

**IOT102 – PROJECT REPORT  
Smart Mirror**

**Group:** 6

**Student Name:** Lê Nguyên Phương, Lê Nguyễn Hữu Quốc

**Class:** IOT102\_4\_half 2

# Introduction

A smart mirror is a two-way mirror with an electronic display attached behind the glass. The display can show the viewer various kinds of information, such as time, date, weather, and news updates. This product would be useful for busy individuals who want to multitask and stay informed while on the go. Instead of constantly pulling out a device, one could get informed while finishing daily tasks.

# About hardware

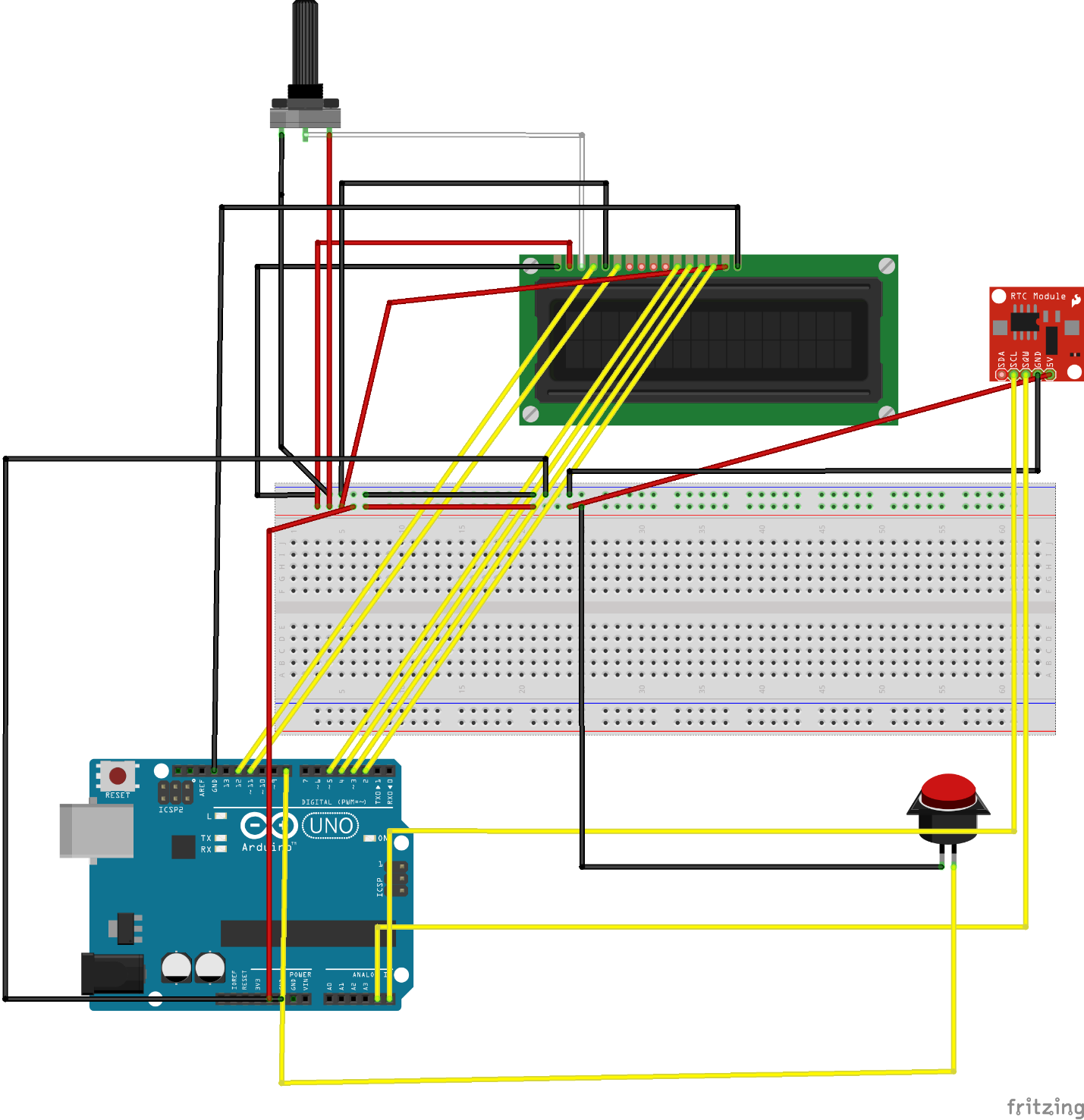
## Components

* Mirror: is used to reflect images;
* Frame: is used to protect the glass and contain the circuitry;
* Circuitry: is used to contain the code, transmit electric current and control signals.

## Properties of components

* Mirror: a transparent glass, with one-way reflective decal sticked on the glass surface. The rest of the glass around the display is blacked out by use of cardboard to ensure that there was a good reflection. Two-way mirror is more recommended because it lets the graphics on the display come through clearer;
* Frame: a solid cardboard box. If possible, use a wooden frame to make it sturdier and more solid. You can also decorate it to make it more eye-catching and beautiful;
* Circuitry:
  + Arduino UNO;
  + Solid core jumper wires;
  + 10KΩ potentiometer;
  + Resistors;
  + Switch;
  + Temperature sensor TMP36;
  + LCD (16×2 character display);
  + USB cable;
  + Power supply.

