CHAPTER 1

PREAMBLE

Cradles provide a safe and comfortable sleeping environment for infants, promoting healthy development and regular sleep patterns. They offer a secure and familiar space for babies to rest and play, and can also be an important cultural symbol representing parental love and care. The scope of this project islands the gap between the parent and the child.

Introduction

We are all too familiar with the phrase IOT, which refers to internet-based impacts that are defined for being less time-consuming as a result of making work simple and accurate. We will create a cradle that will relieve parents' stress and, most importantly, will be safe and secure for the child as far as time and safety are concerned amid the support of "IOT". Therefore, organizing the work on time and obtaining baby care are truly crucial variables. The cradle system will allow parents necessary time to parent for relaxation, as if the parents both the mother and the father were in the same room with the child.

Relevance of the Project

Time management and providing for the needs of the infant are unquestionably crucial considerations. As if the mother and father were to surround the child, the cradle system will allowparents crucial time to parent for relaxation. So, even if a baby has urinated and no one has noticed for a while, there is no need to panic since the cradle system will also notify you to the presence of dust in the cradle. Additionally, the Cradle system has the capacity to characterize it and send a warning if the infant develops a fever or cold.

Our cradle may also be used to refer to any disturbance caused by a stir detector in the cradle for the baby's safety. The proposed approach would assist the parents in taking proper care of their child. Sometimes, it becomes really difficult to spot the potholes on theroads during rainy times as they get full of rainwater and hence it becomes difficult for usto acknowledge them.

- The goal of this endeavor is to create a clever, secure, and advantageous cradle that will undoubtedly make them comfortable. PIR detectors, noise detectors, humidity detectors,
- servo motors, and temperature detectors are the major circuits utilized in this project. The PIR detector measures the baby's movement and sends a signal to the servo motor to cause the cradle to swing back and forth.
- It was intended to implement a smart cradle system that uses cloud services to measure the baby's temperature, gauge the wetness of the mattress, and attach a PIR detector to the cradle. As a result, the design bridges the distance between an infant and the working parent.
- This design's goal is to minimize the physical contact between working parents and their environments while enhancing security, cost effectiveness, and rigidity. The entire system is designed to provide convenience by constantly monitoring every activity of the youngster and providing real-time information and updates to the parent

Scope of the Project

This project proposes the idea of automatic caretaker room for a baby. The concept of an automated infant caretaker chamber is put forth in this proposal. The major goal of this concept is to save parents' hectic schedules time and energy. These days, working people are highly busy. They do not have enough time to provide their infants the attention they need. The entire space is therefore configured such that it can detect the baby's activity and function as needed. Parents can save time and energy by not having to check on their child repeatedly until they receive no information on the child. Using sensors and a microcontroller, this scenario's notion is realized. The sensors that are connected to the microprocessor sense the environment in the room and keep tabs on the infant's behavior. It functions according to the conditions we place on it. The user will be able to see all the data and be informed of the baby's state.

The Smart Cradle System is a project that aims to create a cutting-edge, technologically sophisticated cradle that includes a number of smart elements to give newborns a secure and comfortable resting environment. The project's scope includes creating a cradle with sensors integrated in to track the baby's vital indicators, such as heart rate, respiration, crying, wetness and notify parents or other caretakers if something seems off. The device will also come with a smartphone app to track the infant's sleeping habits remotely and offer tailored sleep advice. With the Smart Cradle System, parents and caretakers will have a dependable and effective tool to ensure their baby's safety and wellbeing while also revolutionizing how infants sleep

Problem Definition

- Traditional baby cradles have the drawback of having few elements to assure the security and comfort of newborns.
- Anxiety and tension can result from parents and caretakers finding it difficult to keep track of the baby's vital signs, such as breathing, heart rate, crying and wetness. Traditional cradles do not offer individualized sleep recommendations, which makes it difficult for parents to establish consistent sleep patterns for their infants.
- Furthermore, it might be challenging to control the temperature and make the baby's environment comforting.
- These restrictions have the drawback of potentially compromising the baby's safety and wellbeing and elevating stress levels in parents and other caretakers. Hence, a solution must be devised for the same.

Problem Explanation

- The system is Arduino grounded that's being designed is aimed to help parents and nourish in child’s care.
- A sound sensor is connived to the regulator which senses sound when baby cries and activates the regulator with its digital affair.
- Temperature and moisture will be covered through detectors and addict will be controlled.
- Humidity Sensor Interfaced for wet discovery.
- Wi- Fi interface sends alert to Phones to get the attention of parent’s nursery.
- An ALCD is connived to the regulator which keeps displaying the status as messages.

The Objective of the Study

The some of the main aspects which we now are working on to achieve are:

- To design and development of a smart baby cradle, the two main things were are concentrating are, the ability to monitor moisture level of bed and emotions of baby.
- Identifying smelly diaper condition and body temperature.

- To make a baby cradle safe and comfortable for the baby with the use of various sensorslike swinging movement.
- Monitor the baby's life, to detect heart rate, bed-wet conditions to keep the baby away froman unhygienic environment.
- To make cradle innovation that is more flexible and less expensive to the Indian market.

There are various other aspects to which we are focused on and are been listed below:

- Designing a system which monitors baby body temperature and increase the pillow temperature according to mother’s body temperature and intimation in times of baby cry.
- Intimation to mobile phone.
- Baby bed wet Detection and Intimation.
- Intimation in times of baby cry.
- Intimation through mobile phone.
- Baby cry detection through Expression detection using machine Learning.
- Vibration sensor to detect the baby fall.
- Whenever the baby cries the cradle will swing.
- Based on the emotions (child crying) message will be sent to the parents.

Existing System

Our aim is to design the smart cradle with today’s advance technologies by doing so eliminates all the drawback present in the previous versions of smart cradle.
Some of the features of the older version has been listed down:

- Mechanical Systems for Pillow Management.
- No Synchronization between child and mother health parameters.
- Short Range i.e. Implementations.
- Costly Implementations.
- It needs GSM connectivity throughout for sending alerts.
- The hygiene condition of the cradle should be taken care of by the parent themselves.

- Although the cradle is capable enough to monitor the baby but it does not comfort the baby as
- much as mother does. Components of cradle are fragile and it needs to be handled carefully.
- Severe damage to the components of the cradle may increase the risk of baby getting a shock.

Our system provides a better timeframe where all the above listed features are handles carefully and advancement of components which are latest and better in all the aspects when compared with the older or the existing system.

Limitations of Existing Systems

- Although the baby can be monitored by the cradle, it cannot provide the same level of comfort.
- To send warnings, it must be connected to GSM at all times.
- The parents themselves should take care of the cradle's hygienic needs without alerts.
- The cradle's parts must be handled with care because they are delicate.
- The likelihood that the baby will receive an electric shock may increase if the cradle's parts suffer severe damage.
- Manual Operated Systems which can be no swing movement, no camera and other various aspect like alerts and sensor monitoring is outdated.
- With all the other factors lead to own the older smart cradle now becomes **costly** and hard to maintain for longer life.

Proposed System

Our system is developed with Arduino and IOT based that is being designed in aimed to help parents in home and nurses in hospital in infant's care.
Features being:

- System starts playing mothers voice automatically when baby cry and stops till the baby stops crying.
- A sound detector is interfaced to the controller which senses sound when baby cries and activates the controller with its digital output.

- Sounds an alarm when mattress gets wet.
- A temperature sensor kept under the bottom cover where the baby sleeps can sense the temperature all time and sends analog signals to the inbuilt ADC of the Arduino controller.
- The digital data can be continuously monitored. A reduction in temperature indicates the wetness in the cover. The controller can be made to activate an alarm, so that his/her cover be changed.
- Sounds an alarm if baby cries for more than a stipulated time indicating that baby needs attention.
- Wi-Fi interface sends alert to android based handsets to get the attention of parents/nurses.
- An ALCD is interfaced to the controller which keeps displaying the status as messages.
- Whenever the baby cries the cradle will swing.
- A sound detector is interfaced to the controller which senses sound when baby cries and activates the controller with its digital output.
- Temperature and Humidity will be monitored through sensors and Fan Will Be Controlled
- Moisture Sensor Interfaced for wet detection
- Wi-Fi interface sends alert to Phones to get the attention of parents/nurses.
- An ALCD is interfaced to the controller which keeps displaying the status as messages**.**
- Automated System, no manual attention required all the time
- Easy to implement

CHAPTER 2

LITERATURE SURVEY

This chapter provides an insight into a few of the papers which were taken as references for the implementation of the project.

2.1 Literature Survey

Case Study – 1: A Wireless Based Real-time Patient Monitoring System.

Authors: Sowmyasudhan S, Manjunath S.

Publications: International Journal of Scientific & Engineering Research

Abstract: The paper demonstrated gives the complete idea on a system which uses four smart sensors to detect the respective four pathological parameters of a baby or a person and interfacing the cell phone to this system to alarm the doctors. The smart sensors sense the biomedical signal of the subject under study (here subject means a baby or a person with pathological disorders) and executes with the controlled coding of the mother processor that is microcontroller and further it hooks the wireless cell phone and using the application software, TEXT application protocol of the cell phone to alarm the respective doctors. The ubiquity of implementation of the wireless standard has been an advantage to both consumers, who may benefit from the ability to roam and switch carriers without replacing phones, and also to network operators, who can choose equipment from many wireless equipment vendors. Wireless also pioneered low-cost implementation of the short message service (SMS), also called text messaging, which has since been supported on other mobile phone standards as well. The standard includes a worldwide emergency telephone number feature. So, the main aim of this project is to get information about the condition of the critical subject which needs to be monitored round the clock, which is in ICU through wireless. The wireless Transmitter sends the details about incubator's temperature, voice level, heart rate and its movement to the destination point which is a doctor’s mobile and these details are displayed on the mobile scree n in the form of SMSs, so that the doctor can analyze the condition from the place where he is sitting. This is the most basic foundation in the area of telemedicine application.

Case Study – 2: Development of Wireless Monitoring System for NICU.

Authors: Joshi, R.K. Kamat, P.K. Gaikwad.

Publications: International Journal of Advanced Computer Research.

Abstract: The research paper depicts a Development of a Wireless Monitoring System for Neonatal Intensive Care Unit (NICU); which is an isolated room for a premature/weak new-born baby. It provides the environmental condition as its mother's belly. Lack of attention to thermoregulation continues to be a cause of unnecessary deaths in the neonatal population. Maintaining a stable body temperature is essential to ensure optimal growth of premature and weak infants. As the temperature and humidity parameters play a vital role during the development of premature weak infants, this research work develops a wireless system which continuously monitors these parameters inside the NICU. The system deploys a set of suitable sensors for the system development. The analogue signals from sensors are processed using a Peripheral Interface Controller (PIC) microcontroller and further transmitted towards the receiving end with the help of Global System for Mobile Communications (GSM) modem using Application Terminal (AT) commands.

Case Study – 3: Dynamic Programming Approach for Newborn's Incubator Humidity Control

Authors: Djaaffar Bouattoura, Pierre Villon, and Gilbert Farges

Publications: IEEE

Abstract: The anatomy, physiology, and biochemistry of the human skin have been studied for a long time. A special interest has been shown in the water permeability of the premature infant's skin, which is known to be an important factor in the maintenance of a controlled water and heat balance. The rate of evaporative heat exchange between the skin surface of a very premature infant and the surrounding incubator air may be so high that evaporative heat loss alone may exceed the infant's total metabolic heat production. However, it has been demonstrated in several investigations published in recent years that basal evaporative water loss can be consistently reduced by increasing the ambient humidity. Nevertheless, the passive humidification system (water reservoir) used in most incubators cannot achieve high and steady humidity levels. Here, the authors propose an active humidification system. The algorithm is based on a combination of optimal control theory and dynamic programming approach.

The relative-humidity (R.H.) regulation is performed in range of 35-90% at 33/spl deg/ C with small oscillations (/spl plusmn/0.5% R.H.) around the reference value (i.e., prescradleed R.H.).

Case Study – 4: Supporting inspection strategies through palpable assemblies.

Authors: Erik Gronvall, Luca Piccini, Alessandro Pollini, Alessia Rullo, Giuseppe Andreoni.

Publications: University of Siena, Communication Science Department.

Abstract: This paper reports an early study on inspection strategies of high- risk systems using ambient computing technologies. Traditionally, the main goal of ambient, pervasive and ubiquitous computing applications is to make the technology transparent or invisible for the users. However, this sort of technological disappearance is not always desirable in particular in presence of any failure in the system. In such an event the user would benefit from the visibility of the system state, and from adopting inspection strategies to detect the error and take, if possible, the necessary correctional measures. The paper presents a study performed in a Neonatal Intensive Care Unit where novel ambient computing technologies and related inspection strategies are currently being designed and assessed in the context of the European project PalCom.

Case Study – 5: Incubator temperature control, effects on the low birthweight infant.

Authors: A DUCKER, A J LYON, R ROSS RUSSELL, C A BASS, AND N McINTOSH.

Publications: ResearchGate.

Abstract: We studied temperature stability in 22 infants of birthweight less than 1500 g in the first four days of life. Infants were nursed in incubators using either air mode control or skin temperature servo control. Data were collected continuously using a computer linked monitoring system. Skin temperature control resulted in a less stable thermal environment than air mode control. Increased thermal stability in the incubator on air mode control may well be beneficial, particularly to sick, very low birthweight infants.

Case Study – 6: Adaptive Sway Control for Baby Bassinet Based on Artificial Metabolic Algorithm.

Authors: Yang Hu, Weihua Gui.
Publications: University of Siena

Abstract: To improve the Household management and decrease the young parents' Labor intensity, a new baby bassinet is made. The kind of bassinet can realize adaptive sway according to baby status. Some sensors can apperceive the movement of baby and other information such as baby cry. Alarm signal can be produced according to baby abnormity status. A kind of artificial metabolic algorithm is proposed in the paper. The algorithm can be applied for adjusting the bassinet swaying extent. The real example can show that baby can reach quiet rapidly and other control device can work efficiently for the baby health.

Case Study – 7: Automatic E-Baby Cradle Swing based on Baby Cry

Authors: Dilip Kumar

Abstract: There is a need to develop a new low-cost indigenous electronic cradle because the existing cradles are imported and costly. This paper presents the design and implementation of a new indigenous low-cost E-Baby Cradle that swings automatically when baby cries, for this it has a cry analyzing system which detects the baby cry voice and accordingly the cradle swings till the baby stops crying. The speed of the cradle can be controlled as per the user need. The system has inbuilt alarm that indicates two conditions – first when the mattress is wet, which is an important parameter to keep the baby in hygienic condition, second when baby does not stop crying with in a stipulated time, which intimated that baby needs attention. This system helps parents and nurses to take care of babies without physical attention.

