الباب الثالث

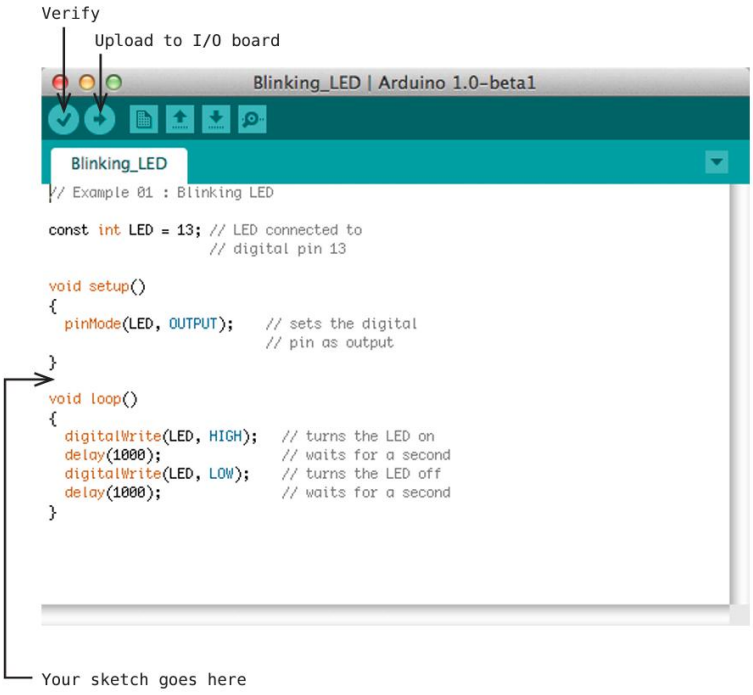
الدائرة العملية وشرحها

1.3 فكرة عن برمجة لوح الأردوينو:

أُستخدم للبرمجة ال(arduino IDE) و هو برنامج يستخدم لغة تشبه إلى حد ما لغة ال(C++)، كذلك يمكن اِستخدام الماتلاب للبرمجة الأردوينو وذلك بعد تحميل دعم الأردوينو على الماتلاب ولكن الأسهل البرمجة ب(arduino IDE)، ولكن يمكن الاستفادة من الماتلاب في عمل السيمولينك وكذلك رسم ال .plot

وصل ال(LED) و اُحتيِجَ لي القول لأردوينو ماذا يفعل وهذا يتم عبر كود و الذي هو عبارة عن مجموعة من الأوامر التي نعطيها إلى المايكروكنترولر لكي نجعله يقوم بما نرغب.

كُتِبَ الكود البرمجي التالي وتمت تسمية السكيتش بإسمBlinking LED



الشكل1.3: كود برمجي لوميض دايود الباعث الضوئي

يجب التأكد من أن البرنامج المكتوب صحيح ولهذا ضُغِط على زر التحقق كما في الصورة أعلاه,إذا كان كل شي صحيح سوف تظهر الرسالة "Done compiling"في أسفل شاشة الarduino IDE هذه الرسالة تعني أن الarduino IDE ترجم السكيتش إلى برنامج قابل للتنفيذ يمكن تشغيله عن طريق بورد الأردينو بعدها يمكن رفع البرنامج إلى لوح الأردينو عن طريق الضغط على زر الرفع هذا سوف يعيد ضبط البورد ويجبره على التوقف عما يفعله والإصغاء إلى الأوامر القادمة من منفذ الUSB.

الarduino IDE سيرسل السكيتش الحالي إلى لوحة الأردوينو الذي سوف تخزن في ذاكرته وفي الأخير يتم تنفيذه.

سوف نشاهد بعض الرسائل تظهر في المنطقة السوداء أسفل الشاشة وفوق تلك المنطقة سنشاهد الرسالة"Done Uploading"لتخبرنا أن العملية تمت بشكل صحيح.

هنالك باعثان ضوئيان معلمان على لوحة هما الRX و الTX يومضان في كل مرة هنالك بايت أرسل أو أستقبل من اللوحة أثناء عملية الإرسال يستمران في الوميض بسرعة.

