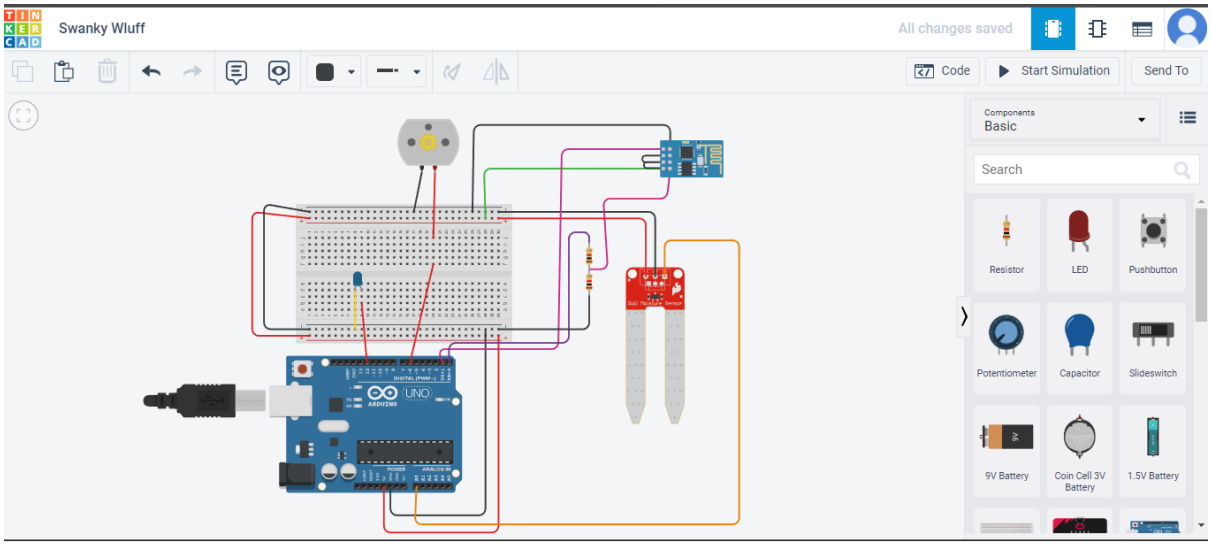
Sprint-1

TEAM ID	**PNT2022TMID44577**
PROJECT NAME	**Project-IoT Based Smart Crop Protection System For Agriculture**
TEAM MEMBERS	**Ramanathan C** **Janarthanan S** **Sithan C** **Bavatharani P**

In this activity you are expected to develop & submit the developed code by testing it

CIRCUIT:



CODE:

```
int moisture_data = 0;


organization = "c5ah4g"
deviceType = "App-1"
deviceId = "13"
authMethod = "token"
```

```
authToken = "12345678"


def myCommandCallback(cmd):
    print("Command received: %s" %
cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led 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()
```

```
void setup()

{

  pinMode(A0, INPUT);

  Serial.begin(9600);

  pinMode(12, OUTPUT);

  pinMode(6, OUTPUT);

}



void loop()

{

  moisture_data = analogRead(A0);

  Serial.println(moisture_data);
```

```
  if (moisture_data < 21) {

    digitalWrite(12, HIGH);

    digitalWrite(6, HIGH);

  } else {

    digitalWrite(12, LOW);

    digitalWrite(6, LOW);

  }

  delay(10); // Delay a little bit to improve simulation
performance


}
  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()
```