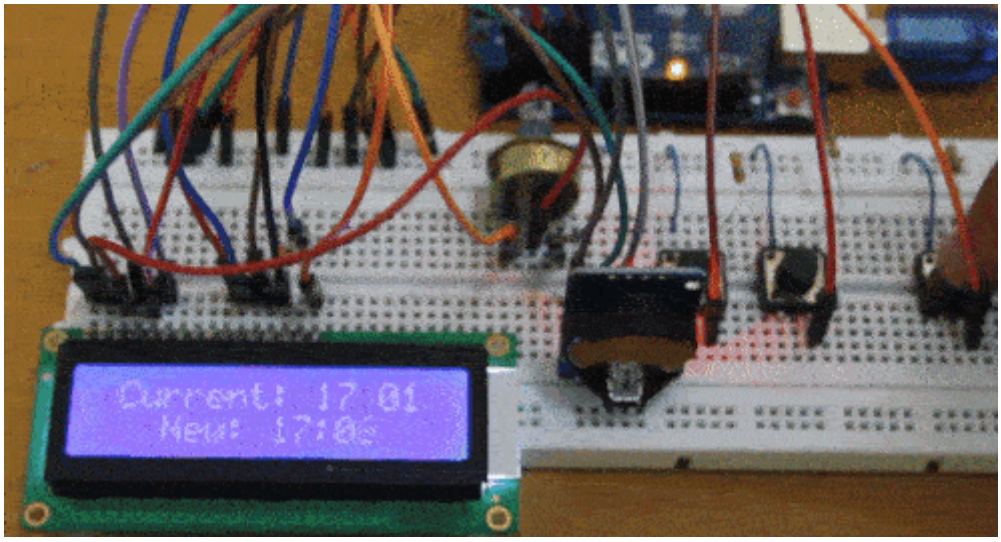
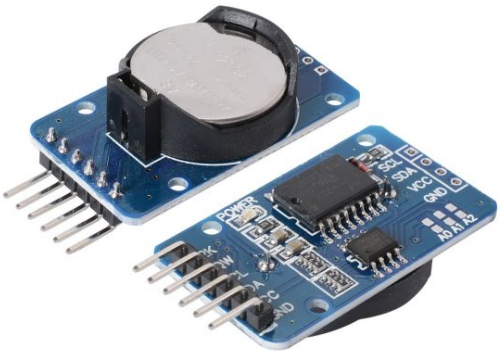
استخدام DS3231 RTC Module مع الاردوينو

سنتعلم في هذا المشروع كيفية إستخدام وحدة RTC “Real Time Clock” وعرض التوقيت على شاشة LCD



