**Chapter III**

**METHODOLOGY**

This chapter contains the procedures and methods that the researchers implemented in order to have a successful outcome of the project. The procedures and methods cover from the procurement of materials, hardware design and implementations, firmware and software developments as well as the test processes.

**3.1 System Flow Chart**

The operation of the system imposed is shown in Appendix A. It contains simplified illustration of the two major parts, the mobile sensor node which may be of any number but for the study, the researcher only made two nodes. The other major part of the system is the receiver node which receives the packets sent by both sensor nodes and dummy nodes.

**3.2 System Block Diagram**

Shown in Appendix B is the physical architecture of the system designed to gather temperature and pulse rate of a patient at an interval set. A single sensor node is composed of an MLX90614 infrared thermometer and an infrared based Grove ear-clip pulse rate sensor. These modules serve as input devices to an ATmega328p microcontroller unit with a CC1101 RF transceiver working at 433 mega hertz frequency band connected into it and to have the data gathered transmitted into the receiver node which is composed of a CC1101 RF transceiver, an Arduino UNO and a personal computer or a laptop. The microcontroller unit serves as a bridge to have the gathered data stored into the PC or the database pool.

**3.3 Design Process**

The overall system is designed with emphasis in cost and practical application but effective and reliable performance and functionality. The process of design requires careful planning and searching to attain an optimum performance of the system. To cater such necessities, hardware and firmware specifications must be set.

**3.3.1 Hardware Design Process**

Hardware design is a vital part for the development of the device to ensure a quality performance to carry out the objectives set. Thorough research and careful selection of materials must be done to have a reliable data acquisition and transfer throughout the network at an optimum period of time. To optimize the life expectancy of a sensor node, the materials used in the nodes must have minimal power consumption. Alternative power source must also be considered because this is also a vital factor to attain a longer life expectancy in each sensor node. The size of the device must also be put into consideration because the sensor nodes are mobile which means that it must be minimized and must not hinder the daily activities of the person being tested.

**3.3.1.1 Microcontroller Unit**

With the advancement of technology, microcontrollers rose into action and became popular simply because it can be programming capability to do a wide range of applications. Majority of the data loggers available in the market does have microcontroller in it to do specific data acquisition tasks. For the device, a programmable ATmega328p microcontroller will be used to do the tasks set. The circuit of an Arduino UNO will be used since its circuit is free and its programming IDE is open source. Also, several supports and forums are available online which would be a great help for the study and could minimize the period of the over-all development process. However, some parts of the Arduino UNO are not usable for the hardware design and may be deemed unnecessary so these parts will be disregarded. Also, the ATmega328p working at 3.3 volts in the Arduino UNO can also be set to use its internal 8 MHz clock which could be a great option in the aim of minimizing the size of the output device.

**3.3.1.2 MLX90614 Infrared Thermometer**

When dealing with data loggers, accuracy and reliability of the data gathered is necessary. Any discrepancy occurrence in the data gathered should be corrected in order to have a true data. A non-contact MLX90614 Infrared Thermometer working at voltage range of 2.4 – 3.6 volts will be used for the sensor node to gather temperature in degrees Celsius (°C). MLX90614 is best in medical usage with an accuracy of 0.5 degrees Celsius (°C). The sensor’s size is also a big contribution in attaining a minimal size of the output device. As shown in Appendix C is interface between MLX90614 to ATmega328P during testing. Both resistors R1 and R2 are designed to have a resistance value of 4.7 kΩ while capacitor C1 has a value of 0.1 uF.

**3.3.1.3 Grove Ear-Clip Heart Pulse Sensor**

Typically, pulse rate can be measured manually by counting the pulse occurrence over a span of time but since the study’s aim is to come up of a device that could measure pulse rate, a pulse rate data logger shall be used which could be easily integrated with a microcontroller. The Grove ear-clip pulse sensor is an infrared based sensor that could detect the changes in the density of blood due to the occurrence of pulse during blood circulations. By determining the time interval of the changes of the blood density in the ear lobe, the sensor can have an accurate value of pulse rate.

In Appendix C, it shows the interface between the grove pulse rate sensor and ATmega328P. Such interface is applicable for the whole device development.

**3.3.1.4 CC1101 RF Transceiver**

To maximize the mobility of the sensor node, the transmission of the data gathered from the sensors can be sent wirelessly depending on the choice of media. The effective transmission range for such media should also be maximized so that a bigger scope of a network will be made without jeopardizing the aim of minimizing the power consumption of each component used in the device. To cater such considerations, a CC1101 RF transceiver operating at 433 MHz was used. The said transceiver is capable of naming each node with a specific unique address which is beneficial for the study. A CC1101 RF transceiver break-out module will be used in the early development of the device for testing and debugging purposes. As soon as a final program code will be met for the operations set, an on-board CC1101 RF transceiver will be used to have a more reliable transmission because the components are already mounted on the PCB board.

The receiver node would also use a CC1101 RF transceiver but would only use an Arduino UNO as microcontroller unit. Since the mentioned transceiver works in 3.3 volts, some pins of the transceiver which receives data from the microcontroller were subjected to a voltage divider to scale down the incoming voltage. Such circuitry is done to prevent damage of the transceiver.

