

CODE FOR TINKERING PROJECT

/*

Sketch generated by the Arduino IoT Cloud Thing "Untitled"

<https://create.arduino.cc/cloud/things/d0c47a73-b159-4e8f-8544-848c2ed02a22>

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

CloudLight led;

CloudTemperatureSensor temperature;

CloudRelativeHumidity humidity;

CloudRelativeHumidity moisture;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.

*/

#include "thingProperties.h"

#include <LiquidCrystal.h>

#include "DHT.h"

#define DHTPIN 4

#define DHTTYPE DHT11

DHT dht(DHTPIN,DHTTYPE);

```
const int LM34 = 34;  
const int LM35 = 4;  
const int motor = 23;  
int percentValue = 0;
```

```
LiquidCrystal lcd(13, 12, 14, 27, 26, 25);  
float t;  
float h;  
float Moisture;  
float temp;  
float permoisture;
```

```
void setup() {  
  
    pinMode(2,OUTPUT);  
    // Initialize serial and wait for port to open:  
    Serial.begin(115200);  
    Serial.println(F("DHTxx test!"));  
  
    dht.begin();  
  
    lcd.begin(16, 2);  
    lcd.print("Automated Crop");  
    lcd.setCursor(0,1);  
    lcd.print("Watering System!");  
    pinMode(motor, OUTPUT);
```

```
delay(2000);
```

```
lcd.clear();
```

```
lcd.print("SoilM%= ");
```

```
lcd.setCursor(0,1);
```

```
lcd.print("T & H = ");
```

```
// This delay gives the chance to wait for a Serial Monitor without blocking if none is found
```

```
delay(1000);
```

```
// Defined in thingProperties.h
```

```
initProperties();
```

```
// Connect to Arduino IoT Cloud
```

```
ArduinoCloud.begin(ArduinoIoTPreferredConnection);
```

```
/*
```

The following function allows you to obtain more information related to the state of network and IoT Cloud connection and errors the higher number the more granular information you'll get.

The default is 0 (only errors).

Maximum is 4

```
*/
```

```
setDebugMessageLevel(2);
```

```
ArduinoCloud.printDebugInfo();
```

```
}
```

```
void loop() {
```

```
  ArduinoCloud.update();
```

```
  DHT_SENSOR_READ();
```

```

// Your code here

int value = analogRead(LM34);
int val = analogRead(LM35);
Moisture = value * 500.0 / 4095.0;
//temp =-(3000-val)*110.0/4095.0;

permoisture=(Moisture)*100/500;

lcd.setCursor(6,0);
lcd.print(permoisture);


lcd.setCursor(8,1);
lcd.print(t);


lcd.setCursor(13,1);
lcd.print(h);


lcd.setCursor(13,0);


if (isnan(h) || isnan(t))
{
    Serial.println(F("Failed to read from DHT sensor!"));
}


if (Moisture < 400){
    digitalWrite(motor, HIGH);

```

```

    lcd.print("ON ");
}
else {
    digitalWrite(motor, LOW);

    lcd.print("OFF");
}
delay(2000);
Serial.print("Moisture");
Serial.print("TEMP");
Serial.println(t);
Serial.println (permoisture);

}

/*
    Since Led is READ_WRITE variable, onLedChange() is
    executed every time a new value is received from IoT Cloud.
*/
void onLedChange() {
    // Add your code here to act upon Led change
}

void DHT_SENSOR_READ()
{
    //int valu = analogRead(LM34);
    //Moisture = valu * 500.0 / 4095.0;
    //temp = -(3000-val)*110.0/4095.0;
    h=dht.readHumidity();
    t=dht.readTemperature();

    moisture=permoisture;
}

```

```
temperature=t;
```

```
humidity=h;
```

```
Serial.print(temperature);
```

```
Serial.print(humidity);
```

```
Serial.print(moisture);
```

```
delay(1000);
```

```
}
```