اسم الكتاب: تعليم الاردوينو خطوة بخطوة.

تأليف: إيمان مؤمن عبد العليم.

النوع: كتاب تعليمي.

تصميم الغلاف: مؤسسة برديس.

تنسيق داخلي : مروان الصياد.

الدار: دار اليانور للنشر الإلكتروني.

رقم تواصل الدار: 01151293168.

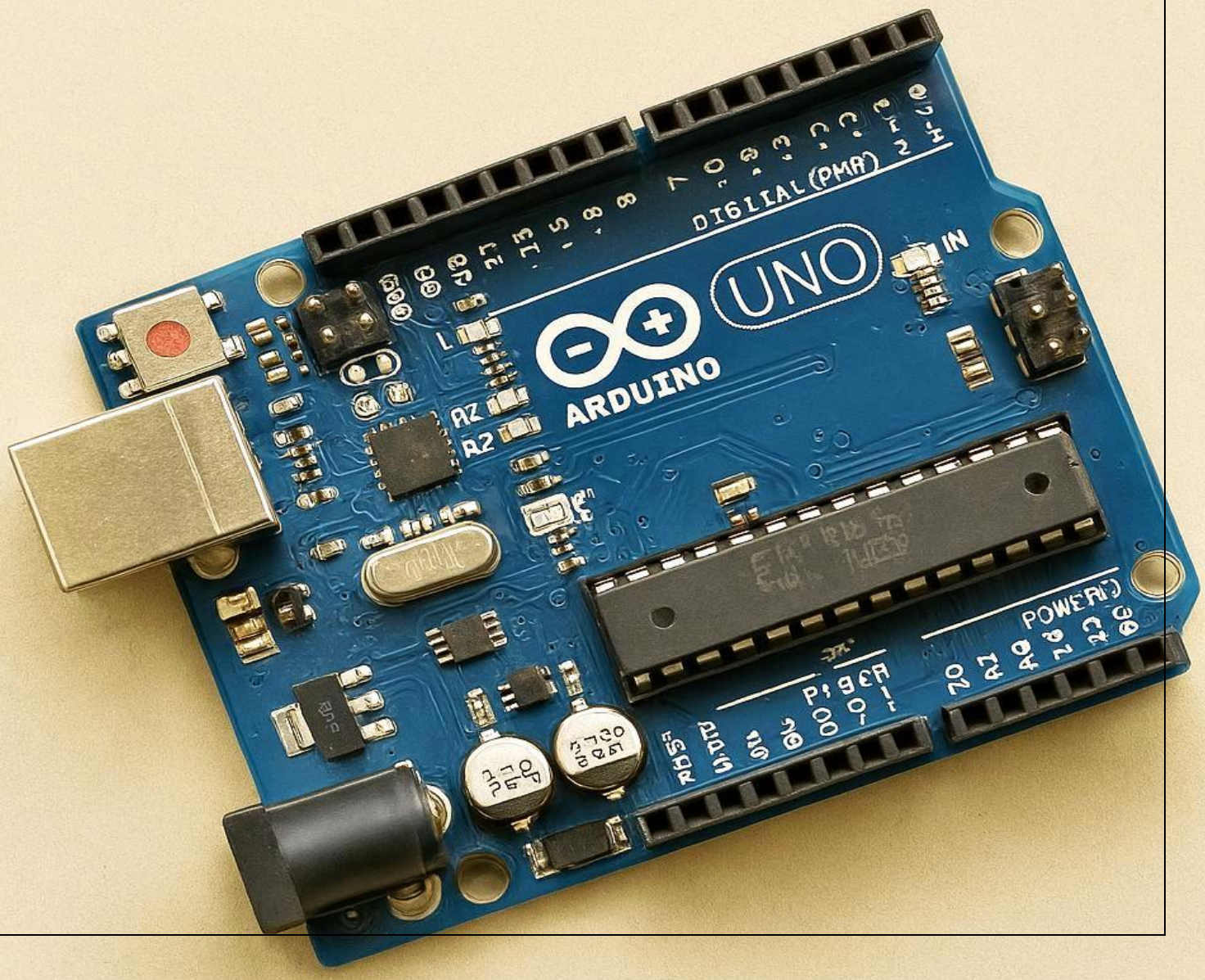
جميع حقوق النشر محفوظة©

يمنع مانعًا بانًا الأقتباس أو إعادة النشر سواء بالطباعة، أو النشر الإلكتروني، أو التصوير الضوئي للمحتوى، أو أي جزء منه إلا بأذن كتابي من الناشر والمؤلف.

ومن يخالف ذلك يعرض نفسه المساءلة القانونