# MEDICATION REMINDER AND TRACKING SYSTEM

DEVELOPED BY JOSIAH BATSON

ADVISOR: DR. STEVEN BARRETT

# THE ISSUE

- Forgetting Medication
- Memory Impairing Diseases
- Taking to much of a proscribed medication
- Medication theft

# THE COMPETITION

- Configuration requirements
- No tracking systems
- Low security
- Tradeoffs between each design


# TECHNICAL SPECIFICATIONS

- Dispense medication and lock if medication is taken
- Needs to remind patient if medication not taken
- Programmable by pharmacist for different specs
- Keep record of whether medication is taken or not
- Transmit these records to a doctor
- Warning System

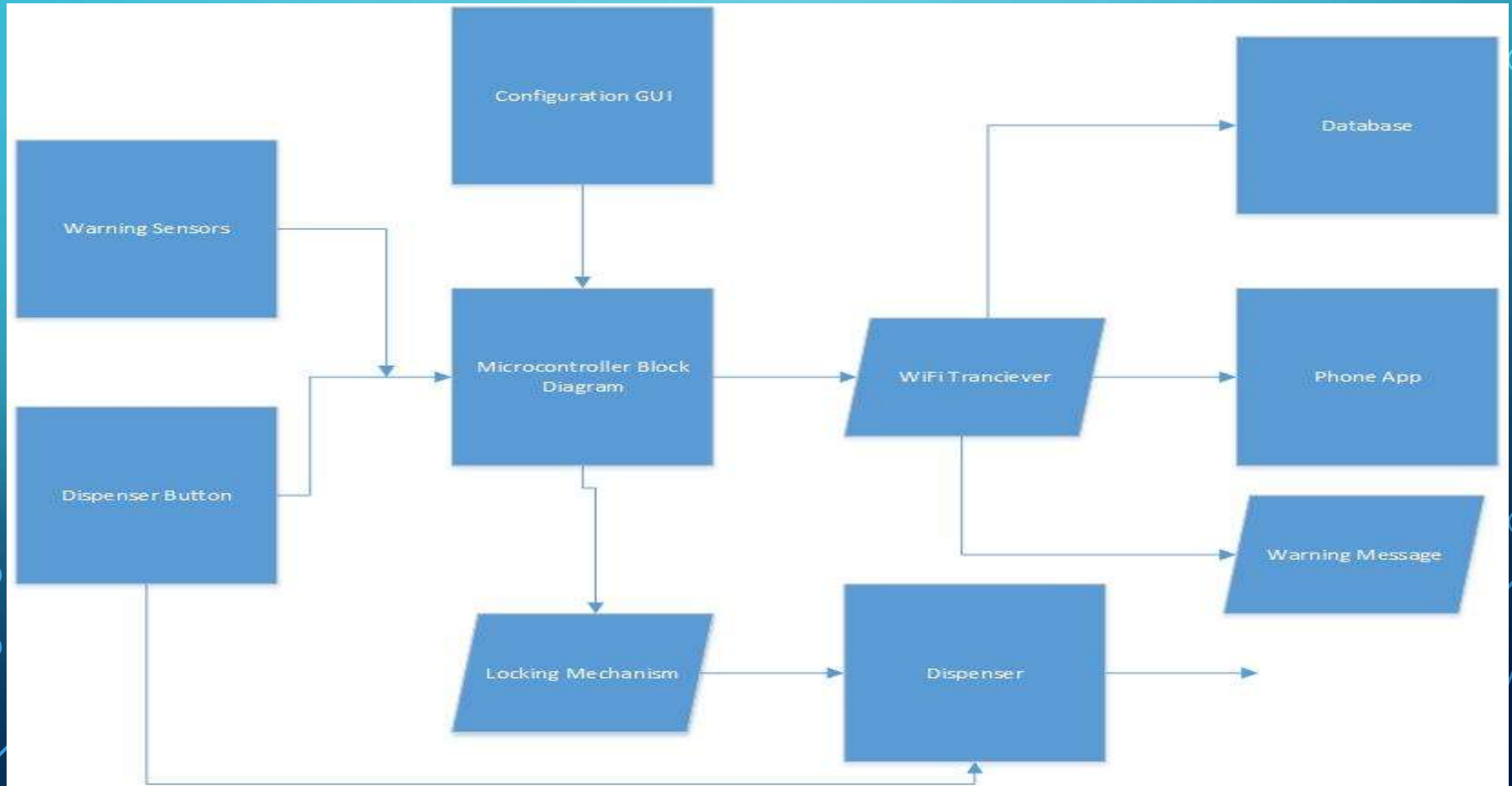
# OTHER CONSIDERATIONS

- Safety
- Online database security
- No available smartphone
- Personnel usage

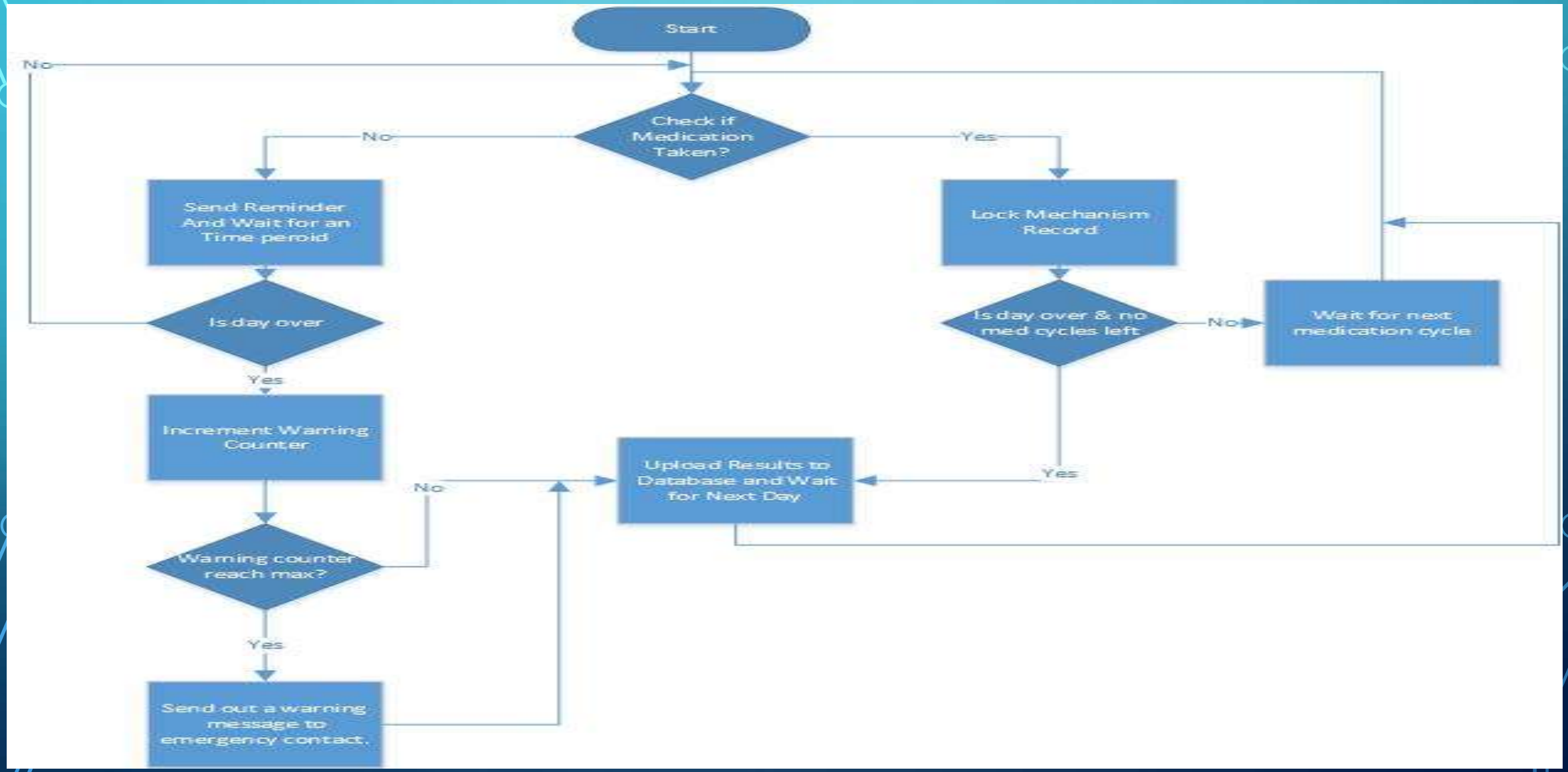
# PROCESS

- Decided on components and app
- Implemented warning systems for msp430
- Implemented data uploads for ESP-07
- Communication between both
- Debugging
- Dispenser development

