Smart Cradle System

PR1101- Automation Project

Final Report

PREPARED BY

Bandaru Eshwar (2020BTechCSE018)

Koppunoor Bhanu Prakash Reddy (2020BTechCSE041)

Bhupathi Sanjay Kumar (2020BTechCSE096)

FACULTY GUIDES

Dr. Hanuman Prasad Agarwal

Mr. Divanshu Jain

Department of Computer Science engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University Jaipur

NOVEMBER 2022

CERTIFICATE

This is to certify that the project work entitled "**Smart Cradle System**" submitted by **Bandaru Eshwar (2020BTechCSE018), Koppunoor Bhanu Prakash Reddy(2020BTechCSE041), Bhupathi Sanjay Kumar (2020BTechCSE096),** towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science Engineering** of JK Lakshmipat University Jaipur is the record of work carried out by them under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted.

Dr. Hanuman Prasad Agarwal
Assistant Professor
Department of Electrical
&Electronics Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Mr. Divanshu Jain
Assistant Professor
Department of Electrical
& Electronics Engineering
Institute of Engineering & Technology (IET)
JK Lakshmipat University Jaipur

Date of Submission:

ACKNOWLEDGEMENTS

We would like to express our heartfelt gratitude to Dr. Hanuman Prasad Agarwal, Assosiate Professor, Institute of Engineering & Technology and Mr. Divanshu Jain, Assosiate Professor, Institute of Engineering & Technology for their valuable time and guidance that made the project work a success.

Sincerely yours,

Bandaru Eshwar (2020BTechCSE018)

Koppunoor Bhanu Prakash Reddy(2020BTechCSE041)

Bhupathi Sanjay Kumar (2020BTechCSE096)

ABSTRACT

The main objective of the project to design Smart Cradle system. Because the society consists high number of working couples. It is becoming difficult for them to take care of their baby and the job. So, the need of these kind of smart devices is very high for this society.

TABLE OF CONTENTS

INTRODUCTION

PROBLEM STATEMENT:

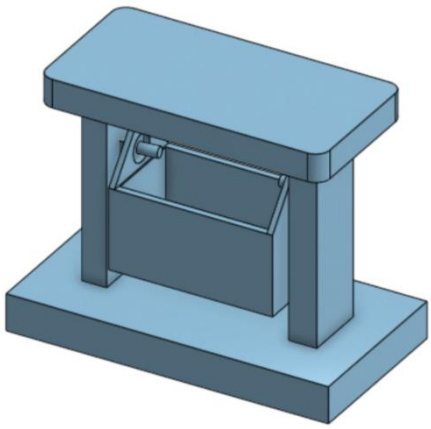
It’s becoming very hard to handle both baby and work at the same time. We also have baby takers for taking care of children. But many frauds are detected as we see in newspapers etc.

BACKGROUND:

Today's lifestyle is changing rapidly. most of the work. Parents find it a little difficult to cope with work Babysit. I can't watch over my child all the time. It’s hard after working long hours. to soothe the baby Manual mount rotation may not be possible such cases. Then with the help of a babysitter Thoughts about baby safety also come to mind. Therefore, you need a product that fills this gap between parent and baby. As far as time passes there had been lot of fraud detected in these kinds of centres from newspaper and news channels, we get the awareness about it. We are creating a smart cradle it contains the feature of temperature monitoring, toy control, moisture monitoring and camera integration. For leisure time such as lunch or tea break. A monitoring system that allows you to monitor your baby's condition in real time. Time has been proposed to resolve these issues. baby monitor. A system consisting of a video camera and a microphone. Limitation of Coverage. Ready to send data. Inform parents of urgent situations, thereby shortening them. The time it takes to deal with such scenarios. baby in general. Crying because you're hungry, tired, sick, or need diapers changed. So, they go home and check on the baby. Also called sudden infant death syndrome (SIDS). Death in crib because more babies die from SIDS in their crib. It occurs in infants under 12 months of age. Most SIDS deaths are in infants under 6 months of age Year. Experts still don't know what causes SIDS, but However, you can reduce the risk by putting your baby to sleep on a hard floor. surface (children's mattress). In addition, babies should not sleep on a pillow or other soft surface. Researchers I don't understand why sleeping on such a surface increases the risk. But they warn that it can be dangerous. For example, in 2003 it was shown to lull infants to sleep. Seemed to be on soft ground instead of hard ground Five times his risk of SIDS. overheat too should be avoided while sleeping. babies need to keep warm while sleeping, but not at extreme temperatures warm. The risk of SIDS increases in winter and cold weather, because parents’ layer or underlay their babies. The ceiling becomes heavy and overheats. For this reason, yes, if the room temperature is comfortable for an adult Also suitable for babies.