**3.3.1.5 Power Supply**

The device has lots of power supply options. The device can be first powered through the VCC and Ground port in the UART (Appendix D) section of the device which is also the section used in the firmware development of the device. The second option is through the 3.3 volt voltage regulator (Appendix D).However, such regulator needs a higher value of input voltage. Typically, the 3.3 volt regulator needs around at least 4.5 volts as the input voltage. The third and the ideal supply for the device is the ADP1621 SEPIC Converter shown in (Appendix D) with its circuit as well as its corresponding components used in the circuit. The said circuit supplies a constant 3.3 volts output despite the diminishing value of input voltage ranging from 5 – 3 volts. Such circuit is ideal for mobile data loggers because of its capability of regulating and boosting input voltages.

**3.3.1.6 Power Source**

Since the main power supply needed for the device needs input voltage ranging from 3 to 5 volts, a mobile phone battery is a good option for such application. It is also advantageous to the device for it would contribute to a lesser device volume. Mobile phone batteries also a rechargeable which would lessen the monetary expenses in a long period rather than buying replacement batteries.

**3.4 Hardware Development**

Hardware development follows to realize the specifications set in the previous section. Careful construction must be abided in order to attain a well quality device.

**3.4.1 Initial Testing**

Upon the arrival of the items procured to accomplish the device, each component was tested to ensure a fully functional output. Individual tests were done especially the sensors namely MLX90614 Infrared Thermometer and Grove Pulse Rate Sensor and the CC1101 RF transceiver to test its functionality, accuracy and effective range. The tests were initially done using the sample codes provided in the forums online uploaded into an Arduino UNO microcontroller.

The MLX90614 Infrared Thermometer was tested in terms of data accuracy. Temperature data of a person was gathered by using two instruments which is by using MLX90614 Infrared Thermometer and by using a standard thermometer available in the medical laboratories and drug stores. The data gathered is essential to get the average temperature difference between the two instruments which will be used into the MLX90614 program algorithm to correct such deficiencies.

To verify the effective range of the CC1101 transceiver, a receiver node and a dummy node were used to communicate. The dummy node sends stream of data from one (1) to twenty (20). It is expected that the same data shall be received by the receiver node. The farthest location of the dummy node from the location of the receiver node without dropping any number from the stream shall be considered as the effective range. To determine the distance in meters, Google Earth application was used.

**3.4.2 Printed Circuit Board Layout and Fabrication**

With the aid of a computer aided design tool, the circuits gathered in each of the components in the hardware design process were integrated into a single board. CadSoft EAGLE PCB Design Software was used in the designing of the PCB because of its wide range of available libraries of components. A two layer PCB board was used in the designing of the board to ensure a minimized size. The final design of the board was fit into a 5 x 5 cm two layered board.

The figure shown in Appendix E is the planned arrangement of the components of the device layout. Each part is numbered with a matching table for its corresponding function in the device. A red layout means that the component is on the top layer of the PCB. On the other hand, a blue layout means that it is on the bottom of the board.

Figures (a) and (b) in Appendix F are the final PCB layout generated thru the software used. The red layout is the top layer of the PCB while the blue one is the bottom layer. The gerber files of the design were then forwarded into the company named Seeed Technology Co., Ltd. which is currently located in China. Initially, the researcher decided to have the PCB done but due to the errors during manufacturing, the researcher then decided to have it fabricated overseas to avoid recurring errors during fabrication in order to minimize monetary expense. The fabrication and shipping for the PCB nearly took two weeks.

The figures (c) and (d) in Appendix F are the layout for the receiver node. The board above was designed to be a shield which could be put on top of the Arduino UNO.

The figures (e) and (f) in Appendix F are the images of the final PCB fabricated by Seeed Technology Co., Ltd. Final checking of the PCB was done whether there were errors during fabrication or none.

**3.4.3 Device construction**

After having the receiver node circuit board and the fabricated board for the sensor node checked, device construction followed to realize the device.

**3.4.3.1 Sensor Node**

Figure (a) in Appendix G is the final prototype arrangement of the sensor node following the assigned arrangement in the hardware design. Since the MLX90614 Infrared Thermometer does not need contact to get temperature value, it is located near the ear canal of the person.

**3.4.3.2 Receiver Node**

Figure (b) in Appendix G shows the receiver node with a shield for the voltage divider circuit to step down the voltage coming from the microcontroller into the CC1101 RF transceiver from 5 volts to 3.3 volts.

**3.5 Firmware Development**

Finally, to attain the desired system function, the sensors namely MLX90614 Infrared Thermometer and Grove Pulse Rate Sensor, the microcontroller unit of both sensor node and receiver node and the transmission of the data gathered thru the CC1101 RF Transceiver were programmed using the Arduino IDE.