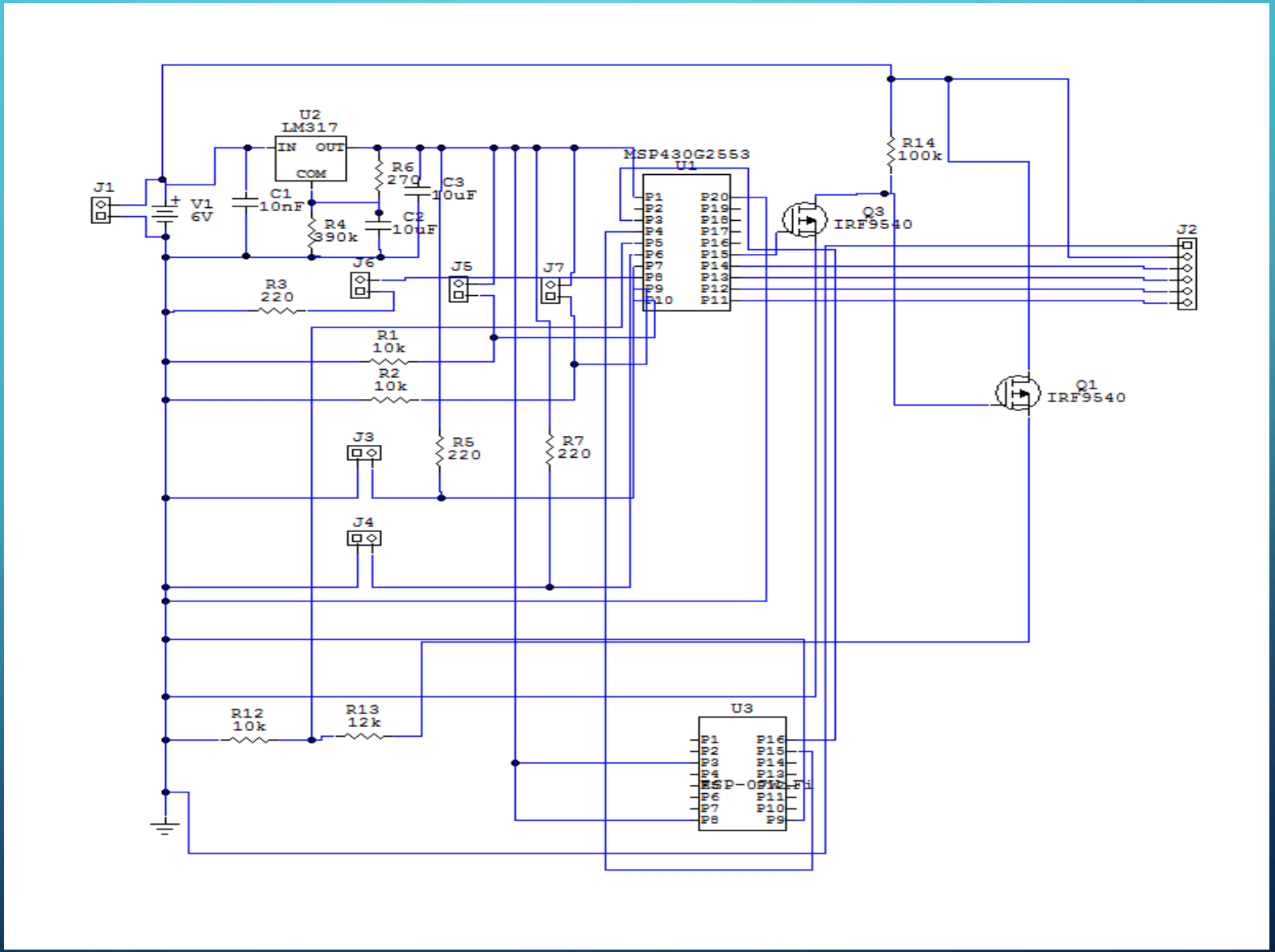
# PROJECT BLOCK DIAGRAM



# MICROCONTROLLER BLOCK DIAGRAM



# THE SCHEMATIC

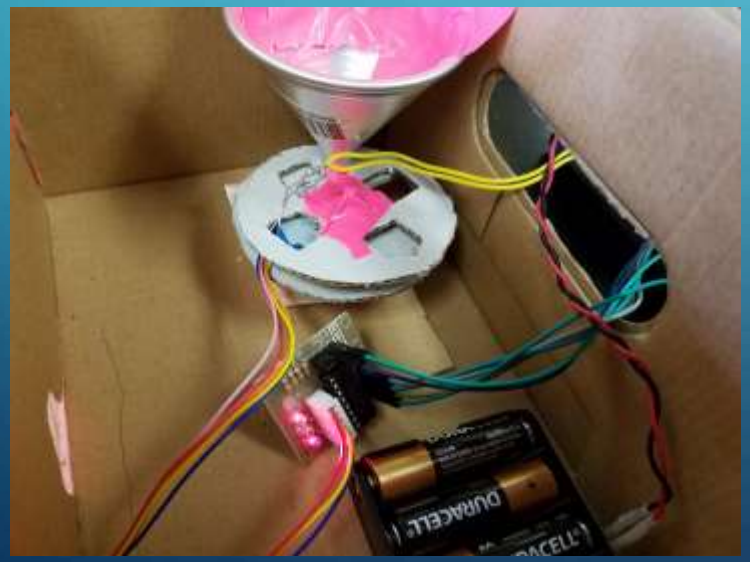


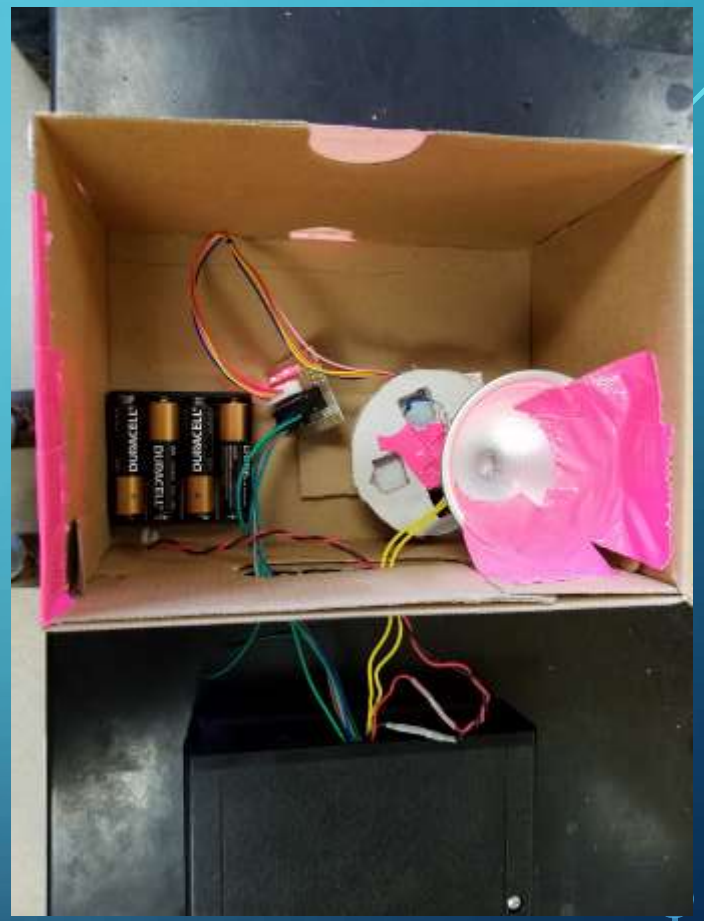
# DESIGN CHANGES/TRADEOFFS

- Solenoid planned for dispenser initially
- Delay scheme rather than interrupts
- Wasn't able to develop phone app

# DISPENSING AND LOCKING

- Stepper Motor
