IDEATION PHASE-1
LITERATRUE SURVEY

DATE	19 NOVEMBER 2022
TEAM ID	PNT2022TMID47145
PROJECT NAME	IOT BASED SAFETY GADGET FOR CHILD SAFETY MONITORING AND NOTIFICATION
Maximum marks	2

CHILD MONITORING SAFETY DATA ANALYSIS

Table of Contents

Table of Figures

Abstract

Parents need ways to better monitor the child, current systems only consider the safety aspect of the child. This project presents a system that will take both, safety, and health of the` child into account to provide better assurance to the parents. This device contains a microcontroller that is combined with a heart rate sensor, a light sensor, an accelerometer, and a GPS that work together to monitor the activity and the location of the child. The system is supported by an android application which will be used by the parent to monitor children. The paper talks about the achievements of the project.

The overall percentage of child abusements filed nowadays in the world is about 80%, out of which 74% are girl children and the rest are boys. For every 40 seconds, a child goes missing in this world. Children are the backbone of one's nation, if the future of children was affected, it would impact the entire growth of that nation. Due to the abusements, the emotional and mental stability of the children gets affected which in turn ruins their career and future. These innocent children are not responsible for what happens to them. So, parents are responsible for taking care of their own children.

1.0. Introduction

1.1. Problem

The satisfaction of parents can only be achieved when they are assured that their child can grow in a safe environment. To be convinced that their child is healthy and protected is one of the major concerns to parents in this developing day and age. The number of children staying in towns are increasing significantly due to the rapid development of industrialization and urbanization in countries and in this fast pacing environment, busy parents cannot always depend on teachers or guardians to look after their child. From the Global Missing Children Network, *Figure 1* below shows the estimate of missing children per year in some of the countries [1].

In **Australia**, an estimated **20,000** children are reported missing every year.
Australian Federal Police, National Coordination Centre.

In **Canada**, an estimated **45,288** children are reported missing each year.
Government of Canada, Canada's Missing – 2015 Fast Fact Sheet.

In **Germany**, an estimated **100,000** children are reported missing each year.
Initiative Vermisste Kinder.

In **India**, an estimated **96,000** children go missing each year.
Bachpan Bachao Andolan, Missing Children of India.

In **Jamaica**, an estimated **1,984** children were reporting missing in 2015.
Jamaica's Office of Children's Registry

In **Russia**, an estimated **45,000** children were reported missing in 2015.
Interview with Pavel Astakhov MIA "Russia Today", Apr. 4, 2016.

In **Spain**, an estimated **20,000** children are reported missing every year.
Spain Joins EU Hotline for Missing Children, Sep. 22, 2010.

In the **United Kingdom**, an estimated **112,853** children are reported missing every year.
National Crime Agency, UK Missing Persons Bureau.

In the **United States**, an estimated **460,000** children are reported missing every year.
Federal Bureau of Investigation, NCIC.

Figure 1: Estimate of Missing Children from the Global Missing Children Network

-

Safety of their child is not the onl+/3+/3-y worry parents face; majority of their concern goes to health and wellbeing of the child. Research shows that more than 80% of children between age 11 to 17 do not get significant physical activity [2]. These children tend to develop variety of health problems as adults, such as cardiovascular disease, musculoskeletal disorders etc. [3].

The first important thing for parents is to feel safe about their children. Parent cannot be monitoring their child's activities 24 x 7 hours in this busy fast pacing world. Although there is no substitute for proper childcare which would include constant monitoring, the truth is constant monitoring of child is not always feasible. And they cannot depend on teachers to give individual attention to a child in school.

1.2. Project Aim and Objective

To help overcome some of these worries, technology can provide with an answer to guarantee their child's safety and health. In this day and age, parents may choose to use a monitoring technology to check and track their child's activities while the child is unsupervised. This can greatly reduce stress and give them more freedom while the child is not around. New monitoring and controlling methodologies are being developed as technology is evolving.

This paper aims to develop a system for supervising children using IoT- based monitoring system that keeps track of the child's health, location, and activity. The overall view of the concept is designed using three distinct modules: the parent module, the child module, and a 3D printed lamp.

- Child Module

  This module is attached to the child. It involves multiple number of sensors which collects data from the child such as location and activity of the child. After gathering required data, it periodically transmits the information to the Parent module when requested.

- Parent Module

  This module is basically an android application that requests the information from the Child module, it reads this information and represents it in an orderly manner which is easy for the user to read and understands.

- 3D printed Wi-Fi Lamp –

  This module is a 3D printed furniture that is kept in the house which changes color according to the activity the child is engaged in. It also includes an LCD on it which will display the child's activity.

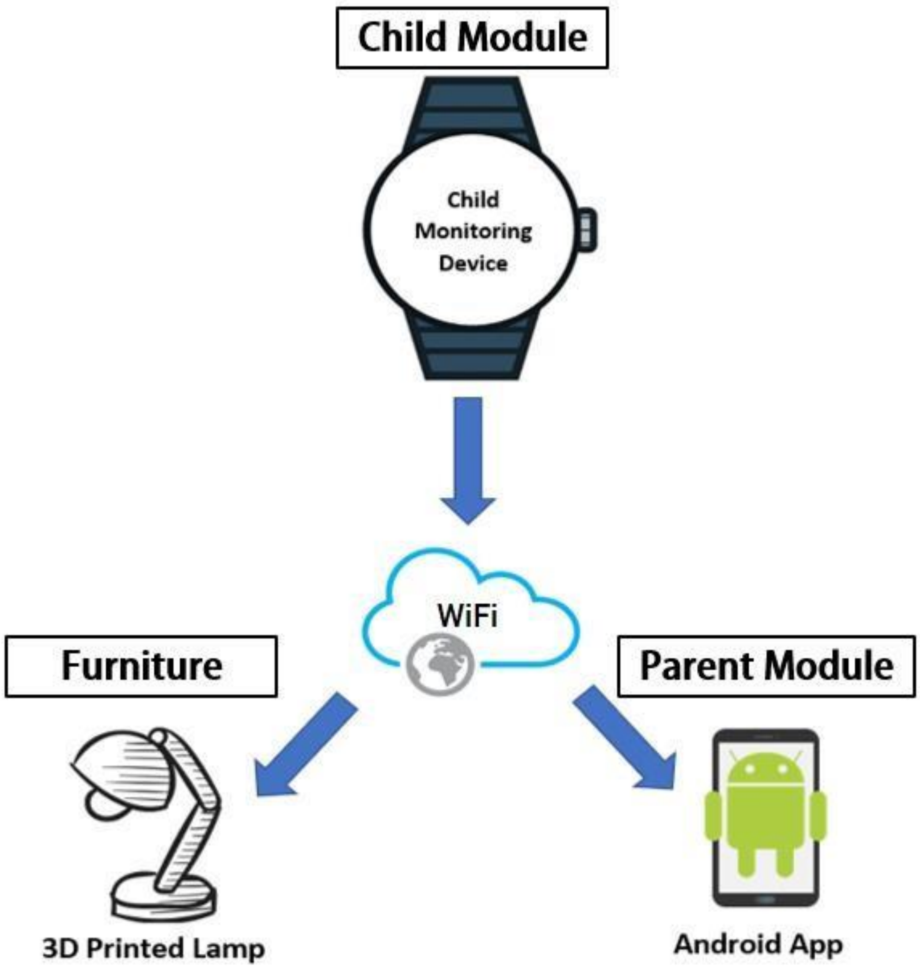

Figure 2: Representation of the 3 parts of the Monitoring System

The Child module is the computational brains of the system as it contains a microchip which gathers relevant data from all the sensors connected and analyses them to deduce the child's activity. It is responsible for the communication to the parent module and the Wi-Fi lamp. Every time the Child module sends status of the child it automatically displays it on the parent module and the LCD on the Wi-Fi Lamp.

1.3. Applications

The monitoring and control system created in this project can be used for many purposes. The main aim of the project is to reduce parents stress by using technology to monitor the child. Further applications can be built upon the same core idea such as adding extra sensor to the Child module to increase the number of functions e.g. Adding a temperature sensor to monitor the child's body temperature.

As public healthcare in many regions currently face crisis due to factors such as shortage of medical staff. Hence, another application for this device is it can be used in retirement homes to monitor the health and wellbeing of old people. Both the family and the caretakers will be able to observe the patient's health and activities. According to their activity, doctors can prescribe proper dosage of medication that will be suitable for them.

If the child goes beyond that particular boundary specified, the respective guardians will receive an alert call using GSM. In our system, we use several components like

1.Temperature sensor

2.Pulse sensor

3.GPS

4.GSM

5.Web camera

6.Raspberry pi microprocessor

If the device moves, out of that boundary the server transfers an alert call by activating the GSM, to the user. The live location of the device will be updated in the server and pinged in the website for every few seconds. The server side coding was written in PHP and the controller side coding was written in Python.

2.0. Literature Review

2.1. Related Projects

This section provides an overview of few other related projects:

A Hybrid Model on Child Security and Activities Monitoring System Using IoT [4]

This project is done to confront some of the issues faced by many children. These issues are observed by utilizing IoT components and sensors to check if the youngster is near an environment of unaccepted behavior. In the event that the child is close to someone with this behavior, at that point the system will send an alarm message to parent that child is in danger. By tracking the location of the child, the guardians/parent can find where he/she is and how they can help the kid from such situation. An Alcohol and Smoke Gas Sensor are used alongside Blood Pressure sensor to check whether the kid in any strange conditions. By using the information gathered by these sensors, it can assist parents or guardians to make appropriate decisions to save the child from that situation. The main idea of this project is to securely track the child from violence situations.

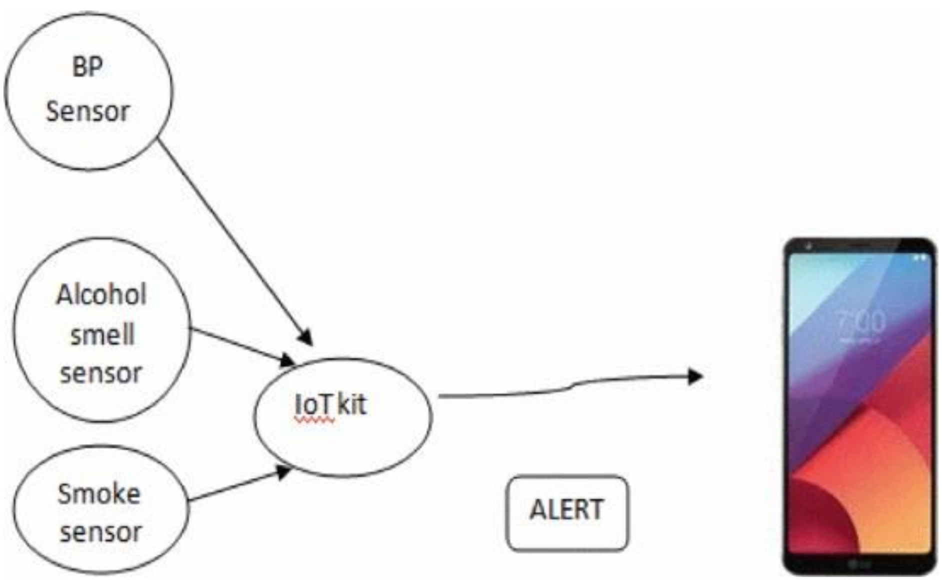

Figure 3:Components in the Child Security and Activity Monitoring System [4]

The blood pressure sensor is used to send alert signals to the parent when the child's blood pressure is high or low. The alcohol smell sensor detects if there are any alcoholic nearby and finally the smoke sensor detects for smokers or any fire accidents.