المكونات المطلوبة

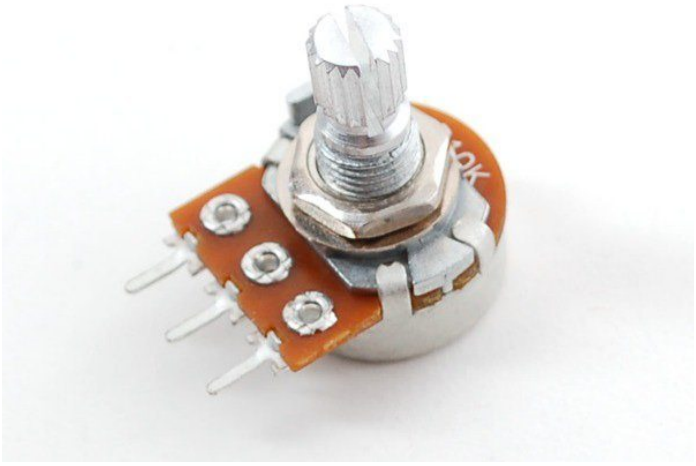
Arduino Uno



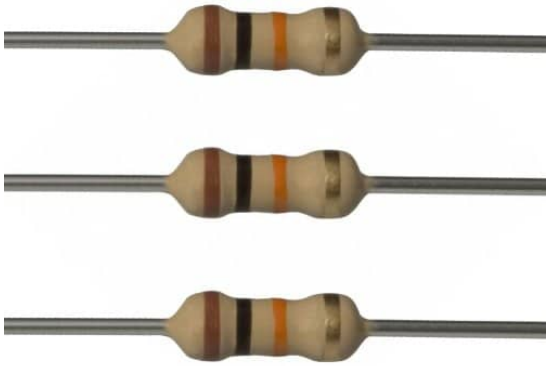
RTC – Real Time Clock

LCD 16×2

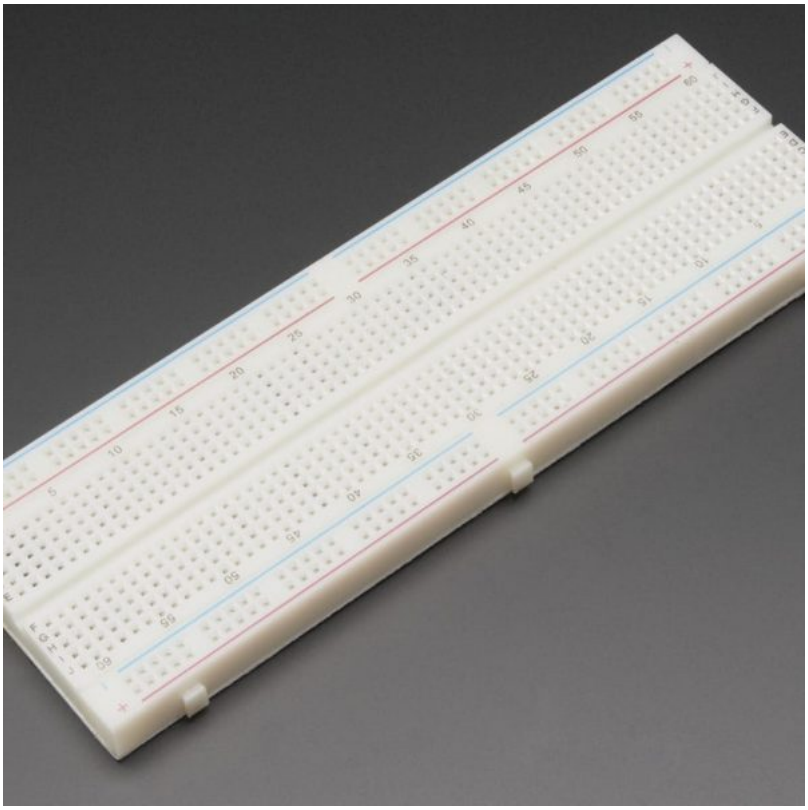
Push Buttons



Potentiometer 10K Ω



10k Ohm Resistors



Breadboard

Wires

الشاشة LCD

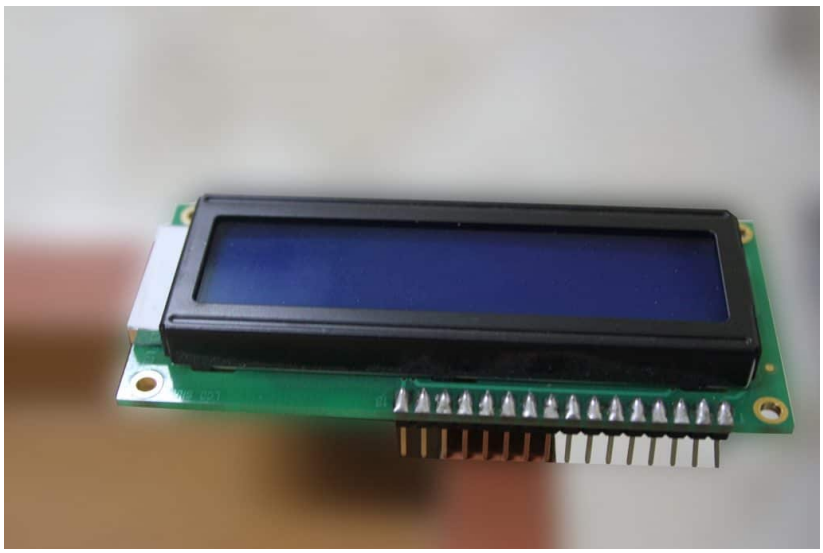
تعمل الشاشة في أحد الحالات التالية:

1- ان تستقبل امر من الارودوينو و تقوم بتنفيذة مثلا: امر مسح الشاشة و امر التهيئة

```
lcd.begin(16,2);
lcd.clear();
```

2- ان تستقبل معلومات من الاردوينو و تقوم بعرضها مثلا : كتابة جملة معينة

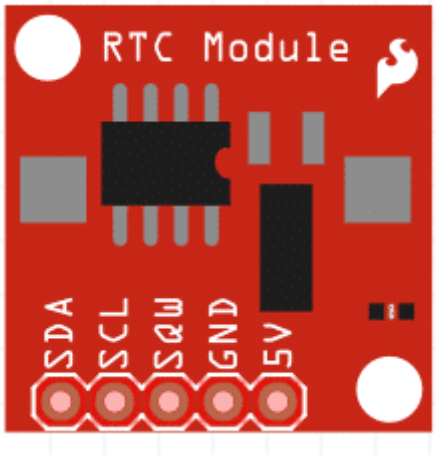
```
lcd.print("Hello");
```



DS3231 RTC Module

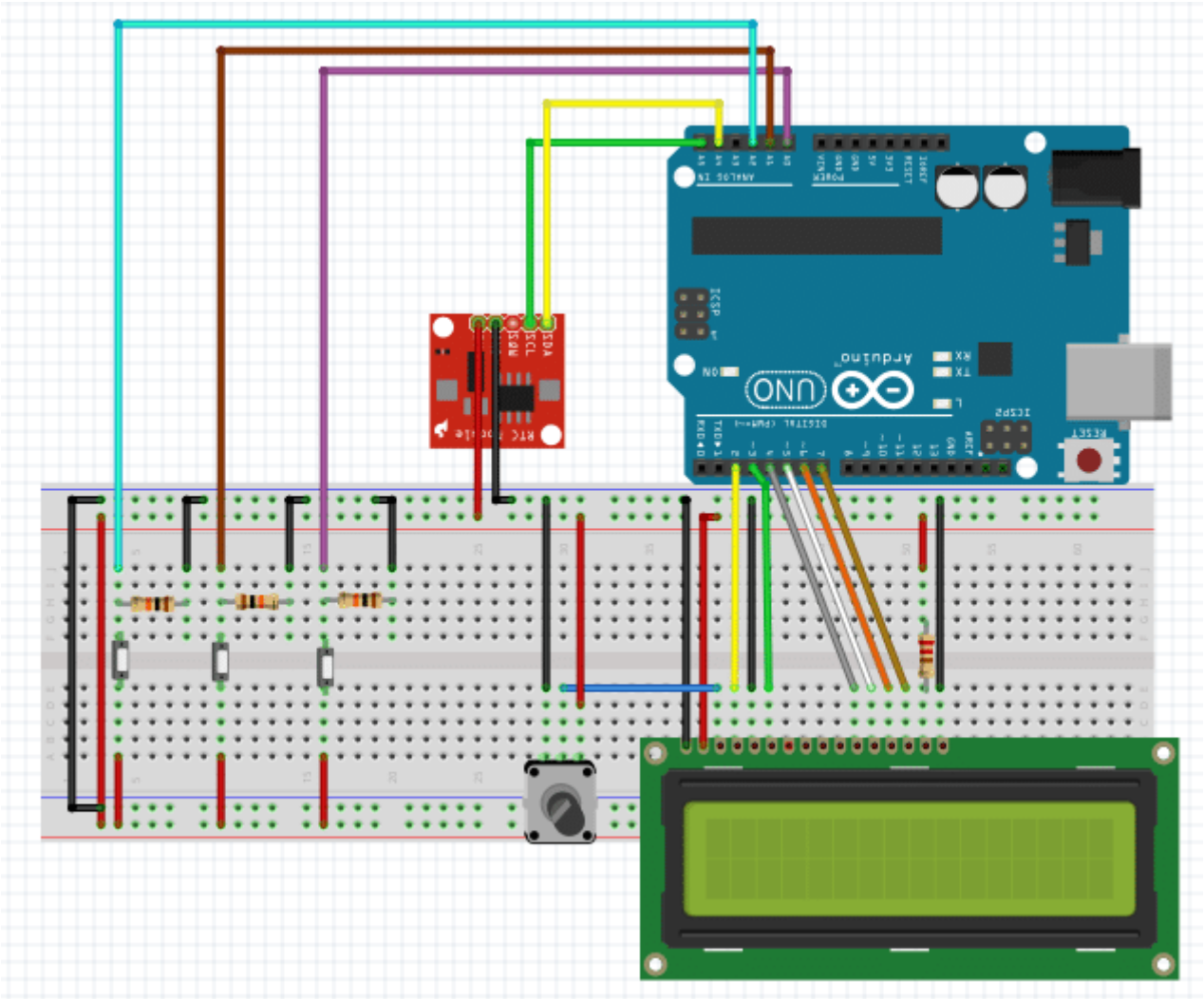
هو عبارة عن جهاز بسيط يتم توصيلة مع الاردوينو بهدف حساب التوقيت. بمعنى أنه يمكننا إستخدامه كساعه لمعرفة الوقت.

