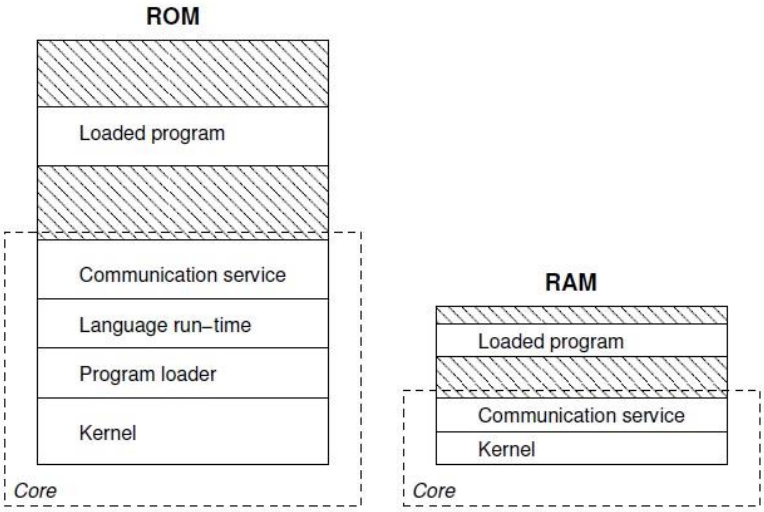


Figure 2.9 - Partitioning into cores and loaded programs [26]

A Contiki system is partitioned into two parts: the core and the loaded programs as shown in Figure 2.9. The partitioning is made at compile time and is specific to the deployment in which Contiki is used. Typically, the core consists of the Contiki kernel, the program loader, the most commonly used parts of the language run-time and support libraries, and a communication stack with device drivers for the communication hardware. The core is compiled into a single binary image that is stored in the devices prior to deployment.

The core is generally not modified after deployment, even though it should be noted that it is possible to use a special boot loader to overwrite or patch the core.

Programs are loaded into the system by the program loader. The program loader may obtain the program binaries either by using the communication stack or by using directly attached storage such as EEPROM. Typically, programs to be loaded into the system are first stored in EEPROM before they are programmed into the code memory.

2.10 Kernel Architecture

The Contiki kernel consists of a lightweight event scheduler that dispatches events to running processes and periodically calls processes' polling handlers. All program execution is triggered either by events dispatched by the kernel or through the polling mechanism. The kernel does not preempt an event handler once it has been scheduled. Therefore, event handlers must run to completion. However, event handlers may use internal mechanisms to achieve preemption.

The kernel supports two kind of events: asynchronous and synchronous events. Asynchronous events are a form of deferred procedure call: asynchronous events are queued by the kernel and are dispatched to the target process some time later. Synchronous events are similar to asynchronous but immediately causes the target process to be scheduled.

Control returns to the posting process only after the target has finished processing the event. This can be seen as an inter-process procedure call and is similar to the door abstraction used in the Spring operating system [26].

In addition to the events, the kernel provides a polling mechanism. Polling can be seen as high priority events that are scheduled in-between each asynchronous event. Polling is used by processes that operate near the hardware to check for status updates of hardware devices. When a poll is scheduled all processes that implement a poll handler are called, in order of their priority.

The Contiki kernel uses a single shared stack for all process execution. The use of asynchronous events reduce stack space requirements as the stack is rewound between each invocation of event handlers.

2.11 Services

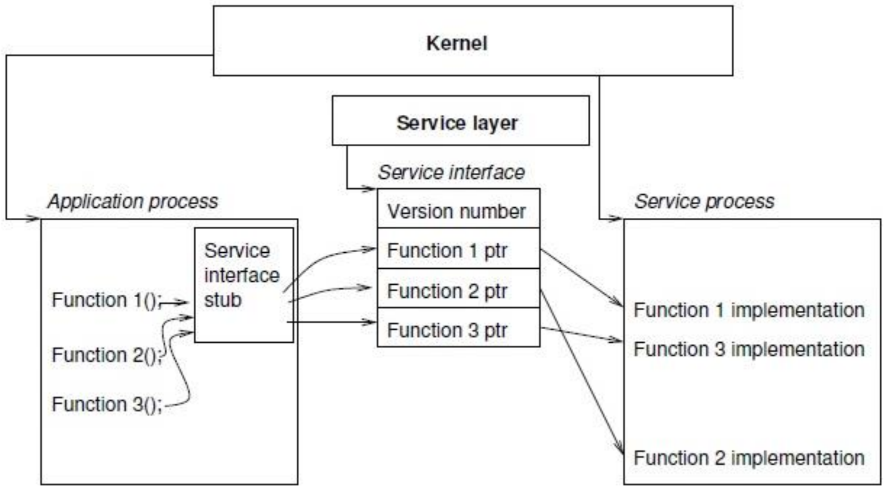


Figure 2.10 - An application function calling a service [26]

In Contiki, a service is a process that implements functionality that can be used by other processes. A service can be seen as a form of a shared library. Services can be dynamically replaced at run-time and must therefore be dynamically linked. Typical examples of services includes communication protocol stacks, sensor device drivers, and higher level functionality such as sensor data handling algorithms.

Services are managed by a service layer conceptually situated directly next to the kernel.

The service layer keeps track of running services and provides a way to find installed services. A service is identified by a textual string that describes the service. The service layer uses ordinary string matching to querying installed services.

A service consists of a service interface and a process that implements the interface. The service interface consists of a version number and a function table with pointers to the functions that implement the interface.

Application programs using the service use a stub library to communicate with the service. The stub library is linked with the application and uses the service layer to find the service process. Once a service has been located, the service stub caches the process ID of the service process and uses this ID for all future requests.

Programs call services through the service interface stub and need not be aware of the fact that a particular function is implemented as a service. The first time the service is called, the service interface stub performs a service lookup in the service layer. If the specified service exists in the system, the lookup returns a pointer to the service interface.

The version number in the service interface is checked with the version of the interface stub. In addition to the version number, the service interface contains pointers to the implementation of all service functions. The function implementations are contained in the service process. If the version number of the service stub match the number in the service interface, the interface stub calls the implementation of the requested function.

2.12 Libraries

The Contiki kernel only provides the most basic CPU multiplexing and event handling features. The rest of the system is implemented as system libraries that are optionally linked with programs. Programs can be linked with libraries in three different ways. First, programs can be statically linked with libraries that are part of the core. Second, programs can be statically linked with libraries that are part of the loadable program. Third, programs can call services implementing a specific library. Libraries that are implemented as services can be dynamically replaced at runtime.

Typically, run-time libraries such as often-used parts of the language run-time libraries are best placed in the Contiki core. Rarely used or application specific libraries, however, are more appropriately linked with loadable programs.

Libraries that are part of the core are always present in the system and do not have to be included in loadable program binaries.

2.13 Communication Support

Communication is considered as a fundamental concept in sensor networks.

In Contiki, communication is implemented as a service in order to enable run-time replacement. Implementing communication as a service also provides for multiple communication stacks to be loaded simultaneously. In experimental research, this can be used to evaluate and compare different communication protocols. Furthermore, the communication stack may be split into different services as shown in Figure 2.11. This enables run-time replacement of individual parts of the communication stack.

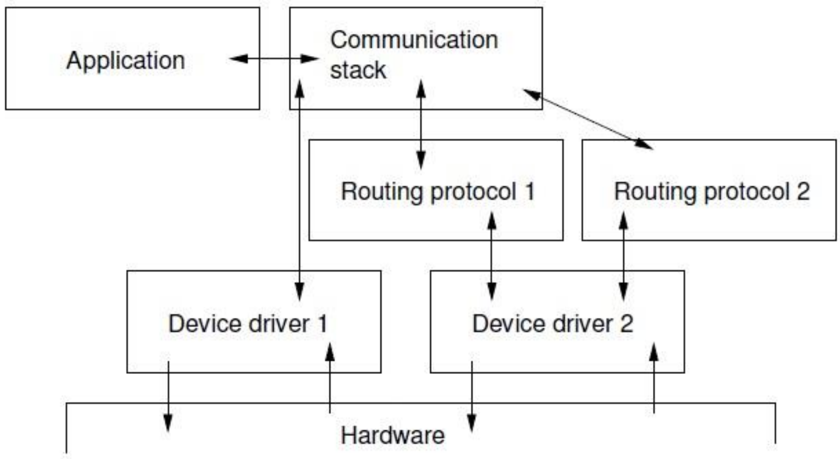


Figure 2.11 - Loosely coupled communication stack [26]

Communication services use the service mechanism to call each other and synchronous events to communicate with application programs. Because synchronous event handlers are required to be run to completion, it is possible to use a single buffer for all communication processing. With this approach, no data copying has to be performed. A device driver reads an incoming packet into the communication buffer and then calls the upper layer communication service using the service mechanisms. The communication stack processes the headers of the packet and posts a synchronous event to the application program for which the packet was destined. The application program acts on the packet contents and optionally puts a reply in the buffer before it returns control to the communication stack. The communication stack prepends its headers to the outgoing packet and returns control to the device driver so that the packet can be transmitted.

2.14 Contiki OS Origins

Contiki was created by Adam Dunkels in 2002 and has been further developed by a world-wide team of developers from Atmel, Cisco, Enea, ETH Zurich, Redwire, RWTH Aachen University, Oxford University, SAP, Sensinode, Swedish Institute of Computer Science, ST Microelectronics, Zolertia, and many others. The name Contiki comes from Thor Heyerdahl's famous Kon-Tiki raft [27].

2.15 Other OSs

There are other Operating Systems that can be used with WSN i.e. Tiny OS, MANTIS, Nano-RK, Lite OS and many others. Each OS has its strong and weakness points Table 2.1

Table 2.1 - Some of the miscellaneous features of the OSs

OS/Feature	Communication Security	File System Support	Simulation Support	Programming Language	Shell
Tiny OS	TinySec	Single level file system	TOSSIM	NesC	Not available
Contiki	ContikiSec	Coffee file system	Cooja	C	Unix-like shell runs on sensor mote
MANTIS	Not available	Not available	Through AVRORA	C	Unix-like shell runs on sensor mote
Nano-RK	Not available	Not available	Not available	C	Not available
Lite OS	Not available	Lite FS	Through AVRORA	Lite C++	Shell that runs on base station

3 Embedded Systems

3.1 Introduction

Computing systems are everywhere. Probably it is not surprising that millions of computing systems that are built annually destined for desktop computers (Personal Computers), workstations, mainframes and servers. What could be surprising is that for a very different purpose millions of computing systems are built every year: they could be embedded within larger electronic devices, carrying out a particular function repeatedly, or completely unrecognized by the user of the device. Creating an accurate definition of such embedded computing systems, or simply the embedded systems, is not an easy task. We may try the following definition: An embedded system is almost any computing system that is not a desktop, laptop, or mainframe computer. That definition isn't perfect enough, but it might be closer as we'll get. We can get better understanding of such systems by testing common examples and characteristics. Such examination will reveal the major challenges that face the designers of such systems [28].

The term embedded systems usually refer to programmed digital chips with high performance in a system to replace computers. And it is dedicated to carry out special functions, while computers are consider a generic user interface.

3.2 What are Embedded Systems?

Embedded systems are fully digital systems where hardware meet the software in a single environment to achieve required integration, where software leads the hardware for the implementation of the required tasks.

Most of the devices that we use in our daily life are built based on embedded systems as air conditioners, microwave, recorders, televisions, mobile phones, cars and traffic lights and, of course, it is noticeable that the embedded systems outnumber computers that only take 5% of the processors market [29].

When designing embedded systems the followings are considered:

- Usually embedded systems are time controlled which means that they must interact with events the moment they occur, and this is what is called real-time processing.

- The errors in programming embedded systems have more serious consequences than those in personal computers.
- Usually built with certain controls related to energy consumption and system feeding.
- Embedded systems must work in harsh weather, climate and environment such as temperature, humidity and dust.
- Embedded systems have fewer system resources and peripheral devices because the embedded system is designed to accomplish a specific goal while the computer is designed for diverse uses in all areas.
- Use special tools and methods to increase the reliability and yield of the system.

Figure 3.1 - An Example of Embedded Controller [29]

Embedded system may consist of:

1. Logical circuit based on an integrated chip PLD (Programmable Logic Device): The use of PLD is much better than using logical circuits because it can represent a required design using one integrated circuit instead of using several separate circuits. PLD segments consists of a large array of programmable logic gates where the designer is the one who determines the function of these gates. PLD segments Feature high performance and low consumption of energy, but it usually produces a huge waste of energy due to partial use of this segments (i.e. it is possible to use 20 logic gate to represent the desired design, but usually PLD contain more than 150 logic gates, this what leads to a rise in energy consumption and a huge waste of the PLD potential).
2. Integrated circuit custom-built to perform a specific function ASIC (Application Specific IC):

The ASIC circuits cover the defects of PLD circuits because it uses only the logic gates required for the design, which reduces energy consumption and prevents wastage happening when using PLD segments.

