SPRINT-4

```
// include the library code:

#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

#define USE_ARDUINO_INTERRUPTS true    // Set-up low-level interrupts for most acurate BPM math.


#define RX 2

#define TX 3


// initialize the library by associating any needed LCD interface pin

// with the arduino pin number it is connected to

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

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


#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);


String AP = "S@THYA IPHONE";      // CHANGE ME

String PASS = "sathyanarayana"; // CHANGE ME

String API = "4M78AALUC1W80V0S";   // CHANGE ME

String HOST = "api.thingspeak.com";

String PORT = "80";

String field1 = "field1";

String field2 = "field2";

String field3 = "field3";
```

SPRINT-4

```
String field4 = "field4";

int countTrueCommand;

int countTimeCommand;

boolean found = false;


const int triggerpin = 4;     //Connect the trigger pin at pin 2

const int echopin = 5;



long time;                  //Variable for storing the time traveled

int S;                      //Variable for storing the distance covered

int inch;


//int ir_sensor1 = A0;

//int ir_sensor2 = A1;

//int thermister = A2;

int buzzer = 7;


SoftwareSerial esp8266(RX, TX);


void setup()

{

  //Serial.begin(9600);

  esp8266.begin(115200);

  //lcd.begin(16, 2);
```

SPRINT-4

```
 lcd.init();
 lcd.backlight();

//  pinMode(ir_sensor1, INPUT);
//  pinMode(ir_sensor2, INPUT);
 //pinMode(panic_switch, INPUT_PULLUP);
pinMode(buzzer, OUTPUT);
// pinMode(led, OUTPUT);
 pinMode(triggerpin, OUTPUT);  //Setting the triggerpin as output pin
 pinMode(echopin, INPUT);

 lcd.setCursor(0, 0);
 lcd.print("SMART DUSTBIN");
 lcd.setCursor(0, 1);
 lcd.print("    SYSTEM      ");
 delay(2000);
 lcd.setCursor(0, 0);
 lcd.print("CONNECTING...   ");
 lcd.setCursor(0, 1);
 lcd.print("                ");
 sendCommand("AT", 5, "OK");
 sendCommand("AT+CWMODE=1", 5, "OK");
 sendCommand("AT+CWJAP=\"" + AP + "\",\"" + PASS + "\"", 20, "OK");
 lcd.clear();
}
```

SPRINT-4

```
 void loop()
 {
digitalWrite(triggerpin, LOW);
delayMicroseconds(2);
digitalWrite(triggerpin, HIGH); //Setting the triggerpin high for 10us to generate a wave
delayMicroseconds(10);
digitalWrite(triggerpin, LOW);
time = pulseIn(echopin, HIGH); //Setting the echopin high to receive the wave
S= time*0.034/2;              //Calculating the distance traveled in cm
inch = time*0.0133/2;
delay(20);
lcd.setCursor(0,1);
lcd.print("LVL VALUE:");
lcd.print(inch);
//Serial.print(inch);
lcd.print(" inc");
  lcd.setCursor(0,0);           // Sets the location at start
  lcd.print("LEVEL:");



 if (inch < 2)
 {
   lcd.setCursor(7,0);           // Sets the location at start
   lcd.print("FULL ");
```

SPRINT-4

```
   digitalWrite(buzzer, HIGH);
 }
  else
  {
   lcd.setCursor(7,0);           // Sets the location at start
   lcd.print("NRML ");
   digitalWrite(buzzer,LOW);
   }


//  if (inch < 10 || level2 == 1 || level3 == 1)
//  {
//    digitalWrite(buzzer, HIGH);
//   delay(10);
//  }
//   else
//   {
//    digitalWrite(buzzer, LOW);
//    }


  //String getData = "GET /update?api_key=" + API + "&" + field1 + "=" + String(inch) + "&" + field2 + "=" +
String(level2)+ "&" + field3 + "=" + String(level3);
    String getData = "GET /update?api_key=" + API + "&" + field1 + "=" + String(inch);
   sendCommand("AT+CIPMUX=1", 2, "OK");
  sendCommand("AT+CIPSTART=0,\"TCP\",\"" + HOST + "\"," + PORT, 3, "OK");
  sendCommand("AT+CIPSEND=0," + String(getData.length() + 4), 2, ">");
  esp8266.println(getData); delay(1); countTrueCommand++;
```

SPRINT-4

```
 sendCommand("AT+CIPCLOSE=0", 2, "OK");
}

void sendCommand(String command, int maxTime, char readReplay[]) {
 Serial.print(countTrueCommand);
 Serial.print(". at command => ");
 Serial.print(command);
 Serial.print(" ");
 while (countTimeCommand < (maxTime * 1))
 {
  esp8266.println(command);//at+cipsend
  if (esp8266.find(readReplay)) //ok
  {
   found = true;
   break;
  }

  countTimeCommand++;
 }

 if (found == true)
 {
  Serial.println("OYI");
  countTrueCommand++;
  countTimeCommand = 0;
```

SPRINT-4

```
  }


  if (found == false)
  {
    Serial.println("Fail");
    countTrueCommand = 0;
    countTimeCommand = 0;
  }


  found = false;
}
```