// MP3 Alarm clock Using Arduino & DFplayer mini, this code also supports DS1307 RTC

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

#include <DS3231.h> //Download DS3231 Library Here (also supports DS1307): http://www.rinkydinkelectronics.com/library.php?id=73

#include <Wire.h>

#include <EEPROM.h>

const int rs = 7, en = 6, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

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

SoftwareSerial DFplayer(13, 8);

DS3231 rtc(SDA, SCL);

# define Start\_Byte 0x7E

# define Version\_Byte 0xFF

# define Command\_Length 0x06

# define End\_Byte 0xEF

# define Acknowledge 0x00

Time t;

int Song\_number;

int volume = 10;

int IsPlaying = 9;

int hh = 0, mm = 0, ss = 0;

int Date = 1, Month = 1, Year = 2020;

const int setting = A0;

const int enter = A1;

const int inc = A2;

const int dec = A3;

const int ext = 10;

bool busy = 0;

bool back = 0;

const char\* set\_month\_str[] = {"Null", "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"};

const char\* set\_day\_str[] = {"Null", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY", "FRIDAY", "SATURDAY", "SUNDAY"};

void setup ()

{

Wire.begin();

lcd.begin(16, 2);

if (EEPROM.read( 0) != 150)

{

for (int i = 0; i <= 255; i++)

{

EEPROM.write(i, 0);

}

EEPROM.write(0, 150);

}

pinMode(IsPlaying, INPUT);

pinMode(setting, INPUT\_PULLUP);

pinMode(enter, INPUT\_PULLUP);

pinMode(inc, INPUT\_PULLUP);

pinMode(dec, INPUT\_PULLUP);

pinMode(ext, INPUT\_PULLUP);

DFplayer.begin(9600);

rtc.begin();

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" WELCOME");

lcd.setCursor(0, 1);

lcd.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

delay(1500);

lcd.clear();

}

void loop ()

{

t = rtc.getTime();

Status();

play();

if (busy == 1)

{

lcd.clear();

lcd.setCursor(4, 0);

lcd.print(rtc.getTimeStr());

lcd.setCursor(4, 1);

lcd.print("PLAYING:");

lcd.print(Song\_number);

if (digitalRead(inc) == 0 && digitalRead(dec) == 0)

{

exe\_cmd(0x0E, 0, 0);

}

}

if (busy == 0)

{

lcd.clear();

lcd.setCursor(4, 0);

lcd.print(rtc.getTimeStr());

lcd.setCursor(3, 1);

lcd.print(rtc.getDateStr());

}

if (digitalRead(setting) == LOW)

{

lcd.clear();

delay(250);

while (true)

{

lcd.setCursor(0, 0);

lcd.print(" ALARM SETTING ");

lcd.setCursor(0, 1);

lcd.print(">>>>>>>><<<<<<<<");

if (digitalRead(enter) == LOW)

{

delay(250);

track\_setting\_menu();

break;

}

if (digitalRead(setting) == LOW)

{

break;

}

}

if (digitalRead(setting) == LOW)

{

delay(250);

while (true)

{

lcd.setCursor(0, 0);

lcd.print(" TIME SETTING ");

lcd.setCursor(0, 1);

lcd.print(">>>>>>>><<<<<<<<");

if (digitalRead(enter) == LOW)

{

delay(250);

time\_setting();

break;

}

if (digitalRead(setting) == LOW)

{

break;

}

}

}

if (digitalRead(setting) == LOW)

{

delay(250);

lcd.clear();

while (true)

{

lcd.setCursor(0, 0);

lcd.print(" DATE SETTING ");

lcd.setCursor(0, 1);

lcd.print(">>>>>>>><<<<<<<<");

if (digitalRead(enter) == LOW)

{

delay(250);

date\_setting();

break;

}

if (digitalRead(setting) == LOW)

{

delay(250);

lcd.clear();

break;

}

}

}

}

delay(50);

}

void PlaySpecSong(int song\_num)

{

exe\_cmd(0x03, 0, song\_num);

delay(500);

exe\_cmd(0x06, 0, volume);

delay(500);

}

void Status()

{

if (digitalRead(IsPlaying) == LOW)

{

busy = 1;

