FINAL YEAR PROJECT REPORT

AUTOMATIC BABY CRADLE

ICT 305 2.0 EMBEDDED SYSTEM

AS2019953

K.M.S.I. RATHNAYAKE

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF APPLIED SCIENCE

UNIVERSITY OF SRI JAYEWARDENEPURA

ACKNOWLEDGMENT

For his kind advice, assistance, and helpful recommendations, I would like to express my sincere gratitude to Dr. M. D. R. Perera, Senior Lecturer, Department of Computer Science, Faculty of Applied Science, University of Sri Jayewardenepura, Gangodawila, Nugegoda, Sri Lanka. I pledge to support the embedded system project wholeheartedly.

CONTENT

1 PURPOSE

A baby is nature's best gift to a loving couple. But today, due to the busy life of the parents, it is not possible to spend time with the children. Therefore, they should find a caregiver to take care of their babies and buy a baby cradle to give their little ones better comfort. But parents should call the caregiver when they want to know the status of the child, and there is a possibility that the caregiver is not doing her job properly. Therefore, there is some risk to the child's safety as well as the child's health. Also, there are several types of automated baby cradles in the market that do not provide better comfort and safety for babies.

So, I created an automatic baby cradle which is a baby cradle that supports parents to look after their child with better comfort and good security. This IoT system is designed by aiming at parents who are going out from home for their job. This system is fully automated and interacts with a user-friendly mobile application and web applications for communicating with the baby’s parents with real-time data.

2 INTRODUCTION

The Internet of Things (IoT) is a network of connected computing devices, mechanical and digital machinery, items, animals, or people that may exchange data across a network without requiring human-to-human or human-to-computer interaction.

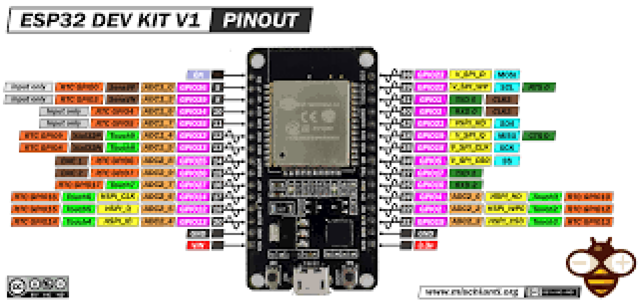
When we add sensors to smart objects, they will become even smarter and offer more alternatives to improve the monitoring process. Smartwatches and other smart devices enable the use of devices that can be worn on the body and linked to the sensor network.

So, this is an automatic IoT baby cradle that can be used to cradle babies of age up to 10 months. Here this project is a Wi-Fi-based project. Other than cradling baby this system fulfills many more functions.

Automatically start the baby's cradle when the baby is crying. After the baby cradle powers on the system, it reads the temperature and humidity values continuously until it powers off. It powers on the fan when the temperature reading increase to 25 Celsius because when the temperature exceeds that value it is unsuitable for the baby. And also, the user can view the humidity level, temperature level, baby cradle state (ON/OFF), and fan state (ON/OFF) through the mobile application and check whether the baby is in a good condition or not. This system provides the facility to monitor the baby through the mobile application whole the time. So, the user does not have to worry about the safety of the baby when they are not around or away from the baby.

3 COMPONENTS

1.1 ESP 32 devkit V1 board



Espressif Systems, the same company that created the well-known ESP8266 SoC, offers the inexpensive ESP32 System on Chip (SoC) Microcontroller. The 32-bit Xtensa LX6 Microprocessor by Tensilica is a replacement for the ESP8266 SoC and features built-in Wi-Fi and Bluetooth. It is available in single-core and dual-core versions. The advantage of ESP32 is that it has inbuilt RF components such as a power amplifier, a low-noise receiver amplifier, an antenna switch, filters, and an RF balun, similar to ESP8266. As a result, it is very simple to construct hardware around the ESP32 since minimal external components are needed. The fact that ESP32 is produced utilizing TSMC's ultra-low-power 40 nm technology is another crucial piece of information to be aware of. Therefore, employing ESP32 should make it very simple to create battery-powered applications like wearables, audio equipment, baby monitors, smart watches, etc.

