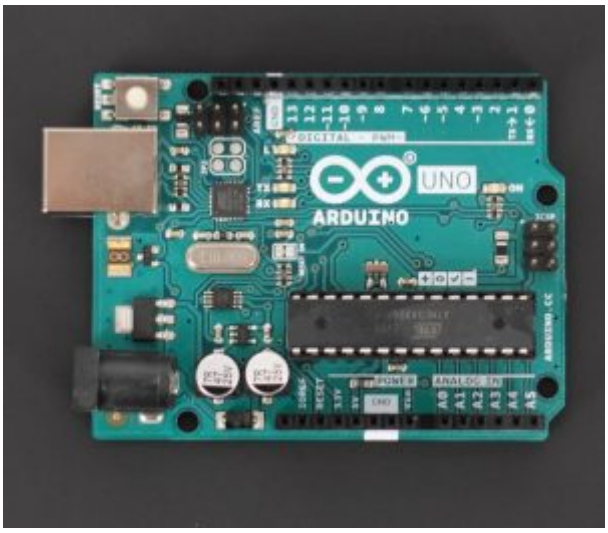
ساعة الكلمات باللغة العربية

في هذا الدرس سنقدم مشروع ساعة تعرض الوقت بالكلمات العربية، بحيث يتم قص كلمات وصف الساعة على لوحة أكليريك، و تضيء الكلمات حسب الوقت الحالي، الساعة تختلف بدقة 5 دقائق ، سيتم برمجة المشروع باستخدام الأردوينو، وتحديد الوقت عن طريق وحدة ساعة الوقت الفعلي.

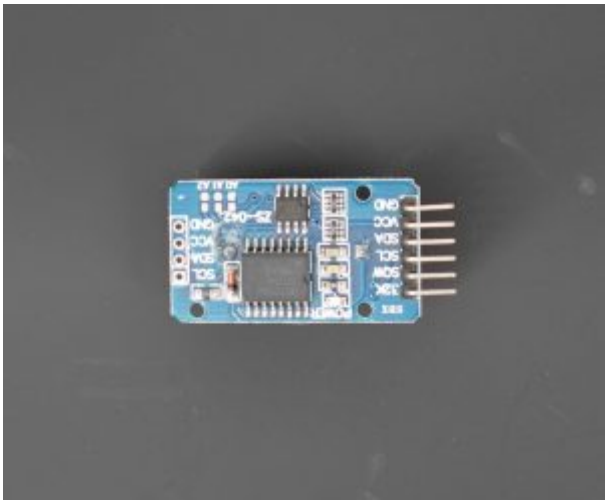
المواد و الأدوات



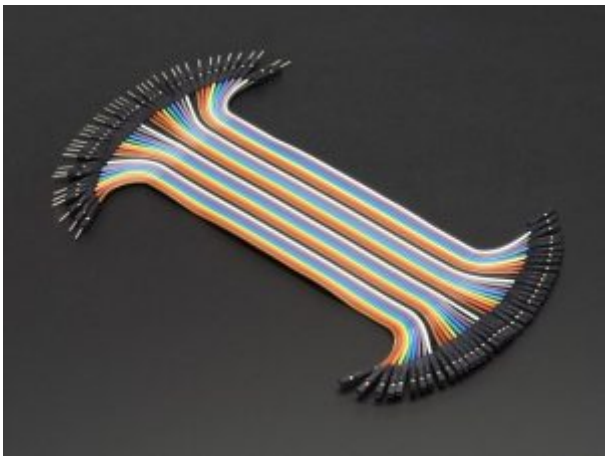
X1 أردوينو

X1 سلك الأردوينو

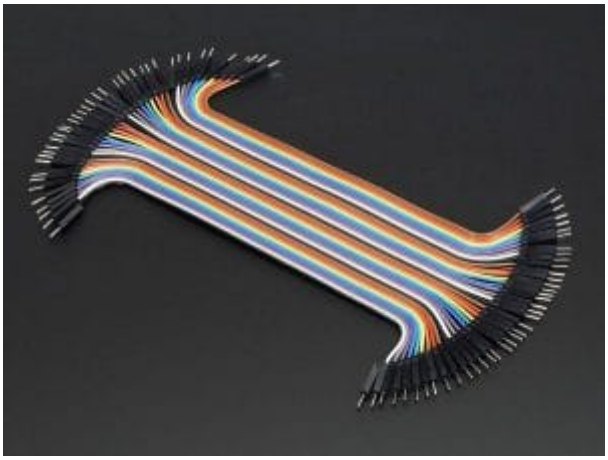
X1 محول طاقة (5فولت ـ2 أمبير)



X1 وحدة الوقت الفعلي



مجموعة أسلاك توصيل (أنثى/ذكر)



مجموعة أسلاك توصيل (ذكر / ذكر)

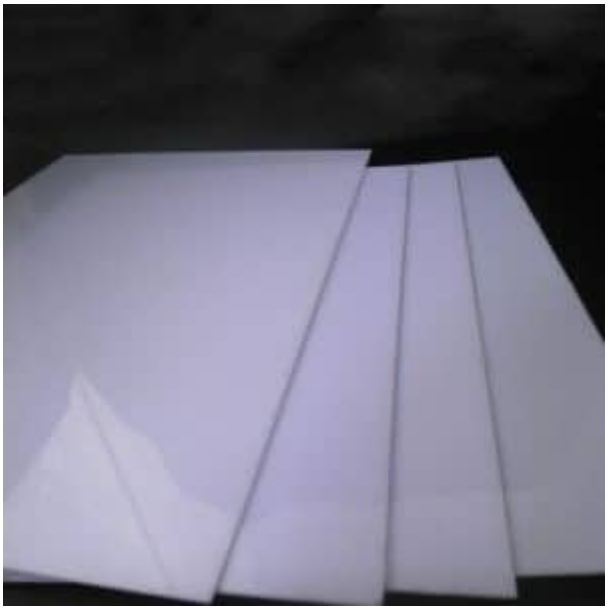
X1 شريط ثنائي القطب يحتوي على (52 وحدة)