}

if (digitalRead(IsPlaying) == HIGH)

{

busy = 0;

}

}

void exe\_cmd(byte CMD, byte Par1, byte Par2)

{

word checksum = -(Version\_Byte + Command\_Length + CMD + Acknowledge + Par1 + Par2);

byte Command\_line[10] = {Start\_Byte, Version\_Byte, Command\_Length, CMD, Acknowledge, Par1, Par2, highByte(checksum), lowByte(checksum), End\_Byte};

for (byte x = 0; x < 10; x++)

{

DFplayer.write(Command\_line[x]);

}

}

void time\_setting()

{

delay(250);

hh = t.hour;

mm = t.min;

A:

lcd.clear();

back = 0;

while (true)

{

lcd.setCursor(0, 0);

lcd.print("SET HOURS [HH]: ");

lcd.setCursor(0, 1);

if (hh < 10) lcd.print("0");

lcd.print(hh);

if (digitalRead(inc) == LOW)

{

hh = hh + 1;

if (hh >= 24) hh = 0;

delay(200);

}

if (digitalRead(dec) == LOW)

{

hh = hh - 1;

if (hh < 0) hh = 0;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

return;

}

delay(50);

}

back = 0;

while (true)

{

lcd.setCursor(0, 0);

lcd.print("SET MINUTE [MM]: ");

lcd.setCursor(0, 1);

if (mm < 10) lcd.print("0");

lcd.print(mm);

if (digitalRead(inc) == LOW)

{

mm = mm + 1;

if (mm >= 60) mm = 0;

delay(200);

}

if (digitalRead(dec) == LOW)

{

mm = mm - 1;

if (mm < 0) mm = 0;

delay(200);

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

if (digitalRead(enter) == LOW)

{

rtc.setTime(hh, mm, 0);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" TIME SAVED ");

lcd.setCursor(0, 1);

lcd.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

delay(1500);

lcd.clear();

break;

}

delay(50);

}

if (back == 1) goto A;

}

void date\_setting()

{

delay(250);

A:

back = 0;

lcd.clear();

int y = t.dow;

Date = t.date;

while (true)

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("SET DAY: ");

lcd.setCursor(0, 1);

lcd.print(set\_day\_str[y]);

if (digitalRead(inc) == LOW)

{

y = y + 1;

if (y > 7) y = 1;

delay(200);

}

if (digitalRead(dec) == LOW)

{

y = y - 1;

if (y < 1) y = 7;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

return;

}

delay(50);

}

B:

back = 0;

lcd.clear();

while (true)

{

lcd.setCursor(0, 0);

lcd.print("SET DATE [DD]: ");

lcd.setCursor(0, 1);

if (Date < 10) lcd.print("0");

lcd.print(Date);

if (digitalRead(inc) == LOW)

{

Date = Date + 1;

if (Date > 31) Date = 1;

delay(200);

}

if (digitalRead(dec) == LOW)

{

Date = Date - 1;

if (Date < 1) Date = 1;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

delay(50);

}

if (back == 1) goto A;

C:

back = 0;

for (int i = 1; i <= 12; i++)

{

if (strcmp(rtc.getMonthStr(), set\_month\_str[i]) == 0)

{

Month = i;

break;

}

}

while (true)

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("SET MONTH: ");

lcd.setCursor(0, 1);

lcd.print(set\_month\_str[Month]);

if (digitalRead(inc) == LOW)

{

Month = Month + 1;

if (Month > 12) Month = 1;

delay(200);

}

if (digitalRead(dec) == LOW)

{

Month = Month - 1;

if (Month < 1) Month = 1;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

delay(50);

}

if (back == 1) goto B;

back = 0;

lcd.clear();

while (true)

{

lcd.setCursor(0, 0);

lcd.print("SET YEAR [YY]: ");

lcd.setCursor(0, 1);

lcd.print(Year);

if (digitalRead(inc) == LOW)

{

Year = Year + 1;

if (Year > 2100) Year = 2100;

delay(200);

}

if (digitalRead(dec) == LOW)

{

Year = Year - 1;

if (Year < 2020) Year = 2020;

delay(200);

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

if (digitalRead(enter) == LOW)