Specifications of ESP32

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz
- 520 KB of SRAM, 448 KB of ROM, and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16 channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC, and RNG.

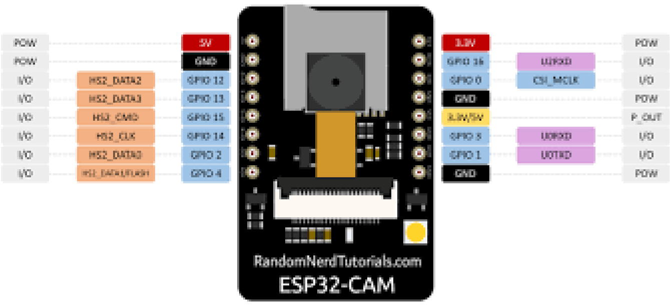
the ESP32 Board consists of the following:

- ESP-WROOM-32 Module
- Two rows of IO Pins (with 15 pins on each side)
- CP2012 USB – UART Bridge IC
- micro–USB Connector (for power and programming)

- AMS1117 3.3V Regulator IC
- Enable Button (for Reset)
- Boot Button (for flashing)
- Power LED (Red)
- User LED (Blue – connected to GPIO2)
- Some passive components

An interesting point about the USB-to-UART IC is that its DTR and RTS pins are used to automatically set the ESP32 into programming mode (whenever required) and also rest the board after programming

1.2 ESP 32 CAM module



ESP32-CAM is a low-cost ESP32-based development board with an onboard camera, small in size. It is an ideal solution for IoT applications,

prototype constructions, and DIY projects. The board integrates Wi-Fi, traditional Bluetooth, and low-power BLE, with 2 high-performance 32-bit LX6 CPUs.

Specifications of ESP32 cam module

- 802.11b/g/n Wi-Fi
- Bluetooth 4.2 with BLE
- UART, SPI, I2C and PWM interfaces
- Clock speed up to 160 MHz
- Computing power up to 600 DMIPS
- 520 KB SRAM plus 4 MB PSRAM
- Supports WiFi Image Upload
- Multiple Sleep modes
- Firmware Over the Air (FOTA) upgrades possible
- 9 GPIO ports
- Built-in Flash LED

Camera specification

The ESP32-CAM includes an OV2640 camera module. The device also supports OV7670 cameras. The OV2640 has the following specifications:

- 2 Megapixel sensor
- Array size UXGA 1622×1200
- Output formats include YUV422, YUV420, RGB565, RGB555 and 8-bit compressed data
- Image transfer rate of 15 to 60 fps

1.3 Sound sensor

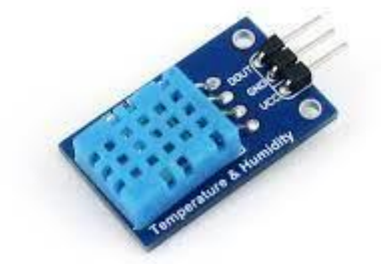
The sound sensor is a simple device that can detect sound. The sound sensors are very simple to use. The sound sensor consists of a Microphone as a transducer, a potentiometer to adjust the intensity, an LM386 low-power audio amplifier, LED, and other passive components like resistors and capacitors.

This sensor includes 3 pins, and they are,

- Pin1 (VCC): 3.3V DC to 5V DC
- Pin2 (GND): This is an aground pin
- Pin3 (OUT): This is an output pin. It provides a high signal when there is no sound and goes LOW when sound is detected.

You can connect it to any digital pin on an Arduino or directly to a 5V relay or similar device.

1.4 DHT 11 sensor



The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins needed). It’s fairly simple to use but requires careful timing to grab data.