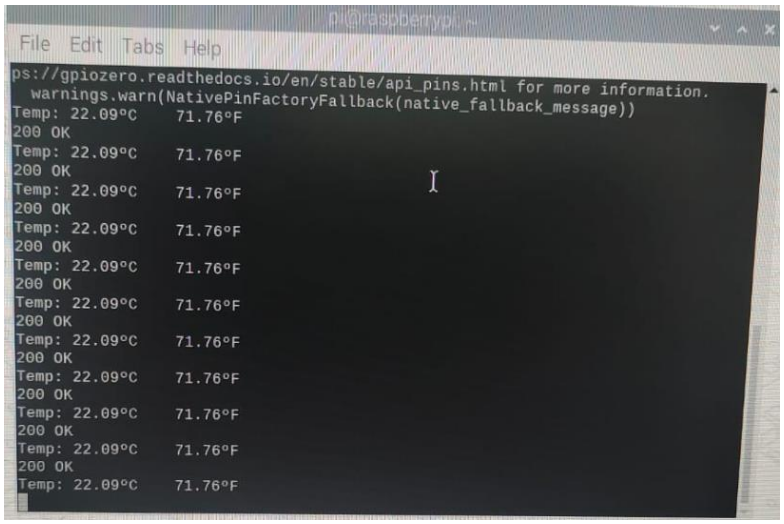
METHODOLOGY

The Internet of Things (IoT) is about connecting real-time data to people who use that data to grow their product portfolios. Both now and in the near future, his IoT mission to make people's lives more convenient continues to be needed. To make the smart cradle we have designed the wooden frame in such a way that all circuit setups can be put far from the baby and the baby can feel comfortable. The 3d model of the prototype is shown the below figure.

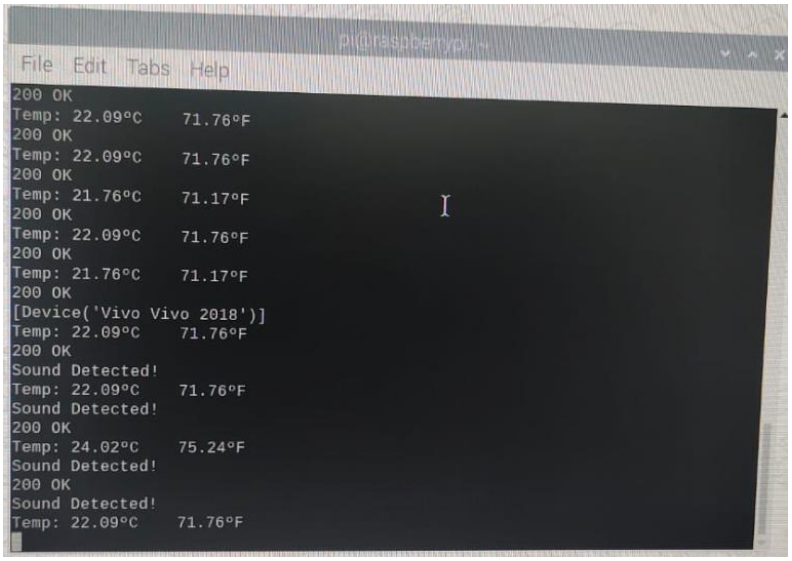


In the smart cradle we have attached the top section for pcb allocation, and the top section contain holes so that the dc motor can go inside. We have kept the servo motor either side of cradle so that the cradle can be swing. The final wood frame setup has been shown in below figure.

