#include <LiquidCrystal.h>

#include <IRremote.h>

enum Mode {

First,

Temperature,

Soil,

Gas,

Light,

Last

};

const int tmpSensorPin = 0;

const int soilMoistureSensorPin = 1;

const int gasSensorPin = 2;

const int lightSensorPin = 3;

const int irRecieverPin = 13;

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

Mode mode;

void setup(){

Serial.begin(9600);

lcd.begin(16, 2);

IrReceiver.begin(irRecieverPin, false);

mode = Mode::Temperature;

}

void loop() {

if(IrReceiver.decode()) {

switch(IrReceiver.decodedIRData.command) {

case 4://|<<

mode = static\_cast<Mode>(mode > Mode::First + 1 ? mode - 1 : mode);

break;

case 6://>>|

mode = static\_cast<Mode>(mode < Mode::Last - 1 ? mode + 1 : mode);

break;

}

IrReceiver.resume();

lcd.clear();

}

switch(mode) {

case Mode::Temperature:

showTemp();

break;

case Mode::Soil:

showSoil();

break;

case Mode::Gas:

showGas();

break;

case Mode::Light:

showLight();

break;

}

delay(50);

}

void showTemp() {

lcd.setCursor(0, 0);

lcd.print("| Soil ->");

lcd.setCursor(0, 1);

lcd.print("TMP: ");

lcd.print(getTemp());

lcd.print(" C ");

}

void showSoil() {

lcd.setCursor(0, 0);

lcd.print("<- Temp Gas ->");

lcd.setCursor(0, 1);

lcd.print("Soil: ");

lcd.print(getSoil());

lcd.print(" % ");

}

void showGas() {

lcd.setCursor(0, 0);

lcd.print("<- Soil Light ->");

lcd.setCursor(0, 1);

lcd.print("Gas: ");

lcd.print(getGas());

lcd.print(" % ");

}

void showLight() {

lcd.setCursor(0, 0);

lcd.print("<- Gas |");

lcd.setCursor(0, 1);

lcd.print("Light: ");

lcd.print(getLight());

lcd.print(" % ");

}

float getTemp() {

int reading = analogRead(tmpSensorPin);

float voltage = reading \* 5.0;

voltage /= 1024.0;

float temperatureC = (voltage - 0.5) \* 100;

return temperatureC;

}

float getSoil() {

int value = analogRead(soilMoistureSensorPin);

return value \* 500.0 / 1023.0 / 4.29;

}

float getGas() {

int value = analogRead(gasSensorPin);

return map(value, 300, 750, 0, 100);

}

float getLight() {

int value = analogRead(lightSensorPin);

return map(value, 0, 942, 0, 100);

}