- Prototype funnel for medication
- Rotating chassis to allow loading and unloading

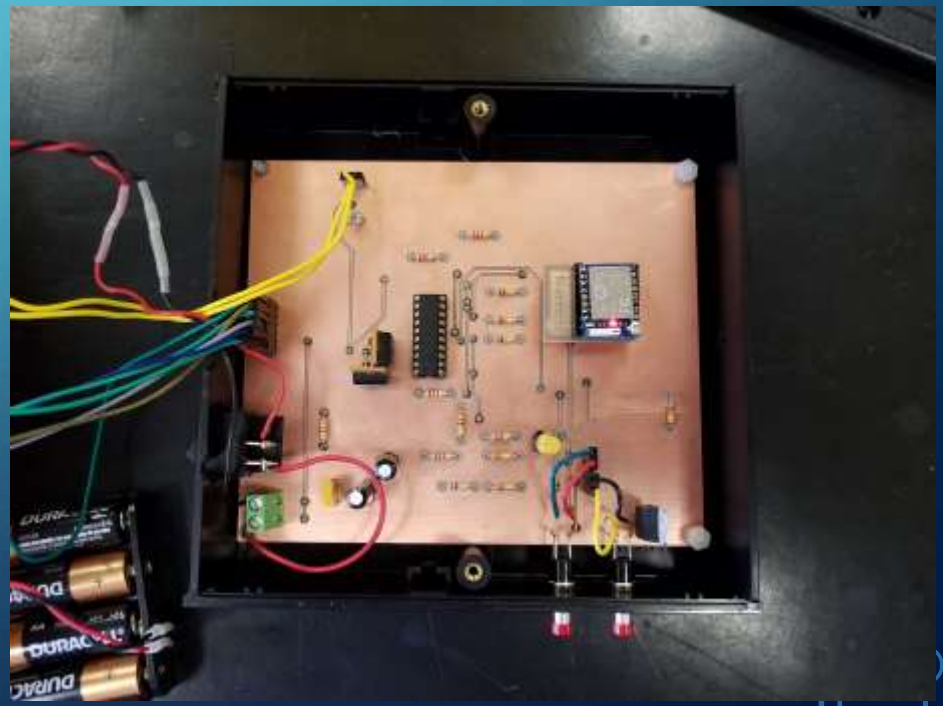


# DISPENSING AND LOCKING OPERATION

# WARNING SYSTEM

- Tracks if medication has not been taken for an extended period of time
- Determine if the device is being tampered with
- Check if emergency button has been pressed
- Check for low battery



