Project Delivery Sprint - 2

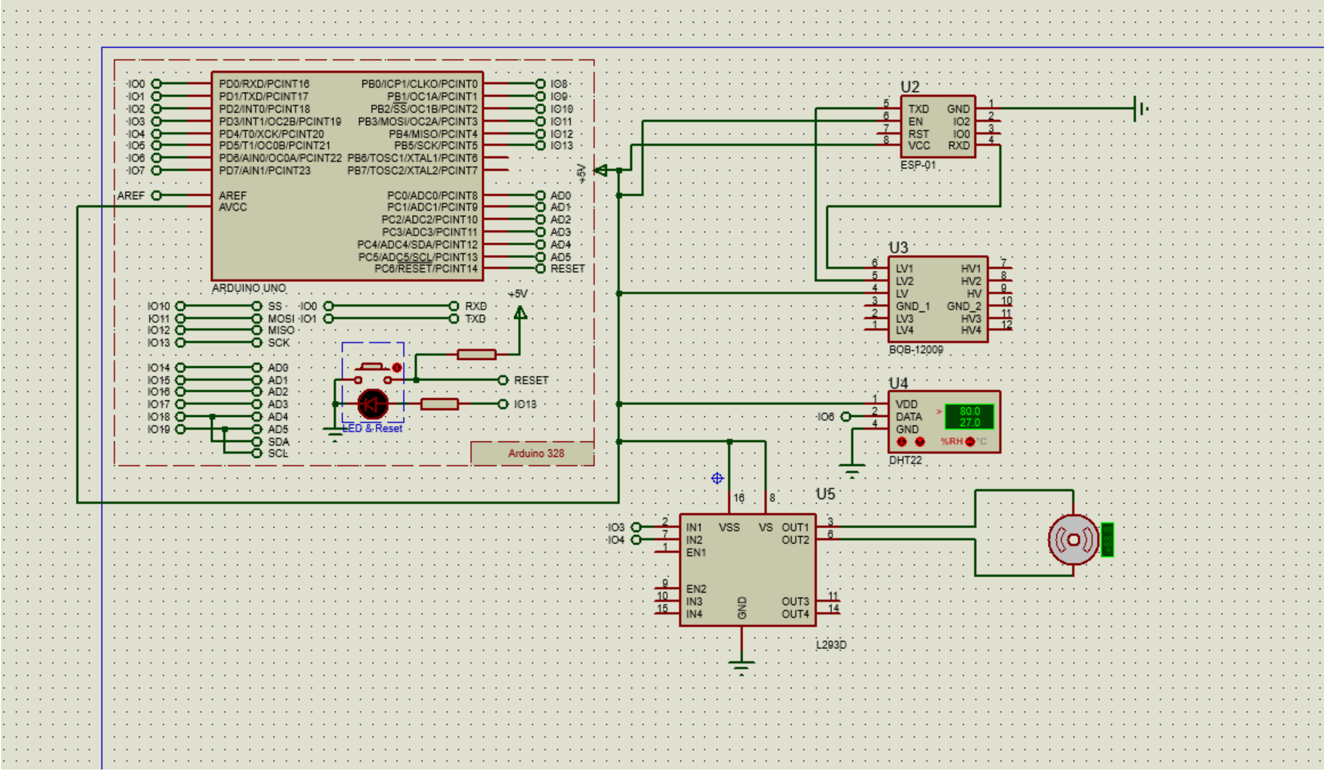
Date	28 Oct 2022
Team ID	PNT2022TMID04704
Project Name	Smart Farmer - IoT Enabled Smart Farming Application

Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task
Sprint-2	I/O interface for Sensors.	USN-3	As a user, I can connect the various sensors like temperature, moisture sensor with Arduino board.

CODE:

```
#include<iWre.h>
#include <DHT.h>;


#define DHTPIN 6
#define m1 3
#define m2 4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);


Variables
int chk;
float hum;
float temp;

void setup()
{
   pinMode(m1, OUTPUT);
   pinMode(m2, OUTPUT);
  Serial.begin(9600);
  dht.begin();
}

void loop()
{
    delay(2000);
    hum = 80;
    temp= 27;
    Serial.print("Humidity: ");
    Serial.print(hum);
    Serial.print(" %, Temp: ");
    Serial.print(temp);
    Serial.println(" Celsius");
    delay(5000);
    temp=35;

    if (temp>30){
      digitalWrite (m1, HIGH);
      delay(5000);
    }


}
```

Circuit Diagram:



Python code To Connect Sensors

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "3nw9vo"
deviceType = "farming"
deviceId = "application"
authMethod = "token"
authToken = "87654321"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("Motor is on")
    elif status == "motoroff":
        print ("Motor is off")
    else :
        print ("please send proper command")

try:
          deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
          deviceCli = ibmiotf.device.Client(deviceOptions)
          #..............................................

except Exception as e:
          print("Caught exception connecting device: %s" % str(e))
          sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()

while True:
        #Get Sensor Data from DHT11

        temp=random.randint(90,110)
        Humid=random.randint(60,100)

        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to IBM Watson")
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```