Specifications

- 3 to 5V power and I/O
- 2.5mA max current use during conversion (while requesting data)
- Good for 20-80% humidity readings with 5% accuracy
- Good for 0-50 °C temperature readings +-2 °C accuracy
- No more than 1 Hz sampling rate (once every second)
- Body size 15.5mm x 12mm x 5.5mm
- 4 pins with 0.1" spacing
- RoHS compliant

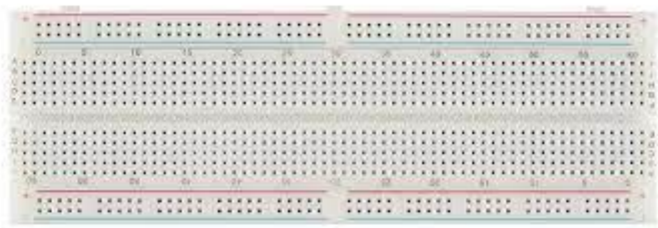
1.5 Servo motor

A Servo motor (or servo) is a rotary actuator that allows for precise control of angular position, velocity, and acceleration. Servos are found in many places, from toys to home electronics to cars and airplanes. Servos also appear behind the scenes in devices we use every day. Servos come in a variety of shapes and sizes for different applications. You may want a large, powerful one for moving the arm of a big robot, or a tiny one to make a robot's eyebrows go up and down. By linking many of these servos together, you can create robots that perform complex real-world operations.

1.6 Fan

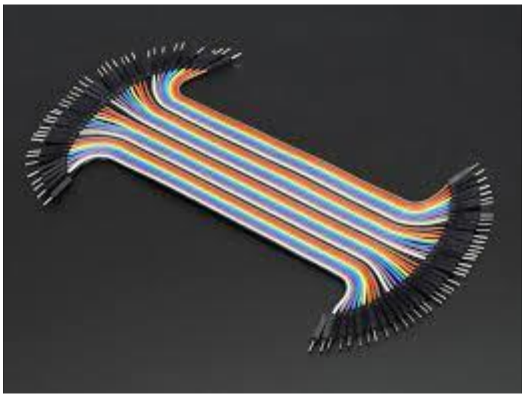
Direct current fans, or DC fans, are powered with a potential of fixed value such as the voltage of a battery. Typical voltage values for DC fans are 5V, 12V, 24V, and 48V. In contrast, alternating current fans, or AC fans, are powered with a changing voltage of positive and of equal negative value.

1.7 Breadboard



A breadboard (sometimes called a plug block) is used for building temporary circuits.It is useful to designers because it allows components to be removed and replaced easily. It is useful to the person who wants to build a circuit to demonstrate its action, then to reuse the components in another circuit.