The smart cradle consists of features like temperature monitoring, camera livestream, moisture detection, toy control and Auto swing by sound detection. When baby cries the sound of baby detected by sound sensor module and the servo motor start running.so that the cradle start swinging. We have put the LM35 Sensor which detects the temperature of the room. When the temperature rises to some threshold, the DC motor connected through L293D motor driver which helps the motor to start rotating, so that the baby cannot feel uneasy due to room temperature. Here, we can see the room temperature in the serial monitor.

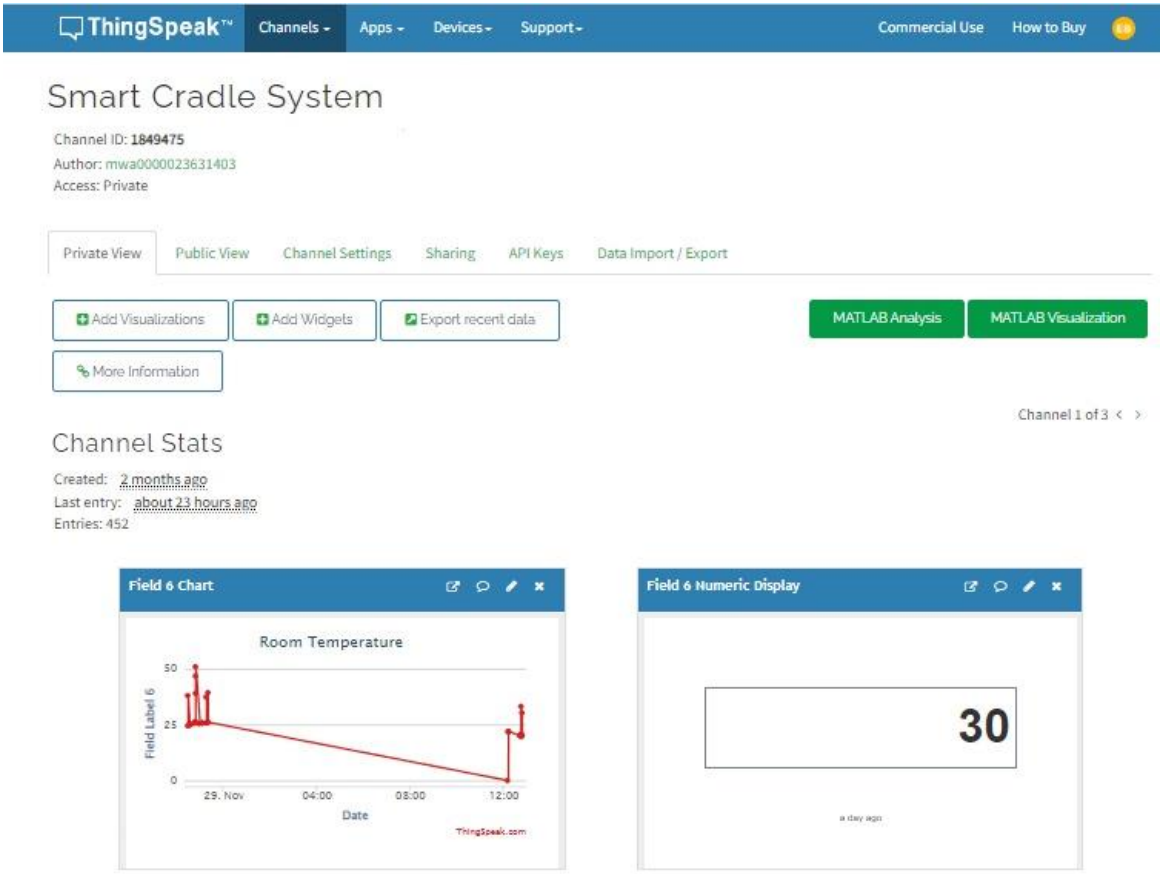


We also have the feature of moisture detection, when the baby makes the bed wet, the sensor detects the moisture and sends the notification to the parent. As we discussed about the 24/7 live stream camera is available, we have the setup of pi camera connected to the raspberry pi 4. The continuous monitoring can be possible for the parent. The Toy control is the DC motor connected to L293D motor driver. When the sound detected by the sound sensor the toy start rotating. So that it helps the baby to calm down and goes into the sleep. Here, we cand see that Sound has been detected in the serial monitor too.