Case Study – 8: IoT Based Smart Cradle for Infants

Authors: SudalaiMuthu T

Abstract: Baby needs attention by all the time, it is highly difficult for parents due to their house hold activities. It is impossible for working women. There are many systems are available to care of baby while the parents in work, but it is a dire need for the complete solution to take care of baby. A smart cradle is designed to address the challenges in taking care of baby.
A proposed Smart Cradle is designed as prototype by utilizing IoT. It assists the parentages to monitoring their kid regardless of whether they are away from home. It identifies each movement of the Infant from any turning of the world. It is an imaginative, shrewd and

defensive Cradle System to sustain a newborn child in a proficient manner. This framework considers all the subtleties required for baby monitoring. The proposed Cradle design uses various sensors includes Humidity, Temperature Sensor, Camera on the Cradle for live video film and Cry Discovery Circuit to break down Cry Patterns. The sensed data stored in Cloud (Thing Speak) and broke down at ordinary interims. The developed prototype experimented in the defined environment. The results are recorded and analyzed.

Case Study – 9: An IoT Smart Cradle for Tracking Babies.

Authors: Rekha S.

Abstract: Today IoT is a common technology. Many applications are designed based on IoT. One such important application is to use mobile phones to watch the activities of babies. If both the parents are working on a farm or if they are busy in their work, then it is difficult for them to nurture their infant. Therefore, we build an IoT-based cradle that lets the parent to track their child even they are far from home. In this paper we presented a smart cradle with continuous live surveillance. When a child cries, the bassinet instantly begins moving. This often warns parents if the infant is crying and if the mattress inside the bassinet is damp. Rotating toy is attached to cradle to entertain baby which may decrease the possibilities of baby cry.

Case Study – 10: Smart Baby Cradle.

Author: Marie R. Harper and Maxine R. Blea.

Abstract: Developed the first automatic rocking cradle which swings side by side on a horizontal axis which replicates the motion same as achieved by human oscillation of cradle. Spring motors are used to provide oscillatory motion to cradle. Springs motors are attached to the cradle of cradle that produce motion same as human efforts. The spring motor is of any known type in which the gear –operating means is easily stopped when the slightest resistance or opposition to its movement is encountered, thereby providing on extremely safe device for use with small children or babies. The advantages of this system is cost effectiveness, safe for small babies as it has mechanism to stop swinging of cradle whenever a resistance is occurred, require less human efforts and presence. The limitation of this system is it does not support video monitoring

Case Study – 11: E-Baby Cradle That Can Swing Automatically

Author: Goyal and Kumar

Abstract: when it detects crying and stops swinging when the crying stops. The speed for the swinging cradle can be controlled based on the user's need. It has an alarm embedded in the system, which notifies the user when two conditions occurred. First, the alarm goes off when the mattress is wet, indicating that the mattress should be changed. Second, when the baby does not stop crying fora certain time, the alarm alerts the parents to attend to their baby. However, it is only applicable when parents are near the cradle, because it only uses a buzzer alarm, the sound of which might frighten the baby. Parents cannot monitor their baby when they are away

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATION

General Description of the System

This project proposes the idea of automatic caretaker room for a baby. The main motive of this idea is to save time and energy of very busy parents. Working people are very busy these days. They do not have enough time to properly take care of their babies. So, the whole room is set up as it can sense the activities of the baby and work according to requirement. Parents can save their time and energy as they don't have to go and check their baby again and again until they don't get any information about baby. The idea of this scenario is accomplished by using sensors and a microprocessor. The sensors attached to the microprocessor sense the room condition and also monitor the activity of the baby. It works based on the condition we apply to it. Entire data will be visible to the user and get notified about the status of the baby.

The first verbal communication of newborn baby with the world is baby's cry. Infant crying is a biological alarm system. An infant crying signal is the attention call for parents or caregivers and motivates them to alleviate the distress.

There is a need to develop a new low-cost indigenous electronic System because the existing mechanical systems are imported and costly.

Hardware and Software Requirements of the System

Various sensors are used like Sound sensor, moisture sensor, heartbeat sensor and temperature sensor. These sensors are responsible for raising flags, If the sensor value exceeds the threshold value, the system will activate the alarm and also notify parents about the specific activity of the baby. All this data will be stored and analyzed at regular intervals on cloud which can be accessed using mobile applications. If the baby is making noise or baby is crying then sound sensor will hear that frequency and it will start swing. Also, Intimation alert will send to parent through the Wi-Fi module. If the baby had wetted the matrices of the cradle, then alert Intimation will send to the parent through the Wi-Fi module. If the body temperature of the baby changes rapidly with comparing atmosphere, then alert Intimation will send to the parents through Wi-Fi module. If baby is moving in cradle or any kind of movement detected by the IR sensor, then alert Intimation will send to the parent through the Wi-Fi module.

Here is list below of hardware and software components which we have included in our cradle system:

Hardware Requirements:

1. Arduino:-

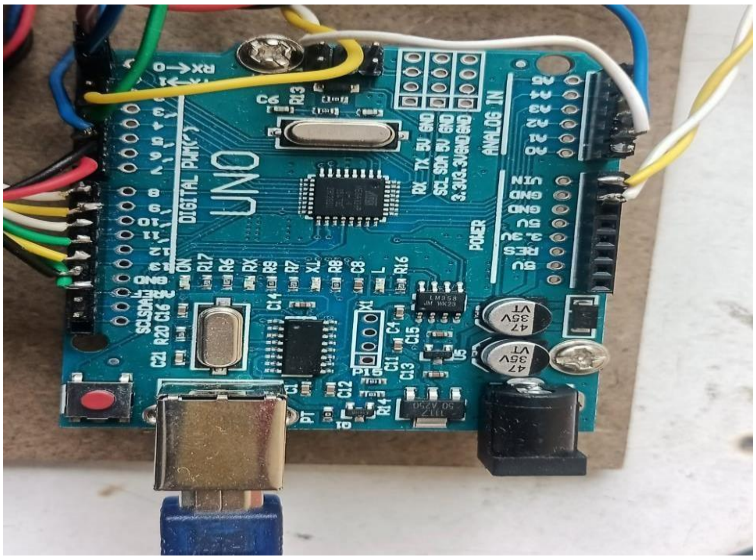

Fig 3.1: Arduino board

2. Motor Driver:-

Fig 3.2: Motor Driver

3. DC Motor

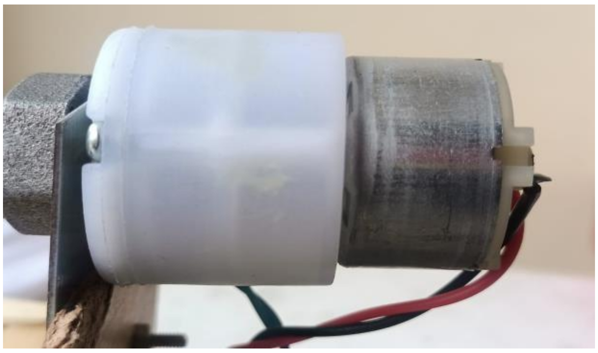

Fig 3.3: DC Motor

4. Temperature Sensor

Fig 3.4: Temperature Sensor

5. Arduino-Board-Camera

Fig 3.5: Camera

6. Sound Sensor

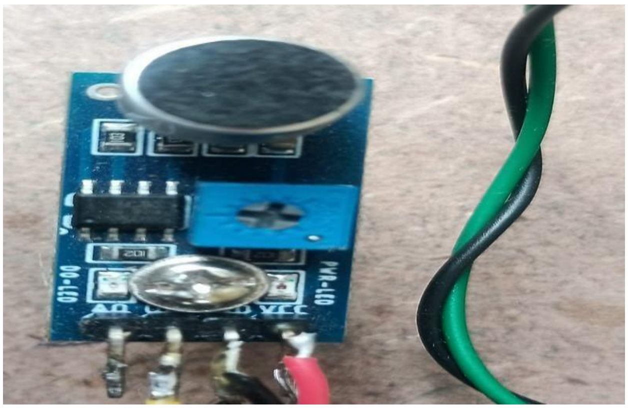

Fig 3.6: Sound Sensor

7. Moisture Sensor

Fig 3.7: Moisture Sensor

8. LCD Display

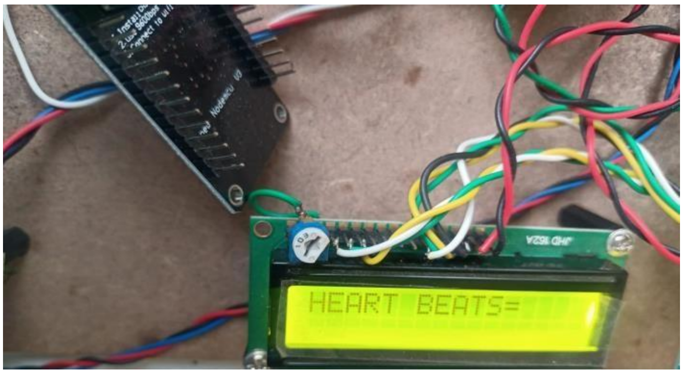

Fig 3.8: LCD

9. NODEMCU

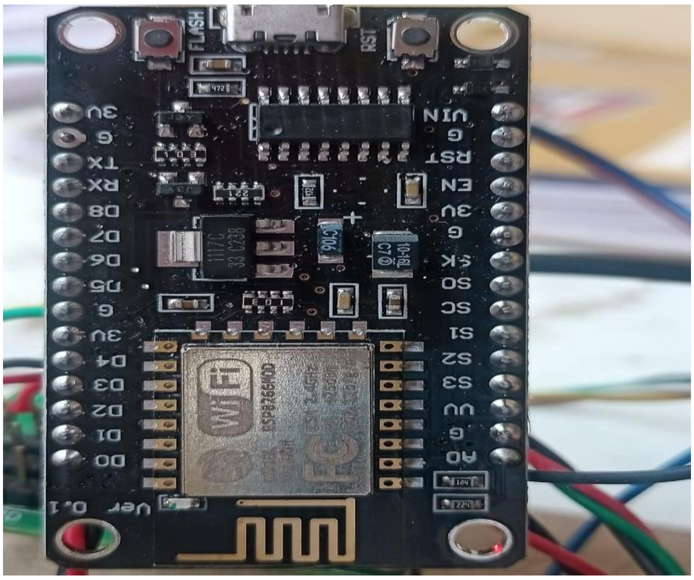

Fig 3.9: Wi-Fi Module

Software Requirements:

1. Arduino IDE
2. Embedded C.

The detailed description of the above listed components is discussed below:

1. Arduino

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control both physically and digitally. Its products are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form or as do-it-yourself (DIY) kits. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

2. Motor Driver:

- Can be used to run Two DC motors with the same IC.
- Speed and Direction control is possible
- Motor voltage Vcc2 (Vs): 4.5V to 36V
- Maximum Peak motor current: 1.2A
- Maximum Continuous Motor Current: 600mA
- Supply Voltage to Vcc1(vss): 4.5V to 7V
- Transition time: 300ns (at 5Vand 24V)
- Automatic Thermal shutdown is available
- Available in 16-pin DIP, TSSOP, SOIC packages

3. DC Motor:

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic; to periodically change the direction of current flow in part of the motor. DC motor has two wires, we can say them positive terminal and negative terminal, when these wires are connected with power supply the shaft rotates. We can reverse the direction of the rotation. L293d chip is very safe to use for DC motor control. This L293D is 16bit chip. Chip is design to control four DC motor, there are two inputs and two outputs for each motor.

There are two Enable pins on l293d. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge we need to enable pin 1 to high.

And for right H-Bridge we need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend working. It’s like a switch.

4. Temperature Sensor:

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°Cat room temperature and ±3⁄4°Cover a full −55 to +150°Ctemperature range. Low-cost is assured by trimming and calibration at the water level.

The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while theLM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and aplastic TO-220 package.

5. Sound Sensor:

The sound sensor is one type of module used to notice the sound. Generally, this module is used to detect the intensity of sound. The applications of this module mainly include switch, security, as well as monitoring. The accuracy of this sensor can be changed for the ease of usage. This sensor employs a microphone to provide input to buffer, peak detector and an amplifier. This sensor notices a sound, & processes an o/p voltage signal to a microcontroller. After that, it executes required processing. This sensor is capable to determine noise levels within DB's or decibels at 3 kHz 6 kHz frequencies approximately wherever the human ear is sensitive. In smartphones, there is an android application namely decibel meter used to measure the sound level.

The working principle of this sensor is related to human ears. Because human eye includes a diaphragm and the main function of this diaphragm is, it uses the vibrations and changes into signals. Whereas in this sensor, it uses a microphone and the main function of this is, it uses the vibrations and changes into current otherwise voltage. Nowadays, a lot of security events are initiated due to some sort of sound which includes gunshots, aggressive behavior, breaking the glass. But cameras with inbuilt sound exposure facilities can add huge value to the security system. Because they give an alert automatically when real and potential incidents occur. Then immediately they activate quick and appropriate actions to reduce the consequences.

6. Moisture sensor:

This is an easy-to-use digital moisture sensor. Just insert the sensor in the soil and it can measure moisture or water level content in it. It gives a digital output of 5V when moisture level is high and 0V when the moisture level is low in the soil. In this project this sensor is used for detecting baby bed wet condition. The sensor includes a potentiometer to set the desired moisture threshold. When the sensor measures more moisture than the set threshold, the digital output goes high and an LED indicates the output. When the moisture in the soil is less than the set threshold, the output remains low.

The digital output can be connected to a micro controller to sense the moisture level. The sensor also outputs an analog output which can be connected to the ADC of a micro controller to get the exact moisture level in the solid.

7. LCD Display:

A 16*2 LCD demonstrates contains two lines likewise; there are 16 characters for each line. Each character is appeared by 5x7 pixel lattice. This LCD includes two registers, specifically, Order and Data. The charges select extras the charge bearings that are given to the LCD. A charge is a rule given to LCD to do a predefined errand like presenting it, clears its screen, sets the cursor position, controls show et cetera. The data enrol saves the data to be appeared on the LCD

LCD is used in wide range application including computer monitors, televisions, instrument panels, aircraft cockpit displays, and indoor and outdoor signage. Small LCD screens are common in portable consumer devices such as digital cameras, watches, calculators, and mobile telephones, including smartphones. LCD screens are also used on consumer electronics products such as DVD players, video game devices and clocks. LCD screens have replaced heavy, bulky cathode ray tube (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and plasma displays, with LCD screens available in sizes ranging from tiny digital watches to huge, big-screen television sets. Since LCD screens do not use phosphors, they do not suffer image burn-in when a static image is displayed on a screen for a long time (e.g., the table frame for an aircraft schedule on an indoor sign). LCDs are, however, susceptible to image persistence. Interfacing an LCD with an Arduino. The 16x2 LCD has a total of 16 pins. As shown in the table below, eight of the pins are data lines (pins 7-14), two are for power and ground (pins 1 and 16), three are used to control the operation of LCD (pins 4-6), and one is used to adjust the LCD screen brightness (pin 3). The remaining two pins (15 and 16) power the backlight.

8. NODEMCU:

NodeMCU is an open-source LUA based firmware developed for ESP8266 Wi-Fi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e., NodeMCU Development board.

NodeMCU Kit/board consist of ESP8266 Wi-Fi enabled chip. The ESP8266 is a low-cost Wi-Fi chip developed by Express if Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 Wi-Fi Module.

There is Version2 (V2) available for NodeMCU Dev Kit i.e., NodeMCU Development Board v1.0 (Version2),

NodeMCU Dev Kit has Arduino like Analog (i.e., A0) and Digital (D0-D8) pins on its board. It supports serial communication protocols i.e., UART, SPI, I2C etc.