X1 سلك لحام

X1 أداة لحام

X1 بكرة سلك

X1 لوح أكليريك أِسود (50*40*3)

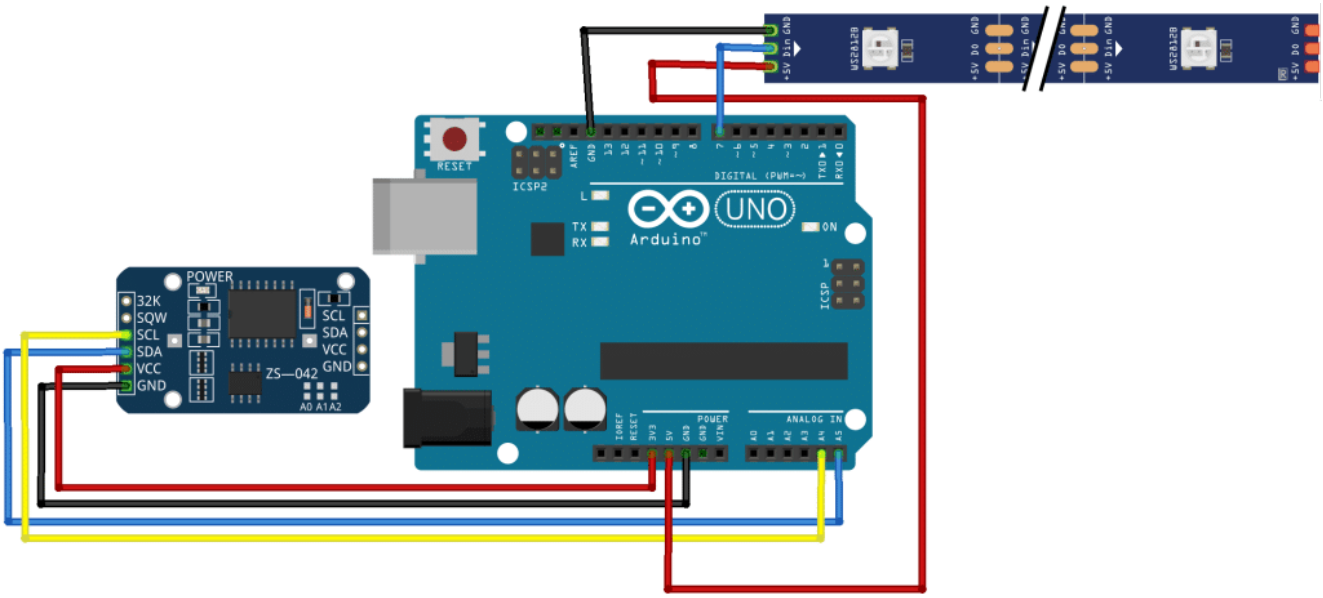


X2 لوح اكليريك أبيض نافذ للضوء(50*40*3)

لحام أكليريك

قاطع الليزر

توصيل الدائرة



التصميم

يمكنك تحميل التصميم من خلال الرابط

أولا: تصميم الواجهة التي تحتوي على الكلمات باللغة العربية

ـ الجزء الأول من الساعة يحتوي على الوقت بالساعات من الواحدة حتى الثانية عشرة وبعدها الوقت بالدقائق و يفصلهما (و/ إلا) ليتم قراءة الوقت بشكل صحيح

