

```

// 11/17/17 EcoDuino Test

#include <dht11.h>
//#include <avr/sleep.h>
#include <avr/wdt.h>
dht11 DHT;
#define MOISTURE_PIN A2 //soil Moisture sensor
#define DHT11_PIN 9 //DHT11

int airHumidity;
int airTemperature;
int soilHumidity;
int Farenheit;
int watering;
int t[10];
int h[10];
int s[10];
int results;
unsigned long previousMillis = 0;

/*
 * Adjust values below for your particular setup and plant
 */

const long interval = 60000; // Change value to change time to check
for watering. 60000 = 1 minute
int waterTime = 1000; // Change length of watering
int moistureLevel = 50; // Adjust to know when to water
/*
 * 0 ~ 300 dry soil
 * 300 ~ 700 humid soil
 * 700 ~ 950 in water
 */

void setup() {
  Serial.begin(9600);

  // Make sure pump is off
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
}

void loop() {
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

```

```

    Serial.print(F("Current millis:"));
    Serial.println(currentMillis);
    Serial.print(F("Previous millis:"));
    Serial.println(previousMillis);

    getData();

    Serial.println(F("-----"));
    Serial.print(F("Air Temperature:"));
    Serial.print(airTemperature);
    Serial.println(F("°F"));

    Serial.print(F("Air Humidity:"));
    Serial.println(airHumidity);

    Serial.print(F("Soil Humidity:"));
    Serial.println(soilHumidity);
    Serial.println(F("-----"));

    if (soilHumidity <= moistureLevel) {
        Serial.println(F("Pump on"));

        pumpOn();

        delay(waterTime);
        Serial.println(F("Pump off"));

        pumpOff();
    }
}

void pumpOn()
{
    digitalWrite(5, HIGH);
    digitalWrite(6, HIGH);
}

void pumpOff()
{
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
}

void getData() {
    int chk;

    // get 10 data samples
    for (int i = 0; i < 10; i++) {
        chk = DHT.read(DHT11_PIN);
    }
}

```

```

    t[i] = DHT.temperature;
    h[i] = DHT.humidity;
    s[i] = soilHumidity = analogRead(MOISTURE_PIN);
}

// get temperature avg
insertionSort(t, 10);
results = results * 9 / 5 + 32;
airTemperature = results;
results = 0;

// get humidity avg
insertionSort(h, 10);
airHumidity = results;
results = 0;

// get soil noisture avg
insertionSort(s, 10);
soilHumidity = results;
results = 0;
}

void insertionSort(int arr[], int length) {
    int i;
    int j;
    int tmp;
    for (i = 1; i < length; i++) {
        j = i;
        while (j > 0 && arr[j - 1] > arr[j]) {
            tmp = arr[j];
            arr[j] = arr[j - 1];
            arr[j - 1] = tmp;
            j--;
        }
    }
    // array sorted, now lose low and high values for better average
    arr[0] = 0;
    arr[9] = 0;
    for (i = 0; i < 10; i++) {
        results = results + arr[i];
    }
    results /= 8; // divide by 8 as values 1 and 10 = 0
}

```