Using such serial protocols, we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

9. DC Motor:

A DC motor is any of a class of rotary electrical machines that converts direct current electricalenergy into mechanical energy. The most common types rely on the forces produced by magnetic fields.Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic; to periodically change the direction of current flow in part of the motor. DC motor has two wires, we can say them positive terminal and negative terminal, when these wires are connected with power supply the shaft rotates. We can reverse the direction of the rotation. L293d chip is very safe to use for DC motor control. This L293D is 16bit chip. Chip is design to control four DC motor, there are two inputs and two outputs for each motor

There are two Enable pins on l293d. Pin 1 and pin 9, for being able to drive the motor, the pin 1 and 9 need to be high. For driving the motor with left H-bridge we need to enable pin 1 to high. And for right H-Bridge we need to make the pin 9 to high. If anyone of the either pin1 or pin9 goes low then the motor in the corresponding section will suspend working. It's like a switch.

10. Jumper Wires

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

11. Arduino IDE:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low-cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step-by-step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handy board, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50

Cross-platform - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

Simple, clear programming environment - The Arduino Software (IDE) is easy-to use for beginners, yet flexible enough for advanced users to take advantage of as well.

Open source and extensible software - The Arduino software is published as open-source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Open source and extensible hardware - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

12. Embedded C:

When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempted to write the software in assembly language.

The trouble with projects done with assembly code can is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher-level language like C can directly address these issues. A program written in C is easier to read than an assembly program.

Since a C program possesses greater structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A high order language such as C allows a developer to write code, which resembles a human's thought process more closely than does the equivalent assembly code. The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable.

By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have C compilers available, which support ANSI C.

All of this is not to say that assembly language does not have its place. In fact, many embedded systems (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only way to go. One of the great things about the C language is that it allows you to perform low-level manipulations of the hardware, if need be, yet provides you the functionality and abstraction of a higher order language**.**

Input and Output Requirements

When the baby is made to sleep on the cradle, various sensors are implemented to monitor minute activities of the baby. Various sensors used are Sound sensor, moisture sensor, methane sensor intemperate sensor. If the sensor value exceeds the threshold value, the system will activate the alarm and also notify parents about the specific activity of the baby. All this data will be stored and analysed at regular intervals on cloud which can be accessed using mobile applications. If the baby is making noise or baby is crying, then sound sensor will hear that frequency and it will start swing. Also, Intimation alert will send to parent through the Wi-Fi module. If the baby had wetted the matrices of the cradle, then alert Intimation will send to the parent through the Wi-Fi module. If the body temperature of the baby changes rapidly with comparing atmosphere, then alert Intimation will send to the parents through Wi-Fi module. If baby is moving in cradle or any kind of movement detected by the IR sensor then alert Intimation will send to the parent through the Wi-Fi module

A system based on Arduino is being developed to assist parents and nurses in hospital in the care of infants.

The inputs taken are from the sensor based on the action or the indications of the baby.

Here all the components included acts as both input and output devices as constant monitoring of the of baby is done for effective results to help parent's relief in checking their loved ones.

Using a camera, recognize baby emotions, the controller is connected to a sound detector, which detects sound when the baby cries and activates the controller with its digital output.

Sensors will be used to keep track of the temperature and humidity, and the fan will be in control.

Interfaced moisture sensor for detecting moisture, to grab parents' or nurses' attention, the Wi-Fi interface delivers alerts to phones.

The controller is connected to an ALCD, which keeps sending messages with the current status.

Just to have a difference lcd display can be called as output devices and rest of the sensor can be referred as input/output devices.

Based on th listed condition the algorithms are designed to raise flags in generation on input and output signals

ALGORITHM USED FOR REMOTE MONITORING

1. Go to Blynk cloud services
2. Enter correct credentials to monitor values.
3. Check the values of temperature, humidity, moisture.
4. Open the app to get live feed from Wi-Fi camera.

ALGORITHM FOR CRY DETECTION

1. Start
2. Detect for noise
3. If noise is detected for longer duration (approx. 1 minute) then send message to parent device and glow LED
4. Switch off LED after 10 seconds. The algorithm descales the function of the cry detection.

Results/Output for Cry Detection

1. A microphone was configured with Arduino microcontroller to detect when baby is crying.
2. Initial reading when there was no noise were approximately 0-100. But when the noise of baby
3. crying was detected reading value increased and LED on the Arduino glowed to indicate that the bay is crying.

ALGORITHM FOR MOISTURE/URINE DETECTION

1. Start.
2. Detect wet condition.
3. If noise is detected for longer duration (approx. 1 minute) then send message
4. to parent device and activate alarm.
5. Stop alarm after 10 seconds.
6. The above algorithm descales the function of the urine detection module.

Results/Output for Moisture/Urine Detection

1. A water sensor was configured with Arduino micro- controller to detect urine condition for the baby. Initial reading when the sensor was dry was approximately 200-500.
2. A few drops of water were put on the sensor to make it wet.
3. After 10 seconds of wetting the sensor, there was an alarm tone and LED on the Arduino glowed to indicate the wet condition.

AUTOMATIC SWINGING ACTION PRINCIPLE

1. The main principle of the automatic swinging action of the cradle is as follows:
2. The sound of the baby cry is measured by noise sensor in decibels and it is compared with the threshold value B.
3. The input signal is amplified, converted to digital signal to sound level A which makes the servo motor to rotate at an angle of 180 degree.
4. Sound level (A)= 20 log (Vinput/V0utput) dB
5. Where, V input= Voltage level measured when the baby is crying. V0utput= Average reference level of
6. the voltage when the baby is happy.

1. Language Specification

Embedded C is one of the most popular and most commonly used Programming Languages in the development of Embedded Systems.

Embedded C is perhaps the most popular languages among Embedded Programmers for programming Embedded Systems. There are many popular programming languages like Assembly, BASIC, C++ etc. that are often used for developing Embedded Systems but Embedded C remains popular due to its efficiency, less development time and portability.What is an Embedded System?

An Embedded System can be best descradleed as a system which has both the hardware and software and is designed to do a specific task. A good example for an Embedded System, which many households have, is a Washing Machine.

Embedded Systems can not only be stand-alone devices like Washing Machines but also be a part of a much larger system. An example for this is a Car. A modern-day Car has several individual embedded systems that perform their specific tasks with the aim of making a smooth and safe journey.

Some of the embedded systems in a Car are Anti-lock Braking System (ABS), Temperature Monitoring System, Automatic Climate Control, Tyre Pressure Monitoring System, Engine Oil Level Monitor, etc.

Programming Embedded Systems

As mentioned earlier, Embedded Systems consists of both Hardware and Software. If we consider a simple Embedded System, the main Hardware Module is the Processor. The Processor is the heart ofthe Embedded System and it can be anything like a Microprocessor, Microcontroller, DSP, CPLD (Complex Programmable Logic Device) and FPGA (Field Programmable Gated Array).

All these devices have one thing in common: they are programmable i.e.; we can write a program (which is the software part of the Embedded System) to define how the device actually works.

Embedded Software or Program allow Hardware to monitor external events (Inputs) and control external devices (Outputs) accordingly. During this process, the program for an Embedded System may have to directly manipulate the internal architecture of the Embedded Hardware (usually the processor) such as Timers, Serial Communications Interface, Interrupt Handling, and I/O Ports etc.

From the above statement, it is clear that the Software part of an Embedded System is equally important to the Hardware part. There is no point in having advanced Hardware Components with poorly written programs (Software).

There are many programming languages that are used for Embedded Systems like

- Assembly (low-level Programming Language), C, C++, JAVA
- (high-level programming languages), Visual Basic, JAVA Script
- (Application-level Programming Languages), etc.

In the process of making a better embedded system, the programming of the system plays a vital role and hence, the selection of the Programming Language is very important.

Factors for Selecting the Programming Language.

The following are few factors that are to be considered while selecting the Programming Language for the development of Embedded Systems.

- Size: The memory that the program occupies is very important as Embedded Processors like Microcontrollers have a very limited amount of ROM.
- Speed: The programs must be very fast i.e.; they must run as fast as possible. The hardware should not be slowed down due to a slow running software.
- Portability: The same program can be compiled for different processors.
- Ease of Implementation
- Ease of Maintenance
- Readability

Earlier Embedded Systems were developed mainly using Assembly Language. Even though Assembly Language is closest to the actual machine code instructions, the lack of portability and high number of resources spent on developing the code, made the Assembly Language difficult to work with.

There are other high-level programming languages that offered the above-mentioned features but none were close to C Programming Language.

Introduction to Embedded C Programming Language

Before going in to the details of Embedded C Programming Language and basics of Embedded C Program, we will first talk about the C Programming Language.

The C Programming Language, developed by Dennis Ritchie in the late 60's and early 70's, is the most popular and widely used programming language. The C Programming Language provided low level memory access using an uncomplicated compiler (a software that converts programs to machine code) and achieved efficient mapping to machine instructions.

The C Programming Language became so popular that it is used in a wide range of applications ranging from Embedded Systems to Super Computers.

Embedded C Programming Language, which is widely used in the development of Embedded Systems, is an extension of C Program Language. The Embedded C Programming Language uses the same syntax and semantics of the C Programming Language like main function, declaration of datatypes, defining variables, loops, functions, statements, etc.

The extension in Embedded C from standard C Programming Language include I/O Hardware Addressing, fixed point arithmetic operations, accessing address spaces, etc.

Difference between C and Embedded C

There is actually not much difference between C and Embedded C apart from few extensions and the operating environment. Both C and Embedded Care ISO Standards that have almost same syntax, datatypes, functions, etc.

Embedded C is basically an extension to the Standard C Programming Language with additional features like Addressing I/O, multiple memory addressing and fixed-point arithmetic, etc.

C Programming Language is generally used for developing desktop applications whereas Embedded C is used in the development of Microcontroller based applications.

Basics of Embedded C Program

Now that we have seen a little bit about Embedded Systems and Programming Languages, we will dive in to the basics of Embedded C Program. We will start with two of the basic features of the Embedded C Program: Keywords and Datatypes.

Keywords in Embedded C

A Keyword is a special word with a special meaning to the compiler (a C Compiler for example, is a software that is used to convert program written in C to Machine Code). For example, if we take the Keil's Cx51 Compiler (a popular C Compiler for 8051 based Microcontrollers) the following are some of the keywords:

- bit
- sbit
- sfr
- small
- large

These are few of the many keywords associated with the Cx51 C Compiler along with the standard C Keywords.

Data Types in Embedded C

Data Types in C Programming Language (or any programming language for that matter) help us declaring variables in the program. There are many data types in C Programming Language like signed int, unsigned int, signed char, unsigned char, float, double, etc. In addition to these there few more data types in Embedded C.

The following are the extra data types in Embedded C associated with the Keil's Cx51 Compiler.

- bit
- sbit
- sfr
- sfr16

The following table shows some of the data types in Cx51 Compiler along with their ranges.

Data Type	*Bits (Bytes)*	*Range*
bit	1	0 or 1 (bit addressable part of RAM)
signed int	16 (2)	-32768 to +32767
unsigned int	16 (2)	0 to 65535
signed char	8 (1)	-128 to +127
unsigned	8 (1)	0 to 255
float	32 (4)	±1.175494E-38 to ±3.402823E+38
double	32 (4)	±1.175494E-38 to ±3.402823E+38
sbit	1	0 or 1 (bit addressable part of RAM)
sfr	8 (1)	RAM Addresses (80h to FFh)
sfr16	16 (2)	0 to 65535

Basic Structure of an Embedded C Program (Template for Embedded C Program)

The next thing to understand in the Basics of Embedded C Program is the basic structure or Template of Embedded C Program. This will help us in understanding how an Embedded C Program is written. The following part shows the basic structure of an Embedded C Program.

- Multiline Comments ....Denoted using /*……*/

- Single Line Comments......... Denoted using //

- Preprocessor Directives ....#include<…> or #define
- Global Variables......... Accessible anywhere in the program
- Function Declarations .... Declaring Function
- Main Function..........Main Function, execution begins here
- {
- Local Variables......... Variables confined to main function
- Function Calls ....Calling other Functions
- Infinite Loop ....Like while (1) or for (;;)
- Statements . . . . .
- ….
- ….
- }
    - Function Definitions .... Defining the Functions
- {
- Local Variables......... Local Variables confined to this Function
- Statements . . . . .

….

….

}

Different Components of an Embedded C Program

Comments: Comments are readable text that are written to help us (the reader) understand the code easily. They are ignored by the compiler and do not take up any memory in the final code (after compilation).

There are two ways you can write comments: one is the single line comments denoted by // and the other is multiline comments denoted by /*….*/.

Preprocessor Directive: A Preprocessor Directive in Embedded C is an indication to the compiler thatit must look in to this file for symbols that are not defined in the program.

In C Programming Language (also in Embedded C), Preprocessor Directives are usually represented using #include… or #define….

In Embedded C Programming, we usually use the preprocessor directive to indicate a header file specific to the microcontroller, which contains all the SFRs and the bits in those SFRs.

In case of 8051, Keil Compiler has the file "reg51.h", which must be written at the beginning of every Embedded C Program.

Global Variables: Global Variables, as the name suggests, are Global to the program i.e. they can be accessed anywhere in the program.

Local Variables: Local Variables, in contrast to Global Variables, are confined to their respective function.

Main Function: Every C or Embedded C Program has one main function, from where the execution ofthe program begins.

When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. If a hardware engineer is tasked with designing both the hardware and the software, he or she will frequently be tempted to write the software in assembly language.

The trouble with projects done with assembly code can is that they can be difficult to read and maintain, especially if they are not well commented. Additionally, the amount of code reusable from a typical assembly language project is usually very low. Use of a higher-level language like C can directly address these issues. A program written in C is easier to read than an assembly program.

Since a C program possesses greater structure, it is easier to understand and maintain. Because of its modularity, a C program can better lend itself to reuse of code from project to project. The division of code into functions will force better structure of the software and lead to functions that can be taken from one project and used in another, thus reducing overall development time. A high order language such as C allows a developer to write code, which resembles a human's thought process more closely than does the equivalent assembly code. The developer can focus more time on designing the algorithms of the system rather than having to concentrate on their individual implementation. This will greatly reduce development time and lower debugging time since the code is more understandable.

By using a language like C, the programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly. Many target processors have C compilers available, which support ANSI C. All of this is not to say that assembly language does not have its place. In fact, many embedded systems (particularly real time systems) have a combination of C and assembly code. For time critical operations, assembly code is frequently the only way to go. One of the great things about the C language is that it allows you to perform low-level manipulations of the hardware, if need be, yet provides you the functionality and abstraction of a higher order language.

CHAPTER 4

SYSTEM DESIGN AND ANALYSIS

System Design and Architecture

A system architecture for a baby cradle using IoT would typically consist of several components working together to provide various functionalities.

1. Microcontroller or IoT device: This component forms the heart of the system, responsible for controlling the cradle's movements, collecting sensor data, and communicating with other devices on the network. It could be an Arduino or a Raspberry Pi, equipped with Wi-Fi or Bluetooth connectivity.

2. Sensors: The sensors used in the system could include temperature and humidity sensors to monitor the baby's environment, motion sensors to detect when the baby is awake or asleep, and possibly even a camera to allow the parents to monitor their baby remotely.