– يتم عرض التغير في الوقت كل 5 خمس دقائق

10:20

الساعة الآن

الحادية عشرة الواحدة الثالثة

العاشرة الثانية الخامسة الثامنة

الثانية عشرة التاسعة السابعة

السادسة الرابعة و النصف

إلا الثلث خمس الربع وخمس

إلا خمس عشر النصف دقائق

10:23

الساعة الآن

الحادية عشرة الواحدة الثالثة

العاشرة الثانية الخامسة الثامنة

الثانية عشرة التاسعة السابعة

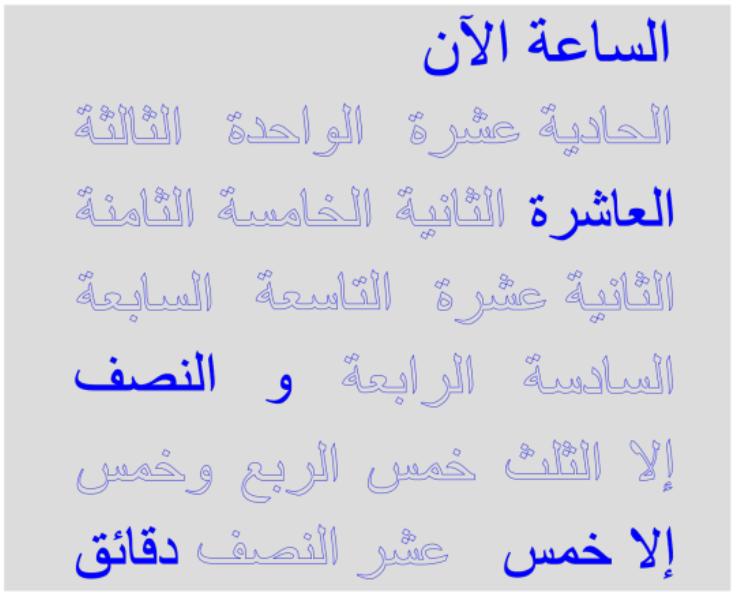
السادسة الرابعة و النصف

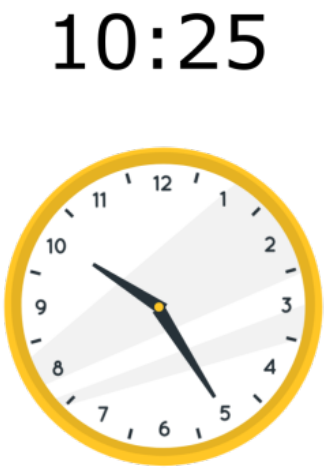
إلا الثلث خمس الربع وخمس

إلا خمس عشر النصف دقائق

– عندما تكون الدقائق 25 ستكون الساعة (و النصف وإلا خمس دقائق)

ـ عندما تكون الدقائق 35 ستكون الساعة (و النصف وخمس دقائق)





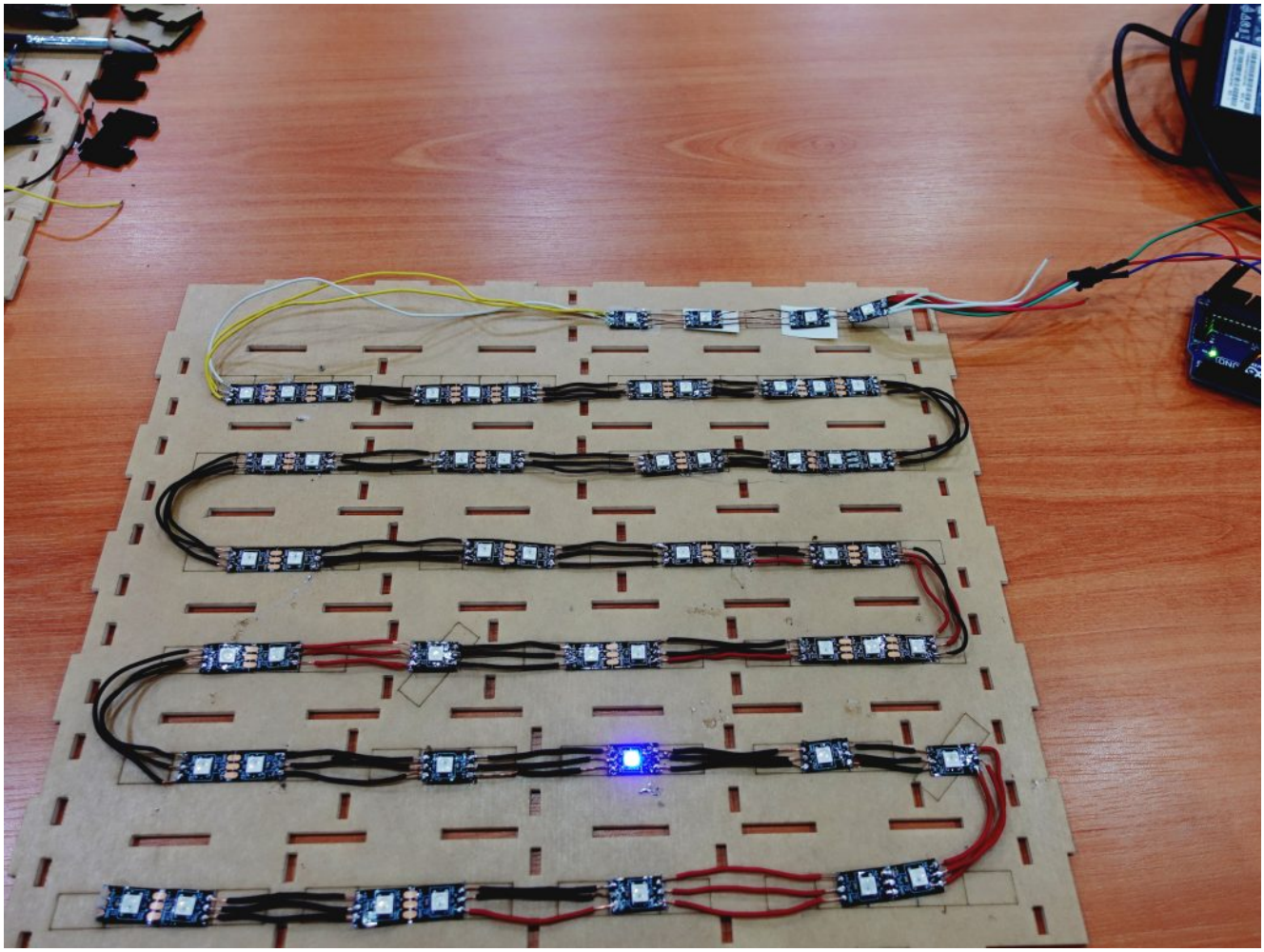
ثانيا: تصميم شبكة حتى تفصل بين الكلمات المتجاورة