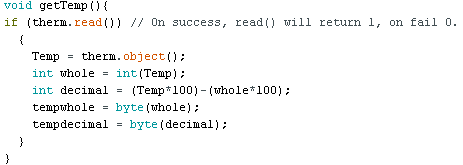
**3.5.1 Sensor Nodes**

**3.5.1.1 MLX90614 Infrared Thermometer**



**Figure 3.1 SparkFun Library Snippet**

As shown in figure 3.1, SparkFunMLX90614 library was used to lessen the burden of manually programming the said sensor. The library is capable of gathering ambient and object temperature but what is only needed is the object temperature of either Farenheit or Celsius. For the system, Celsius temperature system was used since it is the system used in the current location.



**Figure 3.2 Snippet in gathering temperature**

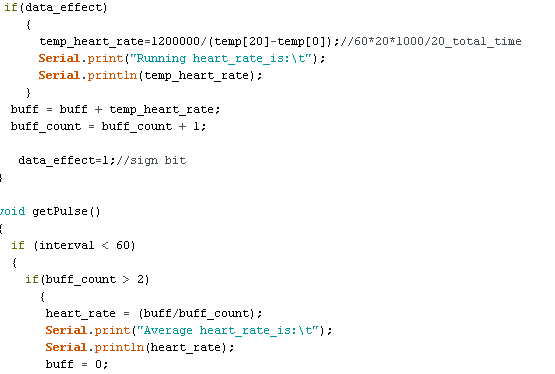
The snippet shown above in figure 3.2 is the program code in getting the object temperature. The temperature gathered is in float data type. The data type needed for sending data thru the transceiver is byte. Such data type doesn’t support decimal values so the whole number and decimal data of the temperature were separated so that it can be easily transmitted thru the transceiver.

**3.5.1.2 Grove Pulse Rate Sensor**



**Figure 3.3 Interrupt based pulse rate sensing**

As shown above in figure 3.3, the pulse rate sensor was attached to the interrupt 1 port of the ATmega328p pin so that whenever a pulse rate occurs, exact millis value shall be measured. A function millis returns the number of milliseconds since the ATmega328P began running the current program. With the millis value measured in each interval of pulse, by dividing it in a minute or 60 seconds, pulse rate per minute is measured.



**Figure 3.4 Calculation program code in gathering Average Pulse Rate**

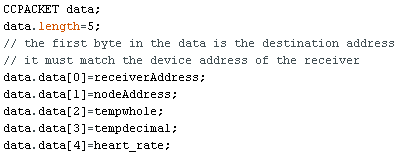
The snippet above, figure 3.4, is the calculation code to get the pulse rate. Running heart rate can be measured by getting the average interval of the twenty (20) samples of pulse interval. After which it will be divided into 60 seconds to attain the pulse rate. Three samples of running pulse rate shall be measured to get the average pulse rate. Such mechanism was done to get a least number of pulse rate value percentage errors. However, if a pulse interval of more than 2000 milliseconds occurs, a pulse rate value of 255 will be sent to notify the receiver node that there may be errors in the pulse rate data gathering in the module. This error may be a hardware failure or reattachment of the ear clip is needed.

**3.5.1.3 CC1101 RF Transceiver**



**Figure 3.5 Interrupt-based CC1101 transceiver**

Figure 3.5 shows on how to ensure that any data sent wireless throughout the network; the transceivers are programmed to be connected in the interrupt 0 pin of the ATmega328p microcontroller. Using interrupts in the system is beneficial to directly stop the looping of the program whenever a data is sent from the sensor nodes. Such data are important and requires immediate processing so that it will be immediately displayed in the serial monitor.



**Figure 3.6 Packet Assignment for the gathered data**

The snippet shown in figure 3.6 is the packet assignment for the data gathered before transmission. The first packet shall always contain the recipient address which is the receiver node with address one (1). Each of the nodes has its own identification address to ensure a proper data transmission. Each of the nodes receives any packet sent in the network but rejects the message whenever the first packet in the message does not coincide with its corresponding identification address.

**3.5.2 Receiver Node**



**Figure 3.7 Program code for displaying received packet**

The program code of the receiver node is somehow similar to the sensor node but does not include the gathering of the data using the sensors. The receiver node’s job only is to write its packets received from the sensor nodes. The snippet above is the program lines for writing the packets received into the serial port. A -1 value will be displayed whenever a 255 value of pulse rate is received. This serves as a warning for the attendant because there is no such negative value of pulse rate. This would tell that the specific sensor node requires immediate maintenance.

**3.6 Debugging, Sensor Data Validation and Synchronization**

After having the sensor nodes and the receiver node programmed. A test run was done to check the functionality of the system. Any errors seen during the testing were carefully noted and shall be corrected again in the firmware development. Also, the pulse rate and the temperature data gathered by the modules were checked if the values were valid. The pulse sensor data were compared with the conventional way of getting such data which is by counting the occurrences of pulse rate manually over a period of time. The temperature data gathered were compared with the data gathered by using a thermometer. Also, the data sent thru the transceivers were also checked to verify if there were data alterations during transmission. As soon as all of the parts are functional, additional dummy nodes were made capable of mimicking the data gathered by the sensor nodes. Such dummies are beneficial for the system to fully show its multinode communication capacity.