3. Actuators: These are responsible for moving the cradle and providing soothing vibrations to help the baby fall asleep. The actuators could be servo motors or stepper motors, controlled by the microcontroller.

4. Power supply: A reliable power supply is crucial to ensure that the system operates smoothly without any interruptions. It could be a battery or a mains power supply, depending on the specific requirements of the system.

5. Communication module: The communication module is responsible for sending and receiving data to and from other devices on the network. It could be Wi-Fi, Bluetooth, or any other wireless communication protocol.

6. Cloud server: To enable remote monitoring and control of the cradle, a cloud server could be used to store and process data collected by the system. This would allow parents to monitor their baby's sleeping patterns and receive notifications if any issues are detected.

Overall, the system architecture for a baby cradle using IoT is designed to be user-friendly, reliable, and secure, providing parents with peace of mind and helping their baby sleep soundly.

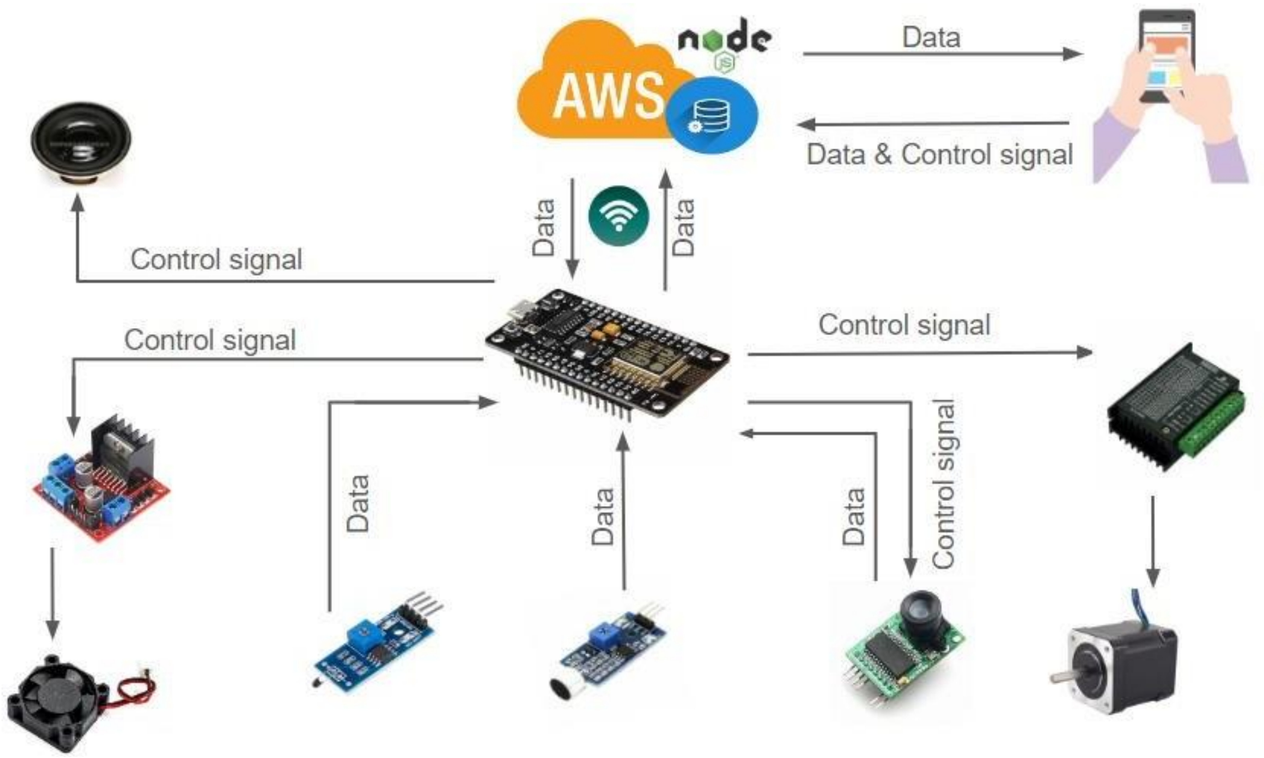

Fig 4.1 : system architecture

Data Flow

The data flow in a smart cradle using IoT typically follows a cycle of data collection, processing, and transmission. Here is a brief overview of the data flow in a smart cradle:

1. Sensor data collection: The smart cradle will have various sensors such as temperature, humidity, motion sensors, and sound sensors that collect data about the baby's environment, movements, and behavior.

2. Data processing: The collected data will be processed by the microcontroller or IoT device in the cradle, which will analyze the data and trigger appropriate actions based on the data.

3. Actuation: Based on the analyzed data, the microcontroller will control the actuators to provide soothing vibrations, gentle swinging, or other actions to help the baby sleep.

4. Data transmission: The processed data, along with any other relevant information, will be transmitted to a cloud server using Wi-Fi, Bluetooth, or other wireless communication protocols.

5. Cloud processing: The cloud server will process the data received from the smart cradle and analyze it to provide insights into the baby's sleeping patterns, environmental conditions, and other relevant information. This information can be accessed by the parents using a smartphone app or a web interface.

6. User interface: The parents can access the information about their baby's sleeping patterns, monitor the baby's environment, and receive notifications about any issues through a user-friendly interface provided by the cloud server.

Overall, the data flow in a smart cradle is designed to be seamless and efficient, providing parents with valuable insights into their baby's health and well-being while also providing a comfortable and soothing environment for the baby to sleep.

There is a cry detection mechanism which detects cry of the baby and send instant mobile app notifications to the user, at the same time some music will be played to soothe the baby. User can use play music option using their mobile application according to their wish at any occasion even if a cry is not detected in order to soothe the baby. Using the temperature sensors, room temperature is deflected and if it exceeds 30 Celsius, mini fan will be turned on automatically. Turn on and off functions of the mini fan can be fully controlled using the mobile application also according to the need of the user. Swing of the cradle is fully controlled by the Mobile application.

Central server (AWS) is used to maintain a database to keep a track of registrations and logins. Whenever the user logs in to the system, mobile application will connect to the server and verify authentications.

Hardware Control Flow

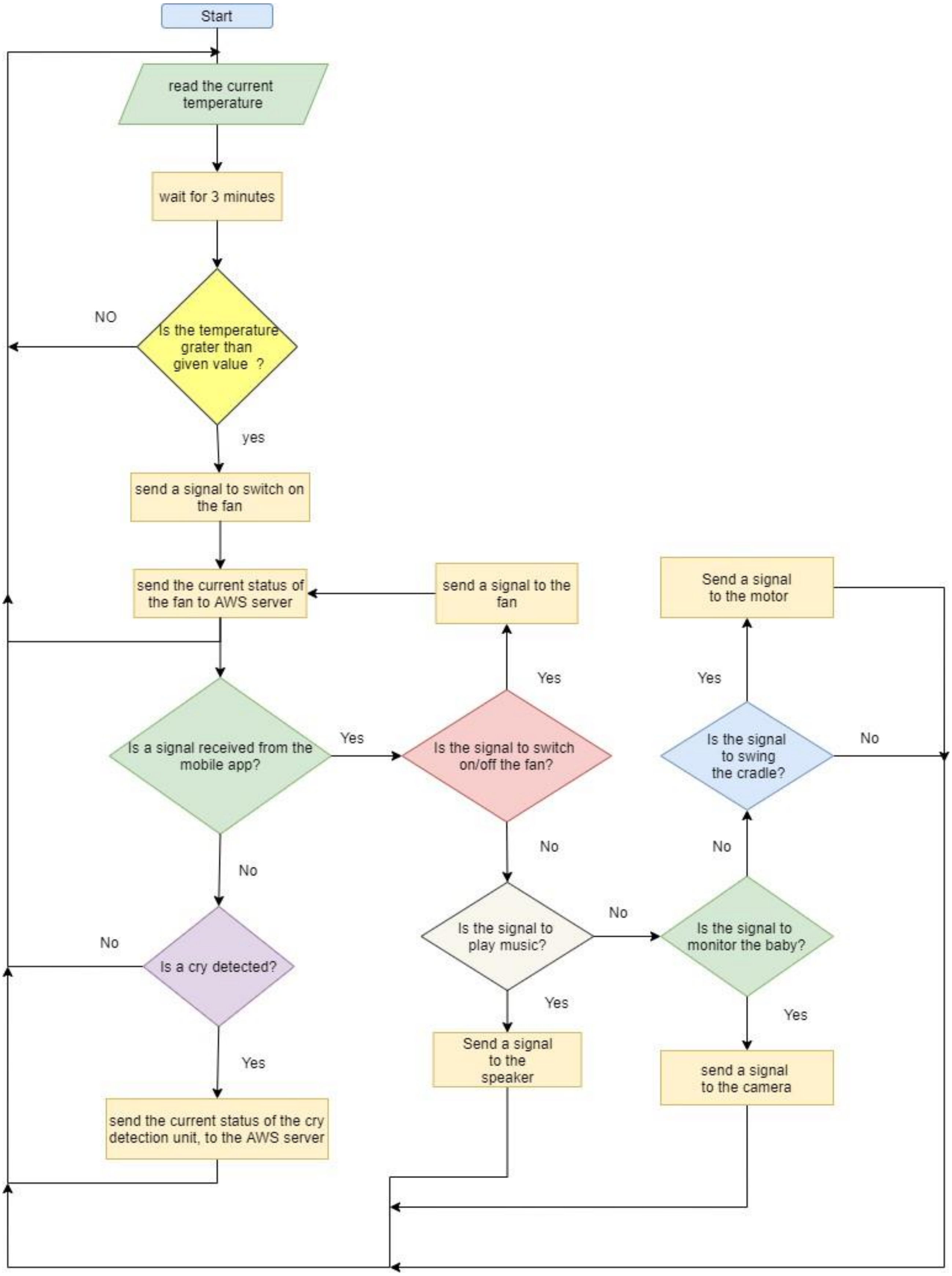

Fig 4.2: Hardware control

Software Control Flow:

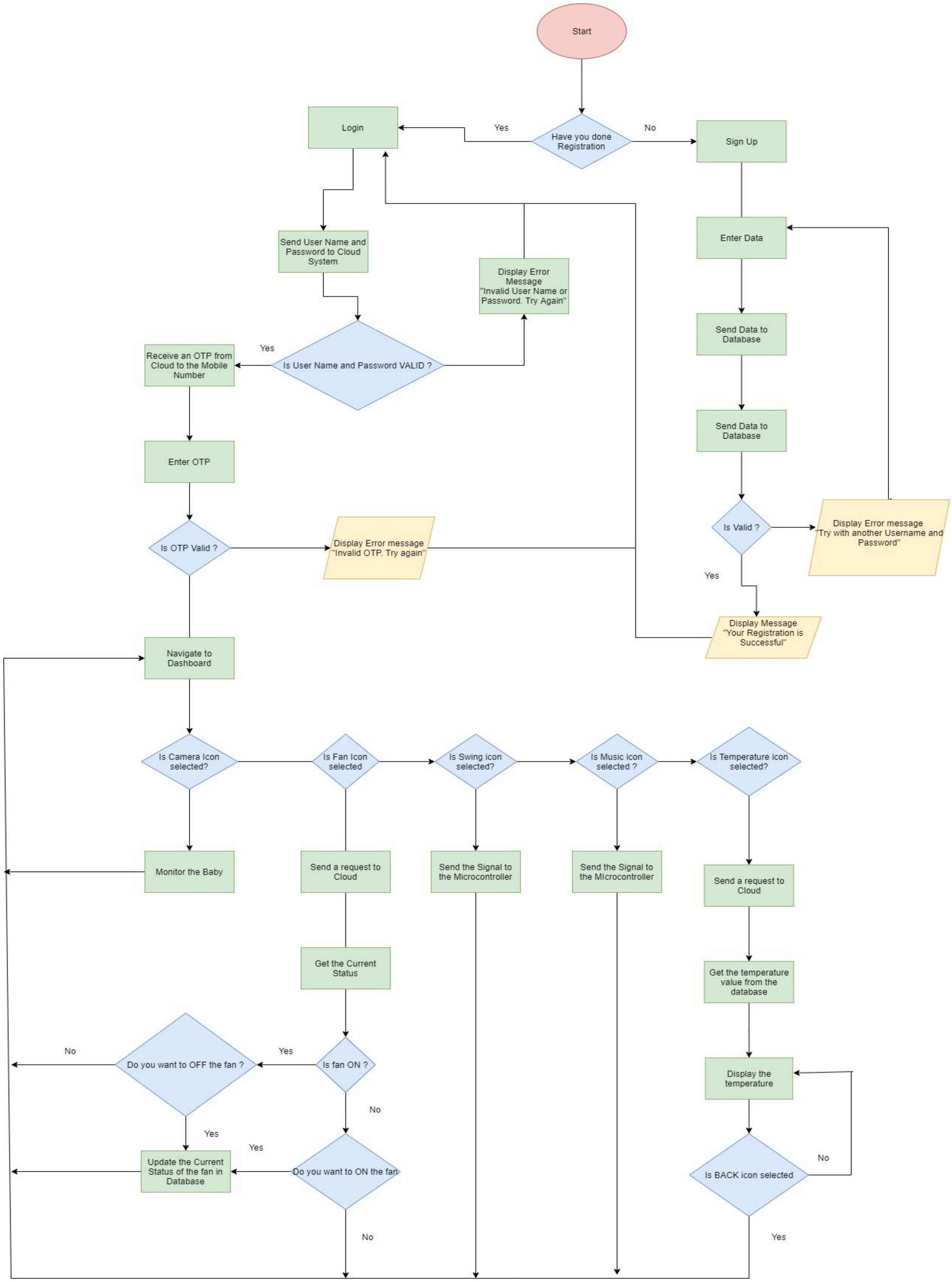

Fig 4.3: software control

E R Diagram

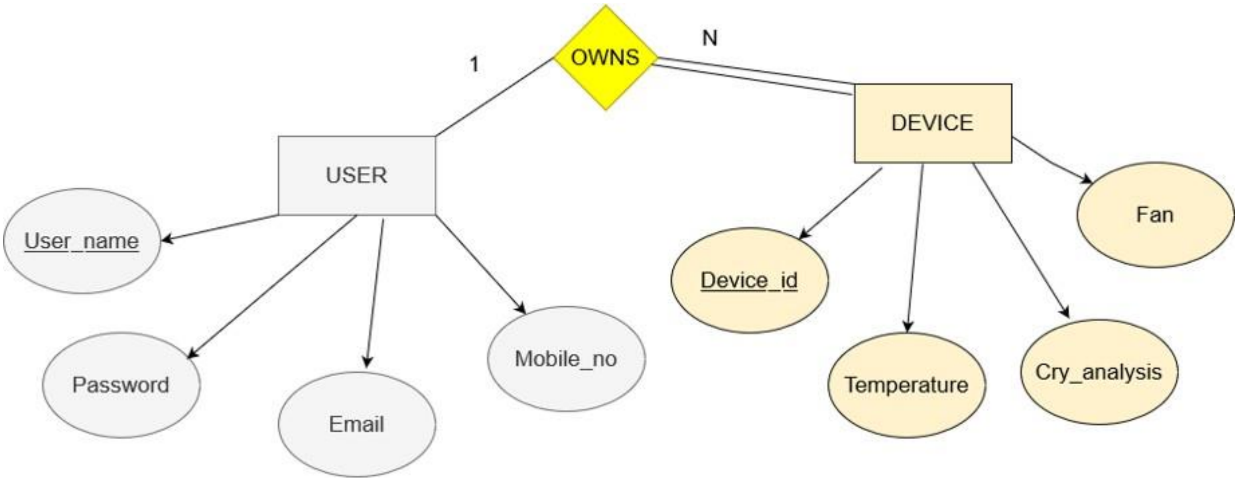

Fig 4.4: ER Diagram

EE R Diagram

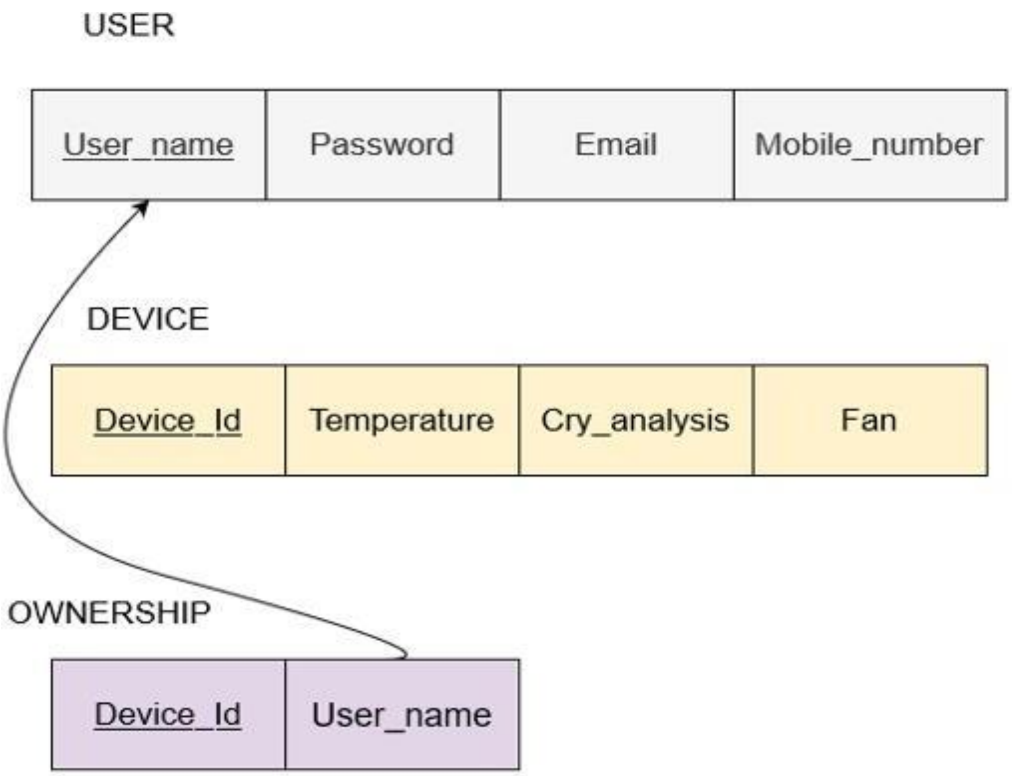

Fig 4.5: EE R Diagram

Use Case

Here are some use case scenarios for a smart cradle using IoT:

1. Sleep monitoring: One of the most common use cases for a smart cradle is to monitor a baby's sleeping patterns. The cradle can be equipped with sensors to monitor the baby's movements, temperature, humidity, and sound levels. The data can be processed and transmitted to a cloud server, where it can be analyzed and used to generate reports on the baby's sleep quality and quantity. Parents can use this information to adjust the cradle's settings to improve their baby's sleep.

2. Remote monitoring: Another use case for a smart cradle is remote monitoring. Parents can access the data collected by the cradle using a smartphone app or a web interface, allowing them to monitor their baby's sleep and environment remotely. This can be especially useful for parents who are away from home or have hired a babysitter.

3. Health monitoring: The smart cradle can also be used to monitor the baby's health. For example, if the baby has a fever, the cradle can be equipped with a temperature sensor to alert the parents if the temperature rises above a certain threshold. Similarly, if the baby is experiencing breathing difficulties, the cradle can be equipped with a sensor to monitor the baby's breathing rate and alert the parents if there are any irregularities.

4. Soothing and comfort: The smart cradle can be designed to provide soothing and comforting movements to help the baby sleep. For example, the cradle can be equipped with actuators to gently rock the baby or provide soothing vibrations. Parents can adjust the settings of the cradle to find the right combination of movements and vibrations to help their baby fall asleep quickly and comfortably.

Overall, a smart cradle using IoT can provide parents with valuable insights into their baby's sleep and health, while also providing a comfortable and soothing environment for their baby to sleep.

USE CASE Diagram

swinging use case

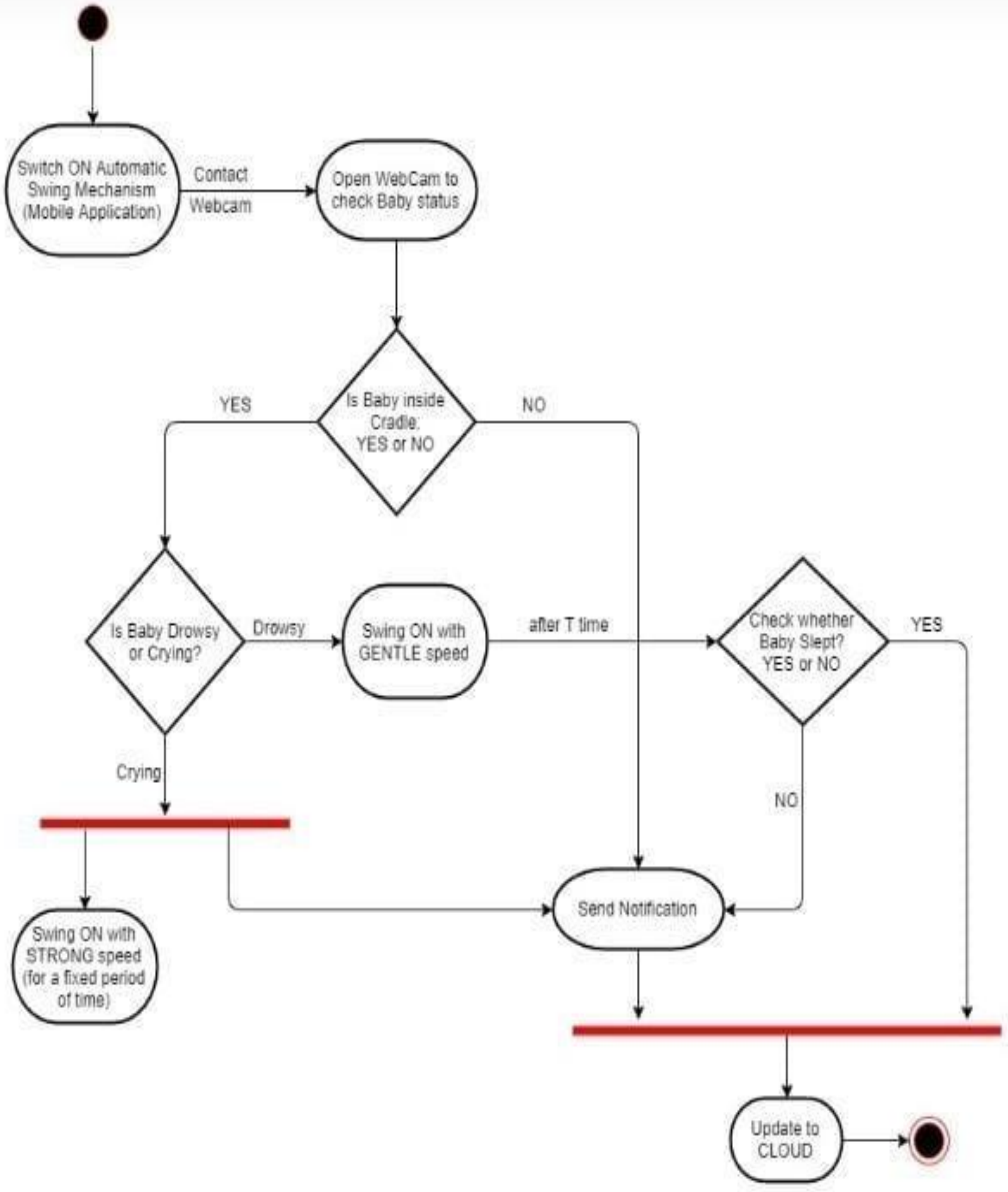

Fig 4.6: Swinging Mechanism

Sound Use Case

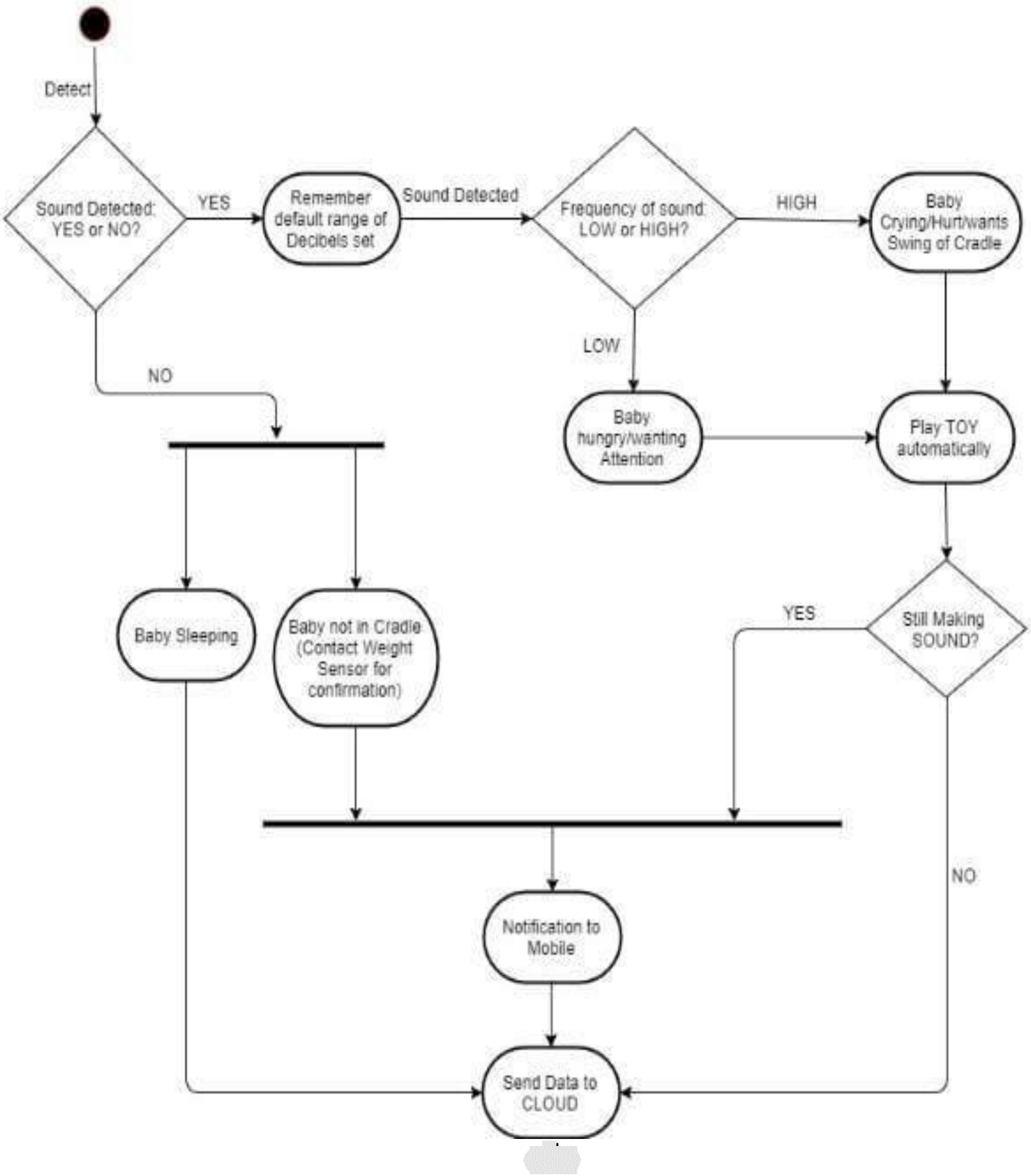

Fig 4.7: Sound Mechanism

Smart cradle Use case

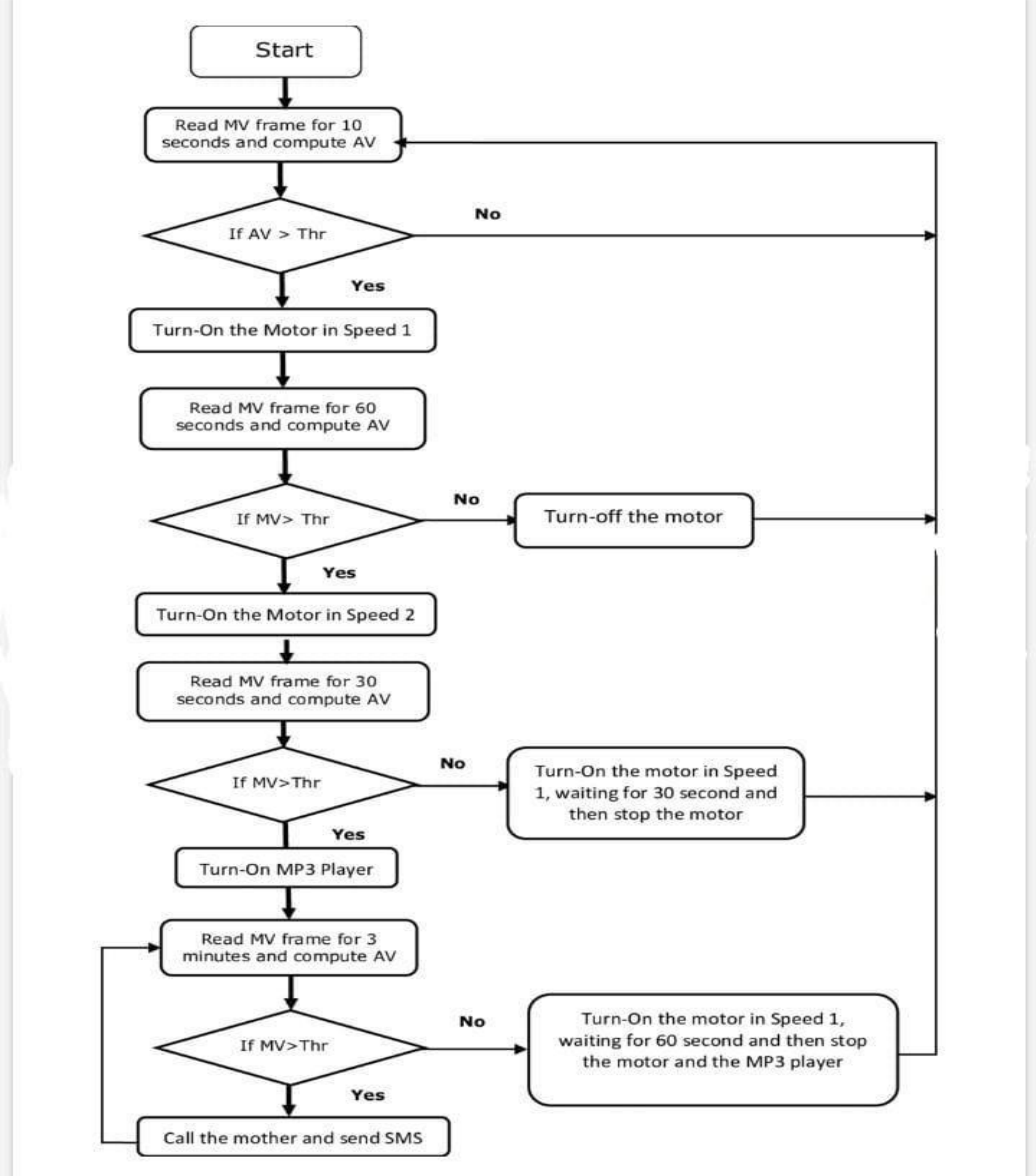

Fig 4.8: Cradle Working Mechanism

CHAPTER 5

IMPLEMENTATION

Construction of the Circuit

The construction of the circuit for a cradle using IoT would typically involve the following steps:

1. Component selection: The first step is to select the appropriate components for the circuit, such as microcontroller or IoT device, sensors, actuators, power supply, communication module, and other supporting components.