1.8 Jumper wires



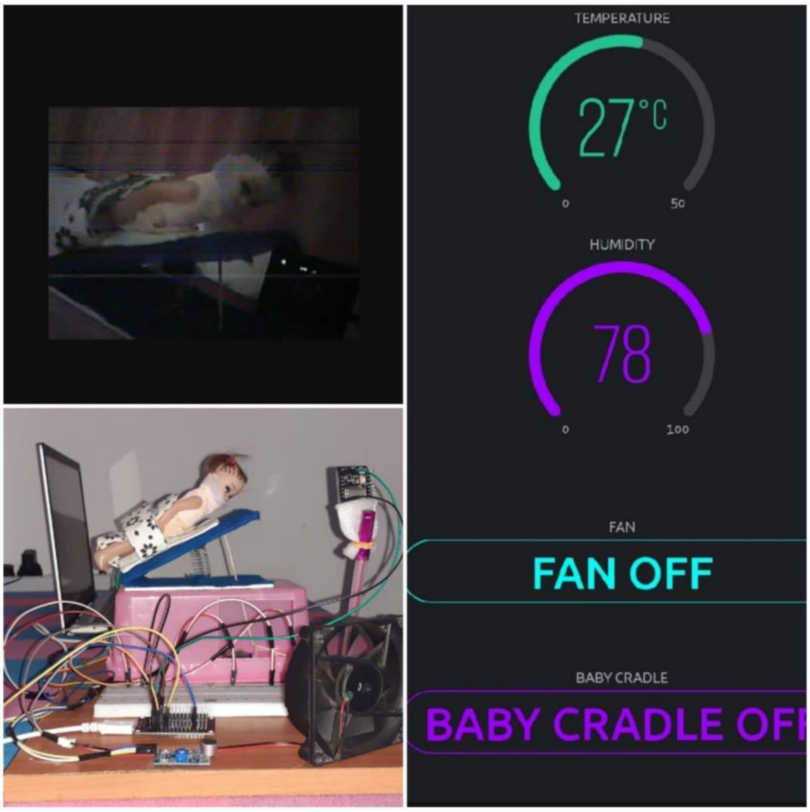
Jumper wires are electrical wires with connector pins at each end. They are used to connect two points in a circuit without soldering. You can use jumper wires to modify a circuit or diagnose problems in a circuit. Although jumper wires come in a variety of colors, they do not actually mean anything. The wire color is just an aid to help you keep track of what is connected to which. It will not affect the operation of the circuit. This means that a red jumper wire is technically the same as the black one. Even so, the colors can be used to your advantage to differentiate the types of connections. For instance, red as ground and black as power.

Types of jumper wires

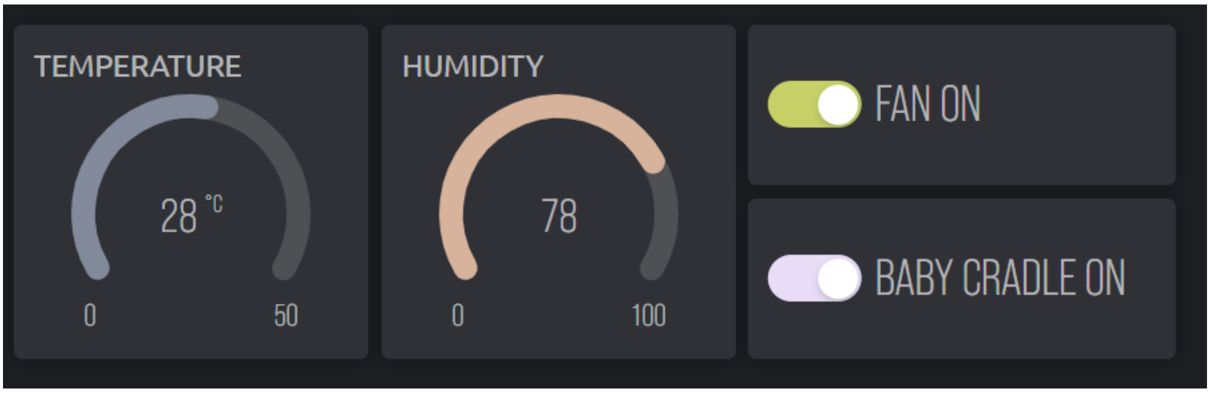
- Male-to-male jumper
- Male-to-female jumper
- Female-to-female jumper

1.9 Newly designed baby cradle with mobile application and web application

This is a newly designed baby cradle prototype that replaces the mother's shoulder used to lean the baby on her shoulder when cradling with many other facilities. Furthermore, I put the DHT 11 temperature and humidity sensor inside the cradle's mattress to get accurate sensor readings So, this system is the best solution for busy parents.



