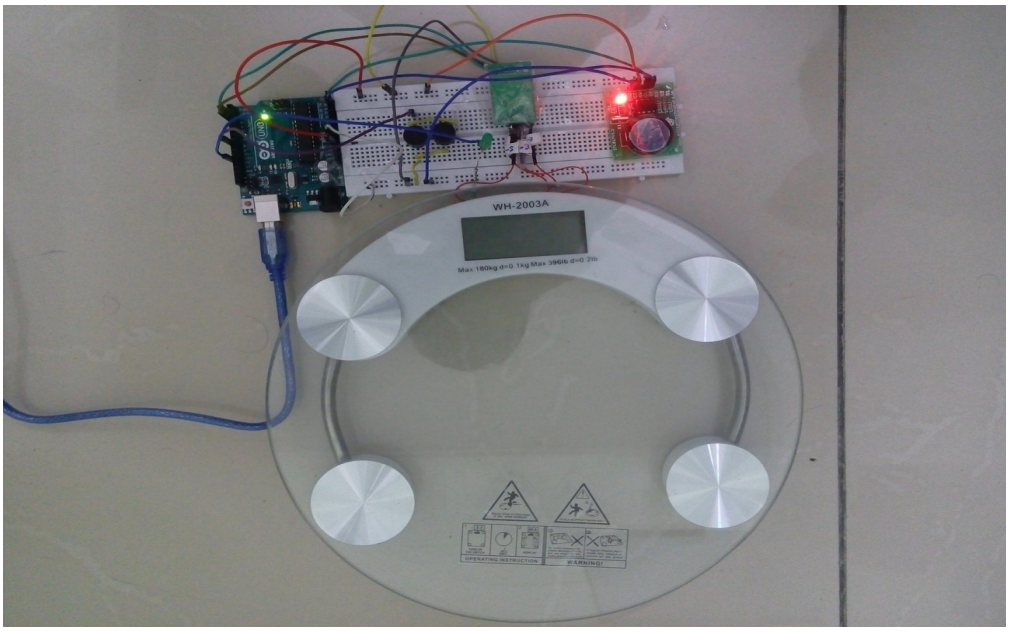
Digital and Logic Circuit

Project Mat Alarm



Group Members:-

1. Rajnish Kumar Robin
2. Sai Sasanka
3. Ritik Bhargava
4. Rajan Choudhary

Introduction:

The Mat Alarm is Specifically designed Alarm clock. It has been experienced that waking up is not hard its the drowsiness that doesn't let you wake up. So by any means if you stay awake for a while and step out of your bed the the drowsiness would end and you will not be dragged into bed. Here our objective is to make the user get off there bed when he is supposed to wake up. To fill this purpose we are using the weight of that person. Once you set Alarm in this device its not possible to shut the alarm off in any conventional way like those we know as Snooze or dismiss button. Lets See the working and Find out how useful it is.

Working:

- The mat alarm is initialized with the present time and the alarm is set.
- As soon as the time matches the buzzer starts ringing. The buzzer continues to ring unless and until the user switches it off.
- Now there is no snooze or 'off' button to switch the alarm off. To switch the alarm off the user has to stand on the mat.
- As soon as the person stands on the mat the buzzer stops ringing
- The main objective is to help the user to get rid of his sleep or drowsiness. For that there's an interval (user defined) which is nothing but the time for which the person has to stand on the mat to switch the alarm off.
- The clock starts counting the interval (say 10sec) as soon as the user stands on the mat and if the interval is not complete and the person steps down, the buzzer will continue to ring.
- The user should stand on the mat for 10sec (interval time) to switch the alarm completely off.
- To avoid situations like using a substitute(an object) to switch the alarm off, It is necessary for the user to set his/her weight in the alarm so that only the user can switch it off and it is also possible that another person of the same weight to switch it off.

This was all about the working procedure of the our Mat Alarm. As we can see there is not any other way of Switching off the alarm. Its impressive to use ones Weight as a Switching Device.
We are using the concept of state machines as a basic tool to implement this entire circuit.
Now, Lets Talk about the components that we have used to design the Mat Alarm.