2. Circuit design: Once the components have been selected, the next step is to design the circuit diagram. The circuit diagram will show how the components are connected and how they interact with each other.

3. PCB design: After designing the circuit diagram, the next step is to design the printed circuit board (PCB) that will house the circuit components. The PCB design will involve placing the components in the appropriate locations and routing the connections between them.

4. PCB fabrication: Once the PCB design is complete, the PCB can be fabricated using standard PCB fabrication processes, such as etching or milling.

5. Component assembly: After the PCB has been fabricated, the components can be assembled onto the board using soldering or other assembly techniques.

6. Firmware programming: Once the circuit is assembled, the firmware for the microcontroller or IoT device can be programmed. This will involve writing code that will control the sensors, actuators, and other components, as well as handling data transmission and communication with other devices on the network.

7. Testing and debugging: After the firmware has been programmed, the circuit can be tested to ensure that it is functioning correctly. Any bugs or issues that are detected can be debugged and resolved.

Overall, the construction of the circuit for a cradle using IoT involves careful component selection, circuit design, PCB design and fabrication, component assembly, firmware programming, and testing and debugging. The goal is to create a reliable and efficient circuit that can provide a comfortable and soothing environment for a baby to sleep while also providing valuable insights into the baby's sleep and health.

Wi-Fi module circuit connection

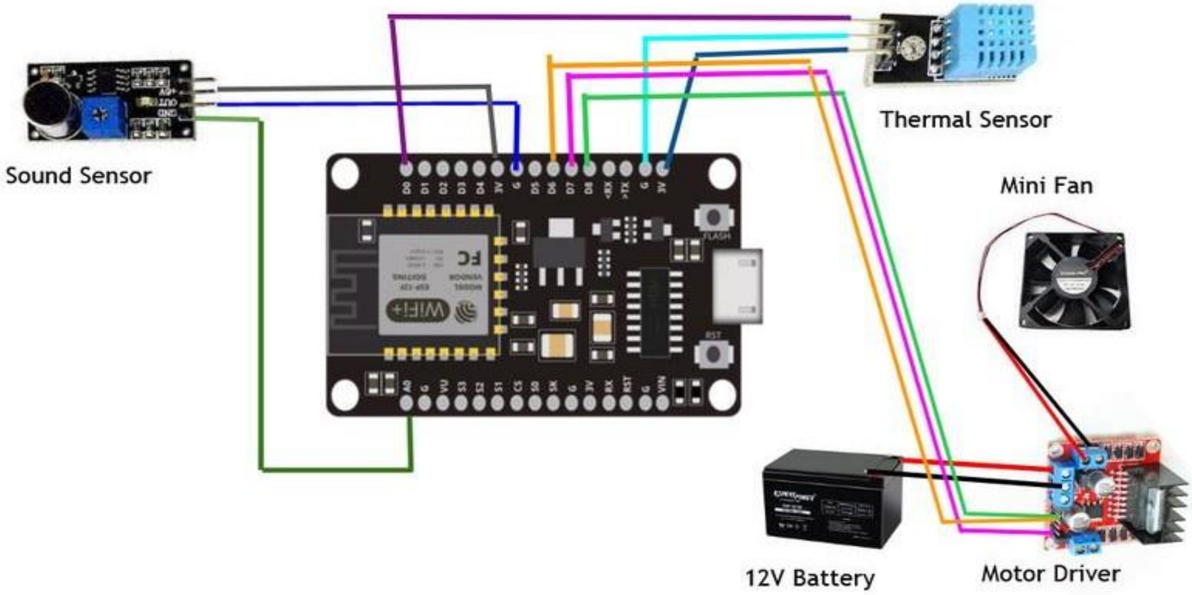

Fig 5.1: Wi-Fi Board Connection

Motor Driver Circuit Connection

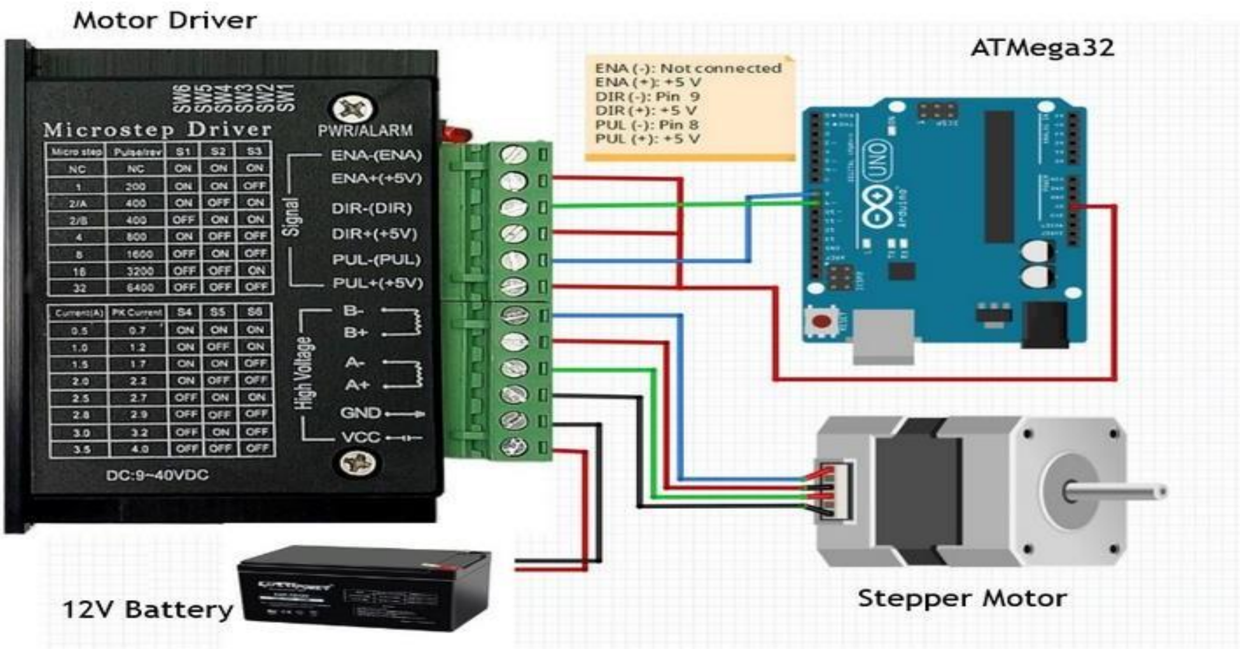

Fig 5.2: Motor Connection

Camera Circuit Connection

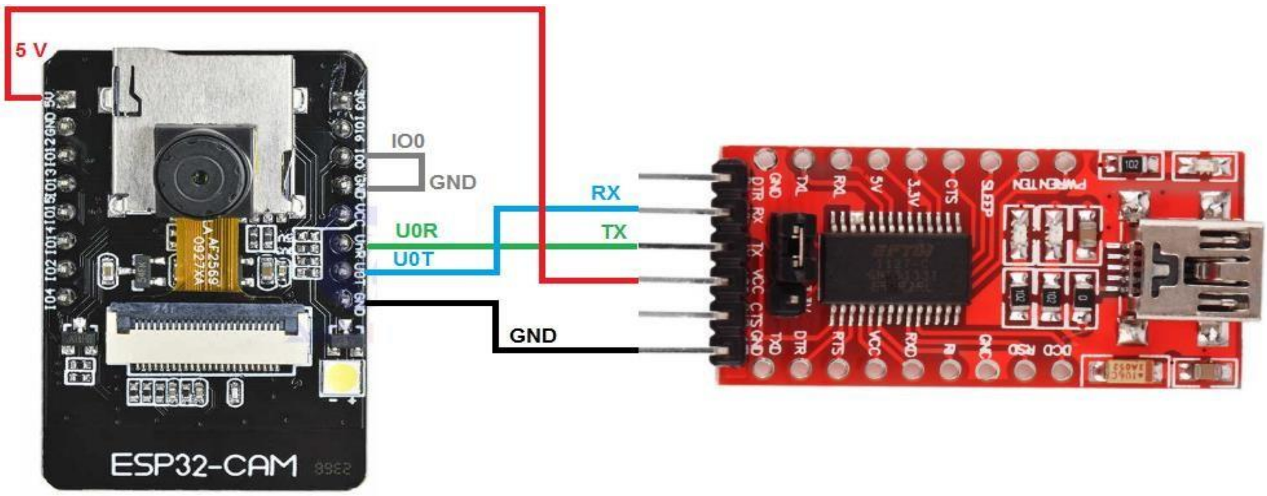

Fig 5.3: Camera connection

Schematic circuit diagram

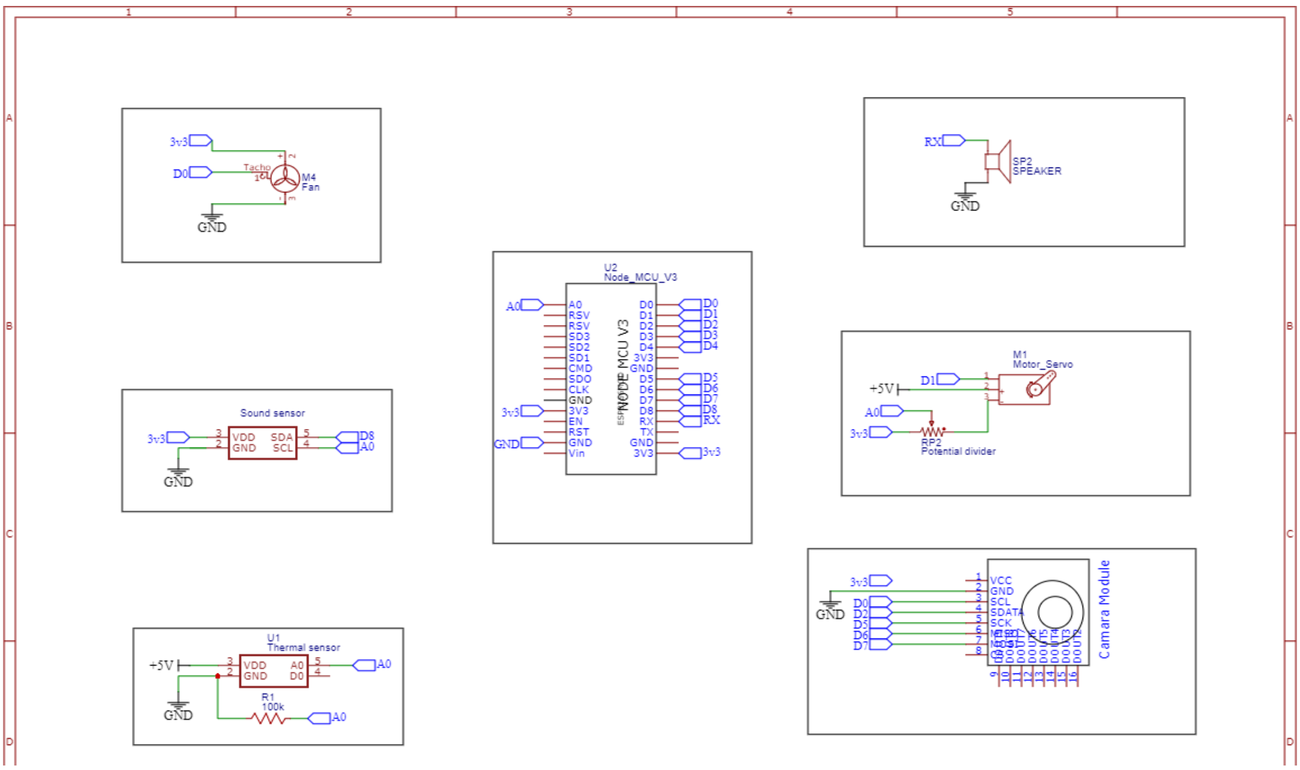

Fig 5.4: Cradle Components Connection

Code for Sensor in Smart Cradle Working Process

Working of infant cradle

```
#include <LiquidCrystal.h> //LCD initialization

#include<DHT.h> //


const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);


int light=4; //Light sensor pin assignment

int sound=3;  //Sound sensor pin assignment


#define Type DHT11  //DHT sensor type definition

int DHT11PIN = 5;   //DHT sensor pin assignment

DHT HT(DHT11PIN, Type);  //DHT sensor initialization

//float Humidity;

float tempC;  //Temperature variable declaration.



int data=A1;//heartbeat sensor

int count=0;


unsigned long temp=0;


//      byte customChar1[8] =
{0b00000,0b00000,0b00011,0b00111,0b01111,0b01111,0b01111,0b01111};

//      byte customChar2[8] =
{0b00000,0b11000,0b11100,0b11110,0b11111,0b11111,0b11111,0b11111};

//      byte customChar3[8] =
{0b00000,0b00011,0b00111,0b01111,0b11111,0b11111,0b11111,0b11111};
```

```
//    byte customChar4[8] =
{0b00000,0b10000,0b11000,0b11100,0b11110,0b11110,0b11110,0b11110};

//    byte customChar5[8] =
{0b00111,0b00011,0b00001,0b00000,0b00000,0b00000,0b00000,0b00000};

//    byte customChar6[8] =
{0b11111,0b11111,0b11111,0b11111,0b01111,0b00111,0b00011,0b00001};

//    byte customChar7[8] =
{0b11111,0b11111,0b11111,0b11111,0b11110,0b11100,0b11000,0b10000};

//    byte customChar8[8] =
{0b11100,0b11000,0b10000,0b00000,0b00000,0b00000,0b00000,0b00000};


#define IN1 6     // left motor

#define IN2 7    // left motor




//int Voice=6;


int i=0;


void setup()

{

  Serial.begin(9600); //Serial communication initialization.

  pinMode(DHT11PIN, INPUT); //DHT11 sensor pin initialization.

  lcd.begin(16,2);  //LCD initialization.

  HT.begin();  //DHT11 sensor communication initialization.

  pinMode(IN1, OUTPUT); //IN1 pin initialization.

  pinMode(IN2, OUTPUT); //IN2 pin initialization.

  pinMode(sound,INPUT); //Sound sensor pin initialization.


  pinMode(light, OUTPUT);
```

```
//  pinMode(VIB, INPUT);

// digitalWrite(Voice,LOW);
  digitalWrite(light,LOW);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);

//      lcd.createChar(1, customChar1);
//      lcd.createChar(2, customChar2);
//      lcd.createChar(3, customChar3);
//      lcd.createChar(4, customChar4);
//      lcd.createChar(5, customChar5);
//      lcd.createChar(6, customChar6);
//      lcd.createChar(7, customChar7);
//      lcd.createChar(8, customChar8);



      Serial.begin(9600); //Initializes the serial communication.
//      lcd.begin(16, 2);
      lcd.print("BABY CRADLE.."); //Displays "BABY CRADLE.." on the LCD.
 delay(1000); //Pauses the program for 1000 milliseconds (1 second).  delay(1000);

}

void loop()
{
   SOUND_MONITOR();    //Calls the SOUND_MONITOR function.
   delay(1000);  //Pauses the program for 1000 milliseconds (1 second).
```

```
  TEMP_MONITOR(); //Calls the TEMP_MONITOR function.
  delay(1000); //Pauses the program for 1000 milliseconds (1 second).
  WET_MONITOR();  //Calls the WET_MONITOR function.
  delay(1000); //Pauses the program for 1000 milliseconds (1 second).
  HEART_BEAT_MONITOR();  //Calls the HEART_BEAT_MONITOR
  function.delay(1000);   //Pauses the program for 1000 milliseconds (1 second).

}

void TEMP_MONITOR()
{
//  Humidity = HT.readHumidity();
  tempC = HT.readTemperature(); //Reads the temperature data from the DHT11 sensorand
stores it in the tempC variable
//  if(Temp_val<10)
//  Temp_val=28.0;
  lcd.clear();  //Clears the LCD display.
  lcd.print("TEMP:"); //Displays "TEMP:" on the LCD.
  lcd.print(tempC);  //Displays the temperature value on the LCD.
  Serial.print("TEMP:");  //Prints "TEMP:" on the serial monitor.
  Serial.println(tempC);  //Prints the temperature value on the serial monitor.
  delay(1000);
  if(tempC>35)  //Checks if the temperature value is greater than 35.
  {
  lcd.clear();  //Clears the LCD display.
  lcd.print("MORE TEMP"); //Displays "MORE TEMP" on the LCD.
  lcd.print("DETECTED..");    //Displays "DETECTED.." on the LCD.
  Serial.print("MORE TEMP");    //Prints "MORE TEMP" on the serial
  monitor.
```

```
  Serial.println();
  delay(1000);

 }

}
void WET_MONITOR()
{
  int WET_val=analogRead(A2); //Reads the analog data from the A2 pin and stores it inthe WET_val variable.
//  WET_val=1023-WET_val;
  lcd.clear();  //Clears the LCD display.
  lcd.print("WET:"); //Displays "WET:" on the LCD.
  lcd.print(WET_val); //Displays the wetness value on the LCD.
  Serial.print("WET:");  //Prints "WET:" on the serial monitor.
  Serial.println(WET_val);  //Prints the wetness value on the serial monitor.
  delay(1000);
  if(WET_val<700)  //
  {
    lcd.clear();
    lcd.print("BABY IS WET..");
    Serial.println("$Baby is Wet#");
//    CRADLE_START();

  }
}
void SOUND_MONITOR()
{
  int Sound_val=digitalRead(sound);  //Declare integer variable.
```

```
    lcd.clear();
    lcd.print("SOUND:"); //Print text on LCD display.
    lcd.print(Sound_val);  //Print variable value on LCD display.
    Serial.print("SOUND:");  //Print text to serial monitor.
    Serial.println(Sound_val);  //Print variable value to serial monitor.
    delay(1000);
    if(Sound_val==LOW)  //Check if input value is low.
    {
     lcd.clear();
     lcd.print("BABY CRYING.."); //Print text on LCD display.
     Serial.println("$Baby Crying...#");  //Print text to serial monitor.
     CRADLE_START();  //Call function to start the cradle.

    }
   }

   void CRADLE_START()
   {
```

digitalWrite(light,HIGH);: Turn on light.
for(i=0;i<2;i++): Start a loop.
digitalWrite(IN1,HIGH);: Set pin 1 high.
digitalWrite(IN2,LOW);: Set pin 2 low.
delay(300);: Wait for 300ms.

```
     digitalWrite(IN1,LOW); //Set pin 1 low.
     digitalWrite(IN2,LOW);    //Set pin 2 low.
     delay(300);
     digitalWrite(IN1,LOW);  //Set  pin  1  high.
     digitalWrite(IN2,HIGH);  //Set  pin  2  high.
     delay(300);
```

```
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
    delay(300);
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);
    delay(300);
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
    delay(300);
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    delay(300);
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);
    delay(300);
  }
  digitalWrite(light,LOW);
  STOP();
}
void STOP()
{

    digitalWrite(IN1,LOW);
    digitalWrite(IN2,LOW);



}
void HEART_BEAT_MONITOR()
```

```
{

  count=0; //Initialize count variable to 0
   lcd.clear(); //Clear LCD display
   lcd.setCursor(0,0); //Set cursor position to (0,0).
   lcd.print("Place The Finger"); //Print text on LCD display.
   lcd.setCursor(0,1); //Set cursor position to (0,1). lcd.print("to
   check HB"); //Print text on LCD display.
  Serial.println("Place The Finger to check HB");  //Print text to serial monitor.
  delay(2000);
// while(digitalRead(start)>0);

    temp=millis();  //Get current time in milliseconds and assign to temp variable.

  while(millis()<(temp+2000))  //Start a loop that runs for 2 seconds.
  {

    if(analogRead(data)<100)  //Check if analog input is less than 100.
     {
      count=count+1;  //Increment count variable.

