#include <TimerOne.h> // Avaiable from http://www.arduino.cc/playground/Code/Timer1

#include<SoftwareSerial.h>

SoftwareSerial mySerial(10, 11);

int DEVICE\_ID = 1315;

int AC\_pin1 = 4; // Output to Opto Triac Analog AC-1

int AC\_pin2 = 5; // Analog AC-2

int fan\_table = 9; // Relay AC-2

int light\_cfl = 8; // Relay AC-1

int light\_table = 6; // DC PWM

volatile int i = 0; // Variable to use as a counter

volatile int j = 0;

volatile boolean zero\_cross1 = 0; // Boolean to store a "switch" to tell us if we have crossed zero

volatile boolean zero\_cross2 = 0;

int dim1\_2 = 0; // led control

int dim1 = 128; // Dimming level (0-128) 0 = on, 128 = 0ff

int dim2\_2 = 0;

int dim2 = 128;

char str[100];

int freqStep = 75; // This is the delay-per-brightness step in microseconds for 50Hz (change the value in 65 for 60Hz)

int pas = 8;

int deviceId[20];

int deviceVal[20];

char deviceName[20][20];

char ch;

int ct = 0;

int ptr;

int l;

bool flag;

int k;

boolean state;

void setup()

{

Serial.begin(9600);

mySerial.begin(9600);

pinMode(AC\_pin1, OUTPUT); // Set the Triac pin as output

pinMode(AC\_pin2, OUTPUT);

pinMode(fan\_table, OUTPUT);

pinMode(light\_cfl, OUTPUT);

pinMode(light\_table, OUTPUT); //..PWM

attachInterrupt(0, zero\_cross\_detect, RISING); // Attach an Interupt to Pin 2 (interupt 0) for Zero Cross Detection

Timer1.initialize(freqStep); // Initialize TimerOne library for the freq we need

Timer1.attachInterrupt(dim\_check, freqStep); // Use the TimerOne Library to attach an interrupt

for (k = 0; k < 20; k++)

deviceVal[k] = deviceId[k] = 0;

flag = false;

Serial.println("WELCOME TO DEVICE 1315 UI");

pinMode(13, OUTPUT);

state = true;

dim1 = dim2 = 128;

digitalWrite(13, state);

l=0;

}

void zero\_cross\_detect() {

zero\_cross1 = true; // set the boolean to true to tell our dimming function that a zero cross has occured

zero\_cross2 = true;

i = 0;

j = 0;

digitalWrite(AC\_pin1, LOW);

digitalWrite(AC\_pin2, LOW);

}

// Turn on the TRIAC at the appropriate time

void dim\_check() {

if (dim1\_2 > 0)

{

if (zero\_cross1 == true) {

if (i >= dim1 && dim1 != 128) {

digitalWrite(AC\_pin1, HIGH); // turn on light

i = 0; // 2reset time step counter

zero\_cross1 = false; // reset zero cross detection

}

else {

i++; // increment time step counter

}

}

}

if (dim2\_2 > 0)

{

if (zero\_cross2 == true) {

if (j >= dim2 && dim2 != 128) {

digitalWrite(AC\_pin2, HIGH); // turn on light

j = 0; // reset time step counter

zero\_cross2 = false; // reset zero cross detection

}

else {

j++; // increment time step counter

}

}

}

}

void setVal()

{

for (k = 0; k < ptr; k++)

{

if (deviceId[k] == DEVICE\_ID)

{

sprintf(str, "\n%d %d %s %d", k, deviceId[k], deviceName[k], deviceVal[k]);

if (strcmp(deviceName[k], "light\_bulb") == 0 && (deviceVal[k] \* 12 != dim1))

{

dim1 = 128 - deviceVal[k] \* 12.8;

dim1\_2 = 255 - 2 \* dim1;

if (dim1\_2 < 0)

dim1\_2 = 0;

}

else if (strcmp(deviceName[k], "fan\_ceil") == 0 && (deviceVal[k] \* 12 != dim2))

{

dim2 = 128 - deviceVal[k] \* 12.8;

dim2\_2 = 255 - 2 \* dim2;

if (dim2\_2 < 0)

dim2\_2 = 0;

}

else if (strcmp(deviceName[k], "fan\_table") == 0)

{

digitalWrite(fan\_table, (deviceVal[k] > 0 ? HIGH : LOW));

}

else if (strcmp(deviceName[k], "light\_cfl") == 0)

{

digitalWrite(light\_cfl, (deviceVal[k] > 0 ? HIGH : LOW));

}

else if (strcmp(deviceName[k], "light\_table") == 0)

analogWrite(light\_table, deviceVal[k] \* 25);

Serial.print(str);

}

}

}

void loop()

{

if (mySerial.available())

{

ch = mySerial.read();

//Serial.print(ch);

if (ch == '[')

{

flag = true;

ptr = 0;

}

else if (ch == ']')

{

setVal();

sprintf(str, "Device1: %d|%d Device2: %d|%d", dim1, dim1\_2, dim2, dim2\_2);

Serial.println(str);

Serial.println("\n-----------------DONE-----------------");

flag = false;

for (k = 0; k < ptr; k++)

deviceVal[k] = deviceId[k] = 0;

//mySerial.write("A");

}

if (flag)

{

if (ch == '{')

{

l = 0;

ct = 0;

deviceVal[ptr] = 0;

deviceId[ptr] = 0;

}

else if (ch == '}')

{

deviceName[ptr][l - 1] = 0;

ptr++;

}

else if (ch == ':')

ct++;

else if (ct == 1)

{

if (ch >= '0' && ch <= '9')

deviceId[ptr] = deviceId[ptr] \* 10 + (ch - '0');

}

else if (ct == 2)

{

if (ch != '"')

deviceName[ptr][l++] = ch;

if (ch == ',')

ct++;

}

else if (ct == 4)

deviceVal[ptr] = deviceVal[ptr] \* 10 + (ch - '0');

}

}

dim1\_2 = 255 - 2 \* dim1;

dim2\_2 = 255 - 2 \* dim2;

if (dim1\_2 < 0)

dim1\_2 = 0;

if (dim2\_2 < 0)

dim2\_2 = 0;

//sprintf(str,"Device1: %d|%d Device2: %d|%d",dim1,dim1\_2,dim2,dim2\_2);

//Serial.println(str);

//delay (100);

}