|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Automatic Embedded Clock System | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Pranav Varadpande B. Tech Electronics & Communication  UIT-RGPV, Bhopal |  | | | |
|  | | | |  |  | | | |
|  | | |  | | |  | | |



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | | |  |
|  | INTRODUCTION | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | |  | | |  | |  |
|  |  | |  | | |  | |  |
|  |  |  | The traditional bells are being used as an alarm in many institutions varying from schools to companies too, i.e. the electromagnetic bell, which is set for fixed time and if we want to modify the settings it’s a very tough task.  The alarm system utilizes the alarm system every phone uses and it enables the organization to easily modify the time for the buzzer or the bell to ring. | | |  |  |  |
|  | | | | |
|  |  |  |  |
|  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | |  |  | |
|  | | Components | | | | |  | |
|  | |  |  | | |  |  | |
|  | 1. Arduino Nano (or any Arduino board) 2. DS1302 RTC Module 3. 16x4 LCD Module 4. 10K Potentiometer (rotary) 5. Pushbuttons 6. PCB Board 7. Connecting Wires 8. Soldering equipment | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | | |  |  | | |  |

#### 

// CONNECTIONS:

// DS1302 CLK/SCLK --> 5

// DS1302 DAT/IO --> 4

// DS1302 RST/CE --> 2

// DS1302 VCC --> 3.3v - 5v

// DS1302 GND --> GND

// introducing all libraries

#include <ThreeWire.h>

#include <RtcDS1302.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);

ThreeWire myWire(8,7,9); // IO, SCLK, CE

RtcDS1302<ThreeWire> Rtc(myWire);

//rtc date time variables

int a,b,c,d,e,f;

//slot hour and seconds variables

int h1,h2,h3,h4; int m1,m2,m3,m4;

//default time

int sh1=10,sm1=0,s1=0;int sh2=13,sm2=00,s2=00; int sh3=13,sm3=50,s3=00; int sh4=17,sm4=20,s4=00;

//string to store bluetooth data

String z;

//clock diagram display

byte customChar[8] = { //cursor(11,0)

B00011,

B00110,

B01001,

B11000,

B10100,

B10000,

B10000,

B11100

};

byte customChar1[8] = { //cursor(11,1)

B10000,

B10100,

B11000,

B10001,

B10000,

B01001,

B00110,

B00011

};

byte customChar2[8] = { //cursor(12,1)

B00100,

B01010,

B10001,

B00000,

B00000,

B00100,

B00100,

B11111

};

byte customChar3[8] = { //cursor(12,0)

B11111,

B00100,

B00100,

B00000,

B00000,

B00000,

B00000,

B00100};

byte customChar4[] = { //cursor(13,1)

B00001,

B00101,

B00011,

B10001,

B00001,

B10010,

B01100,

B11000

};

byte customChar5[] = { //cursor(13,0)

B11000,

B01100,

B10010,

B00011,

B00101,

B00001,

B00001,

B00111

};

void setup ()

{

Serial.begin(9600);

lcd.begin(20,4);

pinMode(A0,OUTPUT);

pinMode(A1,INPUT); //slot

pinMode(A2,INPUT);//hour

pinMode(A3,INPUT);//minute

pinMode(6,INPUT\_PULLUP);

pinMode(10,INPUT\_PULLUP);

pinMode(0,INPUT\_PULLUP);

pinMode(1,INPUT\_PULLUP);

// put your setup code here, to run once:

lcd.begin(20,4);

lcd.createChar(1, customChar);

lcd.createChar(2,customChar1);

lcd.createChar(3,customChar2);

lcd.createChar(4,customChar3);

lcd.createChar(5,customChar4);

lcd.createChar(6,customChar5);

Serial.begin(9600);

Serial.print("compiled: ");

Serial.print(\_\_DATE\_\_);

Serial.println(\_\_TIME\_\_);

Rtc.Begin();

RtcDateTime compiled = RtcDateTime(\_\_DATE\_\_, \_\_TIME\_\_);

printDateTime(compiled);

Serial.println();

if (!Rtc.IsDateTimeValid())

{

// Common Causes:

// 1) first time you ran and the device wasn't running yet

// 2) the battery on the device is low or even missing

Serial.println("RTC lost confidence in the DateTime!");

Rtc.SetDateTime(compiled);

}

if (Rtc.GetIsWriteProtected())

{

Serial.println("RTC was write protected, enabling writing now");

Rtc.SetIsWriteProtected(false);

}

if (!Rtc.GetIsRunning())

{

Serial.println("RTC was not actively running, starting now");

Rtc.SetIsRunning(true);

}

RtcDateTime now = Rtc.GetDateTime();

if (now < compiled)

{

Serial.println("RTC is older than compile time! (Updating DateTime)");

Rtc.SetDateTime(compiled);

}

else if (now > compiled)

{

Serial.println("RTC is newer than compile time. (this is expected)");

}

else if (now == compiled)

{

Serial.println("RTC is the same as compile time! (not expected but all is fine)");

}

}

#define countof(a) (sizeof(a) / sizeof(a[0]))

void printDateTime(const RtcDateTime& dt)

{

char datestring[20];

snprintf\_P(datestring,

countof(datestring),

PSTR("%02u/%02u/%04u %02u:%02u:%02u"),

dt.Month(),

dt.Day(),

dt.Year(),

dt.Hour(),

dt.Minute(),

dt.Second() );

Serial.print(datestring);

a=dt.Second();

b=dt.Minute();

c=dt.Hour();

d=dt.Day();

e=dt.Month();

f=dt.Year();

}

void loop ()

{analogWrite(A0,75);

int s= analogRead(A1);int s1=map(s,0,1023,1,5);//slot

int h=analogRead(A2); int hr=map(h,0,1023,0,24);//hour

int m=analogRead(A3); int mi=map(m,0,1023,0,59);//minute

int a1=digitalRead(6);//slot settings

int b1=digitalRead(10);//save

int c1=digitalRead(0);//reset

int d1=digitalRead(1);//back

lcd.setCursor(11,0);

lcd.write(1);

lcd.setCursor(11,1);

lcd.write(2);

lcd.setCursor(12,1);

lcd.write(3);

lcd.setCursor(12,0);

lcd.write(4);

lcd.setCursor(13,1);

lcd.write(5);

lcd.setCursor(13,0);

lcd.write(6);

if(d>=10){

lcd.setCursor(0,0);

lcd.print(d);}

else if(d<10){

lcd.setCursor(0,0);

lcd.print("0");

lcd.setCursor(1,0);

lcd.print(d);}

lcd.setCursor(2,0);

lcd.print("/");

if(e>=10){

lcd.setCursor(3,0);

lcd.print(e);}

else if(e<10){

lcd.setCursor(3,0);

lcd.print("0");

lcd.setCursor(4,0);

lcd.print(e);}

lcd.setCursor(5,0);

lcd.print("/");

lcd.setCursor(6,0);

lcd.print(f);

lcd.setCursor(0,1);

lcd.print(c);

lcd.setCursor(2,1);

lcd.print(":");

lcd.setCursor(3,1);

lcd.print(b);

lcd.setCursor(5,1);

lcd.print(":");

lcd.setCursor(6,1);

lcd.print(a);

if(a<10){

lcd.setCursor(6,1);

lcd.print("0");

lcd.setCursor(7,1);

lcd.print(a);

}

if(a>10){

lcd.setCursor(6,1);

lcd.print(a);

}

if(b<10){

lcd.setCursor(3,1);

lcd.print("0");

lcd.setCursor(4,1);

lcd.print(b);

}

if(b>10){

lcd.setCursor(3,1);

lcd.print(b);

}

if(c<10){

lcd.setCursor(0,1);

lcd.print("0");

lcd.setCursor(1,1);

lcd.print(c);

}

if(c>10){

lcd.setCursor(0,1);

lcd.print(c);

}

RtcDateTime now = Rtc.GetDateTime();

printDateTime(now);

Serial.println();

if (!now.IsValid()){

// Common Causes:

// 1) the battery on the device is low or even missing and the power line was disconnected

Serial.println("RTC lost confidence in the DateTime!");

}

delay(1000); // ten seconds

Serial.println(a1);

if(a1==0){

lcd.setCursor(0,2);

lcd.print("Default");lcd.clear();

if(s1==1){

lcd.setCursor(0,2);lcd.print("Slot 1");

lcd.setCursor(-4,3);lcd.print("Hour:");lcd.setCursor(5,3);lcd.print("Min:");

lcd.setCursor(2,3);lcd.print(hr);lcd.setCursor(10,3);lcd.print(mi);

if(b1==0){h1=hr; m1=mi;lcd.clear();lcd.setCursor(0,3);lcd.print("saved");}

if(d1==0){lcd.clear();}

}

if(s1==2){

lcd.setCursor(0,2);lcd.print("Slot 2");

lcd.setCursor(-4,3);lcd.print("Hour:");lcd.setCursor(5,3);lcd.print("Min:");

lcd.setCursor(2,3);lcd.print(hr);lcd.setCursor(10,3);lcd.print(mi);

if(b1==0){lcd.clear();lcd.setCursor(0,3);lcd.print("saved");h2=hr; m2=mi;}

if(d1==0){lcd.clear();}

}

if(s1==3){

lcd.setCursor(0,2);lcd.print("Slot 3");

lcd.setCursor(-4,3);lcd.print("Hour:");lcd.setCursor(5,3);lcd.print("Min:");

lcd.setCursor(2,3);lcd.print(hr);lcd.setCursor(10,3);lcd.print(mi);

if(b1==0){lcd.clear();lcd.setCursor(0,3);lcd.print("saved"); h3=hr; m3=mi;}

if(d1==0){lcd.clear();}

}

if(s1==4){

lcd.setCursor(0,2);lcd.print("Slot 4");

lcd.setCursor(-4,3);lcd.print("Hour:");lcd.setCursor(5,3);lcd.print("Min:");

lcd.setCursor(2,3);lcd.print(hr);lcd.setCursor(10,3);lcd.print(mi);

if(b1==0){h4=hr; m4=mi;lcd.clear();lcd.setCursor(0,3);lcd.print("saved");}

if(d1==0){lcd.clear();}

}}

if(c1==0){

lcd.setCursor(0,2);lcd.print("Default");

if(b1==0){lcd.clear(); h1=sh1;m1=sm1;h2=sh2;m2=sm2;h3=sh3;m3=sm3;h4=sh4;m4=sm4;lcd.setCursor(0,3);lcd.print("saved");}

}

if(c==h1||h2||h3||h4&&b==m1||m2||m3||m4){

if(a>0&&a<5){lcd.clear();lcd.setCursor(0,3);lcd.print("Timer is on");digitalWrite(13,HIGH);delay(2000);digitalWrite(13,LOW);lcd.setCursor(0,3);lcd.clear();}

else{

lcd.setCursor(11,0);

lcd.write(1);

lcd.setCursor(11,1);

lcd.write(2);

lcd.setCursor(12,1);

lcd.write(3);

lcd.setCursor(12,0);

lcd.write(4);

lcd.setCursor(13,1);

lcd.write(5);

lcd.setCursor(13,0);

lcd.write(6);

if(d>=10){

lcd.setCursor(0,0);

lcd.print(d);}

else if(d<10){

lcd.setCursor(0,0);

lcd.print("0");

lcd.setCursor(1,0);

lcd.print(d);}

lcd.setCursor(2,0);

lcd.print("/");

if(e>=10){

lcd.setCursor(3,0);

lcd.print(e);}

else if(e<10){

lcd.setCursor(3,0);

lcd.print("0");

lcd.setCursor(4,0);

lcd.print(e);}

lcd.setCursor(5,0);

lcd.print("/");

lcd.setCursor(6,0);

lcd.print(f);

lcd.setCursor(0,1);

lcd.print(c);

lcd.setCursor(2,1);

lcd.print(":");

lcd.setCursor(3,1);

lcd.print(b);

lcd.setCursor(5,1);

lcd.print(":");

lcd.setCursor(6,1);

lcd.print(a);

if(a<10){

lcd.setCursor(6,1);

lcd.print("0");

lcd.setCursor(7,1);

lcd.print(a);

}

if(a>10){

lcd.setCursor(6,1);

lcd.print(a);

}

if(b<10){

lcd.setCursor(3,1);

lcd.print("0");

lcd.setCursor(4,1);

lcd.print(b);

}

if(b>10){

lcd.setCursor(3,1);

lcd.print(b);

}

if(c<10){

lcd.setCursor(0,1);

lcd.print("0");

lcd.setCursor(1,1);

lcd.print(c);

}

if(c>10){

lcd.setCursor(0,1);

lcd.print(c);

}

}

}

}

|  |
| --- |
| CODE |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | WORKING | | | | |  | |
|  | |  |  | | |  |  | |
|  | The main idea would be to provide 5 slots for setting time which will alert people about specific task to be performed, for example for alerting students of college/school about lunch, event start time etc.  Along with slots, the alerting sound system will be a hooter which has the ability to transmit a sound of specific frequency.  A display interface is also connected which shows the current date and time as well as status of timer and slot setup. | | | | | | |
|  | Flow | | |  |  | | |
|  | 1. The LCD will show the current time and date. 2. On pressing switch1, slots will open from which one can select the slot number to be set. 3. The pot1 allows to switch between slots. 4. Pot2 sets the hour and Pot3 sets minute. | | |  | 1. Switch2 sets the value of hour and minute for particular slot. 2. Switch 3 ,on pressing, brings back to the home screen 3. Switch4, sets the default time slots. | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