يستخدم بروتوكول I2C للتواصل مع الاردوينو. وما يميزه أنه يعمل ببطارية خارجية خاصة به مما يتيح له امكانية الحفاظ على حساب التوقيت وحتى عند إنقطاع الكهرباء عن لوحة الاردوينو.
يقوم بحساب التوقيت مثل: الثواني والدقائق والساعات والايام والشهور والسنوات بشكل تلقائي. حيث انه لا يحتاج لإعادة ضبط بسبب إختلاف عدد الايام في بعض الشهور عن الأخرى.



توصيل الدارة

قم بتوصيل الدارة كما هو موضح بالصورة التالية :

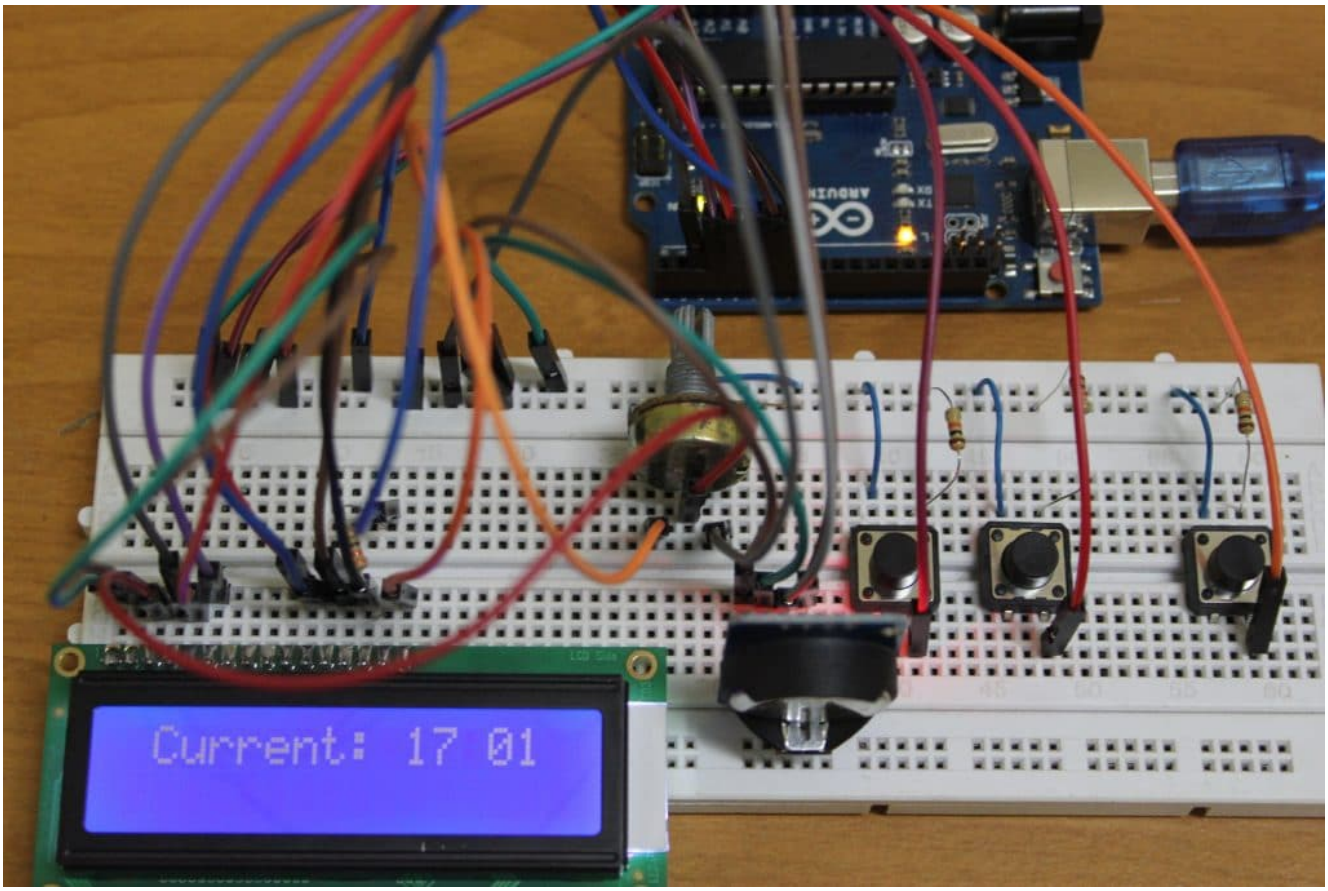


توصيل وحدة DS3231 RTC :

RTC Module	Arduino

5v	Vcc
GND	Gnd
SCL	A5
SDA	A4

تقوم وحدة الRTC بحساب التوقيت و ارساله الى الاردوينو ليتم عرضه على شاشة LCD ، المفاتيح تستخدم لضبط التوقيت في البداية فقط.



البرمجة

قم بتنزيل الكود البرمجي كاملا من خلال الرابط هنا .

شرح البرمجة :

حتى نتمكن من التواصل مع RTC Module بإستخدام الاردوينو، نحتاج الى إستخدام بروتوكول I2C و الذي تتيحه لنا المكتبة Wire.h.