{

rtc.setDate(Date, Month, Year);

if (y == 1) rtc.setDOW(MONDAY);

if (y == 2) rtc.setDOW(TUESDAY);

if (y == 3) rtc.setDOW(WEDNESDAY);

if (y == 4) rtc.setDOW(THURSDAY);

if (y == 5) rtc.setDOW(FRIDAY);

if (y == 6) rtc.setDOW(SATURDAY);

if (y == 7) rtc.setDOW(SUNDAY);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" DATE SAVED ");

lcd.setCursor(0, 1);

lcd.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

delay(1500);

lcd.clear();

break;

}

delay(50);

}

if (back == 1) goto C;

}

void track\_setting\_menu()

{

int track\_no\_add, Hrs\_add, Min\_add, on\_off\_add, track\_setting\_no = 1, mem\_loc = 5, play\_vol\_add;

delay(300);

while (true)

{

track\_no\_add = mem\_loc;

Hrs\_add = track\_no\_add + 1;

Min\_add = Hrs\_add + 1;

on\_off\_add = Min\_add + 1;

play\_vol\_add = on\_off\_add + 1;

lcd.clear();

lcd.setCursor(0, 0);

if (track\_setting\_no < 10)

{

lcd.print("ALARM:");

lcd.print("0");

lcd.print(track\_setting\_no);

}

else

{

lcd.print(track\_setting\_no);

}

lcd.setCursor(11, 0);

if (EEPROM.read( Hrs\_add) < 10)

{

lcd.print("0");

lcd.print(EEPROM.read( Hrs\_add));

}

else

{

lcd.print(EEPROM.read( Hrs\_add));

}

lcd.print(":");

if (EEPROM.read( Min\_add) < 10)

{

lcd.print("0");

lcd.print(EEPROM.read(Min\_add));

}

else

{

lcd.print(EEPROM.read(Min\_add));

}

lcd.setCursor(0, 1);

lcd.print("MP3:");

lcd.print(EEPROM.read(track\_no\_add));

lcd.setCursor(13, 1);

if (EEPROM.read( on\_off\_add) == 1)

{

lcd.print("ON");

}

if (EEPROM.read( on\_off\_add) == 0)

{

lcd.print("OFF");

}

if (digitalRead(inc) == LOW)

{

mem\_loc = mem\_loc + 5;

track\_setting\_no += 1;

if (track\_setting\_no > 5)

{

track\_setting\_no = 1;

mem\_loc = 5;

}

delay(200);

}

if (digitalRead(dec) == LOW)

{

mem\_loc = mem\_loc - 5;

track\_setting\_no -= 1;

if (track\_setting\_no < 1)

{

track\_setting\_no = 1;

mem\_loc = 5;

}

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

track\_no\_add = mem\_loc;

Hrs\_add = track\_no\_add + 1;

Min\_add = Hrs\_add + 1;

on\_off\_add = Min\_add + 1;

play\_vol\_add = on\_off\_add + 1;

track\_setting(track\_no\_add, Hrs\_add, Min\_add, on\_off\_add, play\_vol\_add);

break;

}

if (digitalRead(ext) == LOW)

{

return;

}

delay(50);

}

}

void track\_setting(int track\_no, int Hrs, int Min, int on\_off, int play\_vol)

{

int track\_no\_val, Hrs\_val, Min\_val, on\_off\_val, play\_vol\_val, repeat\_val;

track\_no\_val = EEPROM.read(track\_no);

Hrs\_val = EEPROM.read(Hrs);

Min\_val = EEPROM.read(Min);

on\_off\_val = EEPROM.read(on\_off);

play\_vol\_val = EEPROM.read(play\_vol);

lcd.clear();

delay(250);

A:

back = 0;

while (true)

{

lcd.setCursor(0, 0);

lcd.print("ON / OFF: ");

if (on\_off\_val == 1)

{

lcd.print("ON ");

}

else if (on\_off\_val == 0)

{

lcd.print("OFF ");

}

lcd.setCursor(0, 1);

lcd.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

if (digitalRead(inc) == LOW)

{

on\_off\_val = 1;

delay(250);

}

if (digitalRead(dec) == LOW)

{

on\_off\_val = 0;

delay(250);