إذا ظهرت الرسالة error messageبدلا من done compilingفهنالك مشكلة في الاتصال بين الأردوينو و الكمبيوتر . يجب التأكد أنه اُختيرَ المنفذ التسلسلي الصحيح .

حالما يكون الكود البرمجي في لوح الأردوينو سوف يظل هنالك إلى أن نقوم بتحميل سكيتش أخر. السكيتش سوف يبقى حتى لو تمت إعادة التشغيل أو الإيقاف و هذا يشبه البيانات المحفوظة في قرص الكمبيوتر الصلب .

هنالك بعض الأشياء المهمة التي يجب معرفتها قبل البداية في شرح الكود البرمجي وهي:

1) في بداية البرنامج يجب استدعاء المكتبات التي نحتاجها في البرمجة .

2) بعد المكتبات يجب تعريف المتغيرات التي تستخدم في البرنامج (intger, flot, char).

3) أي شيء بداخل الvoid setup()يتم تنفيذه مرة واحدة لهذا نقوم بتحديد الpins هل هي input or output.وكذلك لعمل أشياء أخرى سيتم التعرف عليه في الكودات البرمجية القادمة إن شاء الله.

4) أي شيء بداخل الvoid loop() يتم تنفيذه بصورة مستمرة لهذا نضع بداخله برنامجنا الأساسي.

5)أي شيء يأتي بعد هذه // العلامة في السطر الواحد يتم تجاهله ونستفيد منه في عمل التعليقات.

وكذلك هنالك أشياء أخرى سنتعرف عليها في الكودات البرمجية القادمة.

بعد هذه التوضيحات يمكن شرح الكود البرمجي.

// Blinking LED

التعليقات مفيدة لكتابة ملاحظات بسيطة و التعليق أعلاه يذكرنا بأن البرنامج لوميض دايود باعث ضوئي LED

constint LED = 13; // LED connected to
// digital pin 13

Constint تعني أن LED عدد صحيح لا يمكن تغيره حيث قيمته ضبطت لتساوي 13 حيث أنه كلما ظهرت كلمة LED يضع الأردوينو محلها الرقم 13 وهذا الأمر الغرض منه تحديد أن الدايود الباعث الضوئي موصل ل.pin13

void setup()

هذا السطر يخبر الأردوينو أن قالب الأوامر البرمجية القادم سوف تتم تسميته .setup()

{

مع فتح هذا القوس المجعد قالب الأوامر البرمجية يبدأ.

pinMode(LED, OUTPUT); // sets the digital
// pin as output

هذا الأمر يخبر الأردوينو كيف يؤكد هذا الpin الحالي, الpins الرقمية يمكن استخدامها كدخل أو خرج. في هذه الحالة نحتاج إلى pin خرج و لفعل ذلك نحدد رقمه و حالته داخل الأقواس .

inputو output هما ثوابتبلغة الأردوينو.

}

إغلاق القوس المجعد يدلل على نهاية وظيفة ال.()setup

void loop()
{

()Loopهو المكان الذي نحدد فيه السلوك الرئيسي للشيء الموصل بالأردوينو وسوف يتم تكراره وتكراره إلى أن يتم إغلاق لوح الأردوينو.

digitalWrite(LED, HIGH); // turns the LED on

مثل ما يقول التعليق ()digitalWrite, قادرة على تشغيل أو إيقاف أي pin تم تثبيته على أنه خرج, أول طرف داخل الأقواس يمثل الpin المحدد في هذه الحالة pin13 كما تم تحديده سابقا بينما الطرف الثاني يمثل حالة الpin (HIGHتعني on بينما LOW تعني off).

delay(1000); // waits for a second

بشكل أساسي يقوم بجعل العملية تظل كما هي لفترة تحدد بالمللي الثانية التي تحدده داخل الأقواس وهنا ألف ملي ثانية يساوي ثانية .

digitalWrite(LED, LOW); // turns the LED off

هذا الأمر يوقف الLED الذي تم تشغيله سابقا. ولماذا نستخدم HIGHو LOW ؟

حسنا. إنه عرف قديم في الإلكترونيات الرقمية حيث أن HIGH تعنين الpinON وفي هذه الحالة لأردوينو سيتم ضبطه على 5V و LOW تعني 0V .