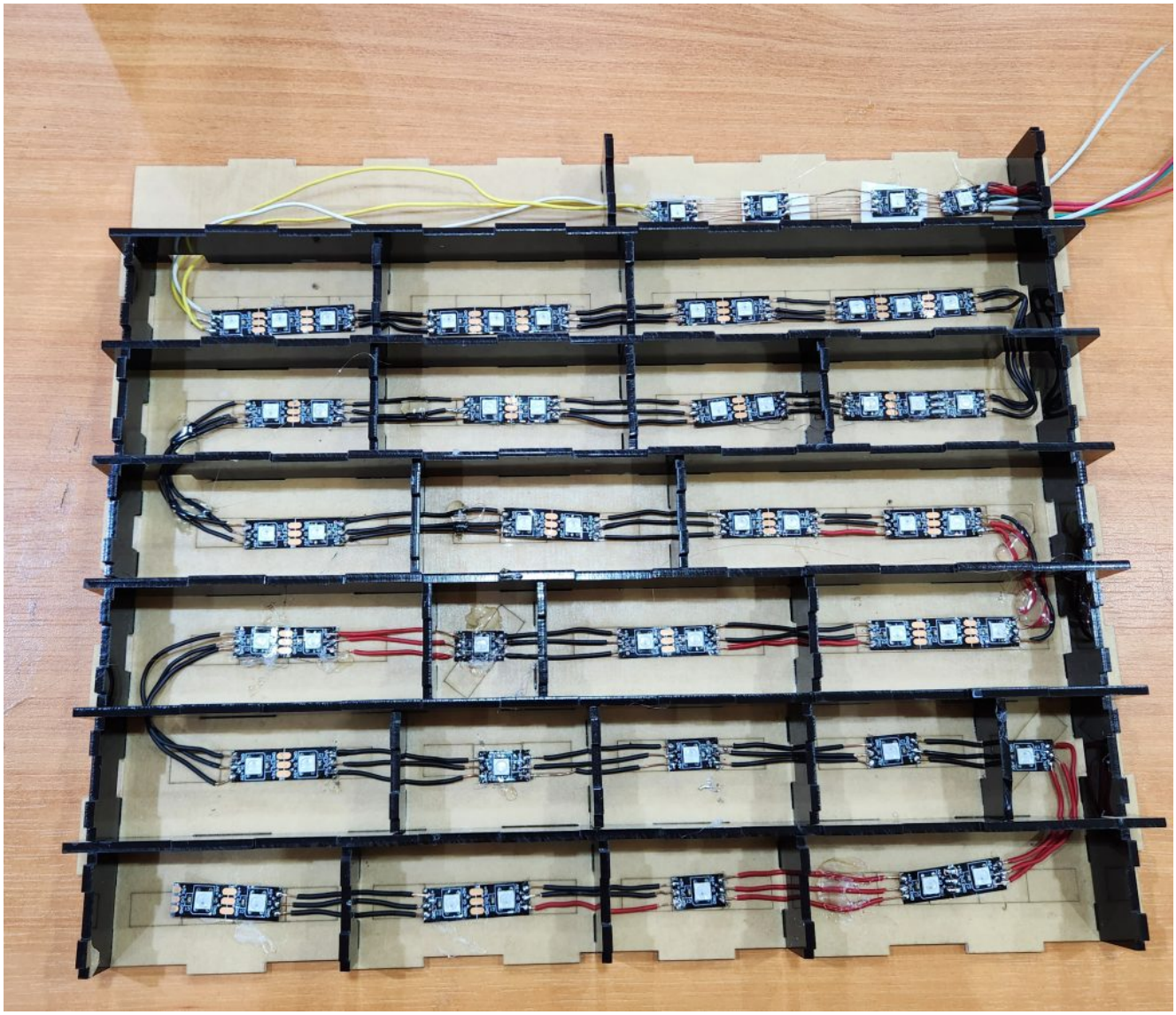
ثالثا: يتم تصميم الصندوق الذي يجمع هذه المكونات

التصنيع

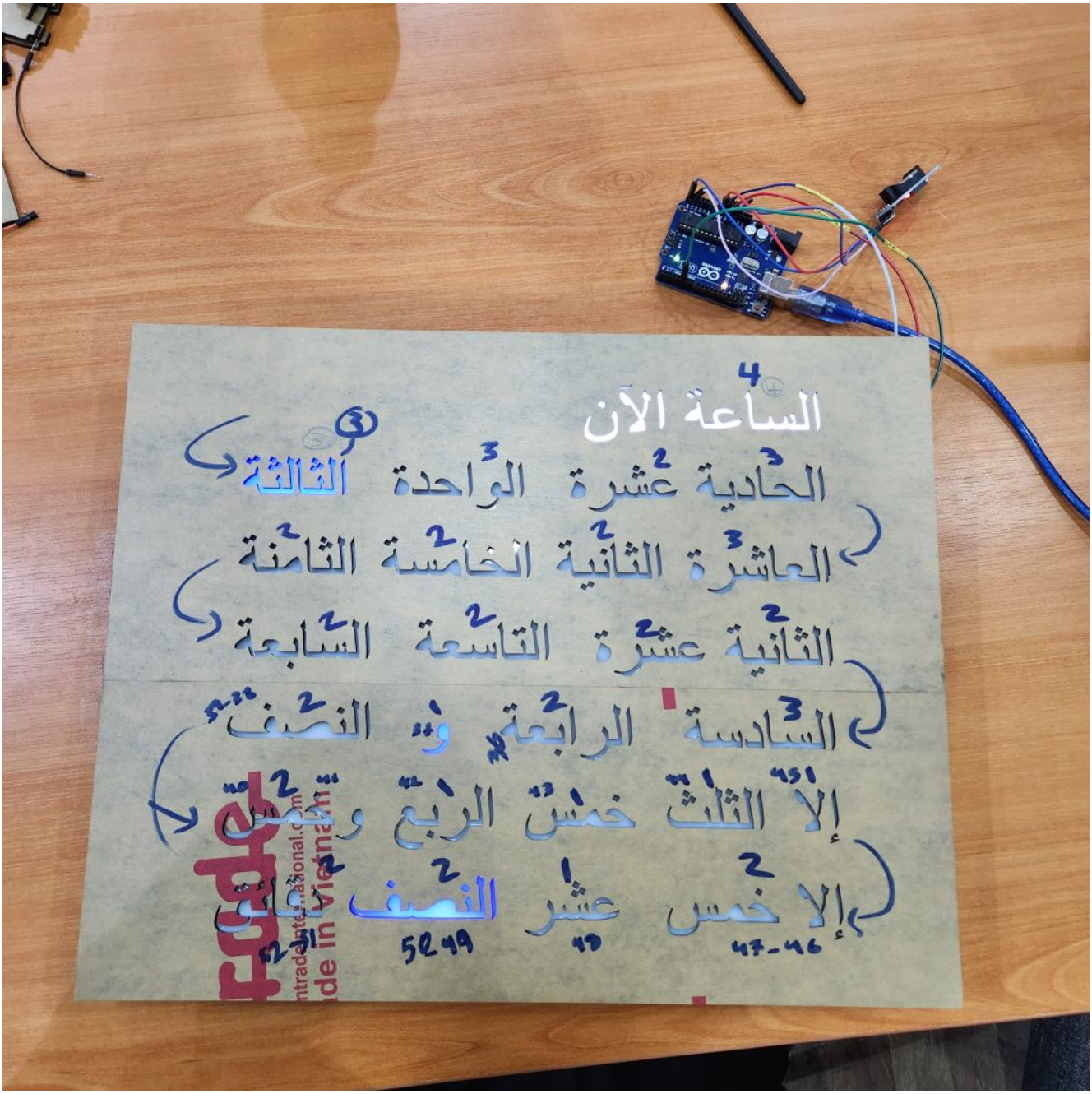
قبل البدء بهذه الخطوة يجب أن التأكد من برمجة و ضبط وحدة الساعة بالانتقال إلى جزء البرمجة في هذا الدرس أو من خلال رابط درس استخدام DS3231 RTC Module مع الاردوينو ، وو كذلك التحقق من شريط الثنائي المشع للضوء