3. Microcontroller or Microprocessor:

   Microcontrollers and microprocessors consume less energy than PLD segments, but in the other hand it has lower speed and programmed using low-level program-ing languages (e.g. Assembly). The microcontrollers use is considered an effective solution because of the great potentials offered compared with their price. Microcontrollers also have low supplying voltages (5V, 3V, 1.5V), and the voltages continued decreasing until it reached lately to 0.8V in some microcontrollers' core.

4. Digital Signal Processor DSP:

   A special kind of Processors with a hardware intended for dealing with complex calculations and numerical arrays. It is faster than microcontrollers. The important purpose of DSP is reading the data quickly on the inputs and doing the complex algorithms and output the results. It is used in audio and video processing in communications, radar, medical devices, mobile phones and power control devices. Some manufacturers are heading to incorporate some of the DSP features in their controllers and thus obtain a microcontroller and digital signal processor in a single chip DSP.

5. Field Programmable Gate Array FPGA:

   The FPGA chip is a solid-state digital logical programmable chip, it means that it could be programmed to do any logical function. Mainly it is programmed using the VHDL language. The main advantages of the FPGA in addition to the low cost is that it provides great flexibility in the design and the ability to be reprogrammed in real time.

3.3 Embedded Systems Manufacturing Stages

1. Identify functions and tasks required from the system with a clear and accurate description for each job far from obscurity.
2. Create the hardware and software models.
3. Test the design using a simulation environments to ensure the design safety and to make sure that the system functions well.

4. Build the system with the final finishing, then a practical test before publishing it in the market.

3.4 Design Considerations

Noise immunity and power consumption and speed are considered very important in the logical design. And also must consider the place and the environment in which the system will work, for example, if the system will work within a factory that contains a large electrical motors or near radio or TV stations, a high immunity against noise integrated circuits must be used.

Immunity means that the circuit is not sensitive to the noise voltages being generated and transmitted undesirably to the electronic system, and if that was not enough a Shielded wires must be used to transmit logical signals [29].

Energy consumption is not considered as a problem in the equipment that feeds via the network using a regulator, but the devices that work on battery must be designed to run using circuits with a few consumption of energy.

Work speed is considered one of the criteria that narrowed the range to choose the type of integrated circuit or processor, therefore studying the specifications given by the manufacturers in order to make the best selection for the logical family that suites the requirements.

An output load factor of a logical gate or logical tool represents number of entries that can be connected with the output together, and must consider this factor carefully to make sure that the output of the circuit will not be forced to drive or operate more than potential logical inputs.

A time-delay in a logical item (gate, for example), which is called propagation delay, represents the time between the moment of applying logical input and the appearance of the corresponding logical output. This delay can sometimes cause problems such as cases of unwanted precarious status or cause the so-called glitches.

3.5 Characteristics of Embedded Systems

The common characteristics that distinguish embedded system from other computing systems are [28]:

- **Single Functioned:** The embedded systems perform specific pre-defined task program repeatedly. For example, a digital camera is always a digital camera. In contrast, a desktop system performs a variety of programs, like spreadsheets, word processors, image processors video games and so on.
- **Tightly Constrained**: A design metrics is a measure of an implementation's features, such as size, performance, power and cost. All computer systems have constraints on design metrics; however, the embedded systems have tight constraints on design metrics. The embedded systems must be sized to fit on a single chip, must perform fast enough to process data in real time, must consume minimum power to extend battery life or prevent the necessity of a cooling fan, and must have less cost.

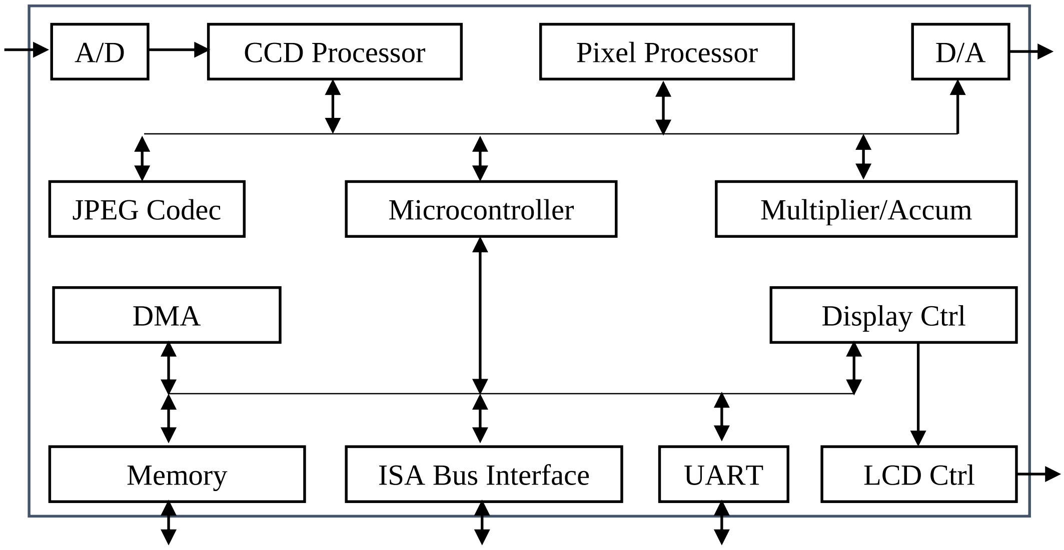


Figure 3.2 - An Embedded System Example (Digital Camera) [28]

- **Reactive and Real Time**: Many embedded systems must react to changes in the system's environment time to time. They must compute certain results in real time without delay. For example, the auto focus camcorders continuously monitors screen and adjust their lenses such that screen is viewed with maximum clarity within no time. In contrast, desktop systems typically focuses on computations with relatively infrequent reactions to input devices.

For example, the digital camera system [28] as shown in Figure 3.2. The A/D and D/A circuits convert the analog images to digital and the digital to analog. The CCD preprocessor which is a charge-coupled device preprocessor. The JPEG codec compresses and also decompresses the image using the JPEG compression standard, to enable compacted storage in the usually limited memory of the camera. The Pixel coprocessor helps rapidly display the images. The Memory controller controls the access to the memory chip which is found in the camera, while the DMA controller enables direct memory access without the need of using the microcontroller. The UART enables the communication with the serial port of a computer for uploading video frames, while the ISA bus interface enables much faster connection with the ISA bus of the computer. The Display ctrl and LCD ctrl circuits control displaying the images on the liquid-crystal display device of the camera. A Multiplier/Accum circuit helps with particular digital signal processing. The heart that beats inside the system is a microcontroller, which is a processor that controls all the activities of all the other circuits.

We can consider each device as a processor designed for a predefined task, while the microcontroller is a more general processor designed for general tasks.

This example illustrates some of the embedded system characteristics described previously. First, it keep performing a single function repeatedly. The system is always a digital camera, it captures, compresses and then stores the frames, decompresses and displays the frames, and uploads the frames. Second, it is and must be tightly constrained. The system should be low cost because consumers should be able to afford such camera. It should be small so it could fits within a camera with a standard size. It must be fast to be able to process numerous images in milliseconds. It has to consume little power so that the battery will last for longer time. However, this particular system does not possess a high level characteristic of being reactive and real-time, because it only needs to respond to the buttons that the user presses, which is considered quite slow for a professional photographer respecting the processor speeds.

3.6 Embedded Systems' Applications

Software tools are used to develop software for designing an embedded system. Debugging tools, such as a stethoscope, trace scope, and sophisticated tools such as an integrated development environment and prototype development tools, are needed for the integrated development of system software and hardware.

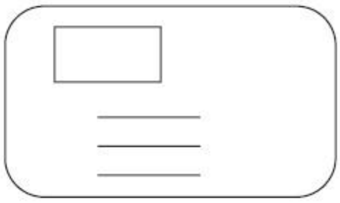
Embedded systems have very diversified applications. A few select application areas of embedded systems are Telecom, Smart Cards, Missiles and Satellites, Computer Networking, Digital Consumer Electronics, and Automotive [30]. Figure 3.3 shows the applications of embedded systems in these areas.

Telecom

- Mobile Computing
- Wireless
- Networking

(a)

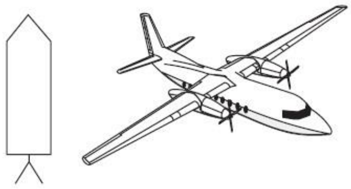
Smart Cards



- Banking
- Telephone
- Security

(b)

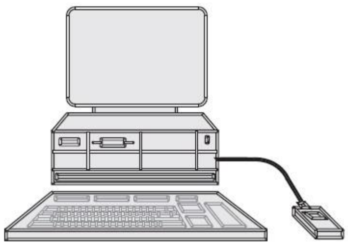
Missiles and Satellite



- Defense
- Aerospace
- Communication

(c)

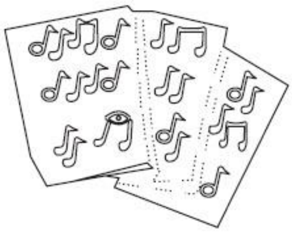
Computer Networking Systems and Peripherals



- Networking Systems
- Image Processing
- Printers
- Networks Cards
- Monitors and Displays

(d)

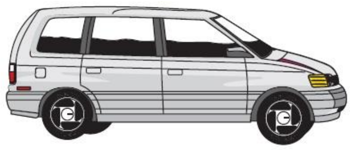
Digital Consumer Electronics



- DVDs
- Set top Boxes
- High Definition TVs
- Digital Cameras

(e)

Automotive



- Motor Control System
- Cruise Control
- Engine/Body Safety
- Robotics in Assembly Line
- Car Entertainment
- Car Multimedia
- Mobile and E-Com Access

(f)

Figure 3.3 - Applications of Embedded Systems in several areas [30]

A few examples of embedded system applications are as follows [30]:

- Point of sales terminals: automatic chocolate vending machine.
- Stepper motor controllers for a robotics system.
- Washing or cooking systems.
- Multitasking toys.
- Microcontroller-based single or multi display digital panel meter for voltage, current, resistance and frequency.
- Keyboard controller.
- SD, MMI and network access cards.
- CD drive or hard disk drive controller.
- The peripheral controllers of a computer, a CRT display controller, a keyboard controller, a DRAM controller, a DMA controller, a printer controller, a laser printer controller, a LAN controller, a disk drive controller
- Fax or photocopy or printer or scanner machine.
- Remote (controller) of TV.
- Telephone with memory, display and other sophisticated features.
- Motor controls systems- for example, an accurate control of speed and position of the d.c. motor, robot and CNC machine; automotive applications such as closed loop engine control, dynamic ride control, and an antilock braking system monitor.
- Electronic data acquisition and supervisory control system.
- Electronic instruments, such as an industrial process controller.
- Electronic smart weight display system and an industrial moisture recorder cum controller.
- Digital storage system for a signal wave form or for electric or water meter reading system.
- Spectrum analyzer.
- Biomedical systems such as an ECG LCD display cum recorder, a blood-cell recorder cum analyzer, and a patient monitor system.
- Computer networking systems, for example, a router, a front-end processor in a server, a switch, a bridge, a hub and a gateway.

- For Internet appliances, there are numerous application system (i) An intelligent operation, administration and maintenance router (IOAMR) in a distributed network and (ii) Mail client card to store e-mail and personal addresses and to smartly connect to a modem or server.
- Entertainment system such as a video game and a music system.
- Banking systems, for example, bank ATM and credit card transactions.
- Signal tracking systems for example, an automatic signal tracker and a target tracker.
- Communication systems such as a mobile communication SIM card, a numeric pager, a cellular phone, a cable TV terminal and a FAX transceiver with or without a graphic accelerator.
- Image filtering, image processing, pattern recognizer, speech processing and a video processing.
- Video games.
- A system that connects a pocket PC to the automobile driver mobile phone and a wireless receiver. The system then connects to a remote server for Internet or e-mail or to a remote computer at an ASP (application service provider).
- A personal information manager using frame buffers in handheld devices
- Thin client [A thin client provides disk-less nodes with remote boot capability]. Application or thin-client accesses to a data center from a number of nodes; in an Internet laboratory accesses to the Internet leased line through a remote server.

3.7 Raspberry Pi

The name Raspberry Pi at the first glance it seems as a delicious pie of berries, but in fact this name became an idle of the most important innovations that started in 2012 as the smallest computer with a price of 25$, which made it the most common word between experts and matures [31].

Raspberry Pi panel represents integrated computer made of a single electronic chip containing traditional computer components such as a single-core CPU with 700 MHz, dual-core GPU with 250 MHz which is capable of running HD movies and 3D games with RAM up to 512 MB, in addition to digital control inputs/outputs which can control various electronic and electrical parts such as Microcontrollers, all these possibilities only on one small chip which known as a "system on a chip" SoC, this small computer runs with an open source OS, such as Linux.