      while(analogRead(data)<100);  //Wait until analog input is greater than or equal to 100
     }
  }
   count=count*6;  //Multiply count variable by 6.

      temp=0;  //Assign 0 to temp variable.
      delay(1000);
```

```
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("HEART BEATS=");  //Print text on LCD display.
    lcd.setCursor(0,1); //Set cursor position to (0,1).
    lcd.print(count);  //Print count variable on LCD display.
    Serial.print(count);  //Print count variable to serial monitor.
    Serial.println(""); //Print blank line to serial monitor.
    delay(1000);  //Wait for 1 second.

}
```

Code for the Baby Cradle Monitor System on Telegram Bot as a Message

```
#include<ESP8266WiFi.h>
#include<WiFiClientSecure.h>
#include<UniversalTelegramBot.h>
#include<ArduinoJson.h>
const char* ssid = "smile123";
const char* password ="123456789";
//  with your network credentials

// Initialize Telegram BOT
#define BOTtoken "6291159697:AAEVm_Q14dQk2V4nuQy_vKuqhOrl27X_2Qw" // your Bot
Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
#define CHAT_ID "1086631129"

X509List cert(TELEGRAM_CERTIFICATE_ROOT);  //This line initializes an X509List object
with the TELEGRAM_CERTIFICATE_ROOT.
WiFiClientSecure client; //This line creates a WiFiClientSecure object for secure
communication.
UniversalTelegramBot bot(BOTtoken, client); //This line creates a UniversalTelegramBot
object with the BOTtoken and client.


char Start_buff[70]; //This line declares a character array named Start_buff with 70
elements.
int i,z;
```

```
char  ch;
int str_len;
 char textmessage[20];



void MESSAGE_SEND();  //This line declares a function named MESSAGE_SEND().
void WAITING();  //This line declares a function named WAITING().
void setup()
{
    // initialize the Serial
    Serial.begin(9600);
    Serial.println("Starting TelegramBot...");

    configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
  client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org

  // Attempt to connect to Wifi network:
  Serial.print("Connecting Wifi: ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

// bot.sendMessage(CHAT_ID, "Bot started up", "");
    bot.sendMessage(CHAT_ID, "BABY CRADLE SYSTEM.");


}
void loop()
{

  WAITING();

}
void MESSAGE_SEND()
{
  bot.sendMessage(CHAT_ID, "SEND START TO CONTINUE");
  // myBot.sendMessage(msg_sender_id1, "WELCOME TO ATM");
```

```
}


char Serial_read(void)
{
      char ch;
      while(Serial.available() == 0);
      ch = Serial.read();
      return ch;
}
void WAITING()
{
   Serial.println("WAITE");

  buffer_clear();
//  msg.text[0]='\0';
//       msg.text[1]='\0';
//        msg.text[2]='\0' ;
//        msg.text[3]='\0';
//         msg.text[4]='\0';
//          msg.text[5]='\0';

 while(1)
 {
      if (Serial.available() > 0)
      {
         //Serial.println("halo");

          while(Serial_read()!='$');
          i=0;
          while((ch=Serial_read())!='#')
          {
            Start_buff[i] = ch;
             i++;
          }
          Start_buff[i]='\0';
       }
       Serial.println(Start_buff);
      bot.sendMessage(CHAT_ID, Start_buff);
      delay(100);


}
}
void buffer_clear()
{
  for(z=0;z<60;z++)
  {
    Start_buff[z]='\0';
//    textmessage[z]='\0';
```

```
  }
}
void buffer1_clear()
{
 for(z=0;z<5;z++)
  {

   textmessage[z]='\0';

  }

}
//void TEST()
//{
// Serial.begin(9600);
//    Serial.println("Starting TelegramBot...");
//
//    configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
//  client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
//const char* ssid = "vivo1802 ";
//const char* password = "password";
//
//// Initialize Telegram BOT
//#define BOTtoken "5339691493:AAEWdnw4-gZCQEAe2dPmKhRVUOWt15QgC-o" // your
Bot Token (Get from Botfather)
//
//  // Attempt to connect to Wifi network:
//  Serial.print("Connecting Wifi: ");
//  Serial.println(ssid);
//
//  WiFi.mode(WIFI_STA);
//  WiFi.begin(ssid, password);
//
//  while (WiFi.status() != WL_CONNECTED) {
//    Serial.print(".");
//    delay(500);
// }
//
//  Serial.println("");
//  Serial.println("WiFi connected");
//  Serial.print("IP address: ");
//  Serial.println(WiFi.localIP());
//
//  bot.sendMessage(CHAT_ID, "NIGHT PATROLLING ROBOT");
//
//
////MESSAGE_SEND();
//WAITING();
//
```

```
////while(1) {
////    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
////
////    while (numNewMessages)
////    {
////      Serial.println("got response");
////     handleNewMessages(numNewMessages);
////      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
////
////
////
////    }

// String text = bot.messages[i].text;
//
////    String from_name = bot.messages[i].from_name;
////    if (from_name == "")
////      from_name = "Guest";
//
//    if (text == "START")
//    {
//
//      bot.sendMessage(CHAT_ID, "Led is ON", "");
//    }

//        if (text.equalsIgnoreCase("START")) {              // if the received message is "LIGHT ON"...
//                                    // turn on the LED (inverted logic!)
//           bot.sendMessage(CHAT_ID, "NON INVESSIVE BLOOD GLUCOSE MONITORING...
"); // notify the sender
////         Serial.println(msg.sender.id);
//          WAITING();
//        }
//        else {                                             // otherwise...
//           // generate the message for the sender
//           String reply;
//           reply = (String)"Welcome " + msg.sender.username + (String)". Try START";
//           bot.sendMessage(msg.sender.id, reply);            // and send it
//
//        }
  // }
    // wait 500 milliseconds
    //delay(500);
//}



void handleNewMessages(int numNewMessages)
{
  Serial.print("handleNewMessages ");
```

```
  Serial.println(numNewMessages);

//  for (int i = 0; i < numNewMessages; i++)
// {
//    String chat_id = bot.messages[i].chat_id;
    String text = bot.messages[i].text;

//    String from_name = bot.messages[i].from_name;
//    if (from_name == "")
//      from_name = "Guest";

    if (text == "START")
    {

      bot.sendMessage(CHAT_ID, "Led is ON", "");
    }

//  }
}

///*
//  Rui Santos
// Complete project details at https://RandomNerdTutorials.com/telegram-esp8266-nodemcu-motion-detection-arduino/
//
// Project created using Brian Lough's Universal Telegram Bot Library: https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot
//*/
//
//#include <ESP8266WiFi.h>
//#include <WiFiClientSecure.h>
//#include <UniversalTelegramBot.h>
//#include <ArduinoJson.h>
//
//// Replace with your network credentials
//const char* ssid = "smile123";
//const char* password = "12345678";
//
//// Initialize Telegram BOT
//#define BOTtoken "5344660356:AAHQfgW0d2RadVZtZB_LNIkp7YKJ3vj_nlQ" // your Bot Token (Get from Botfather)
//
//// Use @myidbot to find out the chat ID of an individual or a group
//// Also note that you need to click "start" on a bot before it can
//// message you
//#define CHAT_ID "1302360775"
//
//X509List cert(TELEGRAM_CERTIFICATE_ROOT);
//WiFiClientSecure client;
//UniversalTelegramBot bot(BOTtoken, client);
```

```
//
//const int motionSensor = 14; // PIR Motion Sensor
//bool motionDetected = false;
//
//// Indicates when motion is detected
//void ICACHE_RAM_ATTR detectsMovement() {
//  //Serial.println("MOTION DETECTED!!!");
//  motionDetected = true;
//}
//
//void setup() {
//  Serial.begin(115200);
//  configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
//  client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
//
//  // PIR Motion Sensor mode INPUT_PULLUP
//  pinMode(motionSensor, INPUT_PULLUP);
//  // Set motionSensor pin as interrupt, assign interrupt function and set RISING mode
//  attachInterrupt(digitalPinToInterrupt(motionSensor), detectsMovement, RISING);
//
//  // Attempt to connect to Wifi network:
//  Serial.print("Connecting Wifi: ");
//  Serial.println(ssid);
//
//  WiFi.mode(WIFI_STA);
//  WiFi.begin(ssid, password);
//
//  while (WiFi.status() != WL_CONNECTED) {
//    Serial.print(".");
//    delay(500);
//  }
//
//  Serial.println("");
//  Serial.println("WiFi connected");
//  Serial.print("IP address: ");
//  Serial.println(WiFi.localIP());
//
//  bot.sendMessage(CHAT_ID, "Bot started up", "");
//}
//
//void loop() {
//  if(motionDetected){
//    bot.sendMessage(CHAT_ID, "Motion detected!!", "");
//    Serial.println("Motion Detected");
//    motionDetected = false;
//  }
//}
```

CHAPTER 6

EXPERIMENTAL RESULTS

Working of Cradle Results

The real-time information from the infant cradle is transmitted to the mobile app, including the infant's temperature, humidity, movement, and sound. The sensors included in the model are responsible for passing information to be driven to the outputdevices for the action of the baby which is presented by the output devices used in the model from where the parents can know their baby presence what exactly the baby is doing it could be viewing with camera or knowing the wetness in the cradle, in case theinfant needs soothing, the parent can activate the cradle's swinging motion. The app alsosends a warning to the parent if any of the sensor systems malfunction. This system canbe connected to the mobile phones and can have control over the cradle and the baby when parents cannot be along baby all the time.