ـ توزيع الثنائي المشع للضوء كما هو موضح بالصورة





ـ تثبيت لوحة الأكليريك التي تسمح بنفاذ الضوء ثم لوحة الكلمات على السطح



البرمجة

أولا: التحقق من ضبط وحدة ساعة الوقت الفعلي و طباعة الوقت على شاشة الإتصال التسلسلي من خلال رفع الشفرة البرمجية التالية

```
#include <RTClib.h>
#include <Wire.h>
RTC_DS3231 rtc;
char t[32];
int dtminutes;
int dthours;
int minutes;
int hours;
void setup()
```

```
{
  Serial.begin(9600);
  Wire.begin();
  rtc.begin();
  rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));
  //rtc.adjust(DateTime(2019, 1, 21, 5, 0, 0));
}
void loop()
{ DateTime now = rtc.now();
  sprintf(t, "%02d:%02d:%02d %02d/%02d/%02d", now.hour(), now.minute(),
now.second(), now.day(), now.month(), now.year());
  Serial.print(F("Date/Time: "));
  Serial.println(t);
  delay(1000);
}
```

ثانيا : التحقق من التحكم بشريط الثنائي المشع للضوء من خلال رفع الشفرة البرمجية التالية

```
#include <FastLED.h>

#define LED_PIN     5
#define NUM_LEDS    64
#define BRIGHTNESS  140
#define LED_TYPE    WS2811
#define COLOR_ORDER GRB
CRGB leds[NUM_LEDS];

CRGB Colors
[13]{CRGB::Gray,CRGB::Orange,CRGB::Maroon,CRGB::Red,CRGB::Cyan,CRGB::Lime,CRGB::Gree
n,CRGB::Purple,
                     CRGB::Yellow,CRGB::White,CRGB::Blue};

void setup() {
    delay( 3000 ); // power-up safety delay
    FastLED.addLeds<LED_TYPE, LED_PIN, COLOR_ORDER>(leds, NUM_LEDS).setCorrection(
TypicalLEDStrip );
    FastLED.setBrightness(  BRIGHTNESS );
}

void loop()
{
for (int j=0;j<13;j++)
  for (int i=0;i<65;i++){
    leds[i]=  Colors[j];
    FastLED.show();
    delay(50);
    leds[i]=CHSV( 0 , 0, 0);
    FastLED.show();
  }
}
```