Child Safety Wearable Device [5]

This device talks about the idea of a wearable gadget for little children. The significant of this device over other devices are that it can be used in any cellphone and does not really require a costly phone. It also be operating by people who are not familiar to technology. The purpose of the gadget is to locate children with ease.

The main idea of the device is to have Communication system between the child's wearable device and parents' mobile phone through SMS. Specific key phrases such as "SOS", "LOCATION", "BUZZ", "TEMPERATURE" is used, to send to the child's wearable device through SMS. The device will answer back according to the SMS sent for example accurate location of the child will be sent when parent messages "LOCATION". Similarly, a bright SOS light and distress buzzer can be toggled on and finally the surrounding temperature of the child can be attained by using the dedicated keywords.

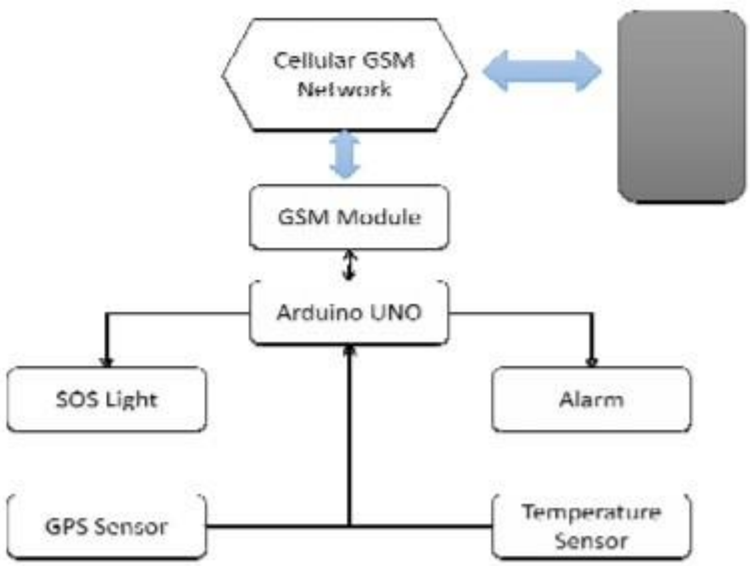

Figure 4: System Overview of the Child Safety wearable device [5]

The project it is based on an Arduino Uno microcontroller and further comprises of a GSM module to send the data collected by other sensors to the parent's phone when requested.

A GPS sensor and the temperature sensor are used as an input device to send location and surrounding temperature to parents' phone.

An SOS light and a buzzer are used as an output to send distress signal from the Childs variable device, signal is activated through SMS from the parent's phone.

Multi-sensor Wearable for Child Safety [6]

This paper talks about a wearable gadget like a wristband which tracks the child every now and then to guarantee their wellbeing. On the off chance that any issue happens it would warn guardians through the mobile phone so that they can take immediate action. The warning will be sent through SMS to the parent's phone. Parents can send SMS with certain catchphrases to the gadget and the gadget answer back.

The gadget can recognize the child's location, it can distinguish surrounding and body temperatures, it can detect humidity and also the heartbeat of the child. For emergency circumstances, the gadget would have a few ways like an alert such as SOS light which will inform nearby observers to care for the child.

This paper is about the wellbeing and security of a child to assist them with recovering from a trouble.

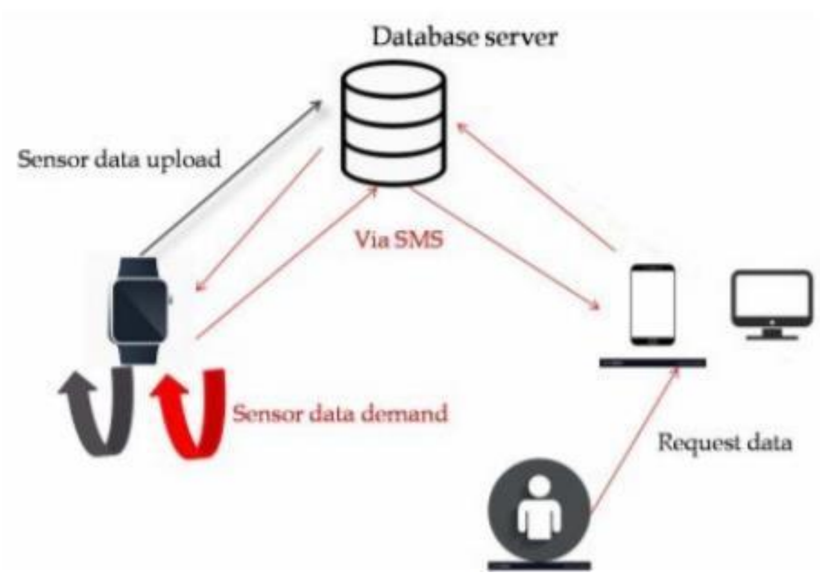

Figure 5: Flow Diagram of the Multi-Sensor Wearable device [6]

How the system works is by monitoring the temperature, humidity, and heartbeat of the child. If these values exceed the threshold value an automatic generated SMS will be sent to alert the parent. Parents can also request for specific data to be sent; this is done by sending key phrases from the parent's phone through SMS.

After gathering all the required data from the sensors, the Arduino microcontroller is programmed to send this information to a database server, and the information will be sent to the parent from this server when requested.

Implementation of Children Tracking System on Android Mobile Terminals [7]

This paper involves around tracking children going to school. The proposed project incorporates a child module and two receiver modules for getting the data about the missing kid periodically. The child module combines an ARM7 microcontroller, a GPS module, a GSM module, and a voice playback circuit. The receiver module comprises of a database and an android phone.

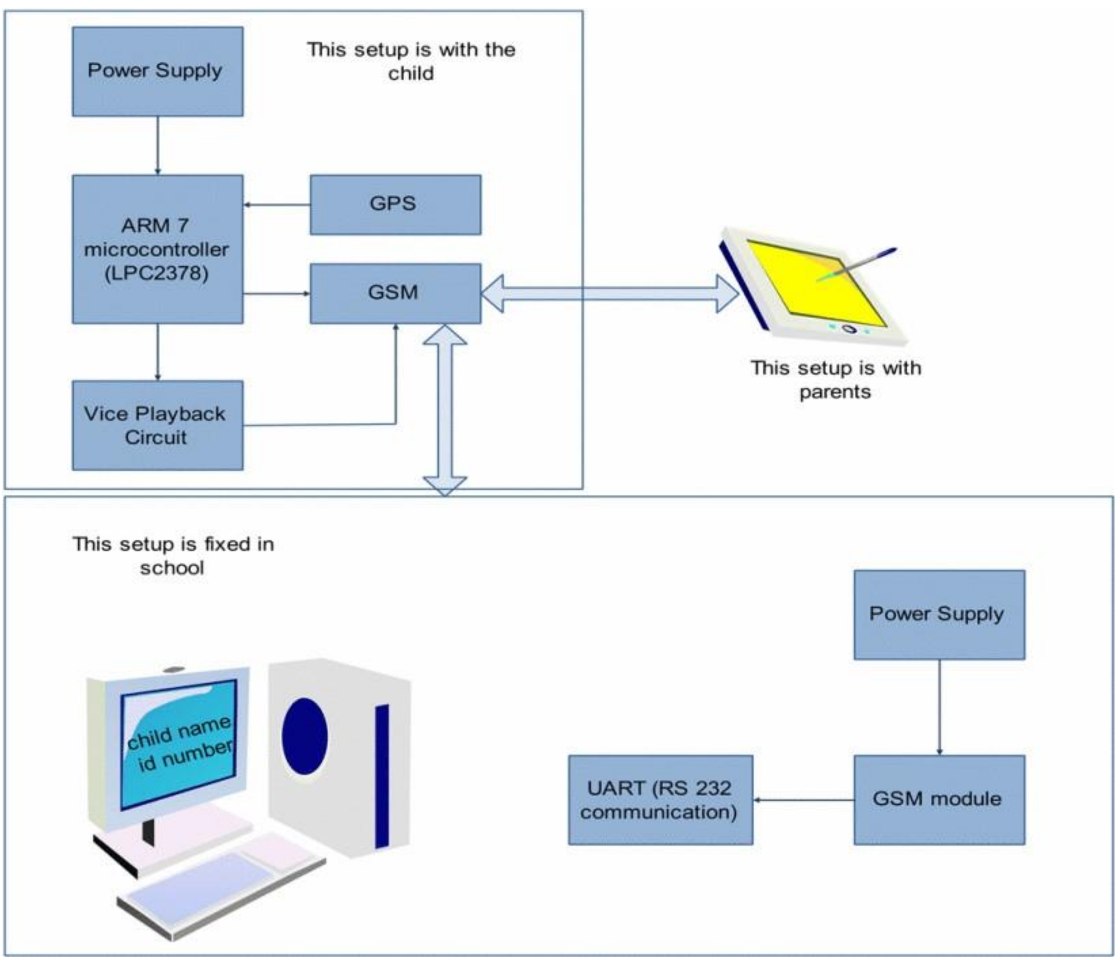

Figure 6: Block Diagram of the Child Tracking System [7]

The Child module is fixed to every single child. A moving child is tracked by GPS and the location is sent to ARM7 microcontroller. The ARM7 microcontroller sends the GPS information (longitude & latitude) to the GSM module. The GSM will send location to the receiver modules. At the point when the child cries, the voice playback circuit is activated by ARM7 microcontroller and information about the child is given through SMS to their parents.

Pervasive Health Monitoring of Special Child using IoT and Cloud Technologies [8]

In this paper IOT and the cloud are combined together to discover a health care service for special children between the age of 4 to 16 years. Taking into account that an extraordinary child faces issues in communicating his/her concerns, this paper proposes a model where a special child would utilize a wearable IoT technology that will detect the blood pressure and pulse from the heart. The information received is captured in the cloud and will be broadcast to the respected doctors and parents of the child in the occasion of a trigger. As many health features are collected the final decision from the algorithm would notify and send the data collected to the respected doctor for assistance.