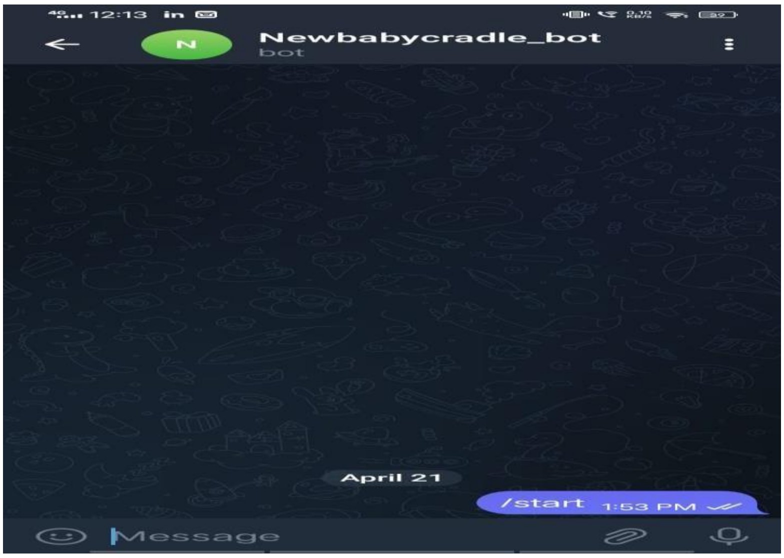

Fig 6.1: Display of Message

Display of the mobile app

In this section, the outcomes of the advanced infant monitoring and facial recognition system under consideration are presented by evaluation from the sensor the results are send to the application present in the mobile from where the parents can know the infant's status right from their mobile phones.

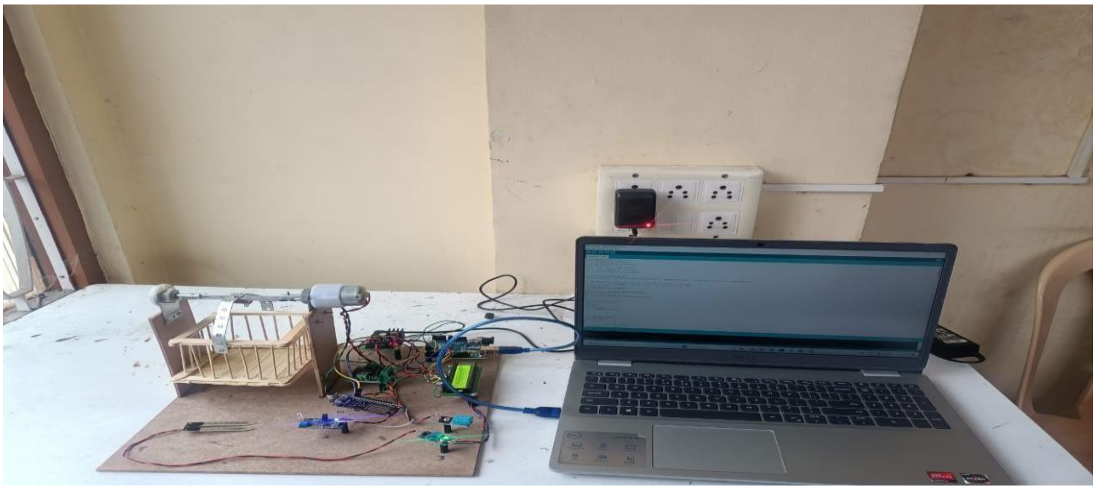

Fig 6.2: Execution of Code and Hardware

First, the components of the hardware system were successfully linked. The connection between the Arduino-Board-Board Mega and all sensors was established. A 12V input was initially applied to the Arduino-Board-Board, then a parallel connection was made between the Arduino-Board-Board and the DC-DC buck converter. By using the buck converter, the voltage was reduced from 12V to 5V, as all sensors and modules, except the Wi-Fi module, function within the voltage range of 3.3-5V.


\

Fig 6.3: Results Performed

Result of Moisture Sensor:

A wet sensor was used to monitor the diaper's wetness by creating a small electrical connection with the moisture or water, causing the circuit to short and sending a signalto the Arduino-Board-Board. The infant's diaper wetness sensor is functioning correctly.

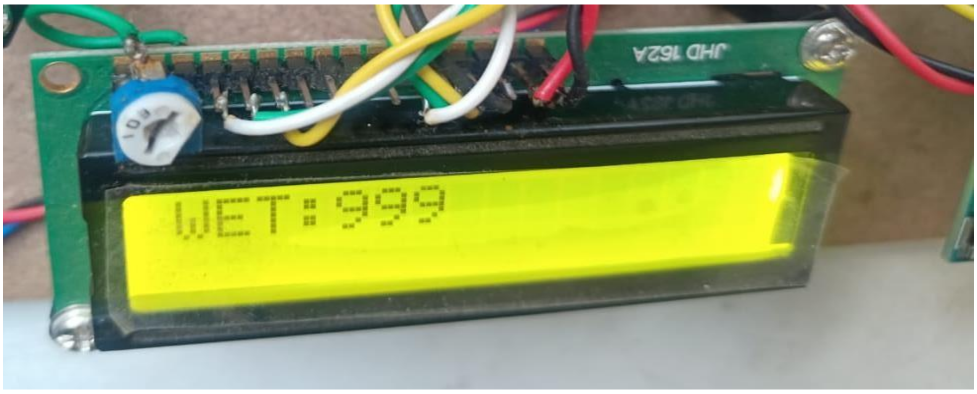

Fig 6.4: Moisture Sensor Working

Result of Heart Beat Sensor

Fig 6.5: Heartbeat Sensor Working

In the baby cradle project with a heartbeat sensor, the heart rate of the baby is measured using the sensor and is displayed only on an LCD screen. The code reads the analog output of the heartbeat sensor and processes the signal to determine the number of heart beats per minute. This information is then displayed on an LCD screen for the caregiver to monitor thebaby's heart rate. The LCD screen is used as the primary output device, and the heartbeat rate is not transmitted or stored anywhere else.

Fig 6.6: Result of heart beats rate

Fig 6.7: Output Temperature

Temperature sensor can measure the ambient temperature around the baby and send an alert if the temperature exceeds a certain limit. This helps to ensure that the baby is comfortable and safe in the cradle. Additionally, the temperature readings can also be displayed on an LCD screen for easy monitoring.

Result of Sound Sensor

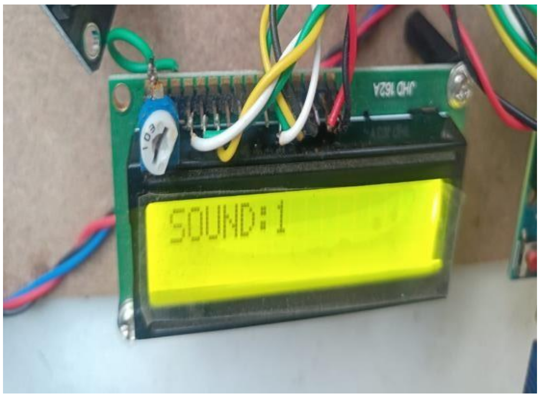

Fig 6.8: Output of Sound Sensor

The sound sensor in the baby cradle project is used to detect if the baby is crying or making noise. It triggers the cradle to start rocking automatically and sends a notification to the parent's phone using the Telegram bot.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

The current implementation of the project, the conclusions that have been drawn are discussed in this chapter. Future enhancements are also discussed. 7.1 Conclusion The development of technology has seen an enormous expansion in recent years, with many new and innovative products being introduced to the market that aim to improve the lives of people in various ways. One such example is the automated infant crib, which is a smart and technologically advanced solution that offers peace of mind to working parents by ensuring the safety and wellbeing of their infants. The crib is not only cost-effective, but also has a range of advanced features that allow for monitoring of the infant's vital signs, such as heart rate, breathing, and movement. This helps parents to ensure that their baby is safe and healthy, even when they are away from home. The automated crib also leverages the power of Wi-Fi technology to provide remote monitoring capabilities, which allows parents to keep an eye on their baby from anywhere in the world. This is especially beneficial for parents who are constantly on the go, and want to be able to check in on their baby from time to time. Furthermore, the system is designed to be highly accessible and easy to use, so that all parents, regardless of their technical expertise, can benefit from this innovative solution. In addition to the automated crib, the proposed IoT-based infant monitoring system using the Arduino-Board and Arduino-Board Mega microcontrollers represents another innovative solution for working parents. This system allows parents to monitor their baby's crying, wet diapers, and presence in the bassinet, and provides real-time notifications in case of any issues. The system also uses face recognition technology to identify the infant and ensure that they are in the crib. This innovative solution provides parents with peace of mind, knowing that their baby is safe and well-cared for even when they are away from home. The development of technology has had a profound impact on society, and the automated infant crib and IoT based infant monitoring system are two examples of how technology can benefit families and improve the quality of life for parents and infants alike. These systems are designed to be cost-effective, accessible, and easy to use, and offer a range of advanced features that help to ensure the safety and well-being of infants.

Future Enhancement

In the future, there are several potential enhancements for a smart cradle using IoT (Internet of Things) technology. Here are a few ideas:

1. Biometric Monitoring: Incorporate biometric sensors into the cradle to monitor vital signs of the baby, such as heart rate, breathing patterns, and body temperature. This data can be transmitted to a mobile app or a centralized monitoring system, enabling parents to keep track of their baby's health in real-time.
2. Sleep Analysis: Implement sleep monitoring features to analyze the baby's sleep patterns, including duration, quality, and sleep cycles. This information can help parents understand their baby's sleep needs and patterns, facilitating better sleep management.
3. Environmental Sensors: Integrate environmental sensors to monitor factors like room temperature, humidity, and air quality. This data can assist parents in creating a comfortable and safe environment for their baby. The cradle can also be programmed to adjust these environmental factors automatically based on the baby's needs.
4. Smart Alerts: Enable the cradle to send alerts and notifications to parents' smartphones or other connected devices. For example, it can notify parents if the baby wakes up, needs a diaper change, or if any abnormal situations are detected, ensuring timely attention to the baby's needs.
5. Remote Access and Control: Develop a mobile app or web interface that allows parents to remotely access and control the cradle. This feature can enable them to soothe the baby by remotely rocking the cradle, playing lullabies, or adjusting the cradle's settings from a distance.

Growth and Development Tracking: Incorporate features to track the baby's growth milestones, suchas weight, height, and feeding patterns. This information can be logged and visualized over time, helping parents monitor their baby's development and share data with healthcare professionals if needed.Integration with Smart Home Systems: Enable integration with existing smart home systems, allowing the cradle to interact with other IoT devices in the house. For example, the cradle can coordinate with smart lights, sound systems, or thermostats to create a calming and comfortable atmosphere for the baby.

6. Parental Insights and Recommendations: Utilize machine learning algorithms to analyze the collected data and provide personalized insights and recommendations to parents. This can include sleep optimization tips, feeding schedules, and suggestions for promoting the baby's well-being and development.

7. Secure Data and Privacy: Implement robust security measures to protect the data collected by the smart cradle, ensuring privacy and confidentiality. Encryption, user authentication, and adherence to data protection regulations can help safeguard sensitive information.

It's important to note that while IoT-enabled smart cradles offer numerous benefits, ensuring the safety, reliability, and privacy of such devices is crucial. Manufacturers should prioritize rigorous testing, adherence to safety standards, and continuous updates to address any potential vulnerabilities.

CHAPTER 8

BIBLIOGRAPHY

[1] Prof.A.D.Anijkaret.al., "General Idea About Smart infant crib" ,"Int.J.of Innovative wisdom andEng., Jan- Feb 2014".

[2] Dr.Andrew Rawicz, Fanchao Yu Liu, Xiago Lu, Kiru Sri, “ Offer for Smart infant crib ”, "Simon Fraser University, 24th jan 2016".

[3] Rajat Arora, Heli Shah, Rohan Arora, “ Smart crib Gear to Ensure Safety of infant in crib ”,"Int.J. of instructional and Futuristic exploration, Mar 2017".

[4] Aquin Nawaz, “ Development of an Intelligent crib for Home and Hospital Use ”, "National Inst. Of Technology, 2015".

[5] Faith Elmas Abdurrahman Yilmaz, Muhammed Garip, “Rocking stir for the infant sleeping on mother's Lap Modeling and prototyping Automatic Swing crib Design ”, "Mechatronics Eng.Dept.Yildiz Tech.University".

[6] SavitaP. Patil ManishaR. Mehtre, “Intelligent infant Monitoring System”, VIT Pune.

[7] "Rachana Palaskar, Shweta Pandey, Ashwini Telang, Akshada Wagh, RameshR. Kagalkar''

[8] Automatic Monitoring And Swinging the infant crib for Infant Care ”."Int.J. of Advanced Research in Computer and Commun.Eng., Dec 2015".

[9] Ronen Luzon, “ child MONITORING SYSTEM ”, "Patent no US 2002/ 0057202AI, May 162002".

[10] CynthiaL. Altenhofen, “ infant MONITORING SYSTEM ”, "Patentno., March 28, 2000".

[11] ”Arduino”,"(Online)Available Https// WWW.arduino.cc, Sep, 2017".

[12] Prof.A.R. Patil, “Smart infant crib an IOT grounded crib Management System”, "2018.

[13] prof. bulletin Anijkaret.al., “General Idea about Smart Baby crib ”, "Int.J. of Innovative Science andEng., Jan- Feb 2014".

[14] Aquib Nawaz, “ Development of an Intelligent crib for Home and Hospital Use ”, "NationalInst.of Technology, 2015".

[15] Tan,L. and Karnjanadecha,M.(2003). "Pitch Discovery Algorithm Autocorrelation Method and AMDF(online) Available From Http//five dots.coe.psu.acth/montri/exploration/Publications/iscit2003_pdapdf(accessed 28/07/2015)".

[16] WWW.amazon.com(2015)."Why Cry Mini BabyCryAnalyzer/ Sleepy Stressed Empty(Online)

http//WWW.amazon.com/WhyCryAnalyzerSleepyStressedirked/dp/B00iALBAD2(Accessed24/02/2015)".

[17] WWW.dx.com(2015)."Wireless Baby Crying Sensor with Maternal Alarm(Online) Available from;http//WWW.dx.com/p/wireless babycryingdetectorwithparentalalarm13394?utm_source=dx &utm_ medium = club & utm crusade = club_skureview_page#. VPD vizqUel(Accessed 24/02/2015)".

[18]WWW.cpsc.gov/en/(2015)."FAQsSafetyStandardfor Children'sToys(online)Availablefromhttp//www.cpsc.gov/Business--Manufacturing/BusinessEducation/ Toy- Safety(penetrated 24/02/2015)".

[19]Wikipedia.org(2015)."unforeseenchilddeatpattern(Onlie)Availablefromhttp//en.wikipedia.org/wiki/sudden_infant_death_syndrome_live/pop up. html(Accessed24/02/2015)