Mobile application

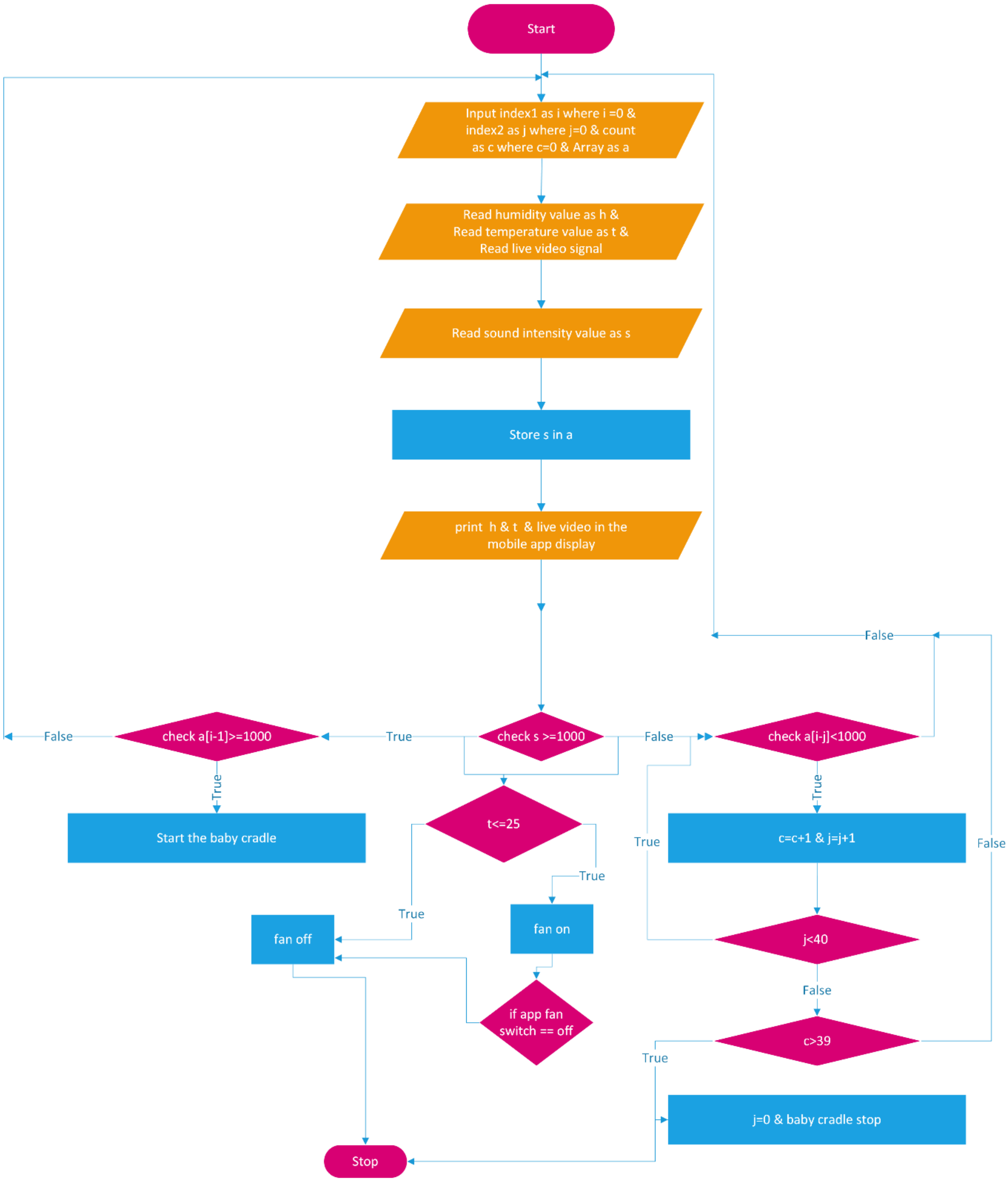


Web application

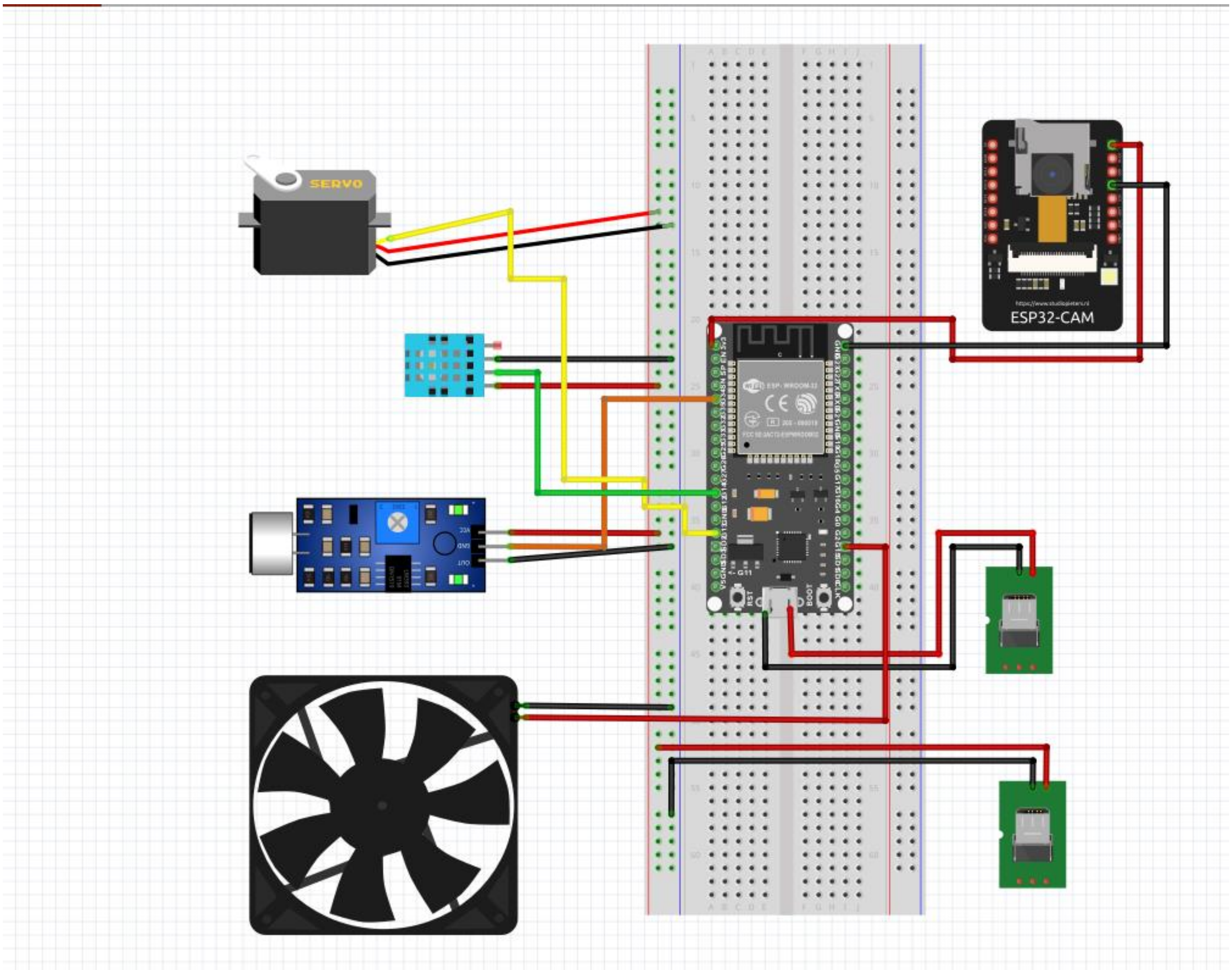
This is my mobile application which is created by using the Blynk console it already has the mobile application and web application. So by using this parents can see their lovely child's condition and their activities through the video screen.