3.8 Raspberry Pi Component

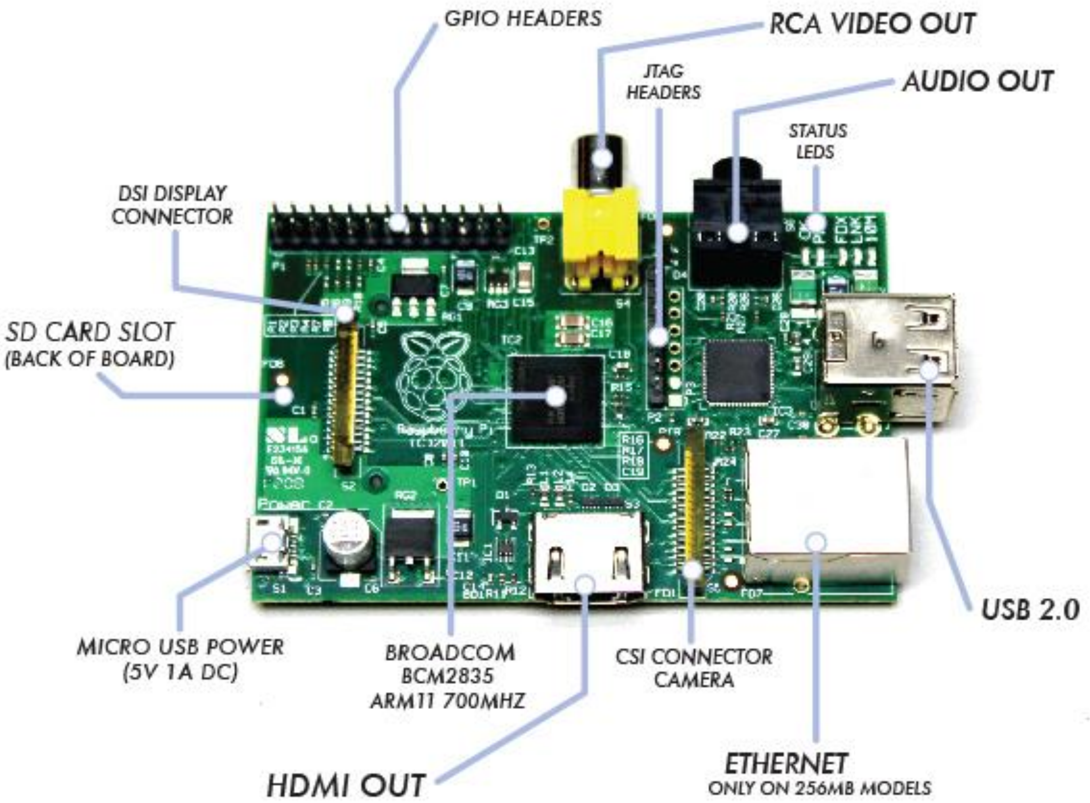


Figure 3.4 - Raspberry Pi component [31]

Table 3.1 - Raspberry Pi Detailed Components

Part Name	Discription
Broadcom BCM2835 SoC	This chip is considered as a full computer, it contains a CPU, RAM and GPU. In a very small square of 1 cm dimension.
HDMI Output	An output to be connected to high definition TV screen, this output handle both video and audio.
RCA video output	It's the same as HDMI, but this one is dedicated to old TV screen without audio.
Audio jack	Audio output with 3.5 mm which is the same standard audio output in different devices.
Ethernet Port	The same RJ45 port that connects Raspberry Pi to computer networks an internet.
2 USB ports	The same USB ports in any regular computer to connect a mouse, keyboard or any other media devices such as flash disks or wireless network card, and it could be connected with any device that connects to a regular computer.
Micro USB port	Which is the power port that provide the Raspberry Pi with power using any 5v mobile phone USB charger.
SD Card slot	A slot to connect the memory card that contains the operating system and the Raspberry Pi files.
GPIO pins	The control set that allows to control different electronic components, which makes Raspberry Pi able to act as a microcontroller.
CSI Camera input	To connect to high definition cameras that made for Raspberry Pi.
DSI Display input	Used to connect to touch screens like the ones used with smart phones and tablets.

3.9 Raspberry Pi OS

Raspberry Pi board works with Linux OS (GNU-Linux), which is a free open source operating system with the freedom to copy, share and even editing it from zero, and that because it is an open source OS which make the source code of the operating system available for everyone and that makes it possible for anyone to derive a new version of their own and ascribe it to themselves, which known as Linux Distribution [31].

3.10 Intel® Galileo and Raspberry Pi Comparison

Table 3.2 - Comparison of Processors & on-Board Features [32]

	Galileo	Raspberry Pi (Model B)
Board Dimensions	10cm x 7cm (slight overlap for power jack)	85.60mm x 56mm x 21mm (with a little overlap for SD card)
Processor	Intel® Quark X1000 – single core	Broadcom BCM2835 – single core
Description of Processor	Quark, described by Intel at IDF2013, is very low power consumption, small form factor, and low cost; ideal for "wearables," and the Internet of Things†.	Per ARM datasheet: For devices such as smart phones, digital TVs, & eReaders, delivering media & browser performance, a secure computing environment.
Architecture	Intel ® Pentium® Class	ARM® ARM1176™
Speed	400MHz	700MHz
Width, Instruction Set	32-bit	32-bit
Real Time Clock	Yes, needs a 3.3v coin cell	No
Cache	16 KB L1 cache	32KB L1 cache & 128KB L2 cache; shared with CPU & GPU
RAM	512KB on-chip SRAM, dedicated for sketch storage & 256MB DRAM, dedicated for sketch storage	512MB SDRAM (shared with GPU)
FLASH Memory	8MB NOR Flash (Legacy SPI), for FW bootloader & sketch storage	No permanent on-board Flash memory
EEPROM	11KB	No
GPU	No	Broadcom Dual-core VideoCore IV® Multimedia co-processor
External Storage	Micro-SD Card (up to 32GB), & support for an external USB2.0 drive	SD-card, & support for an external USB2.0 drive
Video Support	No	HDMI – 1080p RCA (analog), without audio DSI* – for touchscreens
Audio Support	No	HDMI & 3.5mm stereo audio-out jack

Status Indication	LED – Board Power	LEDs for – board power, SD card access, LAN connected, LAN activity, 100Mbps connected
JTAG	10-pin, Mini-JTAG header, to be used with an in-circuit debugger such as 909-ARM-USB-OCD with the 909-ARM-JTAG-20-10 converter (available at www.mouser.com), & with OpenOCD/GDB**for Quark, & GUI.	Yes, headers P2 & P3. (However, there is no current support to debug the Broadcom & SMSC USB/LAN chip.)**
Compatibility	Arduino Shields that fit the Arduino Uno R3 3.3V / 5V shields	Arduino board connects via USB. 3rd party boards enable support for Arduino shields with Pi.
*DSI – Display Serial Interface **OpenOCD support for Quark X1000 may be available, or not be full featured as of this writing.		

3.10.1 Excel in Ease

Galileo boards characterized of being easy to be programmed, because it doesn't need any previous experience in operating systems, it only needs to have the knowledge of Arduino C language.

3.10.2 Superiority in Price

The Raspberry Pi costs about 25$ while the Galileo cost more than the double. While with Raspberry Pi a better CPU with 700 MHz and better Cache memory compared with the 400 MHz CPU with Galileo, without mentioning the HDMI port for high definition video output with Raspberry Pi. In fact one of the factors that makes Raspberry Pi more known and sold is being cheap.

3.10.3 Superiority in Strength

As mentioned before Raspberry Pi has better CPU with 300 MHz more than Galileo's CPU. Both Raspberry Pi and Galileo has 14 digital control pins, but Galileo has extra 8 analog pins while Raspberry Pi doesn't have and that gives Galileo an extra point, but in spite of that an extra analog pins can be added to Raspberry Pi using the Analog to Digital Converter.

Raspberry Pi has a RCA Video output so it allows it to be connected to old TVs which not existed in Galileo.

HDMI port to connect to high definition screen with audio output which also not available in Galileo.

An audio support for Raspberry Pi with the 3.5 mm audio output and the HDMI port, while no audio support available with Galileo.

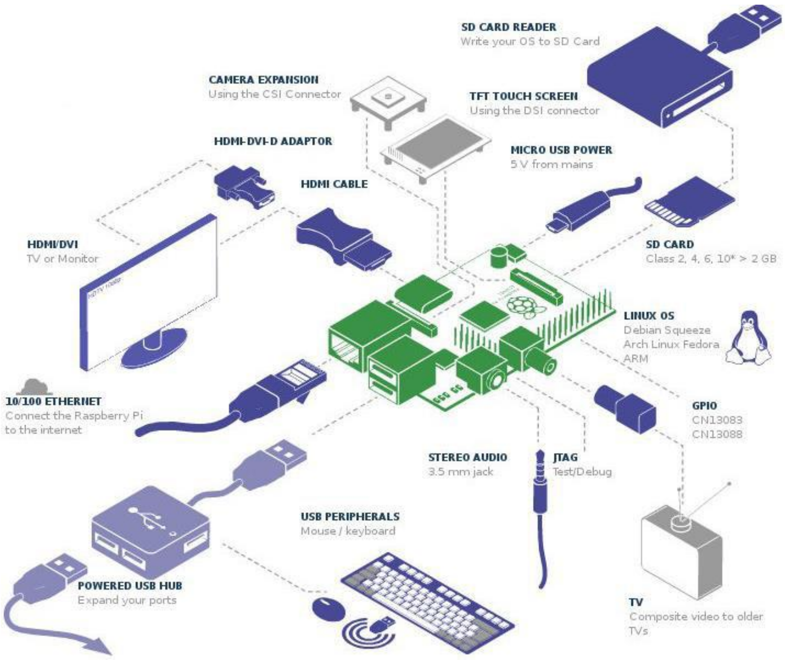


Figure 3.5 - Raspberry Pi possible terminals [31]

3.11 Starting up Raspberry Pi

Raspberry Pi board needs some extra component to start working, some of them is mandatory and the others are not.

3.11.1 SD Memory Card (mandatory)

It is considered the most important extra component, because it is responsible of storing the operating system and all the files that will be used with Raspberry Pi. The memory card should be at least 4 GB up to 32 GB, micro SD cards can be used with an extra adapter to be installed on the board.

Figure 3.6 - SD Cards types

3.11.2 USB Charger Power Source (mandatory)

Raspberry Pi can be powered with any Micro USB cell phone charger, where the charger should be able to provide a tension of 5V and a current of at least 700 mA.

Using a charger that provide less than 700 mA the USB ports in Raspberry Pi might not be able to provide power to the connected devices, for example, some USB devices need 200 mA to function and the Raspberry Pi board needs at least 500 mA that means it all needs at least 700 mA to function well.

Using a charger that provide 1000 mA current would be great.

3.11.3 Mouse and Keyboard (mandatory)

It needs at least keyboard to work with the user interface.

3.11.4 TV or a Screen (mandatory)

As any computer it needs a monitor to interact with it (at least for the first use), so it can be used an old or new TV or using a screen with alternative HDMI port.

3.11.5 HDMI Cable (optional)

It can be used to connect to a high definition TV or screen.

3.11.6 RCA Video Cable (optional)

To connect the Raspberry Pi to an old TV.

4 System Overview

4.1 Introduction

The main idea of this system is to sense temperature and humidity and other factors during different periods of time and store data collected into database for a later use, with the ability to access these data remotely.

The system contains several CC2530 System on Chip (SoC) sensor nodes connected wirelessly to a coordinator which is connected to the central processing and database unit which is Raspberry Pi, it will store the collected data into a database for later use.

4.2 System Architecture

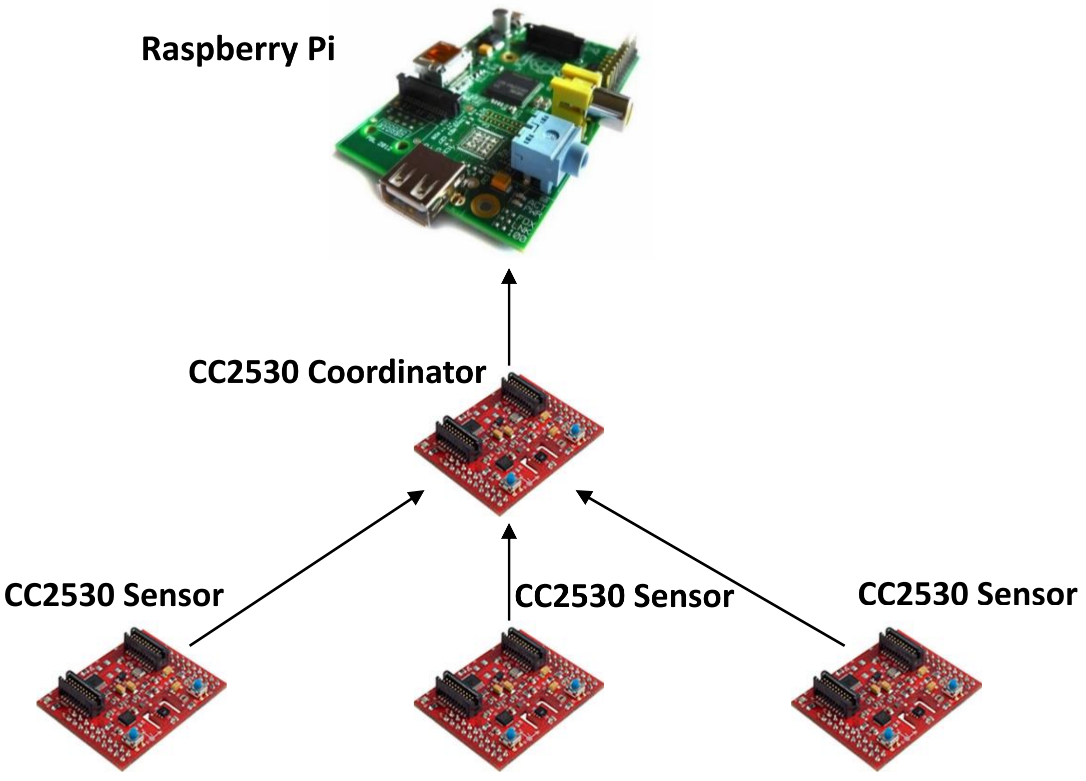


Figure 4.1 - System Architecture

Figure 4.1 presents the system architecture, the three lower CC2530 will collect data and transfer it to the CC2530 Coordinator which will gather the data from all the sensors and transfer it to Raspberry Pi.