ثالثا: بعد التحقق من ضبط الساعة و عمل شريط الثنائي المشع للضوء، نرفع الشفرة البرمجية من خلال الأمر التالي

```
#include <FastLED.h>
#include <RTClib.h>
#include <Wire.h>
#define LED_PIN     5
#define NUM_LEDS    65
#define BRIGHTNESS  200
#define LED_TYPE    WS2811
#define COLOR_ORDER GRB
CRGB leds[NUM_LEDS];
RTC_DS3231 rtc;
int minutes;
int hours;
CRGB
Colors[11]{CRGB::Chartreuse,CRGB::Orange,CRGB::BlueViolet,CRGB::Red,CRGB::Cyan,CRGB:
:Pink,
CRGB::Green,CRGB::Purple,CRGB::Yellow,CRGB::White,CRGB::Blue};
void setup()
{
  delay( 3000 );
  FastLED.clear() ;
  Serial.begin(9600);
  Wire.begin();
  rtc.begin();
  rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));
  FastLED.addLeds<WS2812, LED_PIN, RGB>(leds, NUM_LEDS);
    FastLED.setBrightness(  BRIGHTNESS );
  //rtc.adjust(DateTime(2019, 1, 21, 5, 0, 0));
}
void huors()
{
if (hours==1){
     //28-29-30-31
      leds[28] = CRGB::Black;
      leds[29] = CRGB::Black;
      leds[30] = CRGB::Black;
      leds[31] = CRGB::Black;
      //7-8-9
      leds[7] = CRGB::Blue;
      leds[8] = CRGB::Blue;
      leds[9] = CRGB::Blue;
      FastLED.show();
      }
    if (hours==2){
      //7-8-9
      leds[7] = CRGB::Black;
      leds[8] = CRGB::Black;
      leds[9] = CRGB::Black;
      //18-19
      leds[18] = CRGB::Blue;
      leds[19] = CRGB::Blue;
     FastLED.show();
    }
```

```
    if (hours==3){
      //18-19
      leds[18] = CRGB::Black;
      leds[19] = CRGB::Black;
      //4-5-6
      leds[4] = CRGB::Blue;
      leds[5] = CRGB::Blue;
      leds[6] = CRGB::Blue;
      FastLED.show();
    }

    if (hours==4){
       //4-5-6
      leds[4] = CRGB::Black;
      leds[5] = CRGB::Black;
      leds[6] = CRGB::Black;
      //35-36
      leds[35] = CRGB::Blue;
      leds[36] = CRGB::Blue;
      FastLED.show();
    }
      if (hours==5){//20-21
      //35-36
      leds[35] = CRGB::Black;
      leds[36] = CRGB::Black;
      //20-21
      leds[20] = CRGB::Blue;
      leds[21] = CRGB::Blue;
      FastLED.show();
      }
    if (hours==6){
      //20-21
      leds[20] = CRGB::Black;
      leds[21] = CRGB::Black;
      //32-33-34
      leds[32] = CRGB::Blue;
      leds[33] = CRGB::Blue;
      leds[34] = CRGB::Blue;
     FastLED.show();
    }
    if (hours==7){
      //32-33-34
      leds[32] = CRGB::Black;
      leds[33] = CRGB::Black;
      leds[34] = CRGB::Black;
      //24-25
      leds[24] = CRGB::Blue;
      leds[25] = CRGB::Blue;
      FastLED.show();
    }
    if (hours==8){
     //24-25
      leds[24] = CRGB::Black;
```

```
      leds[25] = CRGB::Black;
      //22-23
      leds[22] = CRGB::Blue;
      leds[23] = CRGB::Blue;
      FastLED.show();
    }
     if (hours==9){
      //22-23
      leds[22] = CRGB::Black;
      leds[23] = CRGB::Black;
      //26-27
      leds[26] = CRGB::Blue;
      leds[27] = CRGB::Blue;
      FastLED.show();
      }
    if (hours==10){
      leds[26] = CRGB::Black;
      leds[27] = CRGB::Black;
      //15-16-17
      leds[15] = CRGB::Blue;
      leds[16] = CRGB::Blue;
      leds[17] = CRGB::Blue;
     FastLED.show();
    }
    if (hours==11){
      //15-16-17
      leds[15] = CRGB::Black;
      leds[16] = CRGB::Black;
      leds[17] = CRGB::Black;
      //10-11-12-13-14
      leds[10] = CRGB::Blue;
      leds[11] = CRGB::Blue;
      leds[12] = CRGB::Blue;
      leds[13] = CRGB::Blue;
      leds[14] = CRGB::Blue;
      FastLED.show();
    }
    if (hours==12){
            //10-11-12-13-14
      leds[10] = CRGB::Black;
      leds[11] = CRGB::Black;
      leds[12] = CRGB::Black;
      leds[13] = CRGB::Black;
      leds[14] = CRGB::Black;
      //28-29-30-31
      leds[28] = CRGB::Blue;
      leds[29] = CRGB::Blue;
      leds[30] = CRGB::Blue;
      leds[31] = CRGB::Blue;
     FastLED.show();
    } }
    void Topast(){
      if (minutes){
```

```
      leds[26] = CRGB::Black;
      leds[27] = CRGB::Black;
      //15-16-17
      leds[15] = CRGB::Blue;
      leds[16] = CRGB::Blue;
      leds[17] = CRGB::Blue;
     FastLED.show();
    }
}

void mints(){
if (minutes==0){
      leds[43] = CRGB::Black;
      leds[51] = CRGB::Black;
      leds[52] = CRGB::Black;
}

if ((minutes==5)  or (minutes==55)){
      leds[43] = CRGB::Blue;
      leds[51] = CRGB::Blue;
      leds[52] = CRGB::Blue;
      leds[48] = CRGB::Black;
}
if ((minutes==10)  or (minutes==50)){
  leds[43] = CRGB::Black;
  leds[42] = CRGB::Black;
  leds[48] = CRGB::Blue;
  leds[51] = CRGB::Blue;
  leds[52] = CRGB::Blue;
}
if ((minutes==15)  or (minutes==45)){
  leds[42] = CRGB::Blue;
  leds[48] = CRGB::Black;
  leds[44] = CRGB::Black;
  leds[51] = CRGB::Black;
  leds[52] = CRGB::Black;
}
if ((minutes==20)  or (minutes==40)){
    leds[42] = CRGB::Black;
    leds[44] = CRGB::Blue;
    leds[51] = CRGB::Black;
    leds[52] = CRGB::Black;
    leds[38] = CRGB::Black;
    leds[39] = CRGB::Black;
     leds[40] = CRGB::Black;
    leds[41] = CRGB::Black;
}
if (minutes==30) {

    leds[46] = CRGB::Black;
    leds[47] = CRGB::Black;
    leds[38] = CRGB::Black;
    leds[39] = CRGB::Black;
```

```
    leds[51] = CRGB::Black;
    leds[52] = CRGB::Black;
    leds[49] = CRGB::Blue;
    leds[50] = CRGB::Blue;
}
if (minutes==25) {
    leds[44] = CRGB::Black;
    leds[38] = CRGB::Blue;
    leds[39] = CRGB::Blue;
    leds[46] = CRGB::Blue;
    leds[47] = CRGB::Blue;
    leds[51] = CRGB::Blue;
    leds[52] = CRGB::Blue;
}

if (minutes==35) {
leds[38] = CRGB::Blue;
    leds[39] = CRGB::Blue;
    leds[49] = CRGB::Black;
    leds[50] = CRGB::Black;
    leds[40] = CRGB::Blue;
    leds[41] = CRGB::Blue;
    leds[51] = CRGB::Blue;
    leds[52] = CRGB::Blue;
}
}
void pastto(){
  if (minutes=35 and minutes>=5) {
  leds[37] = CRGB::Blue;
  leds[45] = CRGB::Black;}
  if (minutes<=55 and minutes>=40) {
  leds[45] = CRGB::Blue;
  leds[37] = CRGB::Black;}
if (minutes=0) {
  leds[45] = CRGB::Black; }
}
void loop()
{
  leds[0] = CRGB::White;
  leds[1] = CRGB::White;
  leds[2] = CRGB::White;
  leds[3] = CRGB::White;
  DateTime now = rtc.now();
   minutes = ("%02d", now.minute() );
    minutes=(minutes/5)*5;
   hours = ("%02d", now.hour() );
 if (hours>12){
  hours=hours-12;
  if (minutes>35 and minutes<=55)
  hours=hours+1;}
  mints();
//Serial.print (hours);
//Serial.print (":");
```

```
//Serial.println (minutes);
delay(300);
huors();
pastto();
}
```