# THE CODE

- Photo-resistors and battery used threshold values to track
- Emergency button used user input
- Not enough medication taken used tracking variables
- Overall, uploaded serially to ESP

```
void lightwarning() {
  photosensor1 = analogRead(A4); // Reads the voltage level from
  photosensor2 = analogRead(A5); // Reads the voltage level from
  if((photosensor1<threshold)||(photosensor2<threshold))
  {
    testled(); // Temporary for testing
    warningmsg = 2; // Indicates emergency breakin
    warningalert(warningmsg);
  }
}
```

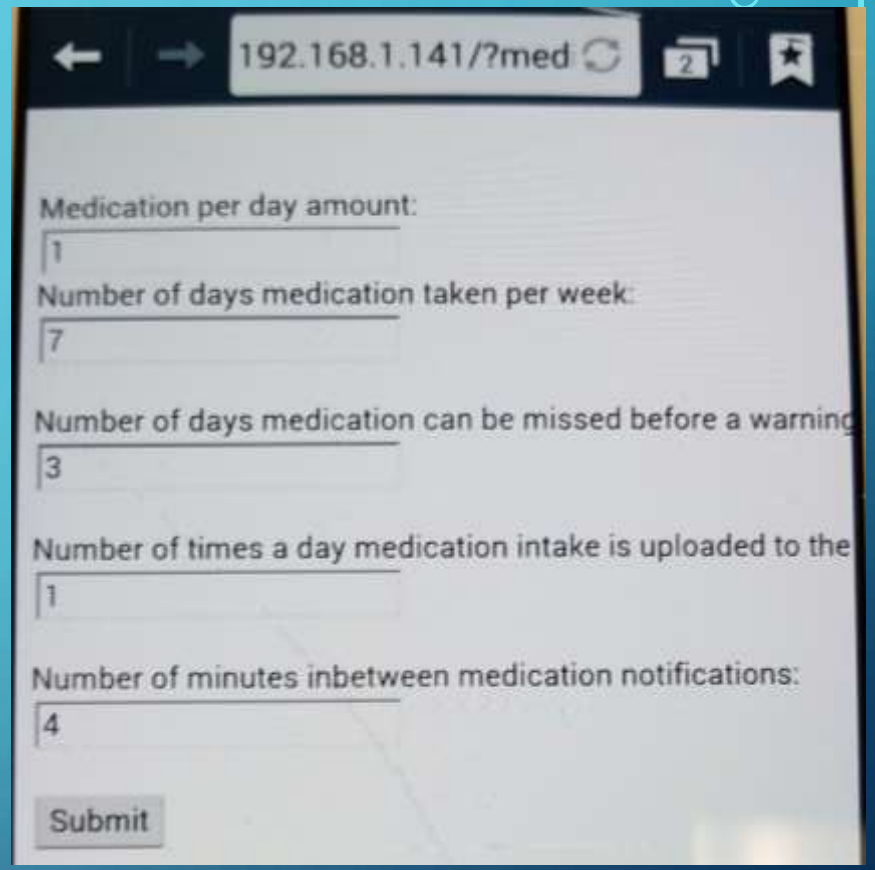
```
void checkbattery() // Function for checking value of battery
{
  digitalWrite(P1_7, HIGH);
  batteryVoltage = analogRead(A3)*(3.0/1024);
  if((batteryVoltage<=batteryThreshold)&&(justchecked>=60000))
  {
    testled(); // Temporary for testing
    warningmsg = 3;
    justchecked = 0;
    warningalert(warningmsg);
  }
  digitalWrite(P1_7, LOW);
  delay(10);

}
```

```
void checkbuttons() {
  buttonState1 = digitalRead(P2_1); // Read the state of dispensing button
  if((buttonState1 == HIGH)&&(lockstatus==false)) // If dispensing button is pressed
  // and lock is not engaged then dispense medication
  {
     dispense(); // Function dispenses pill
     medamountcurrent += 1; // Incriment to indicate another medication has been taken
    if(medamountcurrent>=medamountthreshold)
    {
      lockstatus = true;
    }
  }
  buttonState2 = digitalRead(P2_2); // Read the state of emergency button
  if((buttonState2 == HIGH)&&(medamountcurrent<(medamountthreshold+2))) // If emergency button has been pressed
  // and current medication amount is less than 2 plus the threshold dispense medication
  {
    dispense(); // Function dispenses pill
    medamountcurrent +=1; // Incriment to indicate another medication has been taken
    warningmsg = 6; // Indicates emergency button pressed
    warningalert(warningmsg); // Send out warning message
  }
}
```