4.3 System Activity

The sensor nodes and coordinator have CC2530 boards, these boards support LPRF (Low Power RF), and they also support the ZigBee, ZigBee PRO, and ZigBee RF4CE which are IEEE 802.15.4 standard. They also have a 101dB link budget and excellent receiver sensitivity. In addition to supporting general low power wireless communication by way of excellent RF performance, selectivity, and an industry-standard enhanced 8051 MCU core, the CC2530 can be coupled with one of TI's standard's compliant or proprietary networking protocol stacks (RemoTI, Z-Stack, or SimpliciTI) [33] to ease development.

4.4 CC2530 LPRF Protocols

Low Power RF protocols depend on the implementing topology of the sensors which could be point to point and Star Network, or Mesh network.

Figure 4.2 shows each protocol belongs to the mentioned topology.

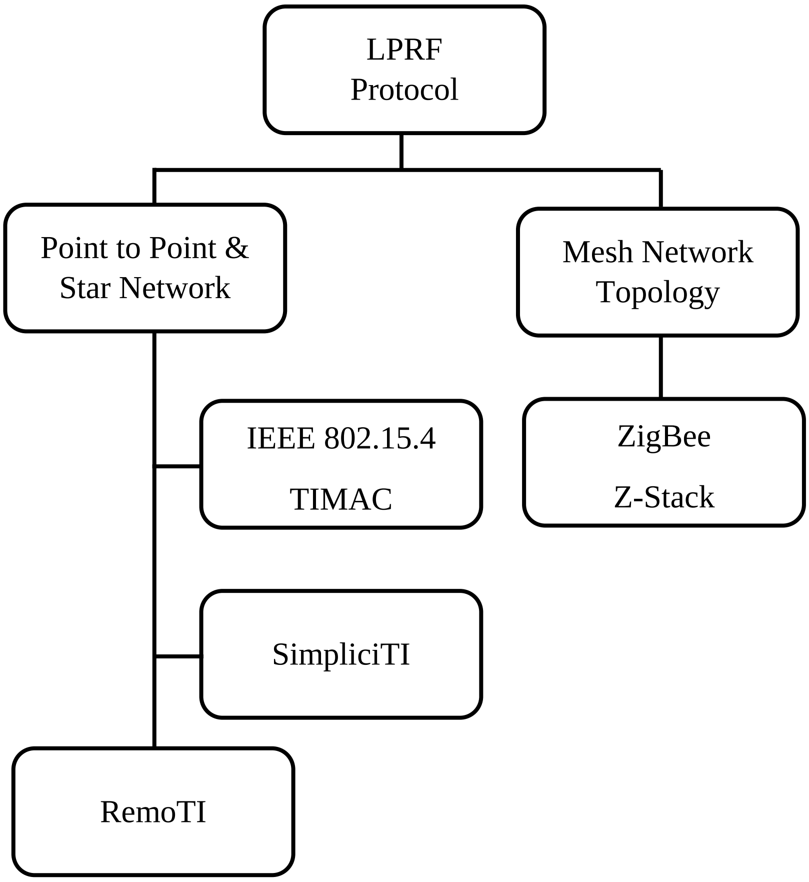


Figure 4.2 - CC2530 LPRF Protocols [33]

4.5 LPRF Protocol Software

- Z-Stack: (ZigBee Protocol stack from TI)
  - Mesh networking.
  - It has the Golden Unit certification for ZigBee-2006, ZigBee-2007 and ZigBee PRO.
- TIMAC
  - A wireless protocol which considered a standard for battery or main powered nodes.
  - Suitable for law data rate required applications.
  - Support IEEE 802.15.4 – 2003/2006.
- SimpliciTI
  - Simple protocol for low power RF network and specifically for the small RF networks.
  - Low data rate and data cycle which make it suitable for battery operated devices with long battery life.
- RomoTI (Remote Control)
  - RF4CE has been built based on the TIMAC, which is based on the IEEE 802.15.4 protocol stack which runs millions of devices worldwide.

4.5.1 ZigBee Z-Stack

The key benefits are [33]:

- Self-healing (for Mesh networks).
- Low cost for nodes.
- Ease of deployment (plus low installation cost).
- Support for large networks (containing hundreds of nodes).
- Suitable for monitoring and control applications.
- Standardized protocol.

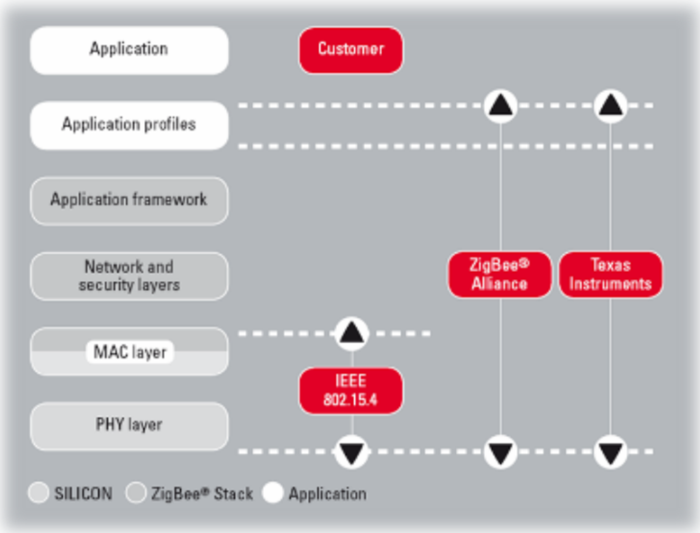


Figure 4.3 - ZigBee in network layers [33]

4.5.2 SimpliciTI

The key benefits are:

- Low Power: it is a TI proprietary protocol for the low-power RF networks.
- Low Cost: it uses < 8K FLASH, 1K RAM memory depending on configuration.
- Flexible: simple star with extender and/or with p2p communication.
- Simple: Utilizes a very basic core API.
- Save Power: Supports sleeping devices.

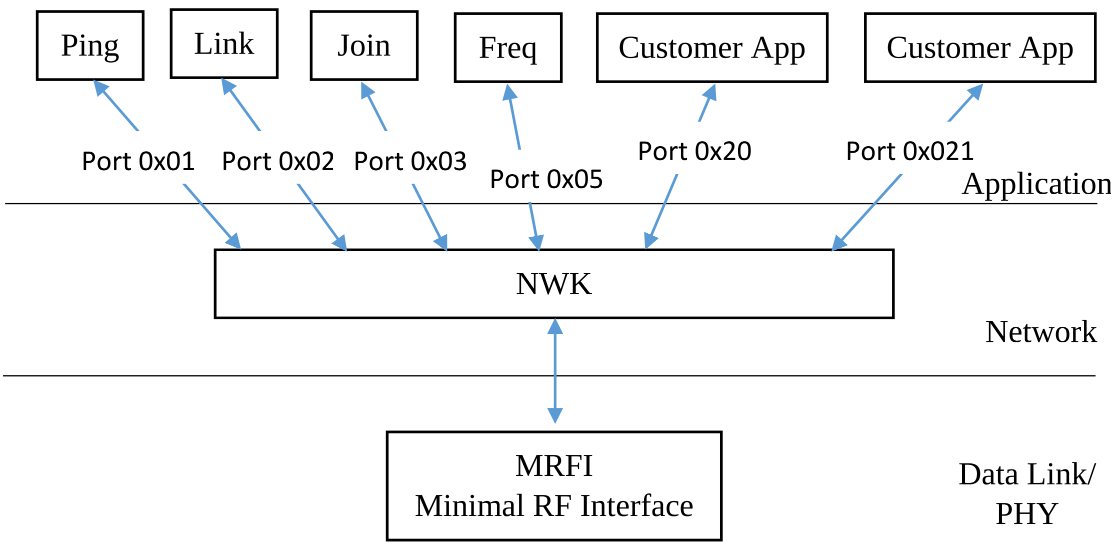


Figure 4.4 - SimpliTI in network layers [33]

4.5.3 RemoTI (RF4CE)

The RemoTI protocol is based on IEEE 802.15.4 and include a thin network layer and a command interface.

The key benefits are:

- Graceful frequency for multi-channel operation to avoid any interference.
- A secure transactions mechanism.
- A power saving mechanism for the power efficient implementations.
- Simple and intuitive pairing mechanism.

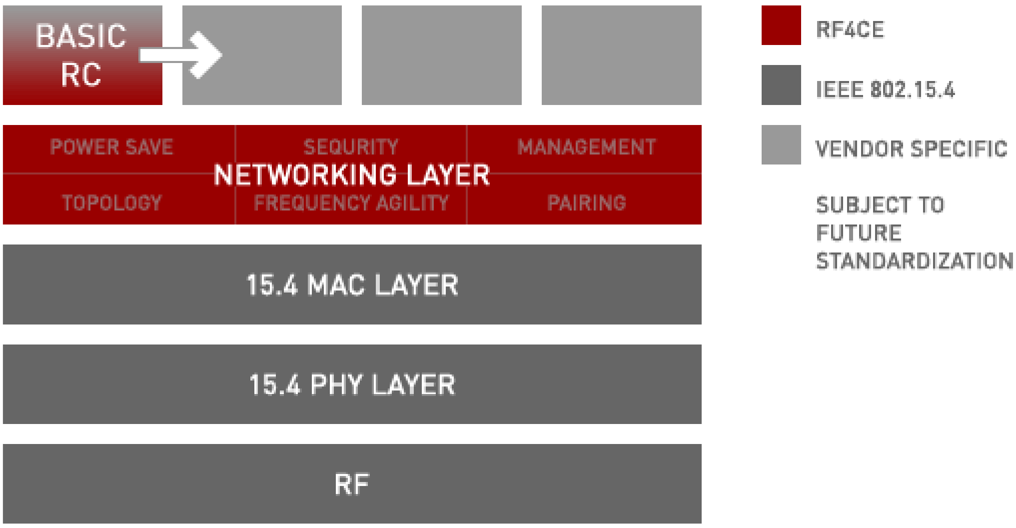


Figure 4.5 - RemoTI (RF4CE) in network layers [33]

4.6 Wireless Frequency Regulations Worldwide

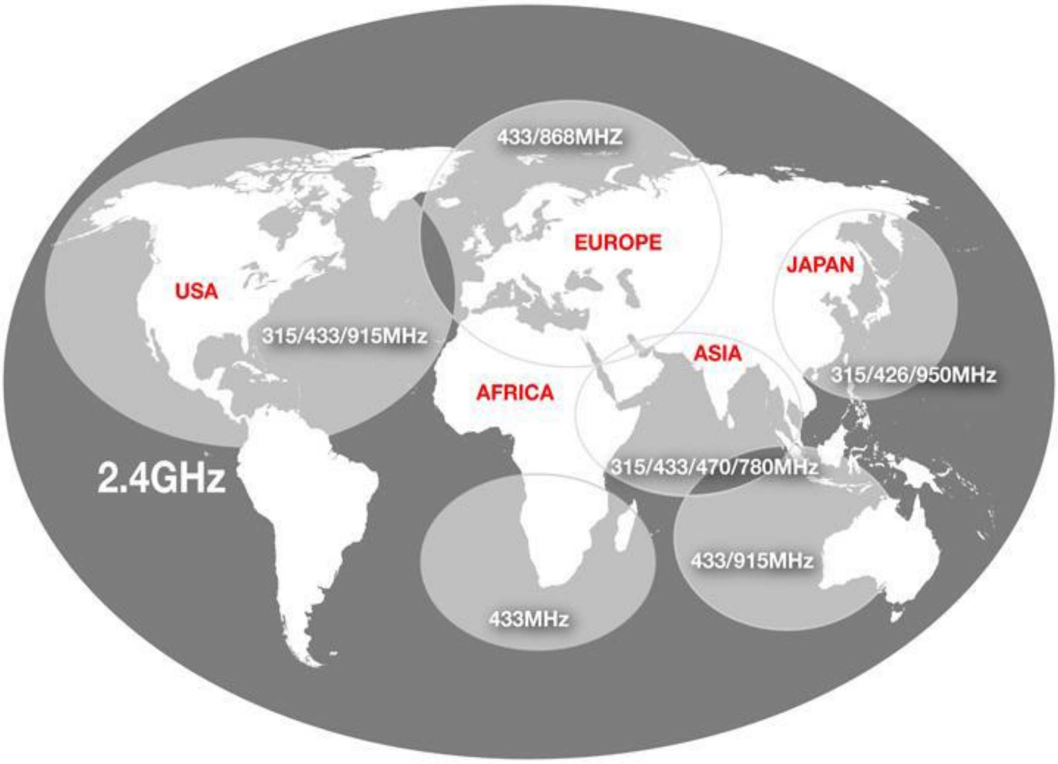


Figure 4.6 - Worldwide wireless frequencies regulations [33]

The 2400 - 2483.5 MHz band is available for license-free operation in most countries [33].