delay(1000); // waits for a second

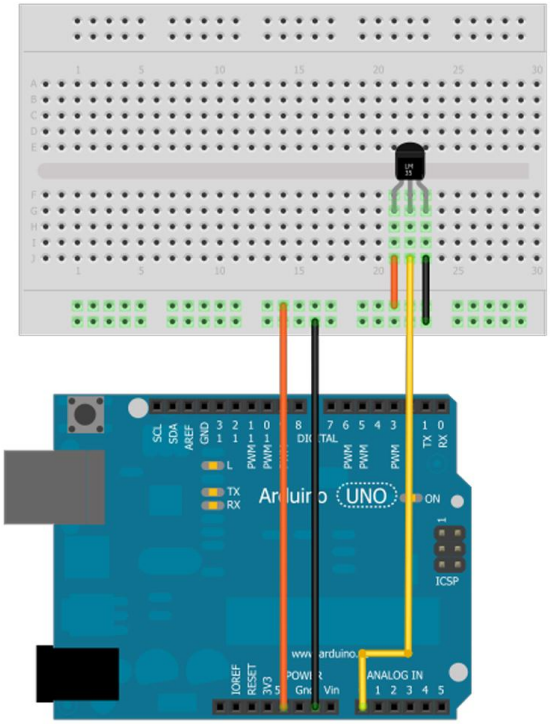
هنا سيتأخر لثانية أخرى و دايود الباعث الضوئي سيكون مغلق .

```
}
```

إغلاق القوس المجعد يدلل على نهاية وظيفة الحلقة المغلقة.

2.3 توصيل الحساس LM35مع لوح الأردوينو :

وُصِل حساس الحرارة (LM35) مع الأردوينو كما يوضح بالشكل أدناه .



الشكل2.3: طريقة توصيل الحساس LM35 مع الاردوينو

يحتوي الحساس على ثلاث أرجل وهي:

1) منفذ الدخل و يتم توصيله بجهد ثابت بين 2.2 و 5.5فولت .

2) منفذ الخرج وهو المنفذ الذي تحصل منه على قراءة الحساس .

3) منفذ أرضي ويتم وصله بأي نقطة أرضي GND .

الشكل3.3: أطراف الحساسLM35

ومن المعروف أن الأردوينو جهاز رقمي بينما الحساس (LM35) يعطي خرج تماثلي (يعرف الدخل أو الخرج التماثلي بأنه أي فرق جهد تبدأ قيمته من صفر و قابل لتغير دائما وليس له حدود قصوى , على عكس الدخل الرقمي والذي إما يكون HIGH (1) أوLOW (0)).

الكود البرمجي:

```
float temp;
float temp1;
void setup()
}
Serial.begin(9600);
pinMode(7, OUTPUT);
{
void loop()
}
temp = analogRead(A0);
temp1 = temp * 0.48828125;
```

```
Serial.print("TEMPRATURE = ");
Serial.print(temp1);
Serial.println();
delay(2000);
{
```

تم شرح الفكرة العامة لكتابة الكود البرمجي في كود Blinking LED و في هذا الكود والكودات البرمجية القادمة سيتم شرح الأوامر الجديدة.

في البداية عُرِف الأمران temp و temp1 باستخدام الأمر float بدلا من استخدام الأمر int و ذلك لأن الحساس الحراري يقيس درجة الحرارة بدقة عالية تصل إلى 0.1 درجة مئوية ومن المتوقع أن تكون درجة حرارة الجهد الكهربي الناتج عنه و درجة الحرارة بالكسور العشرية و ليس الأعداد الصحيحة لذلك أُستخدم الأمر float لجعل هذه المتغيرات تقبل قيمة تحتوي على كسور عشريه.

أُستخدم الأمر Serial.begin(9600) في جعل بورده الأردينو تبدأ التواصل مع الحاسب الآلي عن طريق منفذ الUSB وبهذا يمكن للأردينو إرسال أو استقبال بيانات من و إلى الحاسب الآلي .