Components:

In this Project we have used mainly Six components:

1. Arduino uno Board
2. 4 Load sensor
3. HX711
4. Real time clock Module
5. Buzzer
6. Led

Arduino uno Board:

Its a Programmable Microcontrolling device based on ATmega328p. In our Project it has been used to establish a sync between different components. We are using its 2 analog pin A5 and A4 to connect Real time clock and digital pin 2 and 3 for HX711 module. Further digital pin 8 has been used for buzzer and pin 12 for LED. To provide delay to the circuit we are using its very own crystal oscillator of 16 MHz frequency.

Load Sensor:

Load cell is a kind of transducer which converts ones body weight into electrical impulse. It works on the Principle of strain gauge. As some kind of weight gets onto the it there comes the dimensional change in the load cell which leads to the change in resistance of the device as a result the potential across it changes and this change is potential is taken as signal and used to measure the weight of the object or thing lying on it. We have used 4 load cells since each load cell was capable of measuring only 50 kgs. These load cells are connected in the configuration of weight stone bridge.

HX711:

HX711 is an operational amplifier. Since the voltage gain from the output of the load cell is too low to detect. We have used this Module to increase the signal output of Load sensor. It also converts the the analog output of the load cell to digital and then gives the serial data output at its SDA pin which is further connected to Arduino. So the main purpose of using this module is to do the amplification and ADC.

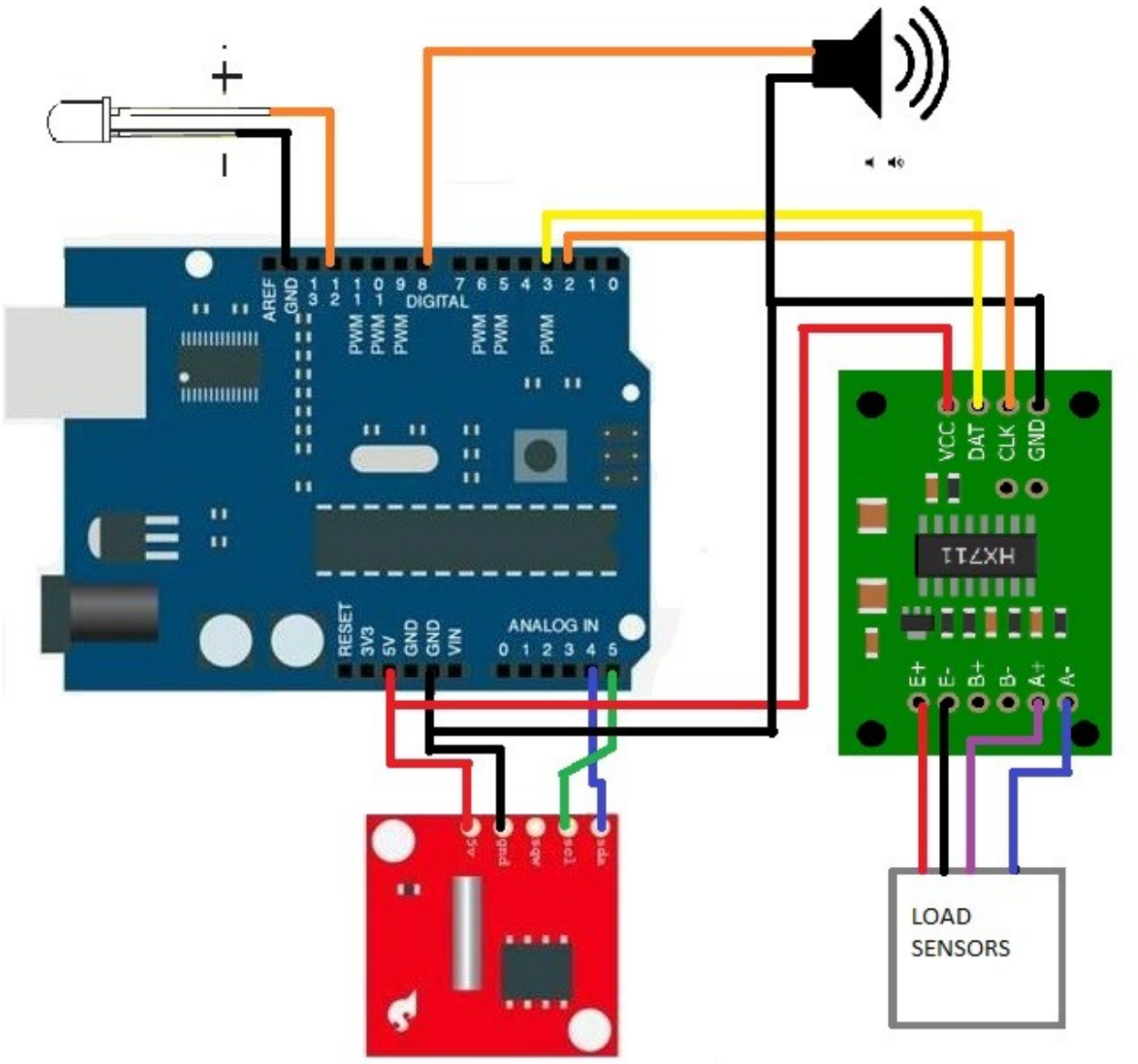
Real Time Clock Module (RTC):

Real time clock module is a basically a clock with some memory. It can store the present time and keep track of time from that point. It's IC is called ds1307 which consists of 56 bytes of ram. It stores the time in sectors like day,year,month,hour,min, second,milliseconds. It uses a crystal oscillator to do this and this crystal oscillator has a frequencyof 32.768 KHz.

Buzzer:

A piezoelectric buzzer is used in this project. It uses the principle of piezoelectricity i.e. if an electrical signal is applied on the piezoelectric material the material oscillates. Using suitable material and electrical signal we can generate sound ranging between 20-20000 cycles/sec(Human hearing range).

This finishes the component part of the project.
Carrying on toward the schematics of the circuit.



As mentioned above, to establish sync between all the compnents we have used programmable Atmega328p. The programming has been done on the arduino IDE. In the program two Libraries have been used one for HX711 Module i.e. <HX711.h> and the second for RTC module i.e. <wire.h>, rest of the functions were made in the source code given below.

#include "HX711.h"

```
#include "Wire.h"
#define DOUT  3
#define CLK  2
#define DS3231_I2C_ADDRESS 0x68
// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return( (val/10*16) + (val%10) );
}
// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return( (val/16*10) + (val%16) );
}
int buzzerPin=8;
int ledPin = 12;
 int limit=50;
 int check=0;
unsigned long previousMillis = 0;
const long inter = 10000;
HX711 scale(DOUT, CLK);

float calibration_factor =-10000;

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 calibration sketch");
  Serial.println("Remove all weight from scale");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press + or a to increase calibration factor");
  Serial.println("Press - or z to decrease calibration factor");
  Wire.begin();
  pinMode(buzzerPin,OUTPUT);
  pinMode(ledPin,OUTPUT);
  // set the initial time here:
  // DS3231 seconds, minutes, hours, day, date, month, year
  setDS3231time(50,50,03,03,01,

  scale.set_scale();
  scale.tare(); //Reset the scale to 0

  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale. Useful in
permanent scale projects.
  Serial.println(zero_factor);
}
void setDS3231time(byte second, byte minute, byte hour, byte dayOfWeek, byte
```

```
dayOfMonth, byte month, byte year)
{
  // sets time and date data to DS3231
  Wire.beginTransmission(DS3231_
  Wire.write(0); // set next input to start at the seconds register
  Wire.write(decToBcd(second)); // set seconds
  Wire.write(decToBcd(minute)); // set minutes
  Wire.write(decToBcd(hour)); // set hours
  Wire.write(decToBcd(dayOfWeek)Sunday, 7=Saturday)
  Wire.write(decToBcd(
  Wire.write(decToBcd(month)); // set month
  Wire.write(decToBcd(year)); // set year (0 to 99)
  Wire.endTransmission();
}
void readDS3231time(byte *second,
byte *minute,
byte *hour,
byte *dayOfWeek,
byte *dayOfMonth,
byte *month,
byte *year)
{
  Wire.beginTransmission(DS3231_
  Wire.write(0); // set DS3231 register pointer to 00h
  Wire.endTransmission();
  Wire.requestFrom(DS3231_I2C_
