

**RP-IPRC MUSANZE**

**DEPARTEMENT: ELECTRICAL AND ELECTRONICS ENGINEERING**

**LEVEL: YEAR III**

**MODULE NAME: PROGRAMMABLE INTEGRATED CIRCUITS (ICs) WORKSHOP**

**MODULE COD**E: ELT 312

**TITLE: MONITORING TIME IN IPRC MUSANZE BY USING REAL TIME CLOCK (RTC) AND LCD(LIQUID CRYSTAL DISPLAY)**

**PREPARED BY: NIYONKURU Jean Baptiste 19RP07704**

**NYIRIMANA Gilbert 19RP10092**

**LECTURER: ISHIMWE Viviane**

**ABSTRACT**

Nowadays people works by referring on the time in order to start and finish their jobs and they prefer digital clocks more than analogue clocks because it is more accuracy, tiny in size and it is easy to use, also digital clock reduce human errors. Different companies manufacture digital clocks because it is found in all kinds of devices such cars, radios, cell phone in order to increase its quality.

Here in IPRC MUSANZE we can monitor time by using liquid crystal display (LCD) and real time clock (RTC) module

In this project the real time is displayed with the help liquid crystal display (LCD) and real time clock

Real Time Clock is used tracks over the Real Time. It displays the time in hours: minutes format.

**PROBLEM STATEMENT**

This project can helps lecturers and students to know the time as clock counts hours, minutes and seconds

This project can help all workers in IPRC MUSANZE to be on time (not coming late) and this an help all workers to work in well organised and competent way.

Also it can help to get an accurate universal time

**BLOCK DIAGRAM**

**ARDUINO UNO**

**LIQUID CRYSTAL DISPLAY**

**RTC**

**RESET PUSH BUTTON**

**DESCRIPTION**

This circuit is composed by different electronics devices. In this part we are going to explain each component and its participation in the circuit:

Some of these components are RTC which stands by Real Time Clock: RTC in this circuit must accurately keep time, even when the device is powered off.

We also use Arduino Uno in this circuit Arduino Uno is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor program you entered on it., turning on an LED, publishing something online. In our circuit the output is liquid crystal display, so the Arduino Uno is used to turning it ON according to the program you entered on it.

The other component used is called push button this is used to set and to reset the time according to your wish.

A Liquid Crystal Display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light- modulating properties of liquid crystal combined with polarizers.

Liquid crystal does not emit light directly, instead using a backlight or reflector to produce images in colour or monochrome. LCDs are available to display arbitrary images (as in general-purpose computer display) or fixed images with low information content, which can be displayed or hidden. For instance: present words, digits, and seven segment displays as in digital clock are all good examples of devices with these displays used displays are widely used in digital clocks as it is used in our circuit.

Also in this circuit we also use a Jumper wires: are used for making connections between items on your breadboard and your Arduino's header pins. We Use them to wire up all your circuits!

**ARDUINO CODE**

// Real time clock and calendar with set buttons using DS1307 and Arduino

// include LCD library code

#include <LiquidCrystal.h>

// include Wire library code (needed for I2C protocol devices)

#include <Wire.h>

// LCD module connections (RS, E, D4, D5, D6, D7)

LiquidCrystal lcd(5,6,7,8,9,10);

void setup() {

pinMode(3, INPUT\_PULLUP); // button1 is connected to pin 8

pinMode(4, INPUT\_PULLUP); // button2 is connected to pin 9

// set up the LCD's number of columns and rows

lcd.begin(16, 2);

Wire.begin(); // Join i2c bus

}

char Time[] = "TIME: : : ";

char Calendar[] = "DATE: / /20 ";

byte i, second, minute, hour, date, month, year;

void DS1307\_display(){

// Convert BCD to decimal

second = (second >> 4) \* 10 + (second & 0x0F);

minute = (minute >> 4) \* 10 + (minute & 0x0F);

hour = (hour >> 4) \* 10 + (hour & 0x0F);

date = (date >> 4) \* 10 + (date & 0x0F);

month = (month >> 4) \* 10 + (month & 0x0F);

year = (year >> 4) \* 10 + (year & 0x0F);

// End conversion

Time[12] = second % 10 + 48;

Time[11] = second / 10 + 48;

Time[9] = minute % 10 + 48;

Time[8] = minute / 10 + 48;

Time[6] = hour % 10 + 48;

Time[5] = hour / 10 + 48;