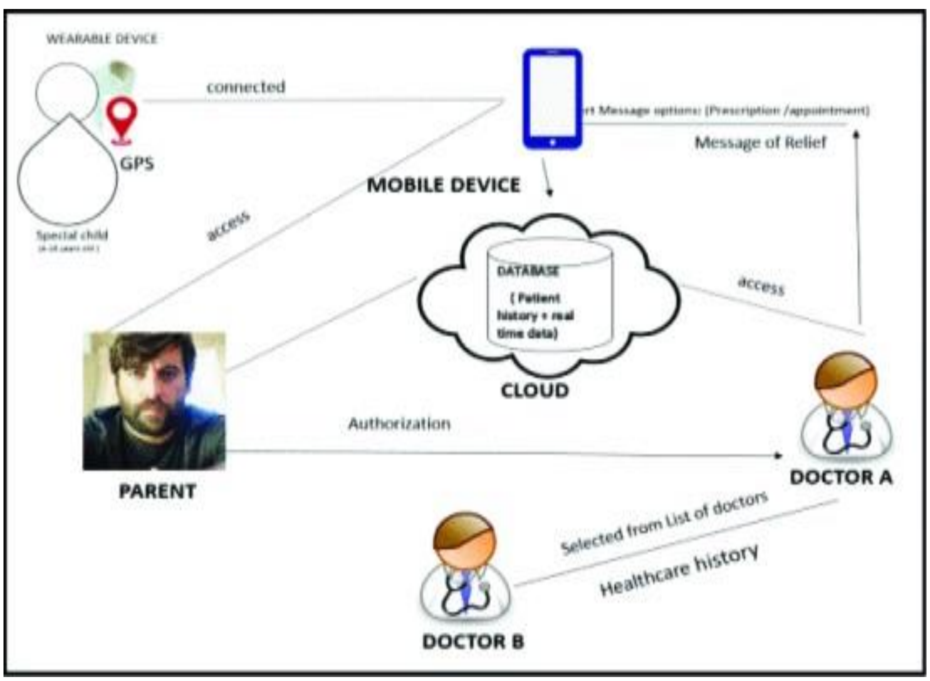

Figure 7: Flow diagram of the Pervasive Health Monitoring system [8]

The IOT modules included in this project are a biosensor, a Bluetooth module, a heart rate sensor, and a blood pressure sensor. An application is used to notify the parents if the blood pressure or the heart rate goes above the common limit. This module also contains an GPS module to help locate the child.

The application consists a unique ID which the parent can give authorization to the doctor so they can also access and track the health data of the child.

Finally, this data is also backed up to the cloud so doctors can see common trends or signals to make better judgment on the child's health.

2.2. Available Systems

This section talks about some of the devices that are available in the market for child health and safety monitoring.

The Leapband by Leapfrog [9]

This is a device that aims to fight childhood obesity. It is a fitness tracked made for children used for the age from four to seven. The device comprises of many activities such as games, challenges, quizzes, and rewards. The device turns exercise into a game for kids to play and enjoy.

Figure 8:The LeapBand [9]

Some features of the LeapBand include:

- Water resistance design
- High resolution screen
- Quiet mode when in school

An accelerometer is used to track the movement of the child to play movement related games.

This device promotes active play, health, and wellness for the child.

X-Doria's KidFit [10]

This device is an activity tracking technology that is targeted for children between age of 5 and 13. How it works is that parents are able to set a goal through the mobile application which is synced to the watch using Bluetooth technology. The progress of the child is tracked and provides feedback via the application.

The wearable device can hold charge up to 7 days and it is waterproof. It can also be used to track the child’s sleep partners just by a push of a button on the device.

The X-Doria KidFit:

- Daily step count
- Daily Distance count
- Daily calorie burnt
- Daily sleep patterns
- Bluetooth 4.0

Figure 9: Labled Diagram of the X-Doria KidFit [10] Features in

The only drawback of the system is that it does not include GPS tracker. The system mainly focuses on the health benefit of the child instead of the safety aspect.

3.0. Methodology

This section provides details about how the project is conducted explaining all the three modules of the system.

3.1 Child Module

This is the device that will be with the child, obtaining information on the child's health and activities. The part consists of range of sensors which are mentioned bellow.

Pulse Sensor:

A Grove Finger clip heart rate sensor is used. This sensor is placed in contact to the child, it outputs heart rate in form of beats per minute.

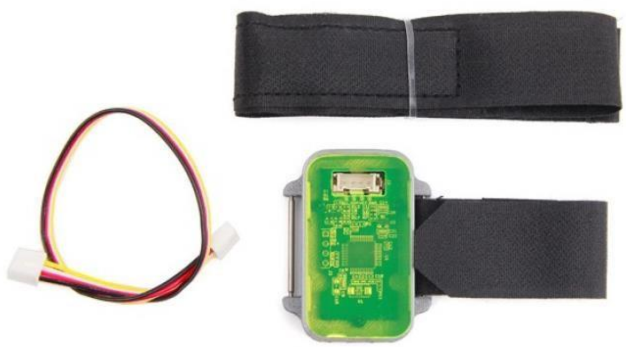

Figure 10: Finger Clip Heart Rate Sensor

Accelerometer:

In this case an MPU 6050, which is an accelerometer and a gyroscope. This device is used to detect the movement of the child.

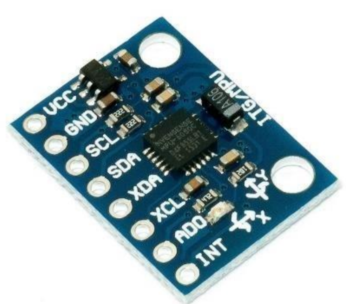

Figure 11: Accelerometer and Gyroscope

GPS Module:

to determine the location of the child a NEO6MV2 GPS module has been used.

Figure 12: GPS module

Light Sensor:

to determine if the child is awake or sleeping the L1B01 light sensor is used.

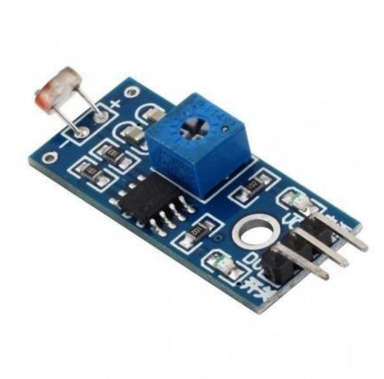

Figure 13:Light Sensor Module

All these four sensor modules are connected to a hub which is an Arduino Mega microcontroller. The microcontroller ultimately decides what activities the child is engaged in.

Figure 14: Arduino Mega Microcontroller

In addition to these sensors a Wi-Fi module is added.

Wi-Fi Module:

An ESP8266 is used to transmit the data that is collected by the Arduino Mega.

Figure 15: Wi-Fi Module

Furthermore, a set of Lithium Ion batteries (ICR 18650) are added to power these devices when it is not connected to the power supply.

Figure 16: Lithium Ion Batteries

All the four sensors which are the pulse sensor, accelerometer, light sensor, and the GPS operate together to define one of the three activities: “Exercising”, “Sleeping” and “Studying”. The data collected by the sensors is forwarded to a microcontroller which then determines the status of the child.

WiFi modules (wireless fidelity) also known as WLAN modules (wireless local area network) are electronic components used in many products to achieve a wireless connection to the internet.

ESP8266 comes with capabilities of

- 2.4 GHz Wi-Fi (802.11 b/g/n, supporting WPA/WPA2),
- general-purpose input/output (16 GPIO),
- Inter-Integrated Circuit (I^2C) serial communication protocol,
- analog-to-digital conversion (10-bit ADC)
- Serial Peripheral Interface (SPI) serial communication protocol,
- I^2S (Inter-IC Sound) interfaces with DMA(Direct Memory Access) (sharing pins with GPIO),
- UART (on dedicated pins, plus a transmit-only UART can be enabled on GPIO2), and
- pulse-width modulation (PWM).

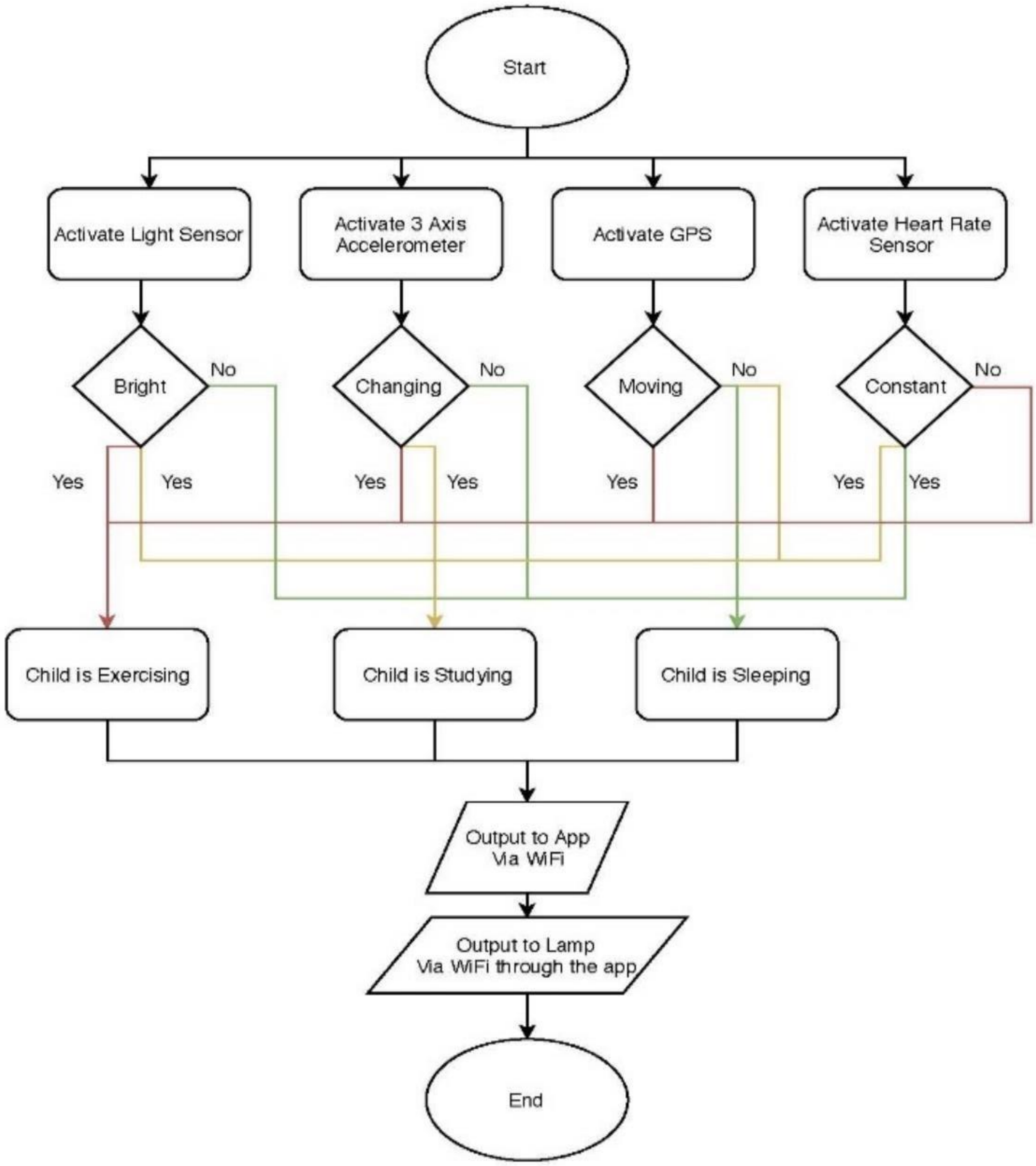

Figure 17: Child Module Flow Chart From the flow chart in *Figure 17* we can see that the activity is:

Exercising If:

- Light Sensor detects light
- Accelerometer detects constant movement
- GPS is moving or stationary

- Pulse sensor detects high pulse

Studying If:

- Light Sensor detects light
- Accelerometer detects limited movement
- GPS is stationary
- Pulse sensor detects constant pulse

Sleeping If:

- Light Sensor detects no light
- Accelerometer detects limited movement
- GPS is stationary
- Pulse sensor detects low and constant pulse

Figure 18 shows a tabular representation of the flow chart in *Figure 17*.