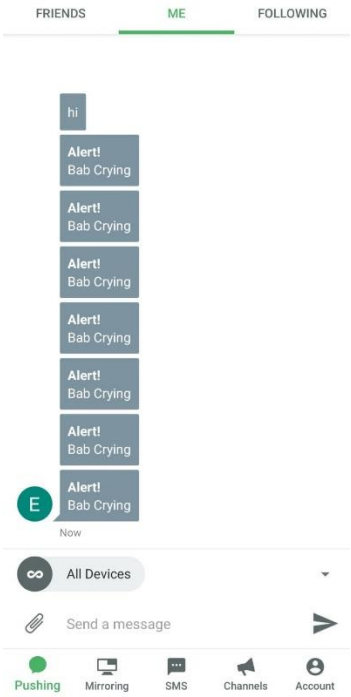


We have connected the cradle to the MATLAB Thing Speak server. ThingSpeak is a platform that offers a variety of services aimed solely at building IoT applications. It provides real-time data collection, graphical visualization of collected data, and the ability to create plugins and apps to work with web services, social networks, and other APIs. Each of these features is described in detail below.

The core element of Thing Speak is the Thing Speak Channel. Channels store data to send to Thing Speak and consist of the following elements:
Eight fields for storing any type of data - These can be used to store data from sensors or embedded devices.
3 location fields - can be used to store latitude, longitude, and altitude. These are very useful for tracking moving devices.
1 Status field - A short message describing the data stored in the channel.
To use Thing Speak, you need to sign up and create a channel. After creating a channel, you can send data and have Thing Speak process and retrieve it. Sign up, set up your channel, and start exploring Thing Speak.



And we used the Push bullet Server for sending the notification to the Parent. Push bullet is a server that sends the notification to the selected devices that sense the data from the microcontroller and sends the data from the sensors to the user as the notification.



The livestream of the pi camera is also done through the cloud. Here the figure shown below is the snapshot of the camera streaming.

This server helps the user to monitor the cradle from their office.

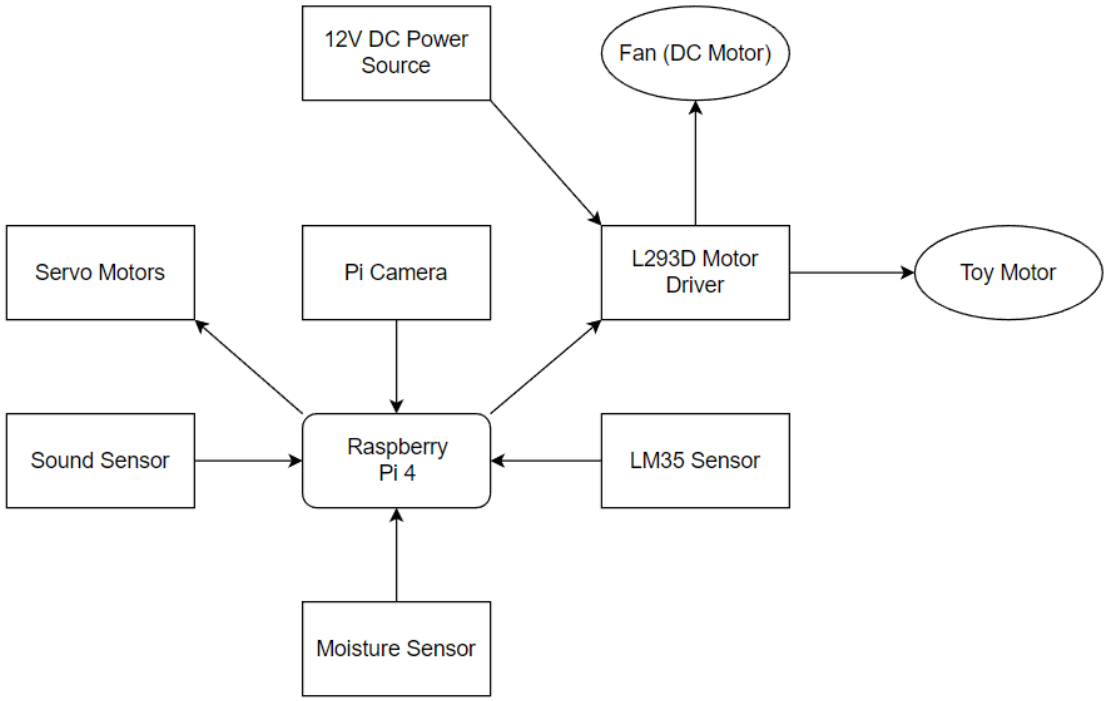
HARDWARE:

1. **Raspberry Pi 4 board:** A single board computer with wireless LAN and Bluetooth connectivity
2. **Raspberry Pi Camera Module:** Pi Camera module is a camera which can be used to take pictures and high-definition video.
3. **LM35 Temperature Sensor:** It is used to measure temperature with an electrical o/p comparative to the temperature (in °C). In our project, we have used to detect the room temperature.
4. **Sound Sensor Module:** A module that detects sound waves through its intensity and converting it to electrical signals. In our project, we have used this to detect the cry sounds of the baby.
5. **Moisture Sensor:** It is used to measure if there is any moisture detected
6. **12V DC Motor:** A DC motor is defined as a class of electrical motors that convert direct current electrical energy into mechanical energy. We have connected this with the LM35 Sensor in which if the temperature rises above 25^0C, then the dc motor should on.
7. **Toy Motor:** A toy motor is an electrical machine that converts electrical energy into mechanical energy. These toy motors are great for DIY hobbies, Science Projects. We have used this to rotate a toy when the baby cries.
8. **L293D Motor Driver Module:** Motor driver acts as an interface between the motors and the control circuits.
9. **SG90 Servo Motor:** Servo motors or "servos", as they are known, are electronic devices and rotary or linear actuators that rotate and push parts of a machine with precision. We have used this for the movement of the Cradle.
10. **12V DC Power Source:** A 12VDC power supply is a device that supplies electrical energy to a load. We have used to give an external voltage for the DC Motors.
11. **MCP3208 (ADC):** The MCP3208-CI/P is an 8 channel, 12bit Analogue to Digital Converter (ADC) with SPI interface in 16 pin DIP package. This ADC combines high performance and low power consumption in a small package by making it as an ideal for embedded control applications.
12. **Resistor:** A resistor is an electrical component that controls or restricts how much electrical current can pass across a circuit in an electronic device.

SOFTWARE:

1. Python 3
2. Raspbian OS
3. HTTP Client
4. URL Client
5. Push Bullet
6. Thing Speak

HARDWARE DIAGRAM:



CONCLUSION

Baby care is a challenging issue everywhere. Given that they represent our future, it is a very vital role. Although a mother's lap is ideal for a baby, this system was created with the needs of the modern world in mind as well as the importance of baby care. In today's busy world, an automatic baby monitoring system is the greatest way for parents to keep an eye on their children. It is merely a method of utilising contemporary technology, with little impact on the parents' regular routines. Our goal, as stated in the introduction, is to create a baby monitoring system. The Raspberry Pi 4 B+ module, a credit card-sized microcomputer that is used to create this system and has many advantages over microcontrollers and Arduino, is used.
The parents' monotony and anxiety can be reduced with effective usage of this technique. This system also supports the baby's safety concern. Even though this system is now in use, it can still be modified and improved.