Calendar[14] = year % 10 + 48;

Calendar[13] = year / 10 + 48;

Calendar[9] = month % 10 + 48;

Calendar[8] = month / 10 + 48;

Calendar[6] = date % 10 + 48;

Calendar[5] = date / 10 + 48;

lcd.setCursor(0, 0);

lcd.print(Time); // Display time

lcd.setCursor(0, 1);

lcd.print(Calendar); // Display calendar

}

void blink\_parameter(){

byte j = 0;

while(j < 10 && digitalRead(3) && digitalRead(4)){

j++;

delay(25);

}

}

byte edit(byte x, byte y, byte parameter){

char text[3];

while(!digitalRead(3)); // Wait until button (pin #8) released

while(true){

while(!digitalRead(4)){ // If button (pin #9) is pressed

parameter++;

if(i == 0 && parameter > 23) // If hours > 23 ==> hours = 0

parameter = 0;

if(i == 1 && parameter > 59) // If minutes > 59 ==> minutes = 0

parameter = 0;

if(i == 2 && parameter > 31) // If date > 31 ==> date = 1

parameter = 1;

if(i == 3 && parameter > 12) // If month > 12 ==> month = 1

parameter = 1;

if(i == 4 && parameter > 99) // If year > 99 ==> year = 0

parameter = 0;

sprintf(text,"%02u", parameter);

lcd.setCursor(x, y);

lcd.print(text);

delay(200); // Wait 200ms

}

lcd.setCursor(x, y);

lcd.print(" "); // Display two spaces

blink\_parameter();

sprintf(text,"%02u", parameter);

lcd.setCursor(x, y);

lcd.print(text);

blink\_parameter();

if(!digitalRead(3)){ // If button (pin #8) is pressed

i++; // Increament 'i' for the next parameter

return parameter; // Return parameter value and exit

}

}

}

void loop() {

if(!digitalRead(3)){ // If button (pin #8) is pressed

i = 0;

hour = edit(5, 0, hour);

minute = edit(8, 0, minute);

date = edit(5, 1, date);

month = edit(8, 1, month);

year = edit(13, 1, year);

// Convert decimal to BCD

minute = ((minute / 10) << 4) + (minute % 10);

hour = ((hour / 10) << 4) + (hour % 10);

date = ((date / 10) << 4) + (date % 10);

month = ((month / 10) << 4) + (month % 10);

year = ((year / 10) << 4) + (year % 10);

// End conversion

// Write data to DS1307 RTC

Wire.beginTransmission(0x68); // Start I2C protocol with DS1307 address

Wire.write(0); // Send register address

Wire.write(0); // Reset sesonds and start oscillator

Wire.write(minute); // Write minute

Wire.write(hour); // Write hour

Wire.write(1); // Write day (not used)

Wire.write(date); // Write date

Wire.write(month); // Write month

Wire.write(year); // Write year

Wire.endTransmission(); // Stop transmission and release the I2C bus

delay(200); // Wait 200ms

}

Wire.beginTransmission(0x68); // Start I2C protocol with DS1307 address

Wire.write(0); // Send register address

Wire.endTransmission(false); // I2C restart

Wire.requestFrom(0x68, 7); // Request 7 bytes from DS1307 and release I2C bus at end of reading

second = Wire.read(); // Read seconds from register 0

minute = Wire.read(); // Read minuts from register 1

hour = Wire.read(); // Read hour from register 2

Wire.read(); // Read day from register 3 (not used)

date = Wire.read(); // Read date from register 4

month = Wire.read(); // Read month from register 5

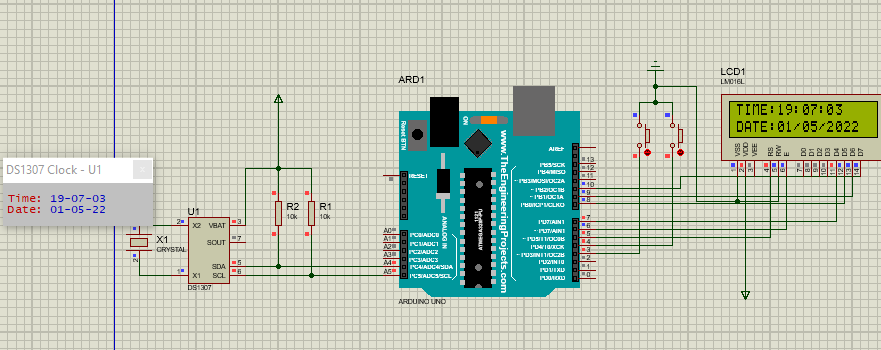
year = Wire.read(); // Read year from register 6