2.4 GHz Prospects:

- Same solution for all markets without SW/HW alterations.
- Large available bandwidth, allows many high data rates and separate channels.
- 100% possible duty cycle.
- More compact antenna solution than below 1 GHz.

2.4 GHz Constrains:

- Shorter range than a sub 1 GHz solution (with the same current consumption).
- Many possible interferers are present in the band.

The ISM bands (Industrial, Scientific and Industrial) under 1 GHz are not world-wide. Limitations vary a lot from region to another and getting a full overview is not an easy task.

Sub 1GHz Prospects:

- Better range than 2.4 GHz with the same output power and current consumption.
- Lower frequencies have better penetration through concrete and steel (buildings and office environments) compared to 2.4 GHz.

Sub 1GHz Constrains:

- No worldwide solution possible. Since different bands are used in different regions a custom solution has to be designed for each area.
- Duty cycle restrictions in some regions.

4.7 Regulation for Unlicensed ISM & SRD Bands

USA/Canada:

- 260 - 470 MHz (FCC Part 15.231; 15.205)
- 902 - 928 MHz (FCC Part 15.247; 15.249)
- 2400 - 2483.5 MHz (FCC Part 15.247; 15.249)

Europe:

- 433.050 - 434.790 MHz (ETSI EN 300 220)
- 863.0 - 870.0 MHz (ETSI EN 300 220)
- 2400 - 2483.5 MHz (ETSI EN 300 440 or ETSI EN 300 328)

Japan:

- 315 MHz (Ultra low power applications)
- 426 - 430, 449, 469 MHz (ARIB STD-T67)
- 2400 - 2483.5 MHz (ARIB STD-T66)
- 2471 - 2497 MHz (ARIB RCR STD-33)

4.8 RF Operation

The Basic RF layer offers a simple protocol for transmission and reception using a two-way RF link. The Basic RF protocol offers service for packet transmission and reception. It also offers secure communication by use of CCM-64 authentication and encryption/decryption of packets. The security features of Basic RF can be compiled in by defining the compile switch SECURITY_CCM [34] in the project file. The compile time inclusion of security features is done to save code space for the applications where security features are not needed.

The protocol uses IEEE 802.15.4 MAC compliant data and acknowledgment packets. However it does not offer a full MAC layer, only a simple data link layer for communication between two nodes.

Basic RF contains only a small subset of the 802.15.4 standard:

- Association, scanning or beacons are not implemented.
- No defined coordinator/device roles (peer-to-peer, all nodes are equal).
- No packet retransmission. This must be taken care of by the layer above Basic RF.

Figure 4.7 shows the Basic RF and the HAL (Hardware Abstraction Layer) operation during initialization, transmission and reception of packets.

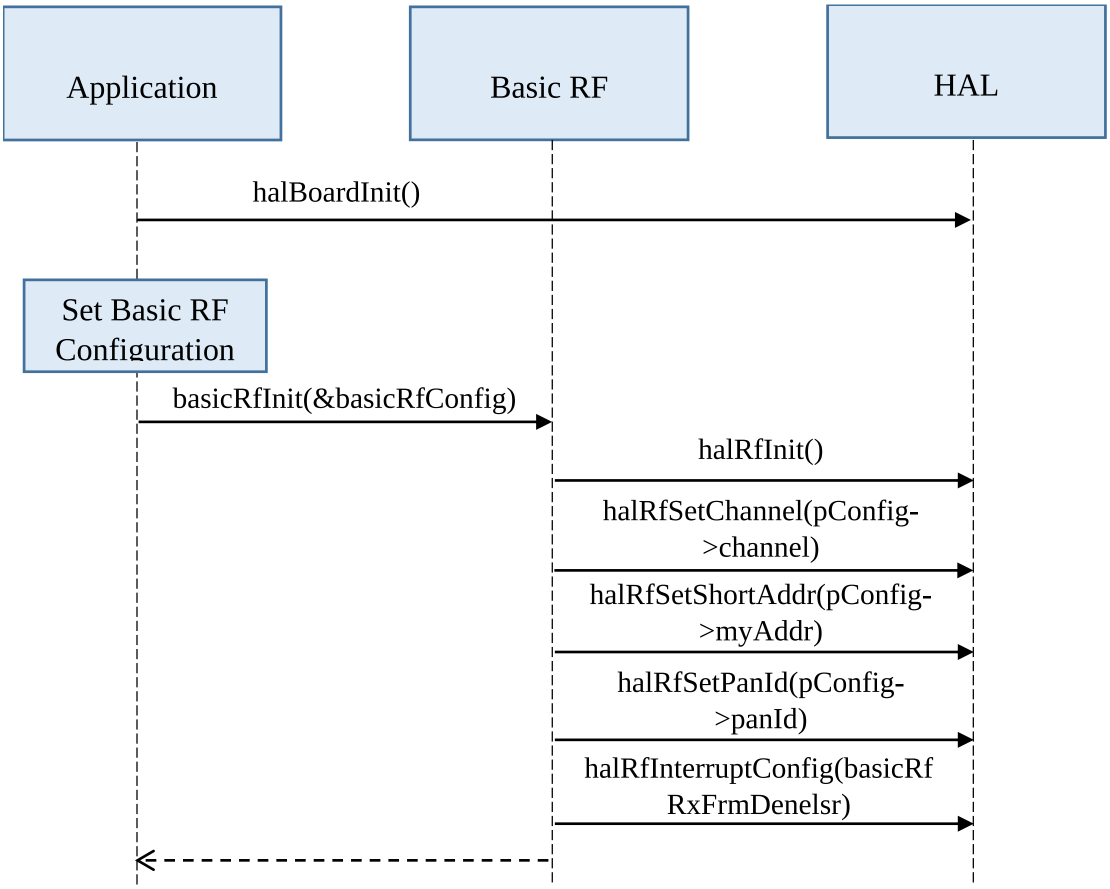


Figure 4.7 - Initialization, packet transmission and reception [34]

5 Implementation

5.1 Devices Description

The system is built using a Raspberry Pi minicomputer as shown in Figure 5.1, and as shown in Figure 5.2, it is connected to CC2530 coordinator that will receive data collected from the CC2530 sensor nodes.

Figure 5.1 - Raspberry Pi

Figure 5.2 - CC2530 Coordinator

A serial connection is made between Raspberry Pi and the CC2530 coordinator using a USB cable as shown in Figure 5.3.

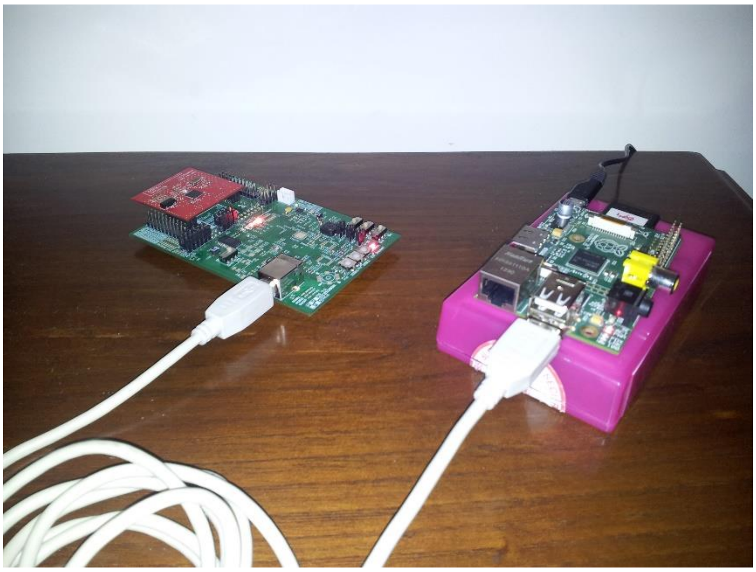


Figure 5.3 - USB connection between coordinator and Raspberry Pi

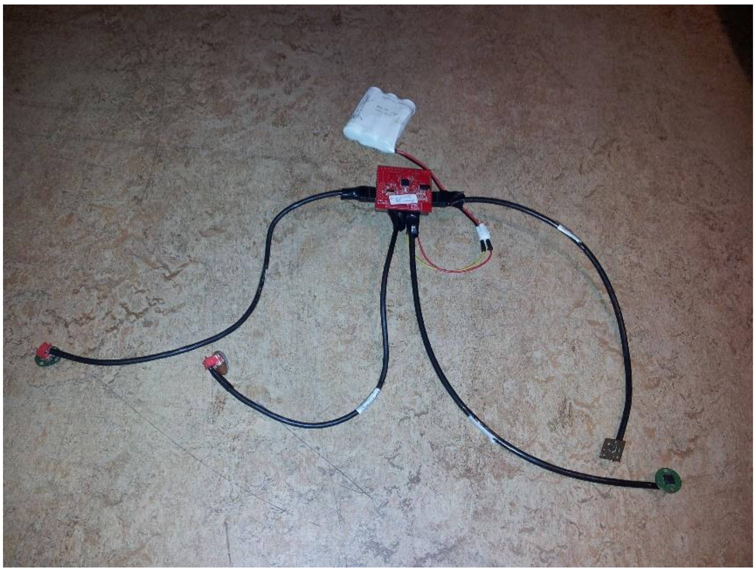
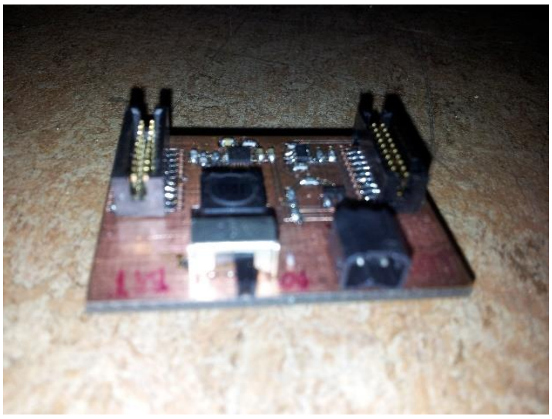


Figure 5.4 - CC2530 Sensor Node

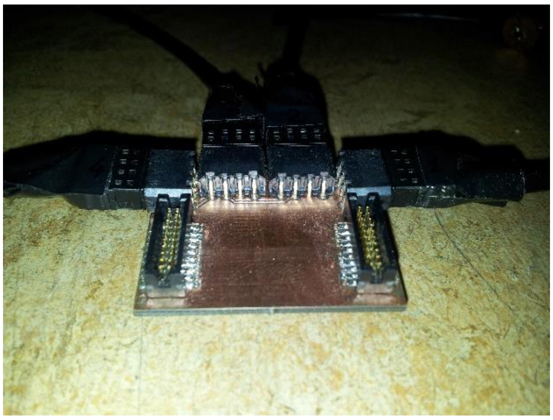
The spread sensors nodes are a CC2530 board as shown in Figure 5.4, and the node is made of a CC2530 RF board connected to a battery board for power supply using a battery, in addition to a sensor's hub board which is used to connect the sensors to it.

All this parts are shown in details in Figure 5.5.