## Schematic design



# About software

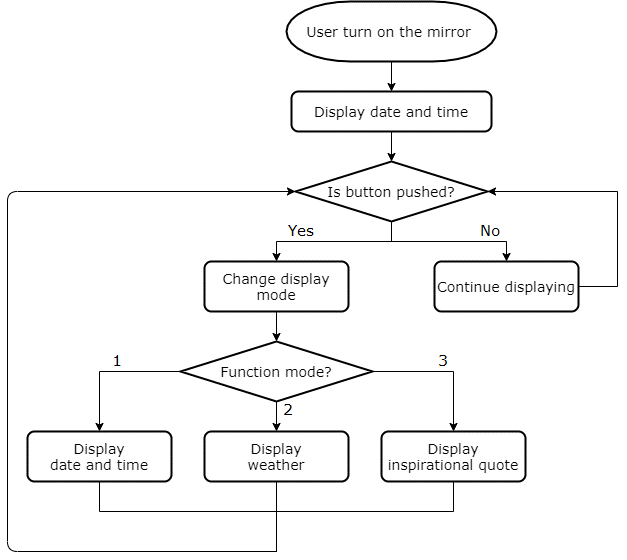
## Requirement analysis

The software have to implement necessary libraries: LiquidCrystal.h, DS1307RTC.h, TimeLib.h.

The software can run the following functions:

* Time and date displaying;
* Temperature displaying;
* Inspirational quotes displaying;

## Flowchart



## Source code

|  |
| --- |
| #include <LiquidCrystal.h> //implement LiquidCrystal and setCursor()  #include <DS1307RTC.h> //provide RTC control functions  #include <TimeLib.h> //implement functions getting real time  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  const int switchPin = 8;  int hits = 0;  int switchState = 0;  int prevSwitchState = 0;  int tempPin = 1;  int temValue;  float tempCelsius;  int countQuotes = 0;  String quotes[14] = {"Love for all", "hatred for none.",  "Whatever you do,", "do it well.",  "What we think,", "we become.",  "Strive for", "greatness.",  "And still,", "I rise.",  "It hurt because", "it mattered.",  "When words fail,", "music speaks."};  void setup()  {  lcd.clear();  lcd.begin(16, 2);  pinMode(switchPin, INPUT);  }  void loop()  {  switchState = digitalRead(switchPin);    if (switchState != prevSwitchState)  {  if (switchState == LOW)  {  hits++;  switch (hits % 3)  {  case 0:  timeDisplay();  delay(300);  break;  case 1:  weatherDisplay();  delay(300);  break;  case 2:  quotesDisplay();  delay(300);  break;  }  }  }  if (hits % 3 == 0)  {  timeDisplay();  delay(200);  }  prevSwitchState = switchState;  }  void timeDisplay()  {  tmElements\_t tm;    lcd.clear();    if (RTC.read(tm))  {  lcd.setCursor(4, 0);  print2digits(tm.Hour);  lcd.write(':');  print2digits(tm.Minute);  lcd.write(':');  print2digits(tm.Second);  lcd.setCursor(3, 1);  print2digits(tm.Day);  lcd.write('/');  print2digits(tm.Month);  lcd.write('/');  lcd.print(tmYearToCalendar(tm.Year));  }  }  void print2digits(int number)  {  if (number >= 0 && number < 10)  {  lcd.write('0');  }    lcd.print(number);  }  void weatherDisplay()  {  temValue = analogRead(tempPin);  tempCelsius = temValue \* 500 / 1023;  lcd.clear();    lcd.setCursor(2, 0);  lcd.print("Temperature: ");    lcd.setCursor(6, 1);  lcd.print(round(tempCelsius));  lcd.write(0xDF);  lcd.print("C");  }  void quotesDisplay()  {  lcd.clear();    lcd.setCursor(0, 0);  String text1 = quotes[countQuotes++];  lcd.print(text1);    lcd.setCursor(0, 1);  String text2 = quotes[countQuotes++];  lcd.print(text2);    if (countQuotes >= 14)  {  countQuotes %= 14;  }  } |

# Implementation and inspection

* Advantages:
  + Cheap and easy to implement;
  + No need to use additional third-party software;
  + No need to use other devices like phone, tablet, laptop.
* Disvantages:
  + The data is taken from the sensor so the accuracy is not high;
  + Not using two-way glass so the graphics on the display is not sharp.
* Development directions:
  + Increase size and make a stronger frame;
  + Add motion sensor to detect people coming;
  + Add more kinds of information, such as humidity, chance of rain;
  + Add feature connecting wifi to get data from Google;
  + Add feature synchronizing data from other platform, such as Google, Facebook.

# Conclusion

Currently, the amount of work to be done is increasing, while our free time is less and less. Owning a smart mirror like this will help us stay informed while still saving time. However, this product has great potential mostly in luxury markets due to current high costs. Smart mirrors can be produced at home quite easily, depending on how complex one wants to make it.