<table>
<tr><th rowspan="2">Activity | Sensor Reading</th><th>Light Sensor</th><th>Accelerometer</th><th>GPS</th><th>Heart Rate Sensor</th></tr>
<tr><th>Bright</th><th>Moving</th><th>Moving</th><th>Constant</th></tr>
<tr><td>Sleeping</td><td>NO</td><td>NO</td><td>NO</td><td>YES</td></tr>
<tr><td>Exercising</td><td>YES</td><td>YES</td><td>YES</td><td>NO</td></tr>
<tr><td>Studying</td><td>YES</td><td>YES</td><td>NO</td><td>YES</td></tr>
</table>

Figure 18: Tabular Representation of the Flow Chart

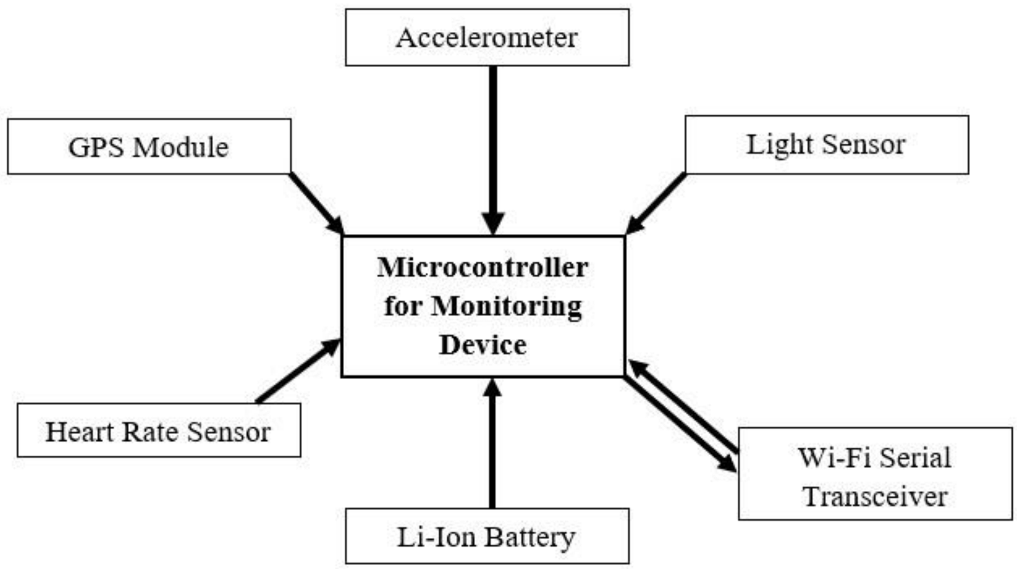

Figure 19: System Block diagram

3.2. Parent Module

This module is an Android application that receives data from the child module through the Wi-Fi module. It reads this data and represents it in an organized manner which is easy for the parent to read and understand.

How this works is after the Arduino mega gets data from the Sensors it transmits to a website using the Wi-Fi module. A mobile application is used to display information from the web page on parents’ mobile phone.

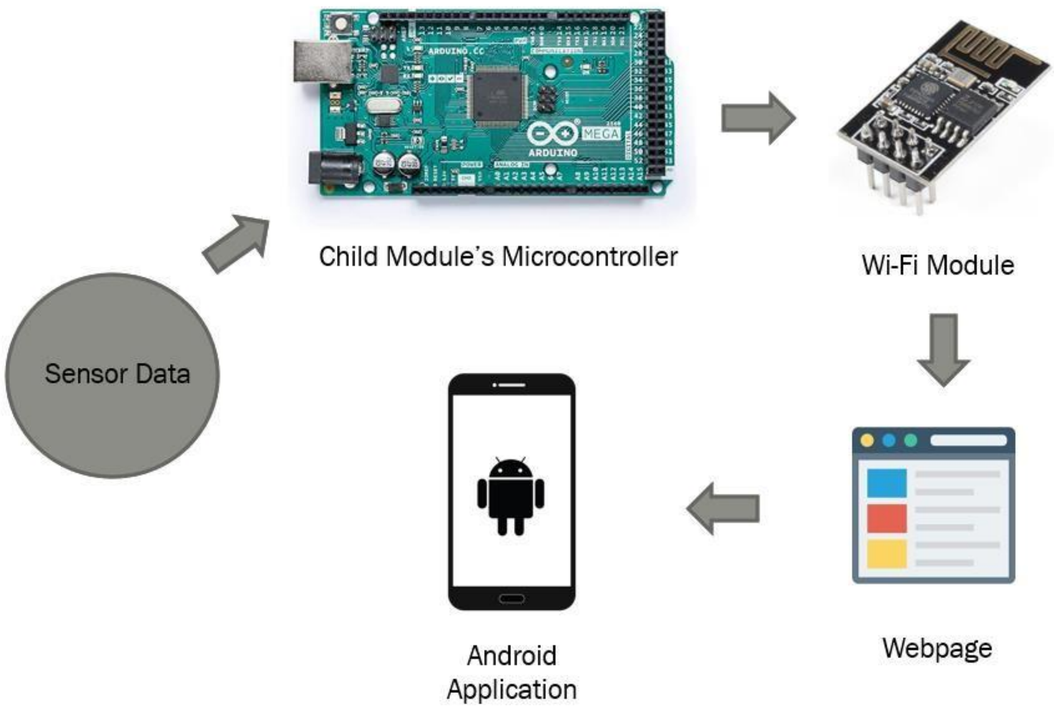

Figure 20: Transmission of Data to the Parent Module

3.3. 3D Printed Wi-Fi Lamp

This module will be a furniture that is kept in the house which changes color according to the activity the child is engaged in, so that parents do not always have to check their phone to see the child's status.

The module works in a similar way to the parent module, after Gathering child's data It is transmitted to the lamp utilizing the Wi-Fi module.

The devices that are included in the lamp are:

Wi-FiModule:
An ESP8266 is used to receive date from the child module.

Figure 21: Wi-Fi module

LCD display:

This display gives a written message of the child’s current activity.

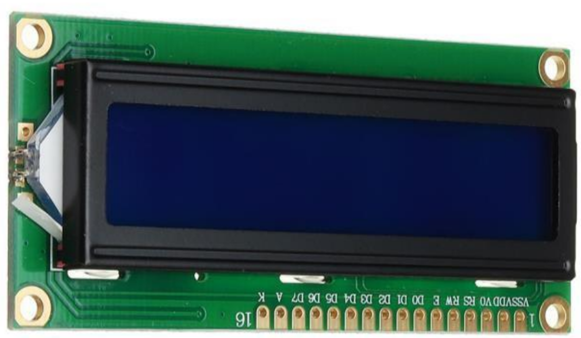

Figure 22: LCD display

RGB LED’s: it is for the lamp to show different color for the different kinds activity the child is engaged in.

Figure 23: RGB LED

Microcontroller:

As the main processing unit for the lamp an Arduino Uno is used.

Figure 24: Arduino Uno Microcontroller

Just like the parent module the lamp works in a similar manner. First the sensors collect the data from the child. The Arduino mega gathers this data and determines the activity the child is doing. The child module's Wi-Fi module sends the activity to the Wi-Fi module of the 3D printed lamp. The Arduino Uno connected with the lamps Wi-Fi module reads the date and turns one of the LED on according to the activity. The flow of the data transfer is represented in *Figure 25* below.

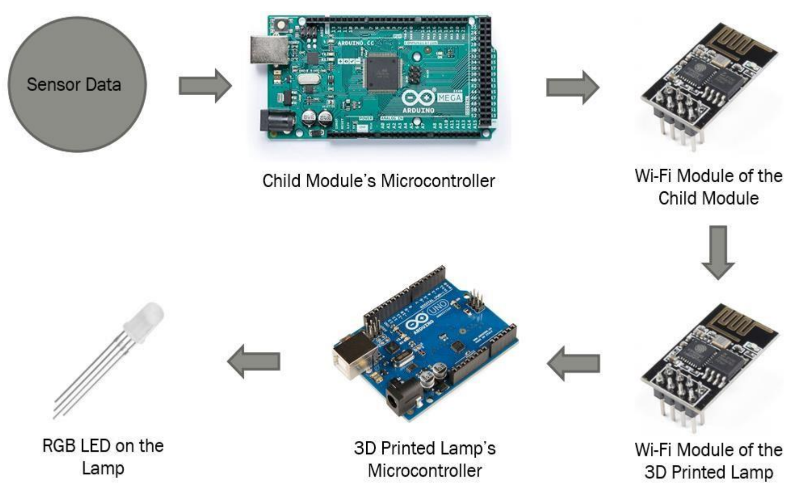

Figure 25: Representation of the 3D printed Lamp

4.0. Result, Analysis and Discussion

Due to the lock down and not enough resources to obtain, all the sensors could not be made to work together. Hence, here each sensor is used separately to show the activity. This section describes the results achieved for each sensor modules.