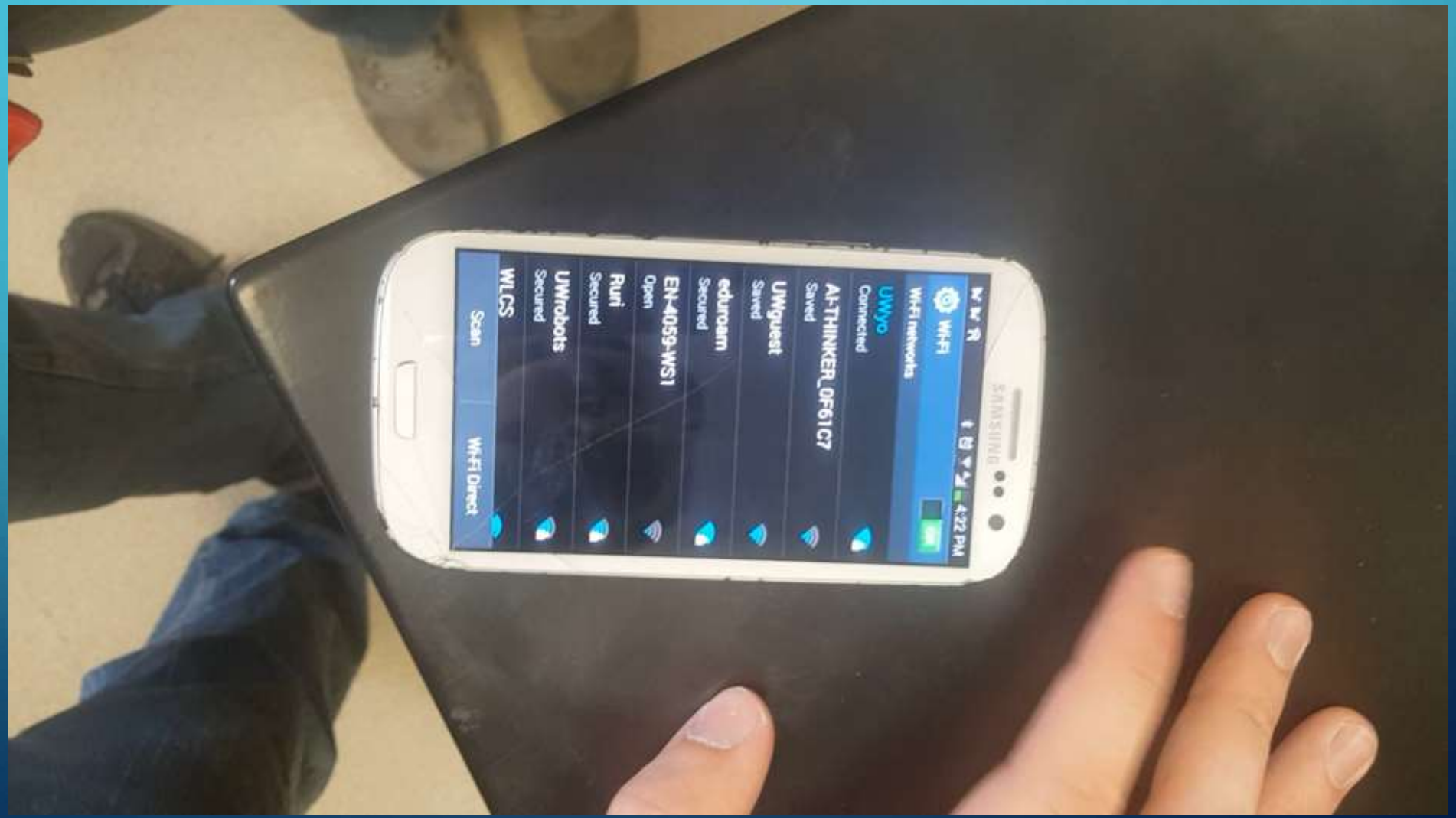
# WARNING SYSTEM OPERATION

# CONFIGURATION

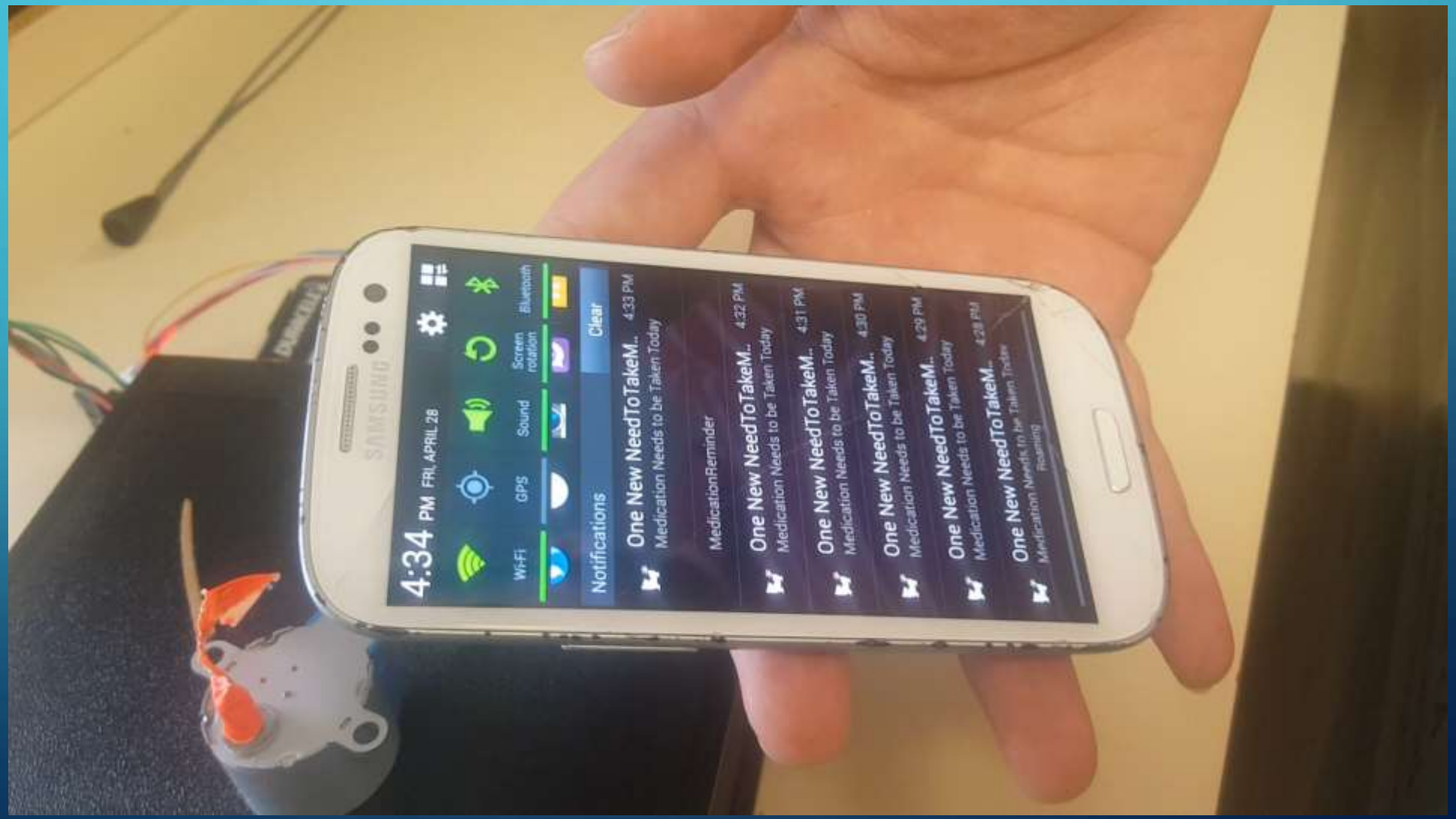
- Configuration by pharmacist
- Needs to change frequency of medication and amount
- GUI accessed by connecting to a web server hosted by ESP-07



# CONFIGURATION OPERATION



# OVERALL OPERATION

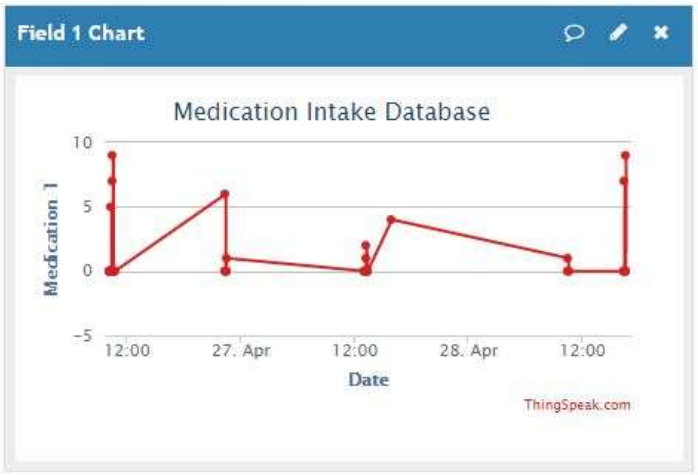


# MEDICATION TRACKING


- Track the history of medication intake
- Upload this information through Wi-Fi to internet of things database