  // request seven bytes of data from DS3231 starting from register 00h
  *second = bcdToDec(Wire.read() & 0x7f);
  *minute = bcdToDec(Wire.read());
  *hour = bcdToDec(Wire.read() & 0x3f);
  *dayOfWeek = bcdToDec(Wire.read());
  *dayOfMonth = bcdToDec(Wire.read());
  *month = bcdToDec(Wire.read());
  *year = bcdToDec(Wire.read());
}
bool alarm()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
  &year);
  byte Min1,Hou1;

  Min1=51;
  Hou1=03;
 if(Min1==minute && Hou1==hour)
  {
    return 1;
  }
  else
```

```
  return 0;
}
bool weight()
{
  if(.453*scale.get_units()>
  {
    return 1;
  }
  else
  return 0;
}
void interval()
  {
    unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= inter) {
    previousMillis = currentMillis;
    check=check+1;
  }
  }
void displayTime()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  // retrieve data from DS3231
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
  &year);
  // send it to the serial monitor
  Serial.print(hour, DEC);
  // convert the byte variable to a decimal number when displayed
  Serial.print(":");
  if (minute<10)
  {
    Serial.print("0");
  }
  Serial.print(minute, DEC);
  Serial.print(":");
  if (second<10)
  {
    Serial.print("0");
  }
  Serial.print(second, DEC);
  Serial.print(" ");
  Serial.print(dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(month, DEC);
  Serial.print("/");
  Serial.print(year, DEC);
  Serial.print(" Day of week: ");
  switch(dayOfWeek){
  case 1:
```

```
    Serial.println("Sunday");
    break;
  case 2:
    Serial.println("Monday");
    break;
  case 3:
    Serial.println("Tuesday");
    break;
  case 4:
    Serial.println("Wednesday");
    break;
  case 5:
    Serial.println("Thursday");
    break;
  case 6:
    Serial.println("Friday");
    break;
  case 7:
    Serial.println("Saturday");
    break;
  }
}

void loop() {

  scale.set_scale(calibration_

  Serial.print("Reading: ");
  Serial.print(.453*scale.get_
  Serial.print(" kg"); //Change this to kg and re-adjust the calibration factor if you follow SI units like a
sane person
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_
  Serial.println();
 //calibration_factor-=10;
  delay(1000);
  displayTime(); // display the real-time clock data on the Serial Monitor,
  /*if(!alarm() && !weight())
  {
    digitalWrite(buzzerPin,LOW);
  }*/
   if(alarm()&& (check==0 || check==1))
   {
    if(!weight())
    {
    digitalWrite(buzzerPin,HIGH);
     Serial.println("Alarm ringing");
    }
    else
```

```
   {
     digitalWrite(buzzerPin,LOW);
     interval();
   }
  }
  else if(alarm()&& check==2)
  {
   digitalWrite(buzzerPin,LOW);
   digitalWrite(ledPin,HIGH);
  }

 if(!weight())
 {
  digitalWrite(ledPin,LOW);
 }

 if(Serial.available())
 {
  char temp = Serial.read();
  if(temp == '+' || temp == 'a')
    calibration_factor += 10;
  else if(temp == '-' || temp == 'z')
    calibration_factor -= 10;
 }

}
```

Working on this project was quite intresting. We got to learn alot of things about Digital and logic circuit. I hope this would help people in getting up early on time, and serve its purpose.

Thank You