لذلك في البداية نقوم بإدراج مكتبة Wire.h ، التي تحتوى على الدوال اللازمة للتواصل بين الاردوينو و الـ RTC Module .كما نقوم أيضا بإدراج مكتبة LiquidCrystal.h التي تحتوي على الدوال الخاصة بشاشة الـ LCD، و التي سيتم إستخدمها لعرض التوقيت.

```
#include <Wire.h>
#include <LiquidCrystal.h>
```

نقوم بعد ذلك بتسمية منافذ الأردوينو التي تم استخدامها في المشروع والخاصة بالـ Push Buttons.

```
#define ENTER A2
```

```
#define UP A1
#define DOWN A0
```

بعد ذلك قمنا بتحديد عنوان خاص بالRTC Module و هو 0x68 و قمنا بتعيين متغير له :

```
#define DS3231_I2C_ADDRESS 0x68
```

أي وحدة تستخدم برتوكول I2C تمتلك عنوان معين ليتمكن الاردوينو من التواصل معه. لان هذا البرتوكول يستطيع التواصل مع عدد كبير من الاجهزة على نفس الـProtocol Bus . لذلك كل واحد منها يمتلك عنوان معين ليستطيع الاردوينو التواصل معه بدون التأثير على باقي الاجهزة الموصلة على الـ Bus.