4.1. Child Module

The Pulse Sensor

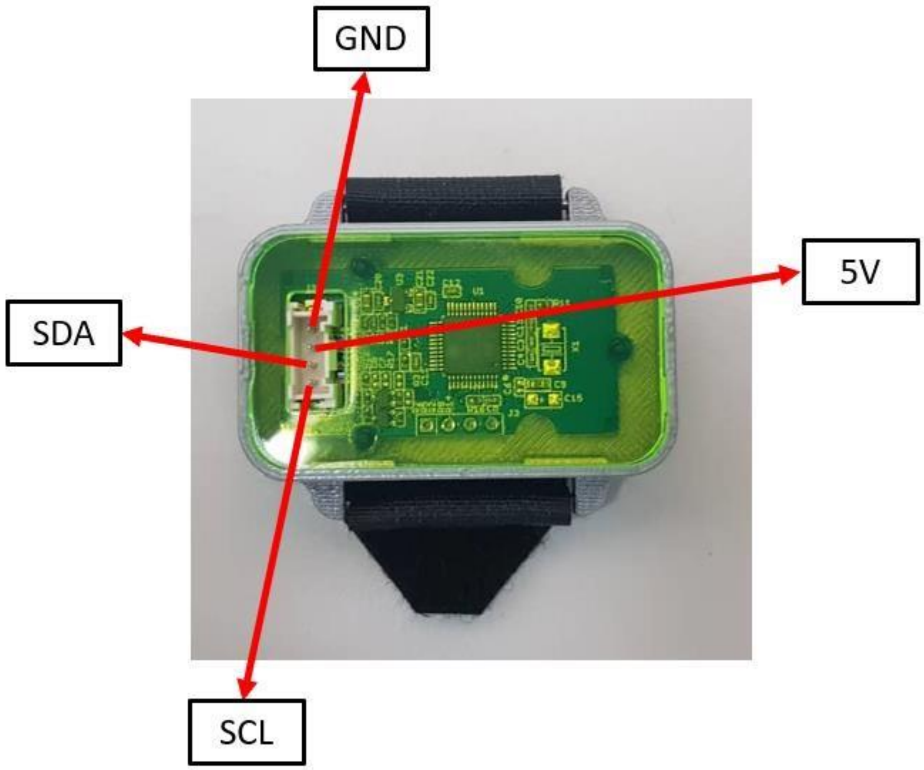

Figure 26: Connections Pins for the Grove Finger clip heart rate sensor

Mentioned on *Figure 26* are the connections for the heart rate sensor these are directly connectedto the Arduino mega board using a breadboard. The first pin is connected to the ground of the Arduino Mega, continued by connecting second pin which is the power source, 5V that is directly connected to the 5V supply from the Arduino Mega. and finally connecting the SDA and SCL pins. The SDA and SCL pins of the sensor are connected to the TX and RX pins off the microcontroller this is to form I^2C (Inter-Integrated Circuit) bus connection. This is used to establish communication between master and slave in this case Arduino Mega is the master and the Grove finger clip heart rate sensor is this slave.

This module can be placed on the wrist and on the forearm to get the heart rate. The heart rate output is in the form of beats per minute.

Figure 27: Placement of Sensor on Forearm

Figure 28: Placement of Sensor on wrist

The heart rate sensor Is used to determine if the child is resting or exercising. Research shows that youngsters heart rate can be as low as 60 beats per minute during sleeping and as high as 220 beats per minute during physical activity [11]. Using this data, the heart rate sensor is Programmed to output:

- ‘Resting’ when heart rate is more than 50 beats per minute and less than 100 beats per minute. Here resting could mean the child is studying or sleeping
- ‘Exercising’ when the heart rate is more than 120 beats per minute and less than 230 beats per minute.

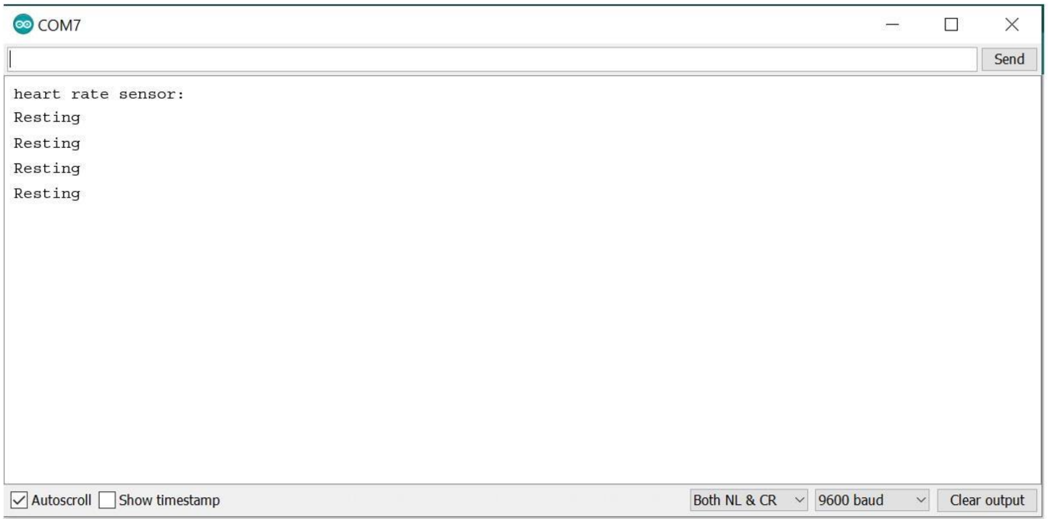

Figure 29:Heart rate sensor's Serial Monitor indicating 'Resting'

Figure 30: Heart rate sensor's Serial Monitor indicating 'Exercising'

Light Sensor

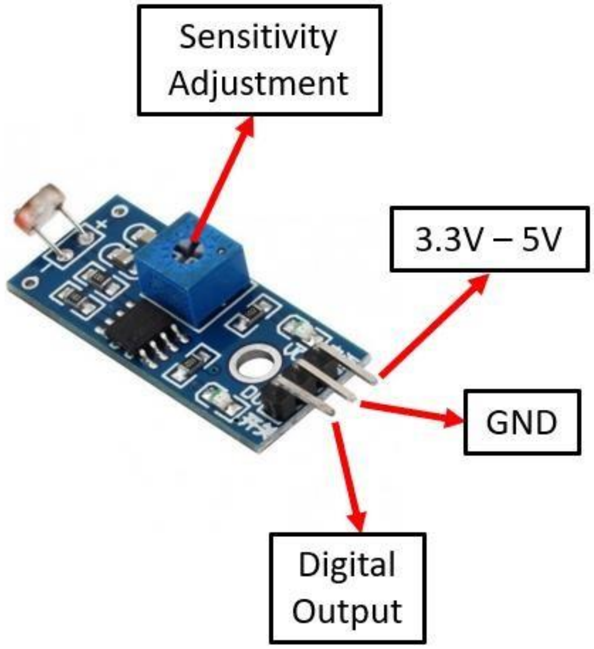

Figure 31: Connections Pins for the Light Sensor

The light sensor can be connected to the 3.3V or the 5 V pin on the Arduino Mega. The middle pin is connected to the ground and the leftmost pin is connected to a digital pin of the Arduino. The sensor can also be altered depending on the intensity of the light by adjusting the sensitivity using a screwdriver.

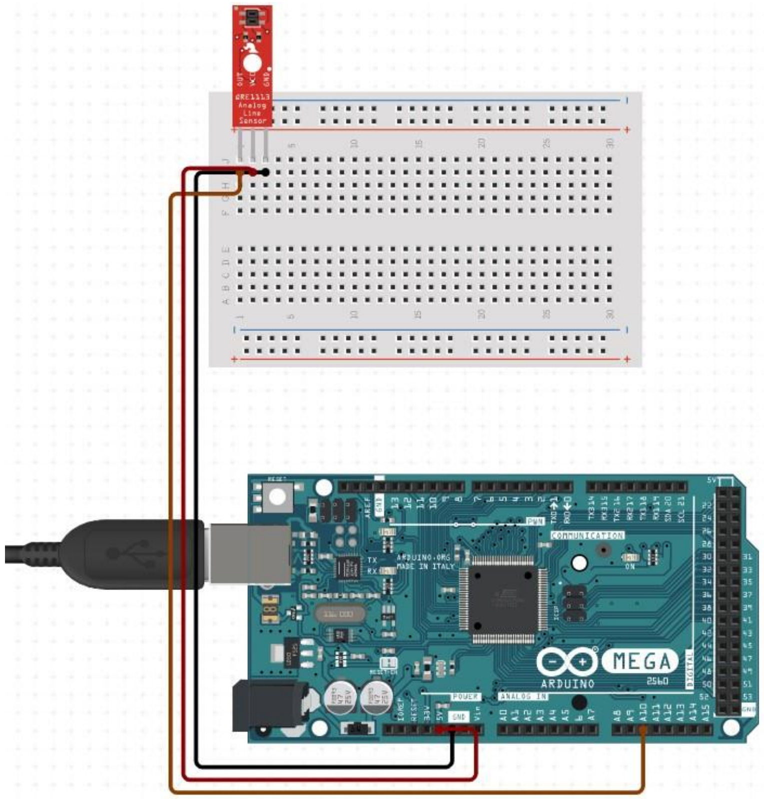

Figure 32: Representation of the connections of Light Sensor to Arduino Mega

The function of the light sensor is to determine if the child is awake or sleeping. This is done by measuring the light intensity surrounding the child. By assuming that the child is awake when there is enough light source near him/her and sleeping when there is limited light source near the child. So, when the lights are on, sensor reading will show child is awake and when lights are off the sensor reading will show child is sleeping. When the sensor outputs ‘awake’ this could either mean child is studying or exercising.

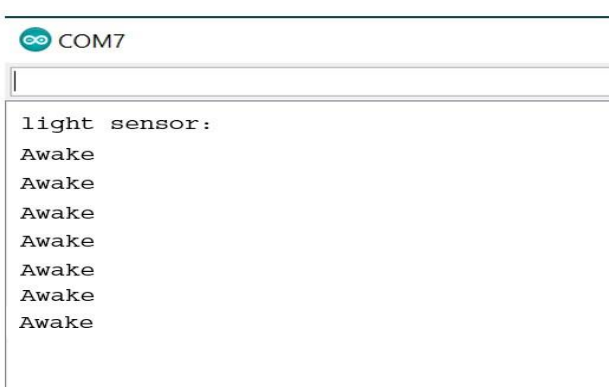

Figure 33:Light sensor's Serial Monitor indicating child is ‘Awake’

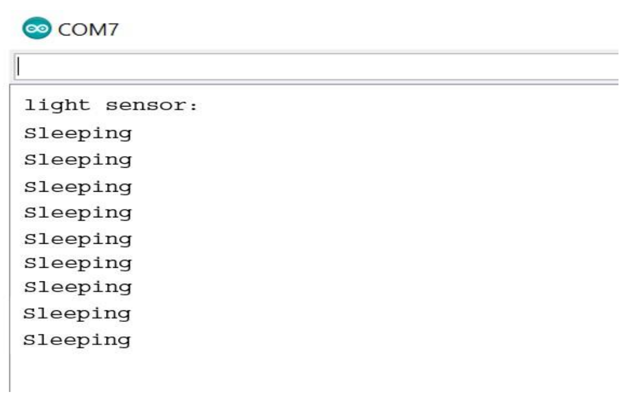

Figure 34: Light sensor's Serial Monitor indicating child is ‘Sleeping’

Accelerometer

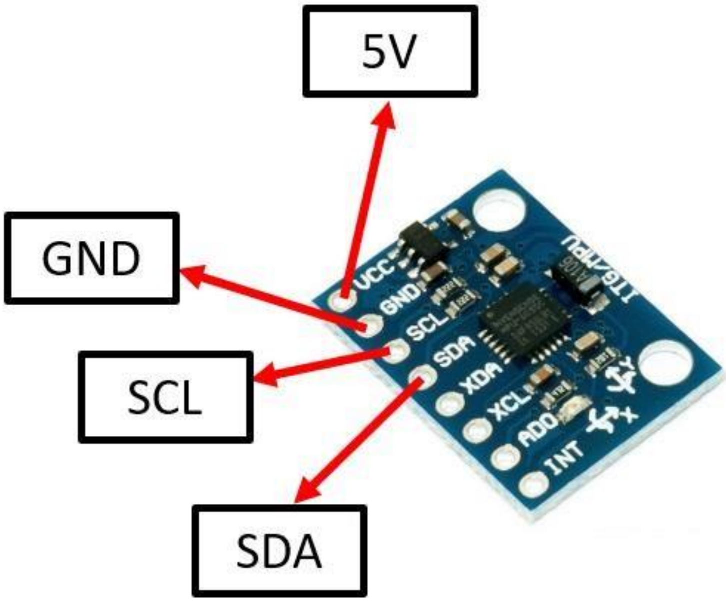

Figure 35: Connecting Pins for the MPU6050

Similar to the heart rate monitor four pins are connected to the MPU. The 5V power supply, the ground, SCL and SDA connections. The SDA and SCL pins of the sensor are connected to the TX and RX pins off the microcontroller this is to form I^2C (Inter-Integrated Circuit) bus connection. This is used to establish communication between master and slave in this case Arduino Mega is the master and the MPU6050 is this slave.