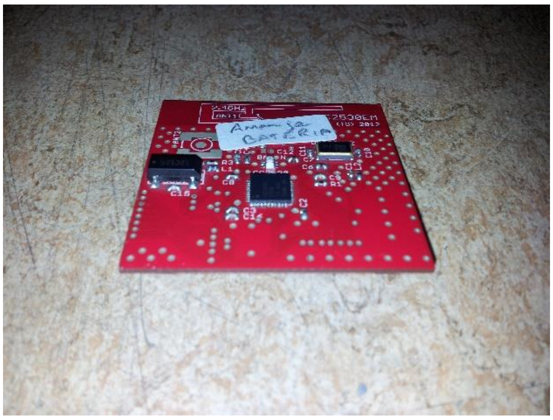


(A)

(B)



(C)



(D)

Figure 5.5 - Sensor's parts. (A) Battery Board, (B) Battery, (C) Sensors' hub, (D) CC2530 board

5.2 Developed Algorithm

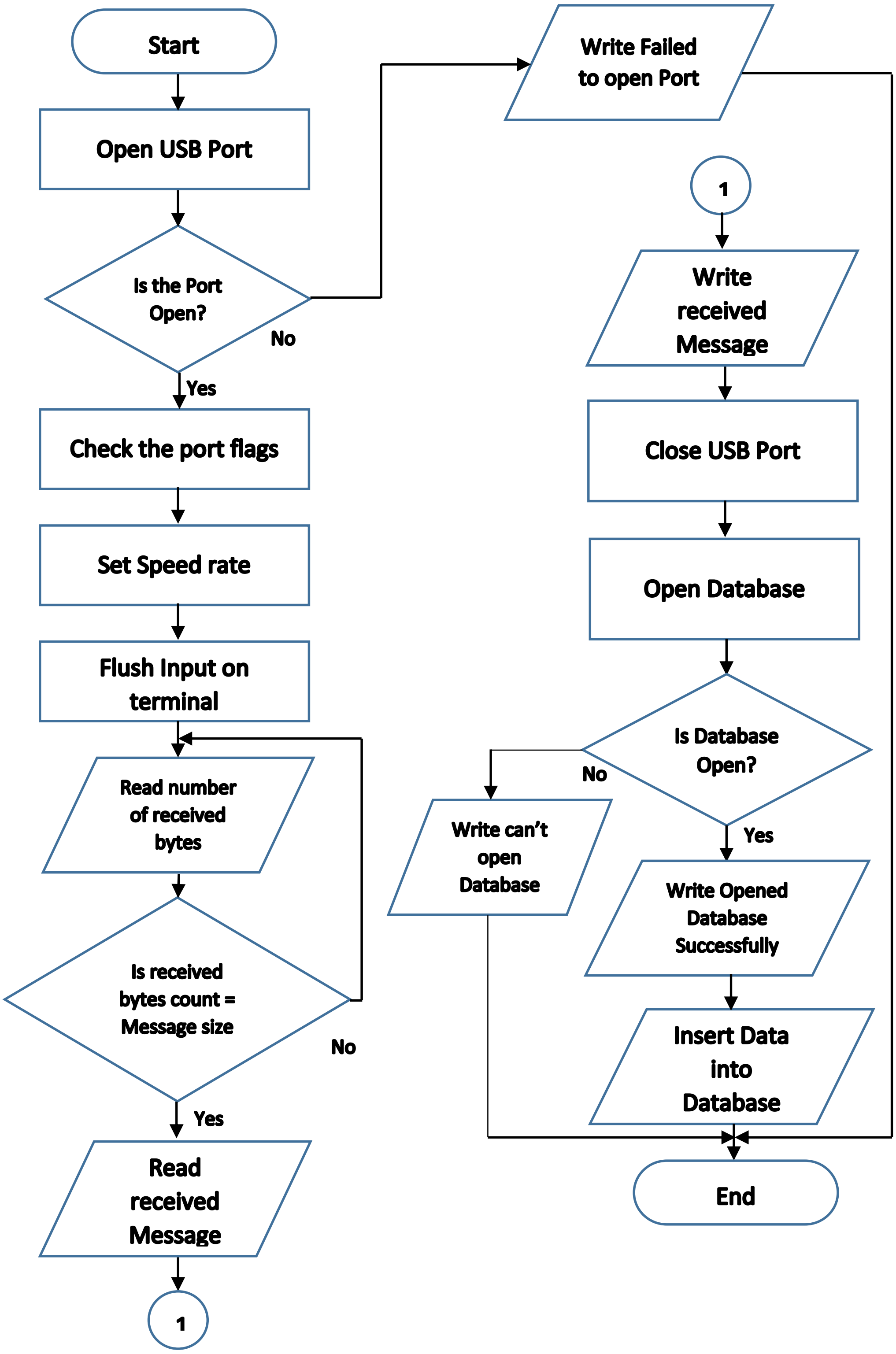


Figure 5.6 - System Algorithm

5.3 Database

Databases and database systems are considered to be an essential component of daily life in modern society: it is easy to encounter with several activities every day that involve some kind of interaction with a database.

For example, going to the bank to deposit or withdraw funds, making a hotel or airline reservation, accessing a computerized library catalog to search for bibliographical items, purchasing something online (such as a toy, book, or computer) chances are that the daily activities will involve someone or some computer program that access a database. Even purchasing items at a regular supermarket often it will automatically updates the database that holds the inventory of grocery items that they had [35].

These interactions are some examples of what is called traditional database applications, where most of the information that are stored and accessed are either numeric or textual. In the past few years, technology development has led to exciting new applications of database systems. The new media technology has made the possibility to store images, audio clips, and even video streamed digitally. These types of files have become an important component of the multimedia databases. Geographic information systems (GIS) can store and analyze maps, satellite images, and weather data. Data warehouses and online analytical processing (OLAP) systems are used in many companies to analyze and extract useful business information from very large databases to help in decision making. While real-time and active database technology is used for controlling industrial and manufacturing processes. The database search techniques are being applied to the World Wide Web (in search engines) to improve the search for information that is needed by users browsing the Internet.

Databases and database technology have a bigger impact on the growing use of computers. It is fair enough to say that databases play a main role in almost all areas where computers are used, including business, electronic commerce, genetics, medicine, engineering, education, law, and library science. The word database is used commonly so that it must be defined what a database is.

A database is a collection of data related to each other. Data, means known facts that are able to be recorded and that have implicit meaning. For example, consider the names, addresses, and phone numbers of the people. This data could be recorded in an indexed address book or it might be stored on a hard drive, using a personal computer or laptop and a software such as Microsoft Excel or Access. This collection of related data with an implicit meaning is knows as database.

The previous definition of database is quite general; for example, the collection of words that make up this page of text could be considered as a related data and that what makes a database. However, the widespread use of the term database is usually more restricted [35]. A database should has the following properties:

- A database represents parts of the real world, sometimes called the mini world or what known as universe of discourse (UoD). Any change to the mini world is replicated in the database.
- A database is a logically united collection of data with some inherited meaning. A random data sorted in a certain way can't be referred to as a database correctly.
- The database is usually designed, built, and then populated with data for a specific purpose. It has an intended group of users and some prepared applications that interest these users.

In other words, a database usually has some data sources, some amount of interaction with the real world events, and contents that actively interest the audience. The database end users may perform business transactions (i.e. a customer buys a camera) or events may happen (i.e. an employee has a baby) that changes the information in the database. In order to make the database accurate and reliable all the time, it must be a 100% reflection of the mini world that it represents; therefore, changes must get reflected in the database as fast and soon as possible.

A database can be of any complexity and size. For example, the list of names and addresses referred to previously may contain only a few hundreds records, each with a simple structure. Furthermore, the computerized catalog of a large library might contain half a million of entries organized with different categories by (Author's last name, subject, and book title) and each category is organized with alphabetical order. A database with a greater complexity and size is maintained by what is called the Internal Revenue Service (IRS) to monitor the tax forms filed by the U.S. tax payers. Assuming that there are 100 million tax payers and every tax payer files as an average are five

forms with approximately 450 characters of information per form, it would be a database of 100 × 106 × 450 × 5 characters (Bytes) of information. If the IRS keeps the previous three returns of each tax payer in addition to the current one, it would be a database of
8 × 1024 bytes (800 Gigabytes). This huge amount of information should be managed and organized so that users can search for, retrieve, and update all the data as needed.

An example of a large commercial database could be ebay.com [35]. It contains data for over 30 million books, CDs, DVDs, videos, games, electronics, apparel, and other products. The database occupies over 2 Terabytes (a Terabyte is 1024 Bytes of storage) and is stored on 200 different computers (servers). About 14 million visitors access ebay.com every day and use their database to make purchases. The database is simultaneously updated as new books and other products are added to the inventory and stock quantities are directly updated as purchases are transacted. About 100 people are responsible for keeping the eBay database up-to-date.

A database may be created and maintained manually or it might be computerized. For example, a library card catalog is type of database that might be created and maintained manually. A computerized database might be created and maintained either by a database management system or by a group of application programs written specifically for that task.

A database management system (DBMS) [35] is a set of programs that enables regular users to create and maintain a database. The DBMS is considered a general purpose software system that makes the processes of defining, manipulating, constructing, and sharing databases among different users and applications easy. To defining a database that involves to specify the structures, data types, and constraints of the data that are going to be stored in the database.

The definition of database or the descriptive information is also stored by the DBMS in some forms of a database catalog or dictionary; what is called meta-data. Constructing a database involve the process of storing the data on some storage media controlled by the DBMS. Manipulating a database involve functions such as querying the database to find specific data, updating the database to replicate changes in the mini world, and generating some reports from the data. Sharing the database allows multiple users and programs simultaneous access to the database.

5.4 Basic SQL

The SQL language might be considered one of the reasons for the commercial success of the relational databases. Because it became a standard for relational databases, users were less concerned about the migration of their database applications from other types of database systems, for example (network or hierarchical systems) [35]for relational systems.

Even if the dissatisfaction of the users with the particular relational DBMS product they were using, so converting to another relational DBMS product will not be too expensive and will not consume time because both systems use the same language standards. Practically, of course, there are a lot of differences between different commercial relational DBMS packages. However, if the user was diligent using only those features which are part of the standard, and if both relational systems were supporting the standard, then to convert between the two systems would be much simple. One extra advantage of having such a standard is when users might write statements in a database application program that can access some data which are stored in two or more relational DBMSs without the need to change the database sublanguage (SQL) because both the relational DBMSs support standard SQL.

The name SQL is the initial letters of Structured Query Language. Originally, SQL was called SEQUEL (Structured English QUEry Language) [35] and the design and implementation was made by IBM Research as an interface for an experimental relational database system which was called SYSTEM R. For now SQL is the standard language for commercial relational DBMSs. A joint effort that made by the American National Standards Institute (ANSI) and the International Standards Organization (ISO) has led to a standard version of SQL (ANSI 1986), known as SQL-86 or SQL1. A revised and much expanded standard which is known as SQL-92 (also referred to as SQL2) was subsequently developed. The next standard that is well-recognized is SQL: 1999, which started out as SQL3. Followed with two updates to the standard which are SQL: 2003 and SQL: 2006 that added XML features among other updates to the language. Another update in 2008 added more features for the database in SQL.

SQL is a comprehensive database language: It has different statements for queries, data definitions, and updates. Therefore, it is both a DDL and a DML. In addition, it has some easy ways to define views on the database, and to specify security and authorization, and to specify transaction controls, also to define integrity constraints. It also has rules to embed SQL statements into a general purpose programming language such as C/C++, COBOL, or Java.

5.5 SQLite Overview

SQLite is a software library that implements a self-contained, zero configuration, server less, and transactional SQL database engine. SQLite is one of the fastest-growing database engines around the world, of course that growth in terms of popularity, not about its size. The source code for SQLite is in the public domain.

5.5.1 What is SQLite?

SQLite is an in process library that implements a self-contained, zero configuration, server less, transactional SQL database engine. It is the one database, which is zero configured, that means like other database it does not need to be configured in the system [36].

SQLite engine is not a standalone process like other databases, it can be linked as required statically or dynamically with any application. The SQLite can access to its stored files directly.

5.5.2 Why SQLite? [36]

- SQLite does not require any system to operate it or a separated server process because it is (server less).
- SQLite is self-contained, which means it does not need external dependencies.
- A complete SQLite database is stored in one single platform disk file.
- SQLite is very small and light weighted, it takes less than 400 KB for fully configured or could be less than 250 KB with optional features omitted.
- SQLite comes with zero configuration, which means it does not need setup or administration.
- SQLite transactions are fully ACID compliant, which allows safe access from multiple threads or processes.
- SQLite supports the majority of the query language features which are in the SQL92 (SQL2) standard.
- SQLite is written in ANSI-C and provides simple and easy-to-use API.

- SQLite is available on UNIX (Mac OS-X, Linux, iOS, Android) and Windows (Win32, WinRT, WinCE).

5.6 SQLite for Raspberry Pi

SQLite is considered an excellent choice of database technologies to work with on Raspberry Pi for a several reasons:

- It is so easy to be installed.
- Python comes with a built in connection support.
- It is free and open source.
- Databases are stored in a single file so it can be directly connected to using the programing code, without the need run a server process.
- The library is small.
- And it is also serious product that it is being used in well-known products such as Mozilla's Firefox web browser.

Of course SQLite might not be the perfect choice for every project but it is more than enough for almost all the projects.

5.7 SQLite Limitations

There are few unsupported features of SQL92 in SQLite [36] which are shown below:

Table 5.1 - SQLite limitation [36]

Feature	Description
RIGHT OUTER JOIN	Only LEFT OUTER JOIN is implemented.
FULL OUTER JOIN	Only LEFT OUTER JOIN is implemented.
ALTER TABLE	The RENAME TABLE and ADD COLUMN variants of the ALTER TABLE command are supported. The DROP COLUMN, ALTER COLUMN, ADD CONSTRAINT not supported.
Trigger support	FOR EACH ROW triggers are supported but not FOR EACH STATEMENT triggers.
VIEWs	VIEWs in SQLite are read-only. You may not execute a DELETE, INSERT, or UPDATE statement on a view.
GRANT and REVOKE	The only access permissions that can be applied are the normal file access permissions of the underlying operating system.

5.8 SQLite Commands

The standard SQLite commands to interact with relational databases are similar to SQL. They are [36] CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into groups based on their operational nature.