تقوم الدالة البرمجية analogRead(pin number) بقرأة فرق الجهد بصورة تماثلية وتستطيع المتحكمه الدقيقة أن تقيس فرق الجهد من 4.8 مللي فولت حتى 5 فولت تقريبا وتقوم بتحويل الإشارة التماثلية إلى قيمة رقمية من صفر إلى 1024 وتسمى هذه العملية بإسم تحويل الإشارة من تماثلية إلى رقمية (Analog to Digital converting).

مثلا إذا كان الجهد الداخل إلى A0يساوي القيم التالية:

4.8 مللي فولت = 1 رقمي

48 مللي فولت = 10 رقمي

480 مللي فولت = 100 رقمي

1 فولت = 208.33 رقمي

2 فولت = 416.66 رقمي

5 فولت = 1024 رقمي

ضُرِب المتغير temp في 0.48828125 وسُجِلت الناتج في المتغير temp1وذلك لتحويل إلى درجة حرارة مئوية ، ولكن قد يتبادر على الأذهان لماذا تم الضرب في 0.48828125 ؟لمعرفة قيمة الفولتية الداخله إلى المدخل التماثلي نستخدم القاعدة أدناه

```
Volt=analogRead(A0)*5/1024
    =analogRead(A0)*0.0048828125
```

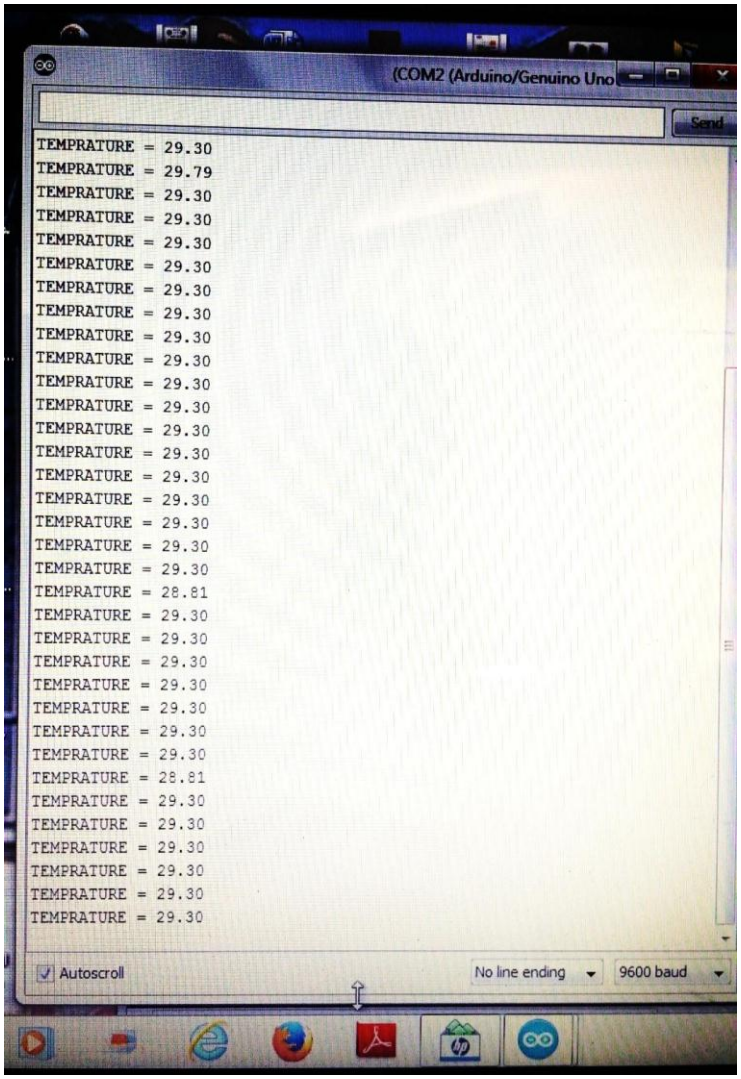
ومن المعروف أن الحساس LM35 يعطي تغير 10 مللي فولت لكل 0.1 درجة مئوية لهذا نقوم بضرب 0.0048828125 في 100 لكي نحول الفولتية إلى درجة حرارة مئوية .

```
Temperature = analogeRead(A0)*0.48828125
```

الأمرSerial.print()اُستخدم لطباعة ما بداخل الأقواس على شاشة الSerial Monitor (سيتم شرحها بعد الانتهاء من شرح الكود البرمجي) ، وعند إضافة ln بعد الprintكما في الأمر Serial.println()بعد كتابة ما باخل الأقواس يبدأ الكتابة في سطر جديد على شاشة الSerial Monitor .

