#include <LiquidCrystal.h>

#define VCC 12

//array to hold our comparators

const int comparators[] = {38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53};

// Constant integers for 4 7-segment display

const int pinA = 22, pinB = 23, pinC =24, pinD = 25,

pinE = 26, pinF = 27, pinG = 28, pinDec = 29, D1 = 2,

D2 = 3, D3 = 4, D4 = 5;

// LCD Constants

const int rs = 35, en = 34, d4 = 33, d5 = 32, d6 = 31, d7 = 30;

const double VOLTAGE\_STEP = 0.75;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

String outVoltage;

void sevenSegmentHelper(char digit, int index);

void sevenSegmentNumbers(char digit);

double comparatorRead();

void setup()

{

//initialize comparator pins as inputs

for (int index = 0; index < 16; index++)

{

pinMode(comparators[index], INPUT);

}

pinMode(pinA, OUTPUT);

pinMode(pinB, OUTPUT);

pinMode(pinC, OUTPUT);

pinMode(pinD, OUTPUT);

pinMode(pinE, OUTPUT);

pinMode(pinF, OUTPUT);

pinMode(pinG, OUTPUT);

pinMode(pinDec, OUTPUT);

pinMode(D1, OUTPUT);

pinMode(D2, OUTPUT);

pinMode(D3, OUTPUT);

pinMode(D4, OUTPUT);

lcd.begin(16, 2);

Serial.begin(9600);

}

void loop()

{

int index;

char voltageChar;

// Get the whole number voltage

outVoltage = (String)comparatorRead(15);

// Adds a 0 to the front of the number if not 10, 11, or 12

if (outVoltage.charAt(1) == '.')

{

outVoltage = "0" + outVoltage;

}

// Iterate through outVoltage and use seven seg helpers

for (index = 0; index < 2; index++)

{

voltageChar = outVoltage.charAt(index);

if (voltageChar != '.')

{

sevenSegmentHelper(voltageChar, index);

}

}

for (index = 3; index < 5; index++)

{

voltageChar = outVoltage.charAt(index);

if (voltageChar != '.')

{

sevenSegmentHelper(voltageChar, index-1);

}

}

Serial.println("Voltage: " + outVoltage);

lcd.setCursor(0,0);

lcd.print("Voltage: " + outVoltage);

}

double comparatorRead(int index)

{

double calculatedVoltage;

if (index >= 0 && digitalRead(comparators[index]) == HIGH)

{

return (VOLTAGE\_STEP + comparatorRead(index-1));

}

return 0;

}

void sevenSegmentHelper(char digit, int index)

{

switch (index)

{

// If first digit

case 0:

digitalWrite(D1, LOW);

digitalWrite(D2, HIGH);

digitalWrite(D3, HIGH);

digitalWrite(D4, HIGH);

digitalWrite(pinDec, LOW);

sevenSegmentNumbers(digit);

break;

// If second digit

case 1:

digitalWrite(D1, HIGH);

digitalWrite(D2, LOW);

digitalWrite(D3, HIGH);

digitalWrite(D4, HIGH);

digitalWrite(pinDec, HIGH);

sevenSegmentNumbers(digit);

break;

// If third digit

case 2:

digitalWrite(D1, HIGH);

digitalWrite(D2, HIGH);

digitalWrite(D3, LOW);

digitalWrite(D4, HIGH);

digitalWrite(pinDec, LOW);

sevenSegmentNumbers(digit);

break;

// If fourth digit

case 3:

digitalWrite(D1, HIGH);

digitalWrite(D2, HIGH);

digitalWrite(D3, HIGH);

digitalWrite(D4, LOW);

digitalWrite(pinDec, LOW);

sevenSegmentNumbers(digit);

break;

}

}

void sevenSegmentNumbers(char digit)

{

switch (digit)

{

// if digit is 0

case '0':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, HIGH);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, LOW);

delay(1);

break;

// if digit is 1

case '1':

digitalWrite(pinA, LOW);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, LOW);

digitalWrite(pinE, LOW);

digitalWrite(pinF, LOW);

digitalWrite(pinG, LOW);

delay(1);

break;

// if digit is 2

case '2':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, LOW);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, HIGH);

digitalWrite(pinF, LOW);

digitalWrite(pinG, HIGH);

delay(1);

break;

// if digit is 3

case '3':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, LOW);

digitalWrite(pinF, LOW);

digitalWrite(pinG, HIGH);

delay(1);

break;

// if digit is 4

case '4':

digitalWrite(pinA, LOW);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, LOW);

digitalWrite(pinE, LOW);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, HIGH);

delay(1);

break;

// if digit is 5

case '5':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, LOW);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, LOW);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, HIGH);

delay(1);

break;

// if digit is 6

case '6':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, LOW);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, HIGH);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, HIGH);

delay(1);

break;

// if digit is 7

case '7':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, LOW);

digitalWrite(pinE, LOW);

digitalWrite(pinF, LOW);

digitalWrite(pinG, LOW);

delay(1);

break;

// if digit is 8

case '8':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, HIGH);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, HIGH);

delay(1);

break;

// if digit is 9

case '9':

digitalWrite(pinA, HIGH);

digitalWrite(pinB, HIGH);

digitalWrite(pinC, HIGH);

digitalWrite(pinD, HIGH);

digitalWrite(pinE, LOW);

digitalWrite(pinF, HIGH);

digitalWrite(pinG, HIGH);

delay(1);

break;

}

}