5.8.1 DDL -Data Definition Language

Table 5.2 – DDL [36]

Command	Description
CREATE	Creates a new table, a view of a table, or other object in database
ALTER	Modifies an existing database object, such as a table.
DROP	Deletes an entire table, a view of a table or other object in the database.

5.8.2 DML -Data Manipulation Language

Table 5.3 – DML [36]

Command	Description
INSERT	Creates a record
UPDATE	Modifies records
DELETE	Deletes records

5.8.3 DQL -Data Query Language

Table 5.4 – DQL [36]

Command	Description
SELECT	Retrieves certain records from one or more tables

5.9 QT (Cross-Platform Application Framework)

Why Qt? Why do programmers choose Qt? The obvious answers is: The single-source compatibility of Qt, its feature richness, its performance in C++ [38], the source code availability, its documentation, and the high-quality technical support. This is all very well, but it misses the most important point that makes Qt successful which is because it is liked by the programmers.

Programmers may like one technology and dislike another, software engineers enjoy technology that feels right, but dislike everything that doesn't. When Qt's original architects faced a problem, they didn't just look for any good solution, or the simplest solution, or a quick solution. They were looking for the right solution, and after that they documented it. They admitted that they made mistakes, and also admitted that some of their design decisions didn't succeed passing the test of time, but they still got a lot of things right, and everything that wasn't right could and can be corrected. This is seen by the fact that a system originally meant to make a bridge between Windows 95 and Unix/Motif now unifies modern desktop operating systems as diverse as Windows, Mac OS X, and GNU/Linux with KDE.

Before Qt becoming this popular and so widely used, the developers of Qt were dedication to find the right solutions that made Qt special. That dedication is very strong today and it affects everyone who keep maintaining and developing Qt. Working on Qt is a responsibility and also a privilege.

One of many things that makes Qt a pleasure to be used is the online documentation feature. But the focus of the documentation is primarily on individual classes.

Nowadays, it is easy to find number of commercial and free Qt applications available for purchase or download. Some of them are specialized for particular vertical markets, while others could be aiming at the mass-market.

The toolkit that Qt uses is a C++ class library with a set of different tools for building multiplatform GUI programs and applications using a "write once, compile anywhere" approach. Qt allows programmers to use a single source tree for the applications that will run on Windows, Mac OS X, Solaris, Linux, HP-UX, and many other versions of Unix operating systems and with X11.A version of Qt it is also available for Embedded systems that use Linux, with the same API.

The Qt toolkit first became publicly available in May 1995. It was developed by Haavard Nord (Trolltech's CEO) and Eirik Chambe-Eng (Trolltech's president) [38]. It kept developing by years with different versions QT1.2 in April 1997, QT2.0 in June 1999, then QT3.0 in 2001, QT4.0 in 2005, and now a days there is QT5.0

5.10 QT Interface Overview

The QT interface is simple and easy to deal, as shown in Figure 5.7.

Creating a new project to design a complete user interface is a simple task due to the framework's rich toolbox and simplicity of use.

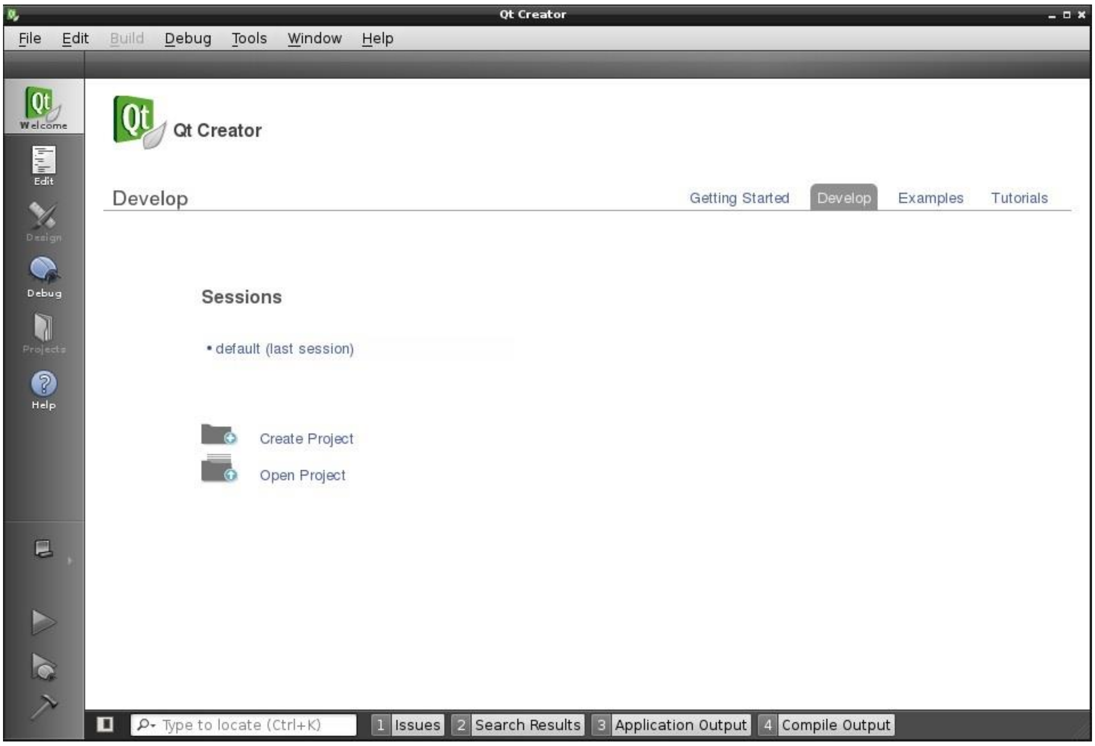


Figure 5.7 - QT interface

5.11 Final User Interface

The final user interface as shown in Figure 5.8, is very simple, by only pressing on Start button the coordinator will start receiving the data from sensor nodes, including (the C2530 MAC Address, Battery Level, Air temperature and humidity, Luminosity, Pressure and Soil Temperature). This received data will be inserted into the database.

And simply by pressing on Stop it will stop receiving the data.

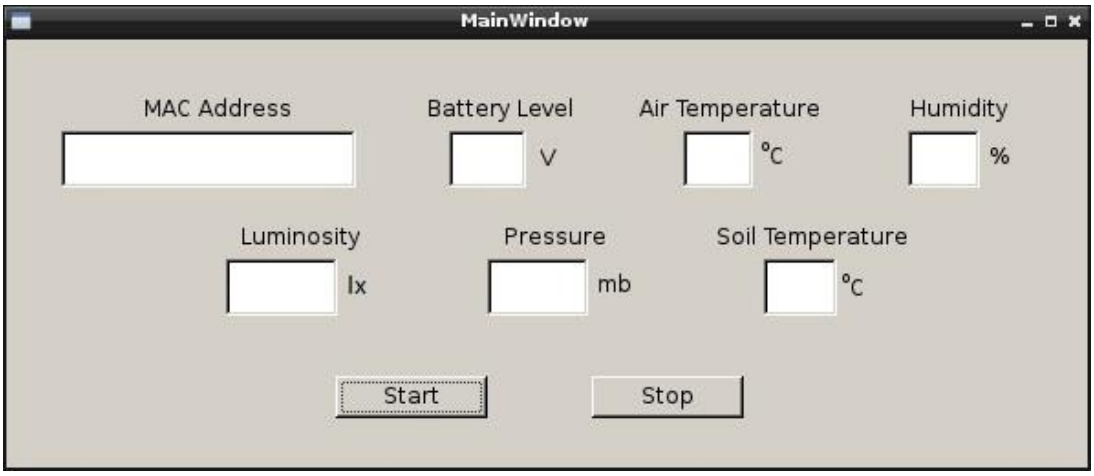


Figure 5.8 - User Interface

6 Tests and Results

6.1 Starting the Application

By pressing on the Start button on the user interface the program will start to receive the collected data from the sensor nodes through the coordinator as shown in Figure 6.1.

Figure 6.1 - Collecting Data

When a message received it will appear on the user interface as shown in the Figure 6.1, each data message source will be identified by the MAC Address of the sensor node transmitter.

The data will vary depending on the sensor node location and the surrounding environment. The following Table 6.1 shows a list of the last 10 messages received in the coordinator, (the test was done using one sensor node).

Table 6.1 - The last 10 received messages

MAC Address	Battery Level	Air Temperature	Humidity	Luminosity	Pressure	Soil Temperature
01:12:00:03:FF:FF:FF:FF	1.14	14.7	61.2	34.559	1007.6	11.1
01:12:00:03:FF:FF:FF:FF	1.14	14.7	61.2	34.559	1007.6	11.1
01:12:00:03:FF:FF:FF:FF	1.14	14.7	61.2	33.119	1007.6	10.8
01:12:00:03:FF:FF:FF:FF	1.14	14.7	61.2	33.119	1007.6	10.8
01:12:00:03:FF:FF:FF:FF	1.14	14.7	61.3	90	1007.5	10.8
01:12:00:03:FF:FF:FF:FF	1.14	14.7	61.3	90	1007.5	10.8
01:12:00:03:FF:FF:FF:FF	1. 14	14.7	61.3	87.839	1007.5	10.8
01:12:00:03:FF:FF:FF:FF	1. 14	14.7	61.3	87.839	1007.5	10.8
01:12:00:03:FF:FF:FF:FF	1. 14	14.7	61.4	78.119	1007.7	11
01:12:00:03:FF:FF:FF:FF	1. 14	14.7	61.4	78.119	1007.7	11

6.2 The Results

The coordinator collects the data received from the sensor nodes, these data are stored in the SQLite database in the Raspberry Pi.

By listing the information stored in the database as shown in Figure 6.2, all the collected data with the results that vary depending on the monitored environment.

```
01:12:00:03:FF:FF:FF:FF|1.144|14.679|61.871|77.399|1007.75|10.875
01:12:00:03:FF:FF:FF:FF|1.144|14.679|61.787|75.239|1007.66|10.812
01:12:00:03:FF:FF:FF:FF|1.144|14.679|61.787|75.239|1007.66|10.812
01:12:00:03:FF:FF:FF:FF|1.144|14.69|61.726|76.32|1007.78|10.937
01:12:00:03:FF:FF:FF:FF|1.144|14.69|61.726|76.32|1007.78|10.937
01:12:00:03:FF:FF:FF:FF|1.145|14.69|61.642|58245|1007.69|10.5
01:12:00:03:FF:FF:FF:FF|1.145|14.69|61.642|58245|1007.69|10.5
01:12:00:03:FF:FF:FF:FF|1.144|14.701|61.588|173.51|1007.66|11.75
01:12:00:03:FF:FF:FF:FF|1.144|14.701|61.588|173.51|1007.66|11.75
01:12:00:03:FF:FF:FF:FF|1.144|14.701|61.527|75.959|1007.69|11.312
01:12:00:03:FF:FF:FF:FF|1.144|14.701|61.527|75.959|1007.69|11.312
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.474|77.399|1007.78|10.937
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.474|77.399|1007.78|10.937
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.443|78.119|1007.76|11
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.443|78.119|1007.76|11
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.39|87.839|1007.57|10.875
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.39|87.839|1007.57|10.875
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.329|90|1007.55|10.875
01:12:00:03:FF:FF:FF:FF|1.144|14.712|61.329|90|1007.55|10.875
01:12:00:03:FF:FF:FF:FF|1.145|14.712|61.276|33.119|1007.63|10.812
01:12:00:03:FF:FF:FF:FF|1.145|14.712|61.276|33.119|1007.63|10.812
01:12:00:03:FF:FF:FF:FF|1.144|14.722|61.245|34.559|1007.63|11.125
01:12:00:03:FF:FF:FF:FF|1.144|14.722|61.245|34.559|1007.63|11.125
sqlite>
```

Figure 6.2 - Data collected in the database

These data can be used to trigger a function depending on the monitored environment.

7 Conclusion and Further Work

7.1 Future Possibilities

As a future possibility this work can be improved to be able to collect more different data depending on the need of the collected data such as Acceleration to sense a car impact against the guardrails of a highway for example, and also, which is more important, to upload the database onto the cloud and keep it synchronized so it will be possible to access the data anytime and anywhere.

Figure 7.1 shows an architecture for a future possibility.

