```
#include <Adafruit_CircuitPlayground.h>
#include <Adafruit_Circuit_Playground.h>
#include <Adafruit_SleepyDog.h>
#include "Timer.h"
#include <DHT.h>

Timer t;
DHT dht(1,11);

//static char vegetablesName[5] = {"other", "radis", "roquette",
"betterave", "legume4"};
static int minEnlightenmentTime[5] = {5, 5, 6, 5, 5};
static int maxEnlightenmentTime[5] = {10, 10, 10, 10, 10};
static int vegetablesMinEnlightenment[5] = {100, 100, 100, 100,
100};
static int vegetablesMaxEnlightenment[5] = {1023, 1023, 1023,
1023, 1023};
static int vegetablesMinSoilMoisture[5] = {35, 35, 35, 35, 35};
static int vegetablesMaxSoilMoisture[5] = {75, 75, 75, 75, 75};
static int vegetablesMinTemperature[5] = {15, 15, 5, 15, 15};
static int vegetablesMaxTemperature[5] = {30, 30, 20, 30, 30};

int LightLogger[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int WaterLogger[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
int TemperatureLogger[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};


int vegetableNum = 0;
int getCensorValues = 1;
int LightProblem = 0;
int WaterProblem = 0;
int TemperatureProblem = 0;
int ProblemLightTime = 0;
int pProblemLightTime = 0 ;
int Sleeping=0;
int DoLights=1;



void SetLightProblem() {
```

```
  int LightLevel = CircuitPlayground.lightSensor();
  Serial.print("l ");
  Serial.println(LightLevel);
  if (LightLevel < vegetablesMinEnlightenment[vegetableNum])
    LightProblem=1;
  else if (LightLevel > vegetablesMaxEnlightenment[vegetableNum])
    LightProblem=2;
  else
    LightProblem=0;
}

void AddLightToLogger() {
  for(int i=1; i<10; ++i)
    LightLogger[i]=LightLogger[i-1];
  LightLogger[0]=CircuitPlayground.lightSensor();
}

void SetMoistureProblem() {
  int MoistureLevel = dht.readHumidity();
  Serial.print("m ");
  Serial.println(MoistureLevel);
  if (MoistureLevel < vegetablesMinSoilMoisture[vegetableNum])
    WaterProblem=1;
  else if (MoistureLevel >
vegetablesMaxSoilMoisture[vegetableNum])
    WaterProblem=2;
  else
    WaterProblem=0;
}

void AddMoistureToLogger() {
  for(int i=1; i<10; ++i)
    WaterLogger[i]=WaterLogger[i-1];
  WaterLogger[0]=CircuitPlayground.readCap(1)*1000./65536.;
}

void SetTemperatureProblem() {
  float TemperatureLevel = CircuitPlayground.temperature();
  Serial.print("t ");
  Serial.println(TemperatureLevel);
```

```
  if (TemperatureLevel < vegetablesMinTemperature[vegetableNum])
    TemperatureProblem=1;
  else if (TemperatureLevel >
vegetablesMaxTemperature[vegetableNum])
    TemperatureProblem=2;
  else
    TemperatureProblem=0;
}

void AddTemperatureToLogger() {
  for(int i=1; i<10; ++i)
    TemperatureLogger[i]=TemperatureLogger[i-1];
  TemperatureLogger[0]=CircuitPlayground.temperature();
}

void SetStripColor(int R, int G, int B, int dt) {
  for (int i=0; i<10; i++)
    CircuitPlayground.strip.setPixelColor(i, R, G, B);
  CircuitPlayground.strip.show();
  delay(dt);
}


void SetLights() {
  if (not CircuitPlayground.slideSwitch())
    return;
  if (LightProblem > 0 && (WaterProblem > 0 && TemperatureProblem
> 0))
    ProblemLightTime = 500;
  else if (LightProblem > 0 || (WaterProblem > 0 ||
TemperatureProblem > 0))
    ProblemLightTime = 1500;
  else if (LightProblem == 0 && (WaterProblem == 0 &&
TemperatureProblem == 0)) {
    SetStripColor(0, 255, 0, 0);
    ProblemLightTime = 0;
  } else
    ProblemLightTime = 750;
  if (LightProblem == 1) {
    CircuitPlayground.strip.setBrightness(2);
```

```
    SetStripColor(143, 110, 110, ProblemLightTime);
    CircuitPlayground.strip.setBrightness(5);
  }
  else if (LightProblem == 2)
    SetStripColor(255, 255, 0, ProblemLightTime);
  if (WaterProblem == 1)
    SetStripColor(255, 75, 0, ProblemLightTime);
  else if (WaterProblem == 2)
    SetStripColor(0, 0, 255, ProblemLightTime);
  if (TemperatureProblem == 1)
    SetStripColor(255, 0, 128, ProblemLightTime);
  else if (TemperatureProblem == 2)
    SetStripColor(255, 0, 0, ProblemLightTime);
}

void SetVegetable() {
  if ((CircuitPlayground.readCap(2)) >= (100))
    vegetableNum=1;
  else if ((CircuitPlayground.readCap(12)) >= (100))
    vegetableNum=2;
  else if ((CircuitPlayground.readCap(6)) >= (100))
    vegetableNum=3;
  else if ((CircuitPlayground.readCap(9)) >= (100))
    vegetableNum=4;
  else if ((CircuitPlayground.readCap(10)) >= (100))
    vegetableNum=5;
}

void SleepFor(float seconds) {
  for (int i=0; i<seconds/4; i++)
    Watchdog.sleep(4000);
}

void setup() {
  CircuitPlayground.begin();
  Serial.begin(9600);
  dht.begin();
  CircuitPlayground.strip.setBrightness(5);
    for(int j=0; j<333; ++j) {
      for(int i=0; i<10; ++i)
```

```
      CircuitPlayground.strip.setPixelColor(i,
CircuitPlayground.colorWheel(((i * 256 / 10) + j) & 255));
      CircuitPlayground.strip.show();
      delay(3);
    }
  for(int i=0; i<10; ++i) {
    LightLogger[i] =
(vegetablesMinEnlightenment[vegetableNum]+vegetablesMaxEnlightenme
nt[vegetableNum])/2;
    WaterLogger[i] =
(vegetablesMinSoilMoisture[vegetableNum]+vegetablesMaxSoilMoisture
[vegetableNum])/2;
    TemperatureLogger[i] =
(vegetablesMinTemperature[vegetableNum]+vegetablesMaxTemperature[v
egetableNum])/2;
  }
  pinMode(3, INPUT);
  t.every(360 * 1000UL, SetMoistureProblem);
  t.every(360 * 1000UL, AddMoistureToLogger);
  t.every(1800 * 1000UL, SetTemperatureProblem);
  t.every(1800 * 1000UL, AddTemperatureToLogger);
  t.every(7200 * 1000UL, SetLightProblem);
  t.every(7200 * 1000UL, AddLightToLogger);
  CircuitPlayground.clearPixels();
}

Timer sl;

void loop() {
  t.update();
  sl.update();
  if (CircuitPlayground.slideSwitch()) {
    SetMoistureProblem();
    SetTemperatureProblem();
    SetLightProblem();
    SetLights();
    if (Sleeping==1)
      Sleeping=0;
  } else {
    CircuitPlayground.clearPixels();
```

```
    SleepFor(1);
    if (Sleeping==0) {
#if defined(USBCON) && !defined(USE_TINYUSB)
  USBDevice.attach();
#endif
      Sleeping=1;
    }
  }
}
```