يتم الحصول على العنوان الخاص بكل موديول من الـ Datasheet الخاصه به :

START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS3231 can operate in the following two modes:

Slave receiver mode (DS3231 write mode): Serial data and clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/$\overline{W}$), which is 0 for a write. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. After the DS3231 acknowledges the

Slave transmitter mode (DS3231 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit indicates that the transfer direction is reversed. Serial data is transmitted on SDA by the DS3231 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit. The slave address byte is the first byte received after the master generates a START condition. The slave address byte contains the 7-bit DS3231 address, which is 1101000, followed by the direction bit (R/$\overline{W}$), which is 1 for a read. After receiving and decoding the slave address byte, the DS3231 outputs an acknowledge on SDA. The DS3231 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode, the first address that is read is the last one stored in the regis-

ثم نقوم بإنشاء المتغير الخاص بشاشة الـ LCD، وتحديد الـ Pins الموصوله بينها وبين الأردوينو :

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
```

ونقوم بالإعلان عن المتغيرات المستخدمه في تخزين حالة الـ Push Buttons، والتي ستيم استخدامها في البداية لضبط التوقيت فقط. كما نقوم أيضا بالإعلان عن متغيرات لحفظ التوقيت الذي سنقوم بضبطه مثل متغير SetM و SetH .

```
//variables for button states
int enterState = 0, enterStateLast = 0, upState = 0, upStateLast = 0, downState = 0,
downStateLast = 0;

//variables for ui
boolean blinkOn = true; //visibility of ':' between hour and minutes
boolean setVisible = false; //visibility of the set time ui

//variables for new time
int setM = 0; //users new minute value
int setH = 0; //users new hour value
```

هذا الموديول يتعامل مع الارقام على النظامBinary Coded Decimal . أي اننا سوف نحتاج لضبط التوقيت في البداية إلى تحويل الأعداد من الصورة Decimal إلى الصورة BCD . لذلك نستخدم الدالة ()decToBcd.

```
//convert normal decimal numbers to binary coded decimals
byte decToBcd(byte val)
{
  return ( (val / 10 * 16) + (val % 10) );
}
```

وعند قراءة قيمة الوقت من الموديول نستخدم الدالة bcdToDec() ، لتحويل التوقيت المرسل من الموديول من الصورة BCD الى الصورة Decimal ، لنتمكن من عرضها على الشاشة LCD.

```
//convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val / 16 * 10) + (val % 16) );
}
```

الدالة setup() نقوم بضبط الإعدادات اللازمة للمشروع، مثل تشغيل بروتوكول I2C و شاشةالـ LCD و قمنا بضبط المفاتيح المستخدمة في ضبط التوقيت كمدخل.

```
//code that runs once at setup
void setup() {
  //start wire and lcd
  Wire.begin();
  lcd.begin(16,2); //(col, rows)

  //intialize buttons as inputs
  pinMode(ENTER, INPUT);
  pinMode(UP, INPUT);
  pinMode(DOWN, INPUT);
}
```

الدالة loop() نقوم بقراءة الـPush Buttons لضبط التوقيت في البداية ثم نقوم بالتواصل مع الـRTC Module و نعرض التوقيت على الشاشة LCD .

```
void loop() {
  checkButtons();
  printTime();
}
```

الدالة setRTCTime() تقوم بإنشاء اتصال بين الأردوينو و الـ RTC Module بإستخدام مكتبة Wire. وظيفة هذه الداله هو ضبط التوقيت الذي سنقوم بإدخاله بإستخدام الـ Push Buttons. في البداية تقوم بإرسال عنوان الجهاز الذي تريد الإتصال معه وفي حالتنا هنا هو RTC Module و الذي قمنا بتحديده في بداية الكود.