Demonstration of our Prototype Video Link:
Smart Cradle System(Baby_Suraksha).mp4

REFERENCES

1. Karanjkar and R. Kumawat, “Design of a Smart Baby Cradle Using Blynk and Local Customer Priorities”, sms, vol. 14, no. 02, pp. 159-165, Jun. 2022.
2. Iarjset.com. 2022. [online] Available at: https://iarjset.com/wp-content/uploads/2020/01/IARJSET.2020.7108.pdf
3. W. A. Jabbar, H. K. Shang, S. N. I. S. Hamid, A. A. Almohammedi, R. M. Ramli and M. A. H. Ali, "IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle," in IEEE Access, vol. 7, pp. 93791-93805, 2019, doi: 10.1109/ACCESS.2019.2928481.
4. S. Joseph, A. Gautham.J, A. Kumar and M. K. Harish Babu, "IOT Based Baby Monitoring System Smart Cradle," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021, pp. 748-751, doi: 10.1109/ICACCS51430.2021.9442022.
5. Bitelectronicsystem.com. 2022. [online] Available at: https://bitelectronicsystem.com/wp-content/uploads/2021/07/IOT-based-smart-cradle-system-with-an-app-for-baby-monitoring.pdf
6. Ieeexplore.ieee.org. 2022. Analysis on IoT Based Smart Cradle System with an Android Application for Baby Monitoring. [online] Available at: https://ieeexplore.ieee.org/abstract/document/9063773
7. Ieeexplore.ieee.org. 2022. Development of an IoT based Smart Baby Monitoring System with Face Recognition. [online] Available at: https://ieeexplore.ieee.org/abstract/document/9454187
8. A.B.TUPKAR, Rushikesh patmase, Sachin Waghade, Sahil Meshram, Wrusham Ambadale, Rajat Thakre, "AN APPROACH ON DEVELOPMENT OF AUTOMATIC BABY CRADLE", International Journal Of All Research Writings, vol. 1, no. 11, pp. 139-144, 2020.
9. Scholar.archive.org. 2022. [online] Available at: https://scholar.archive.org/work/c4lipus4kfcu3hxqmgdxxbaesq/access/wayback/https://ebooks.iospress.nl/pdf/doi/10.3233/APC210268
10. “Doi: 10.17148/iarjset.2021.8781 baby monitoring system for smart cradle ...” [Online]. Available: https://iarjset.com/wp-content/uploads/2021/08/IARJSET.2021.8781.pdf
11. R. Palaskar S. Pandey A. Telang A. Wagh R. Kagalkar "An Automatic Monitoring and Swing the Baby Cradle for Infant Care" International Journal of Advanced Research in Computer and Communication Engineering vol. 4 no. 12 pp. 187-189 2015.
12. "Camera Module - Raspberry Pi" Raspberry Pi 2016 [online] Available: https://www.raspberrypi.org/products/camera-module/