Channel Stats

Created: 12 days ago
Updated: about 14 hours ago
Last entry: about 14 hours ago
Entries: 163




ThingSpeak.com

# THE CODE

- GUI developed with html in Arduino IDE
- Uploads to thingspeak using api, passkeys, and specific message

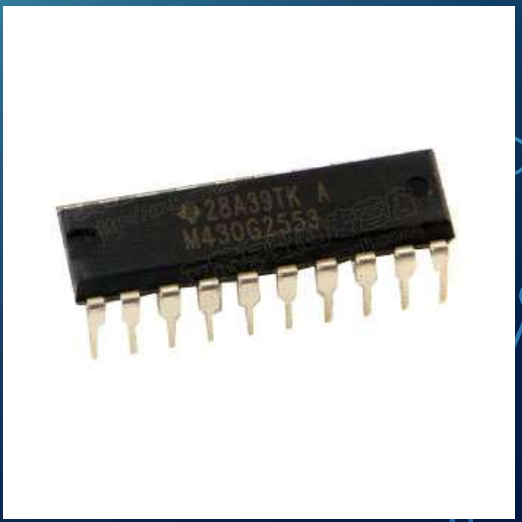
```
void thingSpeakUpload(int medicationTaken) // Function for uploading medication taken amount to thingspeak needs work
{
  i = 0;
  while(i<100)
  {
  if (client.connect(serverThingSpeak,80)) { //  api.thingspeak.com
  String postStr = thingSpeakApi;
  postStr +="&field1=";
  postStr += String(medicationTaken);
  postStr += "\r\n\r\n";

  client.print("POST /update HTTP/1.1\n");
  client.print("Host: api.thingspeak.com\n");
  client.print("Connection: close\n");
  client.print("X-THINGSPEAKAPIKEY: "+thingSpeakApi+"\n");
  client.print("Content-Type: application/x-www-form-urlencoded\n");
  client.print("Content-Length: ");
  client.print(postStr.length());
  client.print("\n\n");
  client.print(postStr);
  //Serial.print("Sent msg, closing");
  client.stop();
  return;
```

```
void medicationTakenMsg(String message) // Function to send notification to your phone if medication has not been taken
{
  i=0;
  while(i <100) {
  if (client.connect(serverPhone, 80)) {
    String json1 = String("{\"event\":\"NeedToTakeMedication\",\"trackers\":{\"message\":\"");
    String json2 = message;
    String json3 = String("\"}}");
    // Make a HTTP request:
    client.print("POST /v1/post HTTP/1.1\n");
    client.print("User-Agent: ESP8266\n");
    client.print("Host: api.instapush.im\n");
    client.print("Accept: *\\*\n");
    client.print("x-instapush-appid: 58f4b1eda4c48a577b967a04\n");
    client.print("x-instapush-appsecret: 05a37cdec0af9a1356551b7369203950\n");
    client.print("Content-Type: application/json\n");
    client.print("Content-Length: ");
    client.print(json1.length() + json2.length() + json3.length());
    client.print("\n\n");
    client.print(json1);
    client.print(json2);
    client.print(json3);
    client.print("\n\n");
    //Serial.println("Sent push, closing connection.");
    client.stop();
    return;
  }
```

# PROBLEMS AND PROCESS

- ESP-07 module took a while to configure
- One function not working correctly
- Many hardware issues and debugging troubles
- MSP430 offboard configuration


```
void tooManyDaysMissed()
{
   if(misseddaycheck >= medicationMissedAllow)
    {
      warningmsg = 5;
      warningalert(warningmsg);
      misseddaycheck = 0;
    }
}
```



# POSSIBLE IMPROVEMENTS

- Secure database
- Personalized App
- Better dispensing
- Better power efficiency
- Fix tracking function

# COST ANALYSIS

| Component | Price | Quantity |
|---|---|---|
| ESP-07 Module | $3.12 | 1 |
| Msp430g2553 | $2.52 | 1 |
| Photo-resistors | $0.95 | 2 |
| Push-button switches | ≈$1.25 | 2 |
| Battery Case | $2.95 | 1 |
| Enclosure | ≈$1.20 | 1 |
| Stepper Motor | $4.95 | 1 |
| Dispensing design | ≈$10-$20 | 1 |
| Total Cost: | $29.14 – $39.14 | |

# CONCLUSION

- Reasonable project
- Got a little behind due to hardware issues
- Improvement to the design is available
- Provided me insight into different design aspects
- Overall, informative experience

# QUESTIONS?