DS1307\_display(); // Diaplay time & calendar

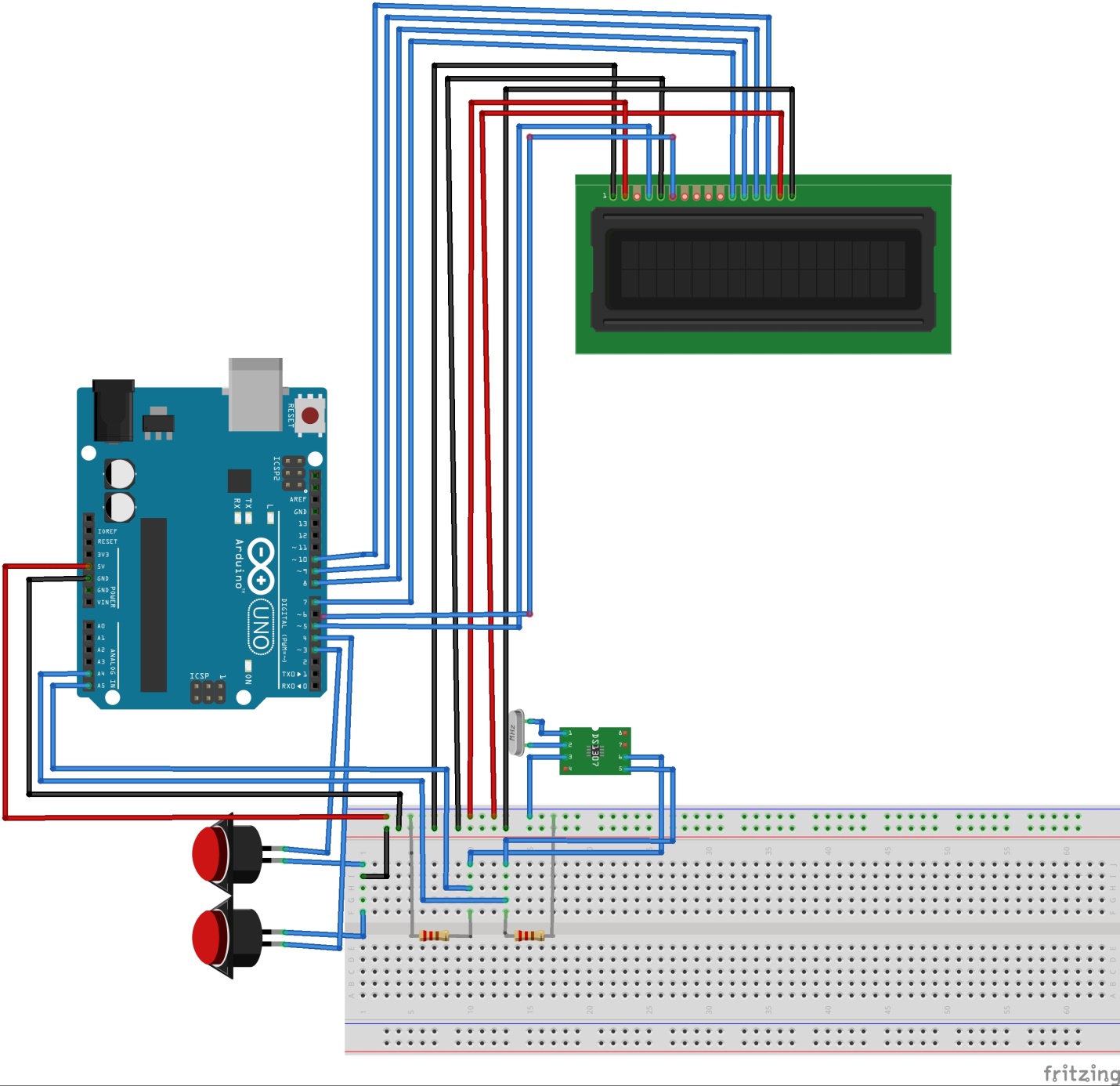
delay(50); // Wait 50ms

}

**PROTEUS CIRCUIT**

****

**FRITZING CIRCUIT**

****