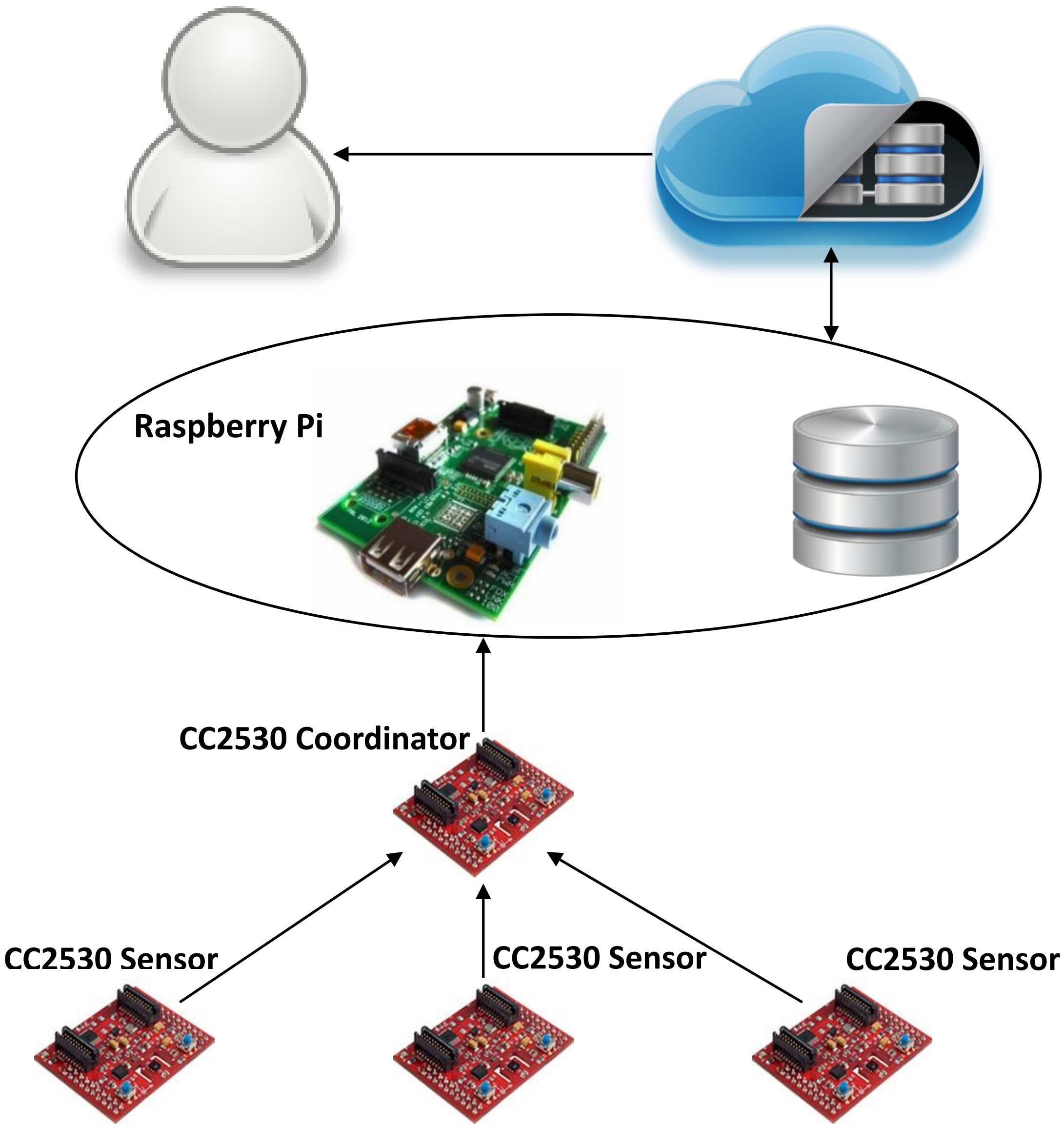


Figure 7.1 - Future work

7.2 Conclusion

This thesis has presented a system architecture for wireless sensor network that is capable of collecting data using a Wireless Sensor Network comprised by CC2530 sensor nodes. By utilizing a single shared controller (Raspberry Pi) that is connected -through a serial connection- to a CC2530 coordinator collecting data, the architecture is able to support flexibility in choosing application or specific communications protocols without sacrificing efficiency.

A C++ program was developed that allows the CC2530 coordinator to receive data from the CC2530 sensor nodes, these sensor nodes were able to collect data such (Temperature, Humidity and Pressure) for air and soil, and they were also equipped with a Luminosity sensor.

Besides receiving, and through using SQLite, a database was developed that can store the received data through the CC2530 coordinator, and using a C++ code enables to insert the received data into the database.

Using QT, a simple graphical user interface (GUI) was developed, that helps the user to start and stop the data collection procedure and also to monitor the received data in real-time.

Several scenarios can be applied on that, depending on what is the need of the wireless sensors network or what the data will be used for.

This thesis aim to monitor environmental changes using a wireless sensors network. Monitoring is needed in our modern life to facilitate human life by collecting data that help making decisions that will serve, improve and make the life more secure and comfortable.

A Raspberry Pi with CC25300 coordinator and sensor nodes were used, and later with using different sensors and a connection to the cloud the monitoring and controlling will be more effective and accessible.

Several years from now, wireless sensors will be a behind the scene technology that has grown enough to impact every aspect of the daily life. Probably all the factories and machines command and the control systems will be switched over to rely on wireless sensing and control points.

Today we can have thoughts about the current electrical grid and what our lives would be like without it. And tomorrow we have to give little thoughts to the wireless sensor network technology and all the other systems that is growing to impact all the aspects of our lives.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," 28 September 2002.

[2] J. Hill, M. Horton, R. Kling and L. Krishnamurthy, "The Platforms Enabling Wireless Sensor Networks," June 2004.

[3] J. L. Hill, "System Architecture for Wireless Sensor Networks," *Phd Thesis, University of California*, 2003.

[4] H. Ritter, J. Schiller, T. Voigt, A. Dunkels and J. Alonso, "Experimental Evaluation of Lifetime Bounds for Wireless Sensor Networks," in *European Workshop on Wireless Sensor Networks*, Istanbul, 2005.

[5] M. Bhardwaj and A. P. Chandrakasan*f*, "Upper Bounds on the Lifetime of Wireless Sensor Networks," 17 March 2001.

[6] "ATMEL," [Online]. Available: http://www.atmel.com/.

[7] S. R. Madden, "The Design and Evaluation of a Query Processing Architecture for Sensor Networks," *Phd Thesis, University of California*, 2003.

[8] Z. Zhou, S. Das and H. Gupta, "Connected K-Coverage Problem in Sensor Networks," 2004.

[9] Q. Li, J. Beaver, A. Amer, P. K. Chrysanthis and A. Labrinidis, "Multi-Criteria Routing in Wireless Sensor-Based Pervasive Environments," *Pervasive Computing and Communications,* vol. 1, 2005.

[10] M. Chen, C. Majidi, D. Doolin, S. Glaser and N. Sitar, "Design and Construction of a Wildfire Instrumentation System Using Networked Sensors," in *Network Embedded Systems Technology (NEST),* Oakland, 2003.

[11] J. Polastre, R. Szewczyk, C. Sharp and D. Culler, "The Mote Revolution: Low Power Wireless Sensor Network Devices," 2004.

[12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: a survey," 2002.

[13] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura and E. Jansen, "The Gator Tech Smart House: a programmable pervasive space," *IEEE Computer*, vol. 38, no. 3, pp. 50-60, 2005.

[14] K. Szlavecz, A. Terzis, S. Ozer, R. Musaloiu-E, J. Cogan, S. Small, R. Burns, J. Gray and A. Szalay, "Life Under Your Feet: An End-to-End Soil Ecology," Microsoft Technical Report MSR TR 2006 90, 2006.

[15] M. Ilyas and I. Mahgoub, Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, Boca Raton: CRC Press LLC, 2005.

[16] M. Demirbas, K. Yian and C. Shyan, "INSIGHT: Internet-Sensor Integration for Habitat Monitoring," in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006.* , Buffalo-Niagara Falls, 2006.

[17] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz and J. Lees, "Deploying a Wireless Sensor Network on an Active Volcano," *Internet Computing, IEEE,* vol. 10, no. 2, 2006.

[18] K. Lorincz, D. J. Malan, T. R. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland and M. Welsh, "Sensor Networks for Emergency Response: Challenges and Opportunities," *Pervasive Computing, IEEE,* vol. 3, no. 4, 2004.

[19] M. Li, D. Ganesan and P. Shenoy, "PRESTO: Feedback-driven Data Management in Sensor Networks," *Networking, IEEE/ACM Transact* , vol. 17, no. 4, 2009.

[20] H. Wang , J. Elson, L. Girod, D. Estrin and K. Yao, "Target classification and localization in habitat monitoring," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference*, 2003.

[21] M. Karpinski, A. Senart and V. Cahill, "Sensor Networks For Smart Roads," in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference,* Pisa, 2006.

[22] K. Sha, W. Shi and O. Watkins, "Using Wireless Sensor Networks for Fire Rescue Applications: Requirements and Challenges," in *Electro/information Technology, 2006 IEEE International Conference*, East Lansing, 2006.

[23] "CC2530 Description & parametrics," Texas Instruments, 2014. [Online]. Available: http://www.ti.com/product/cc2530?keyMatch=CC2530&tisearch=Search-EN.

[24] Contiki OS, [Online]. Available: http://www.contiki-os.org/.

[25] K. Finley, "Out in the Open: The Little-Known Open Source OS That Rules the Internet of Things," 2014. [Online]. Available: http://www.wired.com/2014/06/contiki/?mbid=social_fb.

[26] A. Dunkels, B. Gronvall and T. Voigt, "Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors".

[27] "Contiki," Wikipedia, 2014. [Online]. Available: http://en.wikipedia.org/wiki/Contiki.

[28] F. Vahid and T. Givargis, "Embedded System Design: A Unified Hardware/Software Approach," 19 November 1999.

[29] "Introduction to Embedded Systems," Top Max Tech, 2014. [Online]. Available: http://www.forum.topmaxtech.net/t116735.html.

[30] "McGraw Hill Introduction to Embedded Systems," 2014. [Online]. Available: http://highered.mheducation.com/sites/dl/free/007340456x/167481/sample1.pdf.

[31] A. A. Abdalla, Simply Raspberry Pi, Creative Common V4, 2014.

[32] "Open Source Hardware," Moser Electronics, 2014. [Online]. Available: http://pt.mouser.com/applications/open-source-hardware-galileo-pi/.

[33] T. Instrumentals, "TI Low Power RF," 2010.

[34] T. Instruments, "CC2430 Software Examples User’s Guide," 2009.

[35] R. Elmasri and S. B. Navathe, Fundimentals of Database System, Addison-Wesley, 2010.

[36] tutorialspoint.com, SQLite Toturial.

[37] Python & SQLite Databases.

[38] J. Blanchette and M. Summerfield, C++ GUI Programming with Qt 3, New Jersey: Pearson Education, Inc., 2004.

[39] S. &. U. Computing, "Nanotech Now," [Online]. Available: http://www.nanotech-now.com/smartdust.htm. [Accessed 2014].

[40] U. o. California, "Brainy Buildings Conserve Energy," [Online]. Available: http://coe.berkeley.edu/labnotes/0701brainybuildings.html. [Accessed 2014].

[41] "Wireless sensor network," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Wireless_sensor_network. [Accessed 2014].

Annex

Installing SQLite3

Installation is very straightforward on Raspbian [37] on the Raspberry Pi. Simply open the terminal as the one shown in Figure Annex.1 and type:

sudo apt-get install sqlite3 libsqlite3-dev

By pressing Enter the programs will install themselves.

Figure Annex.1 - Raspberry Pi Terminal

Creating a Database

After installing SQLite3 for the system, now it is easy to create a database called mydatabase.db [37]

Simply by typing:

sqlite3 MyDatabase.db

In the terminal by pressing Enter will see the following as Figure Annex.2.

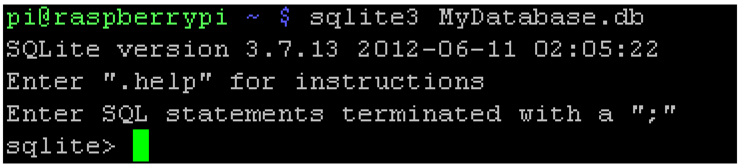


Figure Annex.2 - Creating a Database

Creating a Table

Adding SQLite statement to build a table called Test1 which will contain eight columns (MAC_Address, Battery_Level, Air_Temperature, Humidity, Luminosity, Pressure, Soil_Temperature) some with the type Text and some as Numeric.

Words in upper case are specific to SQL. They don't have to be typed in upper case, it's just a convention, but it does make it a lot easier to keep track of things when writing complicated SQL code.

Type the following after the SQLite prompt as Figure Annex.2 and then press Enter:

BEGIN;

CREATE TABLE My_Data (MAC_Address TEXT, Battery_Level NUMERIC, Air_Temperature NUMERIC, Humidity NUMERIC, Luminosity NUMERIC, Pressure NUMERIC, Soil_Temperature NUMERIC);

COMMIT;

Brackets, commas and semi-colons are very important or SQLite will complain!

- Press Enter
- Type .exit or .quit
- Press Enter to quit SQLite3

Intalling QT on Raspberry Pi

Firstly install the development tools needed by Qt Creator because it would be less heavy for the Pi to download separately.

In the terminal insert the following:

sudo apt-get install qt4-dev-tools

After the download and installation complete went for Qt Creator using the following command:

sudo apt-get install qtcreator

Then installing the following tools for compiling:

```
sudo apt-get install gcc

sudo apt-get install xterm

sudo apt-get install git-core

sudo apt-get install subversion
```

This gives as a result Qt Creator 2.5 with Qt 4.8.1 32 bit