```
//set the time and date to the RTC
void setRTCTime(byte second, byte minute, byte hour, byte dayOfWeek, byte
                dayOfMonth, byte month, byte year)
{
  // sets time and date data to DS3231
```

```
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  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)); // set day of week (1=Sunday, 7=Saturday)
  Wire.write(decToBcd(dayOfMonth)); // set date (1 to 31)
  Wire.write(decToBcd(month)); // set month
  Wire.write(decToBcd(year)); // set year (0 to 99)
  Wire.endTransmission();
}
```

بعد ذلك تقوم بإرسال التوقيت الذي تم ضبطه مع ملاحظة أننا عندما نرسل البيانات الى الموديول نقوم بتحويلها الى الصورة BCD .

```
 Wire.write(decToBcd(minute));
```

بعد إرسال كل البيانات الخاصة بالتوقيت وهي الثواني والدقائق والساعات والأيام والشهور والسنة، نقوم بإنهاء الإتصال لكي يقوم الموديول بمعالجة هذه البيانات في ضبط التوقيت الخاص به.

الدالة readRTCTime() تقوم بعمل إتصال مع الموديول و قراءة معلومات التوقيت منه. وسيتم استخدامها في الحصول على التوقيت وعرضه.
وهي تعمل بنفس اسلوب الداله السابقة مع اختلاف اننا نقوم بقراءة البيانات من الموديول بدلا من إرسالها له، مع ملاحظة اننا في هذه الحاله نقوم بتحويلها من الصورة BCD الى الصورة Decimal .

```
//read the time and date from the RTC
void readRTCTime(byte *second, byte *minute, byte *hour, byte *dayOfWeek,
                 byte *dayOfMonth, byte *month, byte *year)
{
  Wire.beginTransmission(DS3231_I2C_ADDRESS);
  Wire.write(0); // set DS3231 register pointer to 00h
  Wire.endTransmission();
  Wire.requestFrom(DS3231_I2C_ADDRESS, 7);
  // 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());
}
```

الدالة printTime() تقوم بقراءة قيمة التوقيت من الموديول بإستخدام الدالة readRTCTime() ثم تقوم بعرضها على الشاشة LCD مع اضافة كود مهمته عرض التوقيت على الصورة HH : MM .

```
//reads the RTC time and prints it to the top of the LCD
void printTime()
{
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  //retrieve time
```

```
  readRTCTime(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
  //print to lcd top
  lcd.setCursor(0,0);
  lcd.print(" Current: ");
  if (hour<10)
  {
    lcd.print("0");
  }
  lcd.print(hour, DEC);
  if (blinkOn == true)
  {
    lcd.print(" ");
    blinkOn = false;
  }
  else if (blinkOn == false)
  {
    lcd.print(":");
    blinkOn = true;
  }
  if (minute<10)
  {
    lcd.print("0");
  }
  lcd.print(minute, DEC);
  delay(100);
}
```

الجزء السابق مختص بقراءة قيمة التوقيت الذي يقوم الRTC Module بحسابه بدون تدخل من المستخدم . الجزء التالي من الكود مختص بقراءة حالة المفاتيح لضبط التوقيت و ارسالة الى الRTC Module.

الدالة checkButtons() تقوم بقراءة حالة المفاتيح و التي تعمل كالتالي :

المفتاح المسمى UP يقوم بزيادة عداد الدقائق والساعات.
المفتاح المسمى DOWN يقوم بإنقاص عداد الدقائق والساعات.
المفتاح المسمى ENTER يقوم بتطبيق التوقيت الذي تم ضبطه وإرساله الى الـRTC Module.