4 DESIGN OVERVIEW

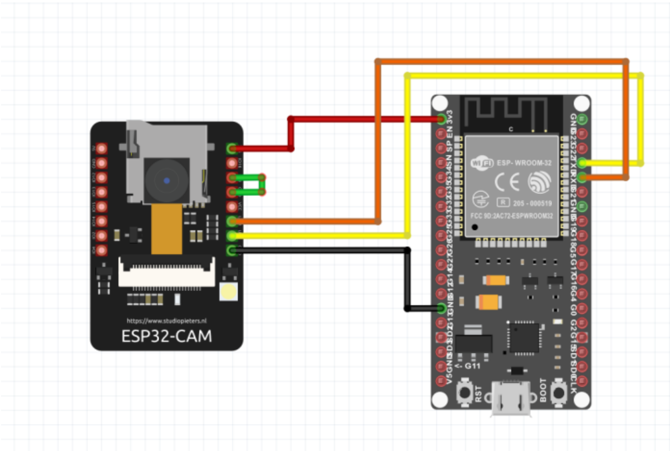
4.1 FLOW CHART



4.2 SIMULATE CIRCUIT DIAGRAM



4.2.1 ESP32 cam module and esp32 board setup circuit diagram



4.3 CODE

```
//AS2019953
//K.M.S.I.RATHNAYAKE

#define BLYNK_PRINT Serial

//FIRMWARE CONFIGURATION OF BLYNK
#define BLYNK_TEMPLATE_ID "TMPL-cO4N03r"
#define BLYNK_DEVICE_NAME "FINAL PROJECT"
#define BLYNK_AUTH_TOKEN "qG03sgcneyH8r-GmSCfdn3AGhtAv1BVh"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <Servo.h>
#include <SPI.h>

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Imalsha";
char pass[] = "12345678";

//define pins
int SOUNDPIN = 34;
#define SERVOPIN 13
#define DHTPIN 14
#define FANPIN 15
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
Servo myservo;

//define variabls
int pos = 0;    // variable to store the servo position
int sensorValue = 0;  // variable to store the value coming from the sensor
int  arrayList[1000] = {}; //declare an array to store sensor values
int stop_sound_count = 0;
int start_sound_count = 0;
static int count = 0;
static int intensity = 1020;

//function which detect sound and swing the baby cradle
void soundSensor() {
  sensorValue = analogRead(SOUNDPIN);// read the value from the sensor
  Serial.print("Sound Sensor Value:");
  Serial.println(sensorValue);
  arrayList[count] = sensorValue;

  if (sensorValue >= intensity) {
    Blynk.virtualWrite(V3, 1);
    for (pos = 0; pos <= 180; pos++) { // goes from 0 degrees to 180 degrees
      myservo.write(pos);              // tell servo to go to position in variable 'pos'
      delay(15);                       // waits 15 ms for the servo to reach the position
    }
    for (pos = 180; pos >= 0; pos --) { // goes from 180 degrees to 0 degrees
      myservo.write(pos);              // tell servo to go to position in variable 'pos'
      delay(15);                       // waits 15 ms for the servo to reach the position
    }
  }
```

```
 else{
  for (int i = 1; i <= 40; i++) {
   if (arrayList[count - i] < intensity) {
    stop_sound_count++;
   }
   if (stop_sound_count >39) {
    Blynk.virtualWrite(V3, 0);
    delay(5000);
   }
  }
  stop_sound_count=0;
 }
 count++;
}
//end function

//function which detect the temperature and humidity and also on/off the fan
void sendTemperatureSensor()
{
 float h = dht.readHumidity();
 float t = dht.readTemperature();
 if (isnan(h) || isnan(t)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
 }
 Serial.print("humidity:");
 Serial.println(h);
 Blynk.virtualWrite(V1, h);
 Serial.print("temperature");
 Serial.println(t);
 Blynk.virtualWrite(V0, t);
```

```
if (t > 25) {
    digitalWrite(FANPIN, HIGH);
    Blynk.virtualWrite(V2, 1);
  }
  else {
    digitalWrite(FANPIN, LOW);
    Blynk.virtualWrite(V2, 0);
  }
}
//end function

//setup function
void setup() {
  pinMode(SERVOPIN, OUTPUT);
  pinMode(FANPIN, OUTPUT);
  Serial.begin(115200);
  Blynk.begin(auth,ssid,pass);
  myservo.attach(SERVOPIN);
  dht.begin();
}
//end function

//loop function
void loop() {
  sendTemperatureSensor();
  soundSensor();
  Blynk.run();
}
//end function
```