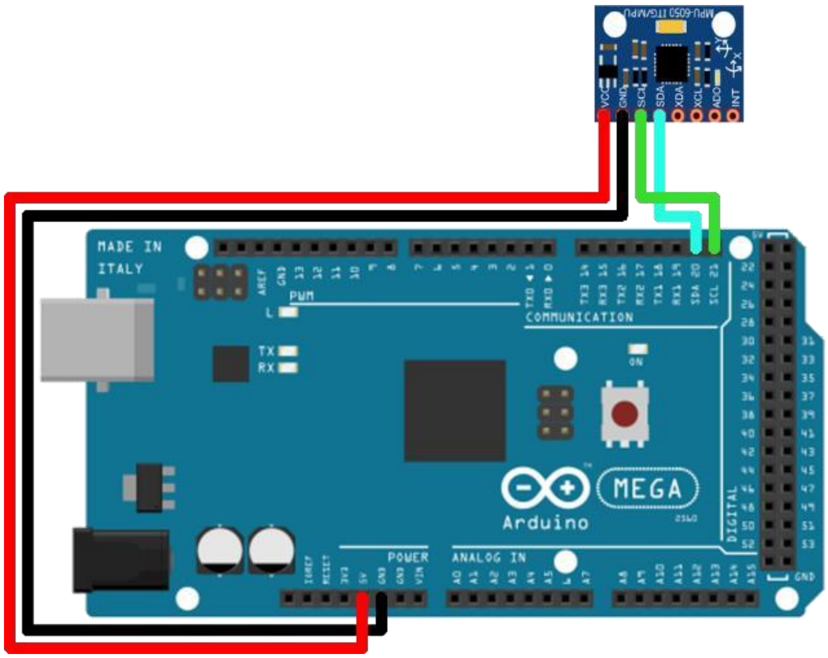

Figure 36: Representation of the connections of MPU6050 to Arduino Mega

The MPU6050 has a built-in accelerometer and a gyroscope. The MPU6050 will be placed horizontally on the child module hence the calibration of the MPU module is done in that position. *Figure 37* shows the representation of the positive axes for the acceleration and gyroscope values after the calibration. The output range of the MPU6050 is -32768 to +32767 for each of the six axes.

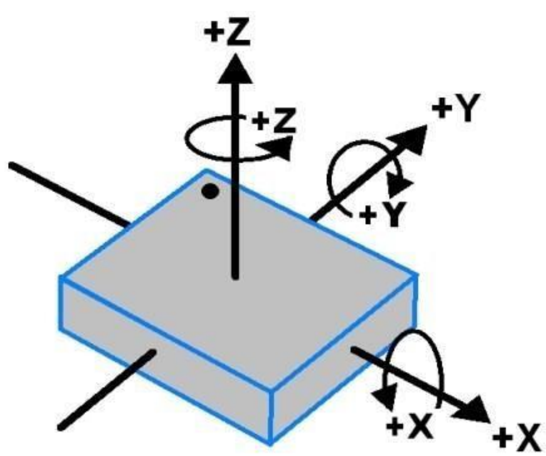

Figure 37: Positive Acceleration and Motion after calibration

The MPU6050 is used to decide if the child is resting or exercising. *Figure 38* shows some of the values when exercising.

a/g:	8040	-1446	11572	20930	-32768	-49
a/g:	14984	3986	14828	16010	-32768	-8
a/g:	12664	-2072	7740	4345	-32768	-213
a/g:	10406	-11844	-1228	-6788	-23483	-197
a/g:	1082	-14178	-5204	-21603	-14702	13
a/g:	3192	-12440	720	-26112	-12456	21
a/g:	2980	-13398	3134	-23280	-10420	-58
a/g:	3494	-8388	2578	-17152	-5563	-153
a/g:	2000	-11974	1264	-9165	1773	-51
a/g:	4674	-8226	4724	-4302	8600	-88
a/g:	8292	-5794	4998	-1372	15127	-84
a/g:	5746	-6572	4958	-1062	18542	-88
a/g:	6474	-7470	2722	-1916	20899	-75
a/g:	9774	-720	9474	-2260	24531	-76
a/g:	6750	-3550	6724	-585	28764	-30
a/g:	15666	5484	15868	3030	32724	-58
a/g:	16148	5994	16360	6279	32767	-78
a/g:	10276	6	10566	9593	32767	7

Figure 38: MPU6050's Serial Monitor representing the Acceleration and gyroscope values

GPS Module

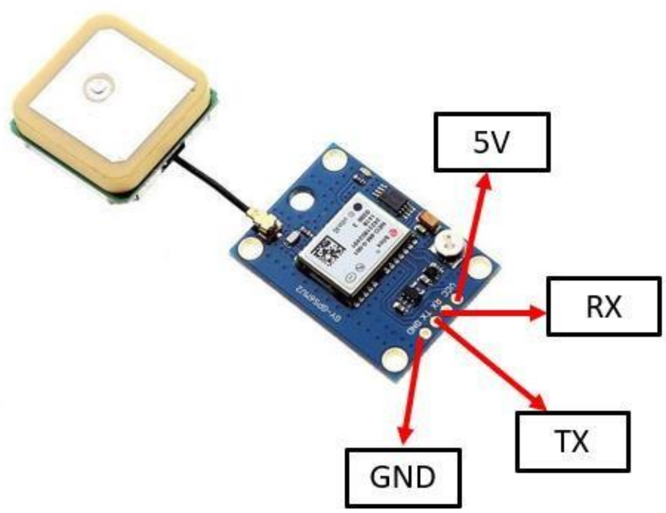

Figure 39:Connecting Pins for the GPS Module

Four connections are needed to operate the GPS module. power supply is through the 5V pin in the Arduino Mega. The RX and TX pins are connected for serial communication (I^2C) And finally the ground.

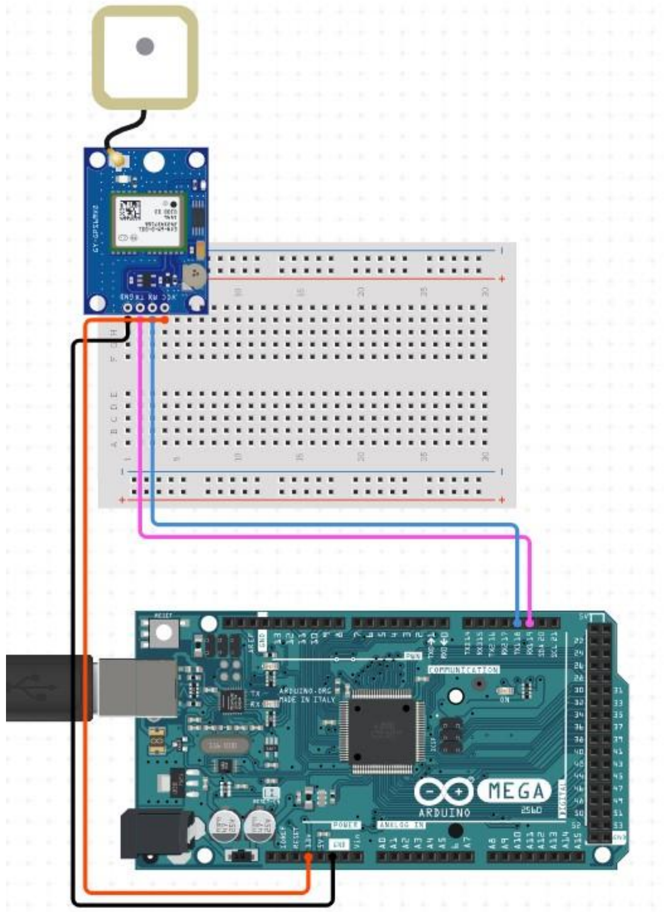

Figure 40: Representation of the connections of GPS module to Arduino Mega

The output of the GPS module provides the latitude and longitudinal coordinates of the child.

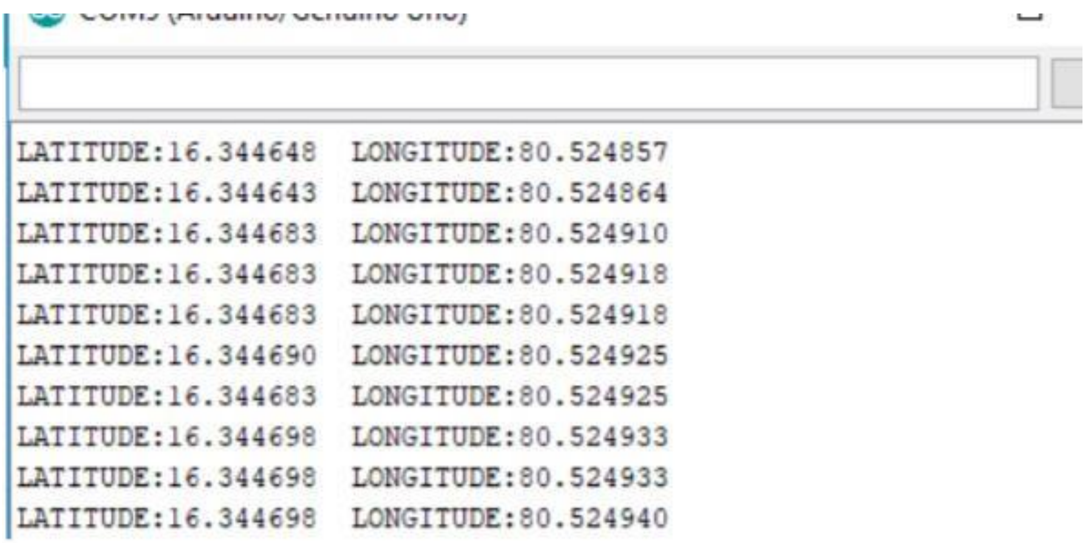

Figure 41: GPS module's Serial Monitor representing in Latitude and Longitude values