شرح البرمجة

نبدأ بتحميل المكتبات شرح طريقة تحميل المكتبات من خلال الرابط

```
#include <FastLED.h>
#include <RTClib.h>
#include <Wire.h>
```

نعرف المتغيرات قيمة كل متغير ظاهرة حسب اسم المتغير

```
#define LED_PIN     5
#define NUM_LEDS    65
#define BRIGHTNESS  200
#define LED_TYPE    WS2811
#define COLOR_ORDER GRB
CRGB leds[NUM_LEDS];
RTC_DS3231 rtc;
int minutes;
int hours;
CRGB
Colors[11]{CRGB::Chartreuse,CRGB::Orange,CRGB::BlueViolet,CRGB::Red,CRGB::Cyan,CRGB:
:Pink,
CRGB::Green,CRGB::Purple,CRGB::Yellow,CRGB::White,CRGB::Blue};
```

في دالة (setup) نكتب أوامر التهيئة الرئيسية الخاصة بشريط الثنائي المشع للضوء و وحدة الوقت الفعلي

```
void setup()
{
delay( 3000 );
FastLED.clear() ;
Serial.begin(9600);
Wire.begin();
rtc.begin();
rtc.adjust(DateTime(F(__DATE__),F(__TIME__)));
FastLED.addLeds<WS2812, LED_PIN, RGB>(leds, NUM_LEDS);
  FastLED.setBrightness(  BRIGHTNESS );
//rtc.adjust(DateTime(2019, 1, 21, 5, 0, 0));
}
```