APPENDECIES

Python Code:

```
from gpiozero import MCP3008
from time import sleep
import RPi.GPIO as GPIO
import time
import http.client as httplib
import urllib
from pushbullet import Pushbullet

key ="1EPX63IN172YAT79"

control = [5,5.5,6,6.5,7,7.5,8,8.5,9,9.5,10]
servo1 = 7
servo2 = 13
channel = 11
channel1 = 3
in1 = 18
in2 = 16
en = 22

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(channel, GPIO.IN)
GPIO.setup(channel1, GPIO.IN)
GPIO.setup(servo1,GPIO.OUT)
GPIO.setup(servo2,GPIO.OUT)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(en,GPIO.OUT)
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
p=GPIO.PWM(en,1000)
p.start(25)
# Set up channel number and SPI chip select device
reading = MCP3008(channel=0)

p1=GPIO.PWM(servo1,45)# 50hz frequency
p2=GPIO.PWM(servo2,45)# 50hz frequency

p1.start(0)# starting duty cycle ( it set the servo to 0 degree )
p2.start(0)# starting duty cycle ( it set the servo to 0 degree )


def callback(channel):
    if GPIO.input(channel):
        pb = Pushbullet("o.H5gGESQCkco3YxVGzsdG21o6M03g8njv")
        print(pb.devices)
```

```
        dev = pb.get_device("Vivo Vivo 2018")
        push = dev.push_note("Alert!","Baby Crying")
        time.sleep(1)
        while True:
            print("Sound Detected!")
            #p1.start(3)# starting duty cycle ( it set the servo to 0 degree )
            #p2.start(3)
            #p1.ChangeDutyCycle(3)
            #p2.ChangeDutyCycle(3)
            for x in range(11):
                p1.ChangeDutyCycle(control[x])
                p2.ChangeDutyCycle(control[x])
                time.sleep(0.06)
            for x in range(9,0,-1):
                p1.ChangeDutyCycle(control[x])
                p2.ChangeDutyCycle(control[x])
                time.sleep(0.06)

    else:
        print("Sound Not Detected")

    time.sleep(1)
        #print (x)

GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=225)  # let us know when
the pin goes HIGH or LOW
GPIO.add_event_callback(channel, callback)  # assign function to GPIO PIN, Run
function on change

def moisture_callback(channel1):
        if GPIO.input(channel1):
                print("water detected")
        else:
                print("NO Water Detected!")

GPIO.add_event_detect(channel1, GPIO.BOTH, bouncetime=300)  # let us know when
the pin goes HIGH or LOW
GPIO.add_event_callback(channel1, moisture_callback)  # assign function to GPIO
PIN, Run function on change

while True:
    # Converts ACD voltage to temperature in Celsius
    temp_c = round((reading.value * 3.3) * 100, 2)
    # Convert Celsius degrees to Farenheit
    temp_f = round(temp_c * 1.8 + 32, 2)
    # Print both temperatures
    print('Temp: {}ºC    {}ºF'.format(temp_c, temp_f))
```

```
    params = urllib.parse.urlencode({'field6': temp_c, 'key':'1EPX63IN172YAT79'
})

    headers = {"Content-typZZe": "application/x-www-form-urlencoded","Accept":
"text/plain"}

    conn = httplib.HTTPConnection("api.thingspeak.com:80")
    conn.request("POST", "/update", params, headers)
    response = conn.getresponse()
    print(response.status, response.reason)
    data = response.read()
    conn.close()

    if(temp_c >20):
        GPIO.output(in1,GPIO.HIGH)
        #GPIO.output(in2,GPIO.HIGH)
    else:
        GPIO.output(in1,GPIO.LOW)
        #GPIO.output(in2,GPIO.LOW)

    sleep(1)  # Wait 1.5 seconds for the next read
```

Camera Live Stream Code:

```
import io
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="""\
<html>
<head>
<title>Smart Cradle System</title>
</head>
<body>
<center><h1>Smart Cradle System</h1></center>
<center><img src="stream.mjpg" width="640" height="480"></center>
</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

    def write(self, buf):
```

```
        if buf.startswith(b'\xff\xd8'):
            # New frame, copy the existing buffer's content and notify all
            # clients it's available
            self.buffer.truncate()
            with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
            self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
                 self.send_header('Content-Type',  'multipart/x-mixed-replace;
boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
                    self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:
            self.send_error(404)
            self.end_headers()
```

```
class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
    output = StreamingOutput()
    #Uncomment the next line to change your Pi's Camera rotation (in degrees)
    #camera.rotation = 90
    camera.start_recording(output, format='mjpeg')
    try:
        address = ('', 8000)
        server = StreamingServer(address, StreamingHandler)
        server.serve_forever()
    finally:
        camera.stop_recording()
```

Moisture Code:

```
import RPi.GPIO as GPIO
import time
from pushbullet import Pushbullet
#import all
channel1 = 3
GPIO.setmode(GPIO.BOARD)
GPIO.setup(channel1, GPIO.IN)
def moisture_callback(channel1):
    if GPIO.input(channel1):
        pb = Pushbullet("o.H5gGESQCkco3YxVGzsdG21o6M03g8njv")
        print(pb.devices)
        dev = pb.get_device("Vivo Vivo 2018")
        push = dev.push_note("Alert!","Change Baby's Diaper")
        time.sleep(1)
        print("water detected")
    else:
        print("NO Water Detected!")

GPIO.add_event_detect(channel1, GPIO.BOTH, bouncetime=300)  # let us know when
the pin goes HIGH or LOW
GPIO.add_event_callback(channel1, moisture_callback)  # assign function to GPIO
PIN, Run function on change

while True:
    time.sleep(1)
```