5 BUDGET

Tool	Number of pieces	Amount	Total price
ESP32 devkit V1 board	1	2050.00	2050.00
ESP32 camera module	1	2200.00	2200.00
Sound sensor	1	200.00	200.00
DHT11 sensor	1	530.00	530.00
Micro (9g) Servo motor	1	500.00	500.00
Two wire fan	1	620.00	620.00
Breadboard	1	340.00	340.00
Jumper wire set	3	280.00	840.00
USB cable	2	500.00	1000.00
Hinge	1	50.00	50.00
Spring	1	50.00	50.00
Doll	1	350.00	350.00
Glue stick	2	200.00	400.00
Double tape	1	150.00	150.00
Cack board	2	120.00	240.00
Total cost			9520.00

6 ISSUES

This automated system was built by using ESP32 devkit v1 which is a controller board. Hence, I cannot get the most correct output from this. There are several problems I had to face.

- There is a sound sensor to get the baby crying noise, so in creating the prototype I placed it under the baby's mattress near the baby's mouth to get more accurate values. But when I run the simulation the sound sensor detected the servo motor Cogwheels sound hence, I cannot stop the baby's cradle when the baby stops crying. So as a solution I remove the sound sensor from the mattress and place it on the board. But it still read the servo motor cogwheels sound.

- In this project, I used a mobile application, so I used the Blynk platform to create my mobile application. But in my Blynk account, could not give permission to use a video streaming gadget because my Blynk account is a free account so to do this I want to upgrade it to the beta version. So, as a solution, I create a mobile application without a video streaming gadget, and I show it in google chrome.

- I create the baby cradle which has a new architecture designed by me that can swing with rhythm after being connected to the Blynk simulator it displays some small delay time, but we can neglect it.

- In this prototype I used a fan to give better comfort to the baby but when running the prototype, it could not work every time at that time I give a small push to the fan wing then it works properly.

7 DISCUSSION

This is an automated IoT Baby Cradle that gives better comfort and security to the child and good satisfaction to their parents. Under the lecture's guidance, I design a new architecture for this project which is to replace the mom's shoulder and swing with rhythm, so it gives better comfort to the baby.

Here I use an external power supply to power up all the sensors and motors to give good condition to my project's all components

Here I used a mobile application and web application to monitor the baby's condition and activities which is called Blynk. By using this platform, I created my mobile application easily. But I am using the free version of it so it does not give permission to create a video streaming part, but I can get it by upgrading to the beta version.

Here I used an ESP32 cam module to stream the baby's activity via Wi-Fi. But sometimes it does not give better quality like showing some lines in the display, gray color dots like that.

8 CONCLUSIONS

An automated IoT baby cradle that gives better comfort and security to the baby and communicates data with their parents via Wi-Fi. The system will swing rhythm when the baby starts crying and check the baby's condition if it is not good then power up the fan from the beginning until the power is off the cradle and always notify the parents via a mobile application. So, the system works efficiently avoiding the breaking rhythm of the cradle. So, the project is successfully completed.

8.1 FUTURE IMPLEMENTATION

So, in this project, we can implement the sound detection part by using AI and Machine Learning because it does not work properly. The sound sensor detects the intensity of the sound either 0 or 1 in digital read and it does not get only the child's sound it also gets the environmental noises. so we want to develop it to identify the baby's sounds.

We can use Blynk upgraded version to create a mobile app with the video streaming gadget.

9 REFERENCE

https://www.youtube.com

https://www.wikipedia.org

https://community.blynk.cc