سنقوم بعمل ثلاث دوال واحدة تعمل على التحكم بثنائي المشع للضوء الخاص بالساعات
وواحدة للتحكم بثنائي المشع للضوء الخاص بالدقائق
وواحدة للتحكم بثنائي المشع للضوء الخاص بكل من (إلا / و)

```
void huors()
```

```
{
if (hours==1){
   //28-29-30-31
    leds[28] = CRGB::Black;
    leds[29] = CRGB::Black;
    leds[30] = CRGB::Black;
    leds[31] = CRGB::Black;
    //7-8-9
    leds[7] = CRGB::Blue;
    leds[8] = CRGB::Blue;
    leds[9] = CRGB::Blue;
    FastLED.show();
    }
  if (hours==2){
    //7-8-9
    leds[7] = CRGB::Black;
    leds[8] = CRGB::Black;
    leds[9] = CRGB::Black;
    //18-19
    leds[18] = CRGB::Blue;
    leds[19] = CRGB::Blue;
   FastLED.show();
    }

  if (hours==3){
    //18-19
    leds[18] = CRGB::Black;
    leds[19] = CRGB::Black;
    //4-5-6
    leds[4] = CRGB::Blue;
    leds[5] = CRGB::Blue;
    leds[6] = CRGB::Blue;
    FastLED.show();
    }

  if (hours==4){
     //4-5-6
    leds[4] = CRGB::Black;
    leds[5] = CRGB::Black;
    leds[6] = CRGB::Black;
    //35-36
    leds[35] = CRGB::Blue;
    leds[36] = CRGB::Blue;
    FastLED.show();
  }
    if (hours==5){//20-21
    //35-36
    leds[35] = CRGB::Black;
    leds[36] = CRGB::Black;
    //20-21
    leds[20] = CRGB::Blue;
    leds[21] = CRGB::Blue;
    FastLED.show();
```

```
  }
if (hours==6){
  //20-21
  leds[20] = CRGB::Black;
  leds[21] = CRGB::Black;
  //32-33-34
  leds[32] = CRGB::Blue;
  leds[33] = CRGB::Blue;
  leds[34] = CRGB::Blue;
 FastLED.show();
}
if (hours==7){
  //32-33-34
  leds[32] = CRGB::Black;
  leds[33] = CRGB::Black;
  leds[34] = CRGB::Black;
  //24-25
  leds[24] = CRGB::Blue;
  leds[25] = CRGB::Blue;
  FastLED.show();
}
if (hours==8){
//24-25
  leds[24] = CRGB::Black;
  leds[25] = CRGB::Black;
  //22-23
  leds[22] = CRGB::Blue;
  leds[23] = CRGB::Blue;
  FastLED.show();
}
 if (hours==9){
  //22-23
  leds[22] = CRGB::Black;
  leds[23] = CRGB::Black;
  //26-27
  leds[26] = CRGB::Blue;
  leds[27] = CRGB::Blue;
  FastLED.show();
  }
if (hours==10){
  leds[26] = CRGB::Black;
  leds[27] = CRGB::Black;
    //15-16-17
  leds[15] = CRGB::Blue;
  leds[16] = CRGB::Blue;
  leds[17] = CRGB::Blue;
 FastLED.show();
}
if (hours==11){
  //15-16-17
  leds[15] = CRGB::Black;
  leds[16] = CRGB::Black;
  leds[17] = CRGB::Black;
```

```
    //10-11-12-13-14
    leds[10] = CRGB::Blue;
    leds[11] = CRGB::Blue;
    leds[12] = CRGB::Blue;
    leds[13] = CRGB::Blue;
    leds[14] = CRGB::Blue;
    FastLED.show();
  }
  if (hours==12){
         //10-11-12-13-14
    leds[10] = CRGB::Black;
    leds[11] = CRGB::Black;
    leds[12] = CRGB::Black;
    leds[13] = CRGB::Black;
    leds[14] = CRGB::Black;
    //28-29-30-31
    leds[28] = CRGB::Blue;
    leds[29] = CRGB::Blue;
    leds[30] = CRGB::Blue;
    leds[31] = CRGB::Blue;
   FastLED.show();
  } }
  void Topast(){
    if (minutes){
    leds[26] = CRGB::Black;
    leds[27] = CRGB::Black;
    //15-16-17
    leds[15] = CRGB::Blue;
    leds[16] = CRGB::Blue;
    leds[17] = CRGB::Blue;
   FastLED.show();
  }
}

void mints(){
if (minutes==0){
    leds[43] = CRGB::Black;
    leds[51] = CRGB::Black;
    leds[52] = CRGB::Black;
}

if ((minutes==5)  or (minutes==55)){
    leds[43] = CRGB::Blue;
    leds[51] = CRGB::Blue;
    leds[52] = CRGB::Blue;
    leds[48] = CRGB::Black;
}
if ((minutes==10)  or (minutes==50)){
leds[43] = CRGB::Black;
leds[42] = CRGB::Black;
leds[48] = CRGB::Blue;
leds[51] = CRGB::Blue;
leds[52] = CRGB::Blue;
```

```
}
if ((minutes==15)  or (minutes==45)){
leds[42] = CRGB::Blue;
leds[48] = CRGB::Black;
leds[44] = CRGB::Black;
leds[51] = CRGB::Black;
leds[52] = CRGB::Black;
}
if ((minutes==20)  or (minutes==40)){
  leds[42] = CRGB::Black;
  leds[44] = CRGB::Blue;
  leds[51] = CRGB::Black;
  leds[52] = CRGB::Black;
   leds[38] = CRGB::Black;
  leds[39] = CRGB::Black;
   leds[40] = CRGB::Black;
  leds[41] = CRGB::Black;
}
if (minutes==30) {
   leds[46] = CRGB::Black;
  leds[47] = CRGB::Black;
  leds[38] = CRGB::Black;
  leds[39] = CRGB::Black;
  leds[51] = CRGB::Black;
  leds[52] = CRGB::Black;
  leds[49] = CRGB::Blue;
  leds[50] = CRGB::Blue;
}
if (minutes==25) {
  leds[44] = CRGB::Black;
  leds[38] = CRGB::Blue;
  leds[39] = CRGB::Blue;
  leds[46] = CRGB::Blue;
  leds[47] = CRGB::Blue;
  leds[51] = CRGB::Blue;
  leds[52] = CRGB::Blue;
}

if (minutes==35) {
leds[38] = CRGB::Blue;
  leds[39] = CRGB::Blue;
  leds[49] = CRGB::Black;
  leds[50] = CRGB::Black;
  leds[40] = CRGB::Blue;
  leds[41] = CRGB::Blue;
  leds[51] = CRGB::Blue;
  leds[52] = CRGB::Blue;
}
}
void pastto(){
if (minutes=35 and minutes>=5) {
leds[37] = CRGB::Blue;
leds[45] = CRGB::Black;}
```

```
if (minutes<=55 and minutes>=40) {
leds[45] = CRGB::Blue;
leds[37] = CRGB::Black;}
if (minutes=0) {
leds[45] = CRGB::Black; }
}
```

في دالة (loop) سنعرف قيمة الوقت و نجري عمليات رياضية كالتالي

ـ تغير الدقائق لأحد مضاعفات الخمسة

ـ قراءة الساعة بصيغة اثنى عشر ساعة

ـ زيادة قيمة الساعة بواحد عندما تكون الدقائق أكبر من 35

استدعاء الدوال الثلاثة الخاصة بالتحكم بالثنائي المشع للضوء

```
void loop()
{
leds[0] = CRGB::White;
leds[1] = CRGB::White;
leds[2] = CRGB::White;
leds[3] = CRGB::White;
DateTime now = rtc.now();
 minutes = ("%02d", now.minute() );
  minutes=(minutes/5)*5;
 hours = ("%02d", now.hour() );
if (hours>12){
hours=hours-12;
if (minutes>35 and minutes<=55)
hours=hours+1;}
mints();
//Serial.print (hours);
//Serial.print (":");
//Serial.println (minutes);
delay(300);
huors();
pastto();
}
```

الساعة الآن
الحادية عشرة
و
الربع
AM
11 16
47