Wi-Fi Module

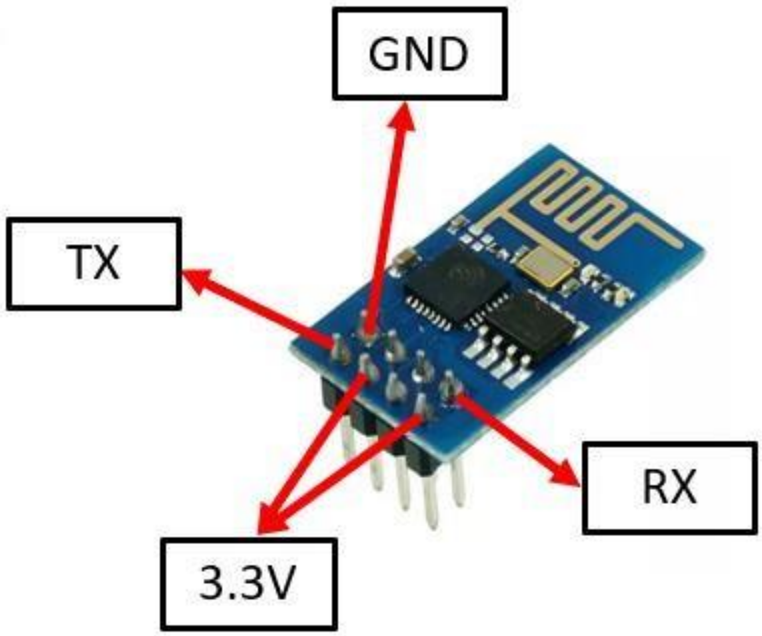

Figure 42: Connecting Pins for the ESP8266 Wi-Fi Module

The ESP8266 has five connections that are made to the Arduino Mega. Starting with the 3.3V pin which is connected to the Vcc and the CH_EN (Chip enabled – Active High) pin. The TX and RX from the ESP8266 is connected to the RX and TX pins of the Arduino Mega. Finally, the ground of that Arduino Mega is connected to the ground of the ESP8266.

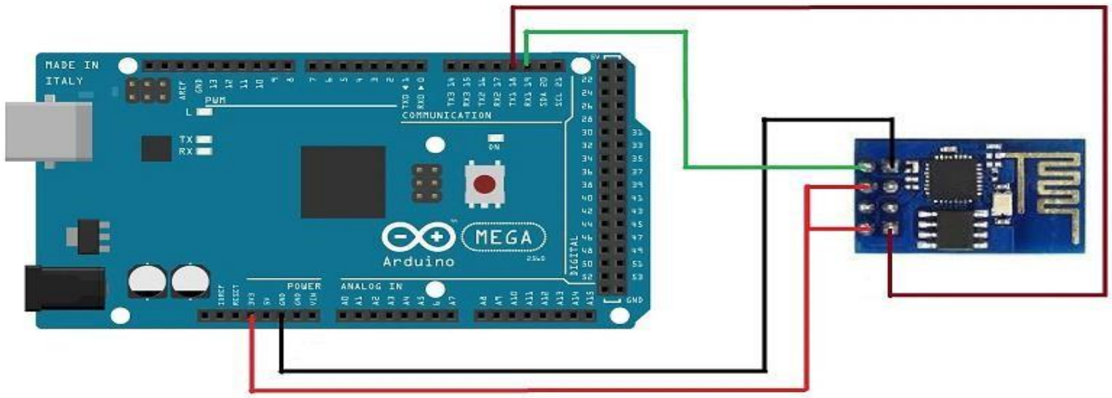

Figure 43: Representation of the connections of ESP8266 to Arduino Mega

After the Arduino Mega collects the data from the sensors it sends it to the Wi-Fi module which will upload the data to a web page. For this the ESP8266 is connected to a network the access tothe internet.

The ESP8266 has a function of its own which allows it to create webpage and provide an IP address to access it. *Figure 44* shows the code to create the webpage which is by using basic HTML.

```
void Send()
{
      webpage = "<h1>Child Monitoring System</h1><body bgcolor=f0f0f0>";//Displays 'Child Monitoring Sysyem' on the page
      webpage = "<h2>Child Status:</h2><body bgcolor=f0f0f0>";//Displays 'Child Status:' on the page
      sendwebdata(webpage);
      webpage=name;
      webpage+=dat;
      sendwebdata(webpage);
      delay(1000);
      webpage = "<h2>Exercising</h2><body bgcolor=f0f0f0>";//webpage will display 'Exercising'
     sendwebdata(webpage);
     client.println("AT+CIPCLOSE=0");
}
```

Figure 44: Code that displays the Webpage

Figure 45 shows the webpage that is accessed through the IP that is provided by the module. The webpage that is created can only be view though a local area network connection.

Figure 45: Webpage Created by ESP8266

4.2. Parent Module

An android application is created to properly represent the data that is displayed on the webpage created by the Wi-Fi module. This application is mainly for the use of the parents, it is built for easy access. The information displayed on the application is organized and easy to interpret.

The application is a representation of the webpage that is created by the ESP8266, this is so that the webpage is more mobile friendly and effortless to use. The application is created by using basic Java on Android Studio platform.

Figure 46:Screenshot of the Android App

4.3. 3D Printed Wi-Fi Lamp

A simulation was created that would demonstrate different colors of the LED for the different activities the child is engaged in.

For this simulation, an RGB LED, a breadboard, switches, a resistor, and an Arduino Uno microcontroller are used. A circuit is made by connecting the three-color terminals of the LED to a switch each. All the switches are connected to a power supply from the Arduino Uno. A resistor is placed between the ground pin of the LED and the ground of the Arduino Uno so that the LED does not overpower and burn out. The LED activity representation is as follows:

- Red LED is displayed to exhibit Exercising
- Blue LED is displayed to exhibit Sleeping
- Green LED is displayed to exhibit Studying

Representation of the LED circuits are shown below.

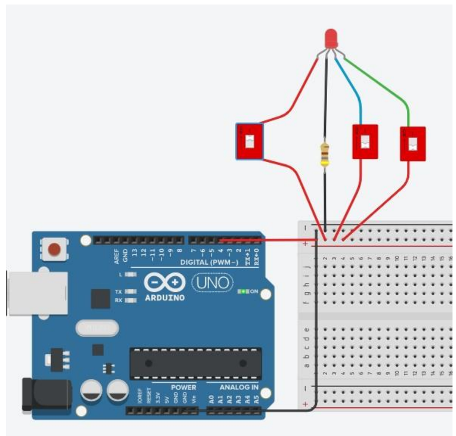

Figure 47: Red LED is on when Exercising

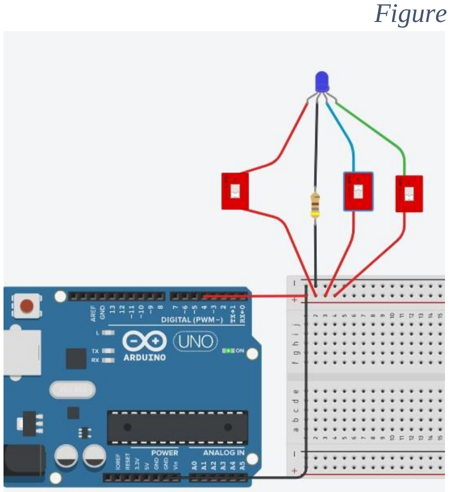

Figure 49:Blue LED is on when Sleeping

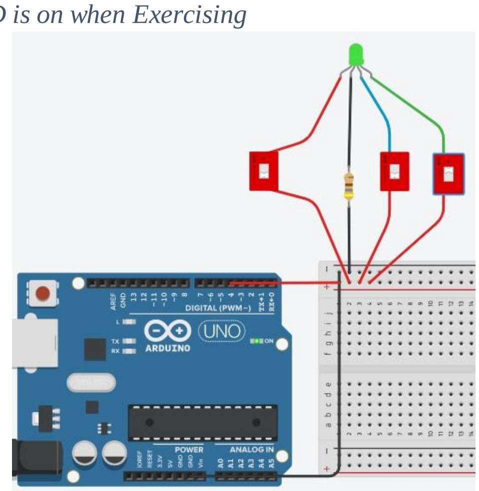

Figure 48: Green LED is on when Studying

5.0. Conclusion and Future Work

Overall, the project met the standards of the objectives in the plan and the aims were achieved. Even though there are many similar systems available at present, these systems use different technology. However, these systems have some limitations such as monitoring only one aspect either the safety or the health of the child. The designed project overcomes this issue by providing safety and keeping the child healthy so that, parents can have complete track of the child's activity and location.

After looking at the results and the analysis of this project we can assume that there are a few limitations considering it.

The first limitation would be that the complete child module consists of seven components this would end up being a huge module when all the components are combined together. Hence, causing a size issue. The child cannot be carrying a huge device with him/her all the time, this would be a hassle and the child would lose his/her patience. For this reason, the child module should be miniaturized so that it can be utilized as a wearable technology for the child e.g. a watch.

Secondly the device proposed is only capable to working in a local area network. Which means that if any one of the devices is connected to a different network the device will now be able to function. The data will not be able to transfer to it. The only device that will be able to operate when switched to another network is the child module as this is the main hub which transmits the data to the rest of the modules. To overcome this issue a GSM module can be introduced, so that the data can be transferred through a mobile network connection.

Even though there are some limitations in the project it can be improved and involved to incorporate some new ideas to the system. Some of the new ideas can be to implement a system that monitors the sleep patterns of the child using the sensors and uploads it to the application in form of a graph, so that parents can keep track of the sleeping habits of the child.

Furthermore, using the accelerometer and the GPS the child’s steps can be counted and the distance moved can be calculated and presented so parents can have a wider idea of the child’s health.

Hence, this device will be suitable for protecting the child by keeping him/her safe and health without troubling the parents.

References

[1] “Missing Children’s Statistics”. globalmissingkids.org https://globalmissingkids.org/awareness/missing-children-statistics/#

[2] Matt Berger. “Children Worldwide Aren’t Getting Enough Exercise”. Helthline.com https://www.healthline.com/health-news/children-around-the-world-dont-get-enoughexercise
(assessed Nov. 20, 2019)

[3] “Commission on Ending Childhood Obesity”. who.int https://www.who.int/end-childhood-obesity/facts/en/
(assessed Sep. 23, 2019)

[4] R. Kamalraj and M. Sakthivel, "A Hybrid Model on Child Security and Activities Monitoring System Using IoT," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, 2018, pp. 996-999, doi: 10.1109/ICIRCA.2018.8596771.

[5] A. Moodbidri and H. Shahnasser, "Child safety wearable device," 2017 International Conference on Information Networking (ICOIN), Da Nang, 2017, pp. 438-444, doi: 10.1109/ICOIN.2017.7899531.