بعد الانتهاء من كتابة الكود البرمجي ورفعه إلى الأردينو يتم الضغط على زر ال Serial Monitor الموجود على شريط الأوامر السريعة (رمز العدسة):

ستظهر صفحة خاصة تستقبل البيانات من لوح الأردينو وتقوم بعرض درجة الحرارة كل ثانيتين (يتم تحديد زمن العرض عن طريق تغير قيمة الdelay) كما سنلاحظ أن الدايود الضوئي الموجود على لوح الأردوينو المكتوب بجانبه Tx بدأ في الوميض بسرعة و ذلك يعني أن الأردوينو يرسل البيانات إلى الحاسب الآلي.



الشكل4.3: شاشة ال Serial Monitor

3.3 توصيل شاشة عرض الكريستال السائل بلوح الأردوينو :

وُصِلت شاشة عرض الكريستال السائل بالمتحكم حيث تستخدم لعرض نتائج القياس على الشاشة ولكن قبل عرض النتائج وُصلت الشاشة بمفردها مع الأردوينو و اختبرت.

وصلت الأطراف (11،12،13،14،6،4) في الشاشة بالمداخل (9،10،11،12،8،7) على الترتيب و وصل الطرفان (3،5) بالأرضي و 2 بالخط الموجب ، وتمت كتابة الكود البرمجي التالي :

```
#include<LiquidCrystal.h>
LiquidCrystallcd(7,8,9,10,11,12);
```

```
void setup()
}
lcd.begin(16, 2);
lcd.print("hello, world!");
{
void loop()
}
lcd.setCursor(0, 1);
lcd.print(millis() / 1000);
{
```

في البداية اُستُدعيت المكتبةLiquid Crystal باستخدام الأمر #include <LiquidCrystal.h>، ومن ثم عُرفت مداخل الأردينو المستخدمة في عملية إرسال البيانات إلى الشاشة بواسطة الأمرLiquidCrystallcd(7,8,9,10,11,12)

الأمر lcd.begin(16,2) يحدد عدد أعمدة و صفوف الشاشة الكريستالية حيث الخانة الأولى داخل القوس 16 تعني عدد أعمدة الشاشة و الخانة الثانية 2 تمثل عدد صفوفها .

الأمرlcd.print()يطبع ما بداخل القوسين .

الأمر lcd.setCursor(0,1)يُموضع مؤشر الشاشة في المكان الذي سوف يعرض فيه النص الذي كتب إلى الشاشة حيث الخانة الأولى داخل القوسين 0 ترمز الأعمدة و الثانية 1 ترمز إلى الصوف و كلهما ترتيبها يبدأ بصفر .

الأمر lcd.print(millis()/1000)سيعرض عدد الثواني منذ أن بدأت لوحة الأردينو في تنفيذ البرنامج الحالي حيث millis()يرجع قيمة لي عدد مللي ثانيةمنذ أن بدأت لوحة الأردينو في تنفيذ البرنامج الحالي(تم تحويلها إلى ثانية بالقسمة على ألف).

4.3 الدائرة النهائية :

في البداية وُصل الباعث الضوئي ذو اللون الأصفر في المخل رقم 1 بينما وصلت (المروحة 5 V) في المخل رقم 2 ، و ثم وُصلت الترانزستورات بالمداخل 3 و 4 و 5 لتشغيل مراوح ال 12 فولت (توصل المداخل بقواعد الترانزستورات التي تعمل كمفاتيح لتشغيل المرحلات) ووصلت البواعث الضوئية ذات اللون الأبيض و الأزرق و البرتقالي على التوازي مع المراوح لكي يَدُلُ على أن المراوح تعمل أم لا و وصلت الشاشة بالمداخل من 7 إلى 12 لي أغراض العرض ، و وصل الطنان بالمدخل رقم 6 و وصل معه الباعث الضوئي الأحمر .