}

if (digitalRead(enter) == LOW)

{

delay(250);

if (on\_off\_val == 1) break;

if (on\_off\_val == 0)

{

EEPROM.write(on\_off, on\_off\_val);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" ALARM OFF ");

lcd.setCursor(0, 1);

lcd.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

delay(1500);

return;

}

}

if (digitalRead(ext) == LOW)

{

delay(250);

return;

}

}

B:

back = 0;

while (true)

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("ENTER TRACK No: ");

lcd.setCursor(0, 1);

lcd.print(track\_no\_val);

if (digitalRead(inc) == LOW)

{

track\_no\_val += 1;

if (track\_no\_val > 255) track\_no\_val = 1;

delay(200);

}

if (digitalRead(dec) == LOW)

{

track\_no\_val -= 1;

if (track\_no\_val < 1) track\_no\_val = 255;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

delay(50);

}

if (back == 1) goto A;

C:

back = 0;

while (true)

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("VOLUME: ");

lcd.setCursor(0, 1);

lcd.print(play\_vol\_val);

if (digitalRead(inc) == LOW)

{

play\_vol\_val = play\_vol\_val + 1;

if (play\_vol\_val > 30) play\_vol\_val = 30;

delay(200);

}

if (digitalRead(dec) == LOW)

{

play\_vol\_val = play\_vol\_val - 1;

if (play\_vol\_val < 1) play\_vol\_val = 1;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

delay(50);

}

if (back == 1) goto B;

D:

back = 0;

while (true)

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("ENTER HOUR: ");

lcd.setCursor(0, 1);

lcd.print(Hrs\_val);

if (digitalRead(inc) == LOW)

{

Hrs\_val += 1;

if (Hrs\_val >= 24) Hrs\_val = 0;

delay(200);

}

if (digitalRead(dec) == LOW)

{

Hrs\_val -= 1;

if (Hrs\_val < 0) Hrs\_val = 23;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

delay(50);

}

if (back == 1) goto C;

E:

back = 0;

while (true)

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("ENTER MINUTE: ");

lcd.setCursor(0, 1);

lcd.print(Min\_val);

if (digitalRead(inc) == LOW)

{

Min\_val += 1;

if (Min\_val >= 60) Min\_val = 0;

delay(200);

}

if (digitalRead(dec) == LOW)

{

Min\_val -= 1;

if (Min\_val < 0) Min\_val = 59;

delay(200);

}

if (digitalRead(enter) == LOW)

{

delay(200);

break;

}

if (digitalRead(ext) == LOW)

{

delay(200);

back = 1;

break;

}

delay(50);

}

if (back == 1) goto D;

EEPROM.write( track\_no, track\_no\_val);

EEPROM.write( Hrs, Hrs\_val);

EEPROM.write( Min, Min\_val);

EEPROM.write( on\_off, on\_off\_val);

EEPROM.write( play\_vol, play\_vol\_val);

lcd.setCursor(0, 0);

lcd.print(" ALARM SAVED ");

lcd.setCursor(0, 1);

lcd.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

delay(1500);

lcd.clear();

}

void play()

{

t = rtc.getTime();

int Hrs = t.hour;

int Min = t.min;

int Sec = t.sec;

int play\_track\_no, play\_hrs, play\_min, play\_on\_off, mem\_loc = 5, play\_vol;

for (int i = 1; i <= 5; i++)

{

int track\_no\_add = mem\_loc;

int Hrs\_add = track\_no\_add + 1;

int Min\_add = Hrs\_add + 1;

int on\_off\_add = Min\_add + 1;

int play\_vol\_add = on\_off\_add + 1;

play\_track\_no = EEPROM.read(track\_no\_add);

play\_hrs = EEPROM.read(Hrs\_add);

play\_min = EEPROM.read(Min\_add);

play\_on\_off = EEPROM.read(on\_off\_add);

play\_vol = EEPROM.read(play\_vol\_add);

mem\_loc = mem\_loc + 5;

if (Hrs == play\_hrs && Min == play\_min && Sec == 0 && play\_on\_off == 1)

{

Song\_number = play\_track\_no;

PlaySpecSong(play\_track\_no);

volume = play\_vol;

}

}

}