// Binary Counter Software Instructions

// Default Instructions

// Enable one of these two #includes and comment out the other.

// Conditional #include doesn't work due to Arduino IDE shenanigans.

#include <Wire.h> // Enable this line if using Arduino Uno, Mega, etc.

//#include <TinyWireM.h> // Enable this line if using Adafruit Trinket, Gemma, etc.

#include "Adafruit\_LEDBackpack.h"

#include "Adafruit\_GFX.h"

Adafruit\_7segment matrix = Adafruit\_7segment();

int ledPin[] = {2,3,4,5,6,7,8,9}; //Array of led output pins (Pin 2 being the value 1)

int modePin = 13; //Input pin for mode switch

int btnPin = 12; //Input pin for pushbutton

int btnState = 0;

int prevBtn = 0;

byte counter = 0;

boolean manual = false;

boolean prevManual = false;

boolean mus = false;

unsigned long firstClick;

unsigned long lastClick;

void setup()

{

for (int i =0;i<8;i++) //Sets all led pins to outputs

{

pinMode(ledPin[i], OUTPUT);

}

pinMode(modePin, INPUT);

pinMode(btnPin, INPUT);

// #ifndef \_\_AVR\_ATtiny85\_\_

Serial.begin(9600);

Serial.println("7 Segment Backpack Test");

//#endif

matrix.begin(0x70);

manual = digitalRead(modePin);

prevManual = !manual;

btnState = digitalRead(btnPin);

prevBtn = !btnState;

lastClick = millis();

}

void loop()

{

manual = digitalRead(modePin);

btnState = digitalRead(btnPin);

if(manual != prevManual) { //If the mods switch changes position

mus = false; //exit museum mode

Serial.println("Mus is false");

Serial.println("Switch from " + prevManual);

counter = 0; //reset the counter

prevManual = manual;

flashOn(); //start reset sequence

delay(250);

flashOff();

delay(250);

flashOn();

delay(250);

flashOff();

delay(250);

}

if(mus){ //If in museum mode

if (btnState == HIGH) {

counter++;

increment(counter);

delay(350);

lastClick = millis();

}

else if ((millis() - lastClick) > 10000) {

//If 10 seconds has passed since last button press

Serial.println("Mus auto mode");

counter++;

increment(counter);

delay(1000);

}

}

else if(!manual) { //If in automatic mode

if(btnState != prevBtn){

if(btnState == HIGH){

firstClick = millis();

}

prevBtn = btnState;

}

else{

if(btnState == HIGH){

if((millis() - firstClick) > 5000){

mus = true;

Serial.println("Mus is true");

}

}

}

counter++;

increment(counter);

delay(1000);

}

else { //In manual mode

if(btnState == HIGH) {

counter++;

increment(counter);

delay(350); //Delay to prevent button bouncing

}

}

}

void increment(byte i) { //Method to increment 7 segment display and leds

matrix.println(i, DEC);

matrix.writeDisplay();

displayBinary(i);

}

void flashOn() { //A method used in reset sequence

matrix.println(255, DEC);

matrix.writeDisplay();

displayBinary(255);

}

void flashOff() { //A method used in reset sequence

matrix.println(0, DEC);

matrix.writeDisplay();

displayBinary(0);

}

void displayBinary(byte numToShow) //Method used to update the leds

{

for (int i =0;i<8;i++)

{

if (bitRead(numToShow, i)==1)

{

digitalWrite(ledPin[i], HIGH);

}

else

{

digitalWrite(ledPin[i], LOW);

}

}

}