[6] U. Chowdhury et al., "Multi-sensor Wearable for Child Safety," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York City, NY, USA, 2019, pp. 0968-0972, doi: 10.1109/UEMCON47517.2019.8992950.

[7] J. Saranya and J. Selvakumar, "Implementation of children tracking system on android mobile terminals," 2013 International Conference on Communication and Signal Processing, Melmaruvathur, 2013, pp. 961-965, doi: 10.1109/iccsp.2013.6577199.

[8] A. Kakkar and A. Garg, "Pervasive Health Monitoring of Special Child using IoT and

Cloud Technologies," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 98-102, doi: 10.1109/ICCS45141.2019.9065375.

[9] "LeapBand™". Leapfrog.com https://store.leapfrog.com/en-us/store/p/leapband/_/A-prod19263

[10] "X-Doria's KidFit is a low-cost fitness wearable for five-year-olds". Engadget.com https://www.engadget.com/2014-06-19-xdoria-kidfit-activitytracker.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAACN4iyntkgtmnNLb08UaQsfXB3isOH7rfFdUDCBJughQI2BTaJMm3_GwQKY_JcvFggUZX_5hw2F-_kjWhpONCBrYKAHnyzJm0hJsfLPLkX-KwE117FvUk2H3ooEHfCKGHgLjd1oK22iwzjHhoMD7Ud3Be8IDcDndiiprXcmMRzh
(assessed Jun. 19, 2014)

[11] "Are You Pushing Your Child's Heart Rate Up Too High?". Clevelandclinic.org https://health.clevelandclinic.org/pushing-childs-heart-rate-high/
(assessed Oct 15, 2015)

Appendix

Heart rate sensor codes

```
#include <Wire.h>
void setup() {
    Serial.begin(9600);
    Serial.println("heart rate sensor:");
    Wire.begin();
}
void loop() {
    Wire.requestFrom(0xA0 >> 1, 1);    // request 1 bytes from slave device
    while(Wire.available()) {          // slave may send less than requested
        unsigned char c = Wire.read();   // receive heart rate value (a byte)

        if ((c >= 120)&&(c <= 230)){
          Serial.println("Exercising");// print Exercising
        }
          else if ((c >= 50)&&(c <= 110){
            Serial.println("Resting");// print Resting
        }

    }
    delay(500);
}
```

MPU6050 Codes

```
#include "Wire.h" // This library allows you to communicate with I2C devices.
const int MPU_ADDR = 0x68; // I2C address of the MPU-6050. If AD0 pin is set to HIGH, the I2C address will be 0x69.
int16_t accelerometer_x, accelerometer_y, accelerometer_z; // variables for accelerometer raw data
int16_t gyro_x, gyro_y, gyro_z; // variables for gyro raw data
int16_t temperature; // variables for temperature data
char tmp_str[7]; // temporary variable used in convert function
char* convert_int16_to_str(int16_t i) { // converts int16 to string. Moreover, resulting strings will have the same length in the debug monitor.
  sprintf(tmp_str, "%6d", i);
  return tmp_str;
}
void setup() {
  Serial.begin(9600);
  Wire.begin();
  Wire.beginTransmission(MPU_ADDR); // Begins a transmission to the I2C slave (GY-521 board)
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0); // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
}
void loop() {
  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H) [MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2, p.40]
  Wire.endTransmission(false); // the parameter indicates that the Arduino will send a restart. As a result, the connection is kept active.
  Wire.requestFrom(MPU_ADDR, 7*2, true); // request a total of 7*2=14 registers

  // "Wire.read()<<8 | Wire.read();" means two registers are read and stored in the same variable
  accelerometer_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x3B (ACCEL_XOUT_H) and 0x3C (ACCEL_XOUT_L)
  accelerometer_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x3D (ACCEL_YOUT_H) and 0x3E (ACCEL_YOUT_L)
  accelerometer_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x3F (ACCEL_ZOUT_H) and 0x40 (ACCEL_ZOUT_L)
  temperature = Wire.read()<<8 | Wire.read(); // reading registers: 0x41 (TEMP_OUT_H) and 0x42 (TEMP_OUT_L)
  gyro_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x43 (GYRO_XOUT_H) and 0x44 (GYRO_XOUT_L)
  gyro_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x45 (GYRO_YOUT_H) and 0x46 (GYRO_YOUT_L)
  gyro_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x47 (GYRO_ZOUT_H) and 0x48 (GYRO_ZOUT_L)
```

```
  // print out data
  Serial.print("aX = "); Serial.print(convert_int16_to_str(accelerometer_x));
  Serial.print(" | aY = "); Serial.print(convert_int16_to_str(accelerometer_y));
  Serial.print(" | aZ = "); Serial.print(convert_int16_to_str(accelerometer_z));
  // the following equation was taken from the documentation [MPU-6000/MPU-6050 Register Map and Description, p.30]
  Serial.print(" | tmp = "); Serial.print(temperature/340.00+36.53);
  Serial.print(" | gX = "); Serial.print(convert_int16_to_str(gyro_x));
  Serial.print(" | gY = "); Serial.print(convert_int16_to_str(gyro_y));
  Serial.print(" | gZ = "); Serial.print(convert_int16_to_str(gyro_z));
  Serial.println();

  // delay
  delay(1000);
}
```

Light Sensor Codes

```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}
void setup() {
  //setup the input pins
  pinMode(Analog, INPUT);
}
void loop() {
  // reads the input on analog pin A0 (value between 0 and 1023)
  int analogValue = analogRead(A0);

  Serial.print("Analog reading = ");
  Serial.print(analogValue);   // the raw analog reading

  // We'll have a few threshholds, qualitatively determined

  if (analogValue < 200) { //threshhold <200 is dark
    Serial.println("Sleeping");
  } else if (analogValue < 800) { //threshhold <800 is bright
    Serial.println("Awake");
  }
  delay(500);
}
```

GPS Module

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

static const int RXPin = 4, TXPin = 3;
static const uint32_t GPSBaud = 4800;

// The TinyGPS++ object
TinyGPSPlus gps;

// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);

void setup()
{
  Serial.begin(115200);
  ss.begin(GPSBaud);
}

void loop()
{

  while (ss.available() > 0)
    if (gps.encode(ss.read()))
      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
    while(true);
  }
}

void displayInfo()
{
  if (gps.location.isValid())
  {
    Serial.print("LATITUDE");
    Serial.print(gps.location.lat(), 6);// shows the latitude coordinated
    Serial.print(F(","));
    Serial.print("LONGITUDE");
    Serial.print(gps.location.lng(), 6);// shows the longitude coordinated
  }
  else
  {
    Serial.print(F("INVALID"));
  }

  Serial.println();
}
```

<u>Wi-Fi Module (ESP8266)</u>

```
#include<SoftwareSerial.h>
SoftwareSerial client(2,3); //RX, TX

String webpage="";
int i=0,k=0;
String readString;
int x=0;

boolean No_IP=false;
String IP="";
char temp1='0';

String name="<p>Circuit Digest</p>";   //22
String dat="<p>Data Received Successfully.....</p>";     //21

void check4IP(int t1)
{
  int t2=millis();
  while(t2+t1>millis())
  {
    while(client.available()>0)
    {
      if(client.find("WIFI GOT IP"))
      {
        No_IP=true;
      }
    }
  }
}
```

```
void get_ip()
{
  IP="";
  char ch=0;
  while(1)
  {
    client.println("AT+CIFSR");
    while(client.available()>0)
    {
      if(client.find("STAIP,"))
      {
        delay(1000);
        Serial.print("IP Address:");
        while(client.available()>0)
        {
          ch=client.read();
          if(ch=='+')
          break;
          IP+=ch;
        }
      }
      if(ch=='+')
      break;
    }
    if(ch=='+')
    break;
    delay(1000);
  }

  Serial.print(IP);
  Serial.print("Port:");
  Serial.println(80);
}

void connect_wifi(String cmd, int t)
{
  int temp=0,i=0;
  while(1)
  {
    Serial.println(cmd);
    client.println(cmd);
    while(client.available())
    {
      if(client.find("OK"))
      i=8;
    }
    delay(t);
    if(i>5)
    break;
    i++;
  }
  if(i==8)
  Serial.println("OK");
  else
  Serial.println("Error");
}
```

```
void wifi_init()
{
        connect_wifi("AT",100);
        connect_wifi("AT+CWMODE=3",100);
        connect_wifi("AT+CWQAP",100);
        connect_wifi("AT+RST",5000);
        check4IP(5000);
        if(!No_IP)
        {
          Serial.println("Connecting Wifi....");
          connect_wifi("AT+CWJAP=\"Santhosh",\"0123456789"",7000);
       // connect_wifi("AT+CWJAP=\"vpn address\",\"wireless network\"",7000);
        }
        else
          {
          }
        Serial.println("Wifi Connected");
        get_ip();
        connect_wifi("AT+CIPMUX=1",100);
        connect_wifi("AT+CIPSERVER=1,80",100);
}

void sendwebdata(String webPage)
{
     int ii=0;
      while(1)
      {
       unsigned int l=webPage.length();
       Serial.print("AT+CIPSEND=0,");
       client.print("AT+CIPSEND=0,");
       Serial.println(l+2);
       client.println(l+2);
       delay(100);
       Serial.println(webPage);
       client.println(webPage);
       while(client.available())
       {
         //Serial.print(Serial.read());
         if(client.find("OK"))
         {
           ii=11;
           break;
         }
       }
       if(ii==11)
       break;
       delay(100);
      }
}

void setup()
{
   Serial.begin(9600);
   client.begin(9600);
   wifi_init();
   Serial.println("System Ready..");
}

void loop()
{
  k=0;
  Serial.println("Please Refresh your Page");
  while(k<1000)
  {
```

```
     k++;
    while(client.available())
    {
     if(client.find("0,CONNECT"))
     {
       Serial.println("Start Printing");
       Send();
       Serial.println("Done Printing");
       delay(1000);
     }
   }
   delay(1);
  }
}


void Send()
{
     webpage = "<h1>Child Monitoring System</h1><body bgcolor=f0f0f0>";//Displays 'Child Monitoring Sysyem' on the page
     webpage = "<h2>Child Status:</h2><body bgcolor=f0f0f0>";//Displays 'Child Status:' on the page
     sendwebdata(webpage);
     webpage=name;
     webpage+=dat;
     sendwebdata(webpage);
     delay(1000);
     webpage = "<h2>Exercising</h2><body bgcolor=f0f0f0>";//webpage will display 'Exercising'
    sendwebdata(webpage);
    client.println("AT+CIPCLOSE=0");
}
```

Android Application Code

```java
package com.example.webviewdemoapp;

import ...

public class MainActivity extends AppCompatActivity {

    private WebView webView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        webView = findViewById(R.id.webView);
        webView.setWebViewClient(new WebViewClient());
        webView.loadUrl("203.217.131.2");// calls the webpage using IP from the WIFI module

        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);
    }
```

```java
@Override
public void onBackPressed(){
    if (webView.canGoBack()){
        webView.goBack();
    } else {
        super.onBackPressed();
    }

}
```