```
//checks if buttons are pressed and responds accordingly
void checkButtons()
{
  //check enter
  enterState = digitalRead(ENTER);
  if (enterState != enterStateLast)
  {
    if (enterState == HIGH)
    {
      if (setVisible == true)
      {
        byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
        readRTCTime(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month,
&year);
        setRTCTime(0, setM, setH, dayOfWeek, dayOfMonth, month, year);
      }
```

```
      else if (setVisible == false)
      {
        showSet();
      }
    }
  }
  enterStateLast = enterState;
  //check up
  upState = digitalRead(UP);
  if (upState != upStateLast)
  {
    if (upState == HIGH)
    {
      if (setVisible == true)
      {
        addMin();
        printSetTime();
      }
      else if (setVisible == false)
      {
        showSet();
      }
    }
  }
  upStateLast = upState;

  //check down
  downState = digitalRead(DOWN);
  if (downState != downStateLast)
  {
    if (downState == HIGH)
    {
      if (setVisible == true)
      {
        subMin();
        printSetTime();
      }
      else if (setVisible == false)
      {
        showSet();
      }
    }
  }
  downStateLast = downState;
}
```

الدالة showSet() تقوم بعرض التوقيت الذي نقوم بضبطه في أسفل الشاشة قبل إرساله الى الموديول لكي لا نقوم بإرسال توقيت خاطئ بشكل غير مقصود.

```
//displays the new time interface in the bottom of the LCD
void showSet ()
{
  //update new time variables to current RTC values
```

```
  byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;
  readRTCTime(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);
    setH = hour, DEC;
    setM = minute, DEC;

  //pints to the LCD
  lcd.setCursor(0,1);
  lcd.print("   New: ");
  printSetTime();
  setVisible = true;
}
```

الدالة printSetTime() تستخدمها الدالة showSet() لعرض التوقيت الذي تم ضبطه في اسفل الشاشة LCD .

```
//prints the new time values on the bottom of the LCD
void printSetTime()
{
 lcd.setCursor(8,1);
 if (setH<10)
  {
    lcd.print("0");
  }
  lcd.print(setH);
  lcd.print(":");
  if (setM<10)
  {
    lcd.print("0");
  }
  lcd.print(setM);
}
```

الدوال التالية مستخدمة في الجزء السابق من الكود وهي تقوم بضبط المتغيرات التي تستخدم في تخزين قيم الدقائق والساعه التي يقوم المستخدم بضبطها بواسطة الPush Buttons .

الدالة addMin() تقوم بزياد عداد الدقائق عند الضغط على المفتاح UP و تقوم أيضا بعمل اختبار لعداد الدقائق فإذا وصل العدد الى 59 تقوم بزيادة عداد الساعات .

```
//adds a minute to new time
void addMin()
{
  if (setM < 59)
  {
    setM++;
  }
  else if (setM == 59)
  {
    setM = 0;
    addHr();
  }
}
```

الدالة subMin() تعمل عكس عمل الدالة السابقة وهي تقوم بإنقاص عداد الدقائق عند الضغط على المفتاح DOWN وتقوم أيضا باختبار عداد الدقائق فإذا وصل الى 0 تقوم بإنقاص عداد الساعات.

```
//subtracts a minute from new time
void subMin()
{
  if (setM > 0)
  {
    setM--;
  }
  else if (setM == 0)
  {
    setM = 59;
    subHr();
  }
}
```

الدالة addHr() نقوم بإستدعائها الدالة addMin() عندما يصل عداد الدقائق الى 59 لتقوم بزيادة عداد الساعات بمقدار واحد.

```
//adds an hour to new time
void addHr ()
{
  if (setH < 23)
  {
    setH++;
  }
  else if (setH == 23)
  {
    setH = 0;
  }
}
```

الدالة subHr() تقوم باستدعائها الدالة subMin() عندما يصل عداد الدقائق الى 0 لتقوم بإنقاص عداد الساعات بمقدار واحد.

```
//subtracts an hour from new time
void subHr ()
{
  if (setH > 0)
  {
    setH--;
  }
  else if (setH == 0)
  {
    setH = 23;
  }
}
```