#include <ShiftRegister74HC595.h>

#include <TM1637Display.h>

// ==== Pinos do 74HC595 ====

#define SDI\_ROUND 39

#define SCLK\_ROUND 41

#define LOAD\_ROUND 43

#define SDI\_TIME1 48

#define SCLK\_TIME1 50

#define LOAD\_TIME1 52

#define SDI\_TIME2 8

#define SCLK\_TIME2 9

#define LOAD\_TIME2 10

// ==== Pinos do TM1637 (TROCADO PARA EVITAR CONFLITO) ====

#define CLK 2

#define DIO 3

// ==== Botões Round ====

#define BTN\_ROUND\_INC 22

#define BTN\_ROUND\_DEC 23

// ==== Botões Time 1 ====

#define BTN\_TIME1\_INC 26

#define BTN\_TIME1\_DEC 27

// ==== Botões Time 2 ====

#define BTN\_TIME2\_INC 30

#define BTN\_TIME2\_DEC 31

// ==== Botões do Cronômetro ====

#define BTN\_START\_STOP 35

#define BTN\_ADD\_TIME 34

#define BTN\_RESET 44

// ==== Instâncias ====

ShiftRegister74HC595<2> srRound(SDI\_ROUND, SCLK\_ROUND, LOAD\_ROUND);

ShiftRegister74HC595<2> srTime1(SDI\_TIME1, SCLK\_TIME1, LOAD\_TIME1);

ShiftRegister74HC595<2> srTime2(SDI\_TIME2, SCLK\_TIME2, LOAD\_TIME2);

TM1637Display display(CLK, DIO);

// ==== Dígitos 7 segmentos ====

uint8\_t digits[] = {

B11000000, B11111001, B10100100, B10110000, B10011001,

B10010010, B10000010, B11111000, B10000000, B10010000

};

// ==== Variáveis ====

int roundValue = 0;

int time1Value = 0;

int time2Value = 0;

unsigned long previousMillis = 0;

bool running = false;

int defaultMaxTime = 120;

int maxTime = 120;

int seconds = 0;

// ==== Funções ====

void showNumber(ShiftRegister74HC595<2>& sr, int num) {

int unidades = num % 10;

int dezenas = (num / 10) % 10;

uint8\_t toPrint[] = {digits[dezenas], digits[unidades]};

sr.setAll(toPrint);

}

void displayTime(int sec) {

int minutes = sec / 60;

int secRemain = sec % 60;

int timeToDisplay = (minutes \* 100) + secRemain;

display.showNumberDecEx(timeToDisplay, 0x40, true);

}

// ==== Setup ====

void setup() {

pinMode(BTN\_ROUND\_INC, INPUT\_PULLUP);

pinMode(BTN\_ROUND\_DEC, INPUT\_PULLUP);

pinMode(BTN\_TIME1\_INC, INPUT\_PULLUP);

pinMode(BTN\_TIME1\_DEC, INPUT\_PULLUP);

pinMode(BTN\_TIME2\_INC, INPUT\_PULLUP);

pinMode(BTN\_TIME2\_DEC, INPUT\_PULLUP);

pinMode(BTN\_START\_STOP, INPUT\_PULLUP);

pinMode(BTN\_ADD\_TIME, INPUT\_PULLUP);

pinMode(BTN\_RESET, INPUT\_PULLUP);

display.setBrightness(7);

display.showNumberDecEx(0, 0x40, true);

showNumber(srRound, roundValue);

showNumber(srTime1, time1Value);

showNumber(srTime2, time2Value);

}

// ==== Loop ====

void loop() {

// Controle Round

if (digitalRead(BTN\_ROUND\_INC) == LOW) {

delay(50);

if (digitalRead(BTN\_ROUND\_INC) == LOW && roundValue < 99) {

roundValue++;

showNumber(srRound, roundValue);

while (digitalRead(BTN\_ROUND\_INC) == LOW);

}

}

if (digitalRead(BTN\_ROUND\_DEC) == LOW) {

delay(50);

if (digitalRead(BTN\_ROUND\_DEC) == LOW && roundValue > 0) {

roundValue--;

showNumber(srRound, roundValue);

while (digitalRead(BTN\_ROUND\_DEC) == LOW);

}

}

// Controle Time 1

if (digitalRead(BTN\_TIME1\_INC) == LOW) {

delay(50);

if (digitalRead(BTN\_TIME1\_INC) == LOW && time1Value < 99) {

time1Value++;

showNumber(srTime1, time1Value);

while (digitalRead(BTN\_TIME1\_INC) == LOW);

}

}

if (digitalRead(BTN\_TIME1\_DEC) == LOW) {

delay(50);

if (digitalRead(BTN\_TIME1\_DEC) == LOW && time1Value > 0) {

time1Value--;

showNumber(srTime1, time1Value);

while (digitalRead(BTN\_TIME1\_DEC) == LOW);

}

}

// Controle Time 2

if (digitalRead(BTN\_TIME2\_INC) == LOW) {

delay(50);

if (digitalRead(BTN\_TIME2\_INC) == LOW && time2Value < 99) {

time2Value++;

showNumber(srTime2, time2Value);

while (digitalRead(BTN\_TIME2\_INC) == LOW);

}

}

if (digitalRead(BTN\_TIME2\_DEC) == LOW) {

delay(50);

if (digitalRead(BTN\_TIME2\_DEC) == LOW && time2Value > 0) {

time2Value--;

showNumber(srTime2, time2Value);

while (digitalRead(BTN\_TIME2\_DEC) == LOW);

}

}

// Controle do cronômetro

if (digitalRead(BTN\_START\_STOP) == LOW) {

delay(200);

running = !running;

while (digitalRead(BTN\_START\_STOP) == LOW);

}

if (digitalRead(BTN\_ADD\_TIME) == LOW) {

delay(200);

if (maxTime < 180) {

maxTime += 60;

if (!running && seconds >= maxTime - 60) {

running = true;

}

}

while (digitalRead(BTN\_ADD\_TIME) == LOW);

}

if (digitalRead(BTN\_RESET) == LOW) {

delay(200);

running = false;

seconds = 0;

maxTime = defaultMaxTime;

while (digitalRead(BTN\_RESET) == LOW);

}

// Atualização do cronômetro

unsigned long currentMillis = millis();

if (running && currentMillis - previousMillis >= 1000) {

previousMillis = currentMillis;

if (seconds < maxTime) {

seconds++;

} else {

running = false;

}

}

displayTime(seconds);

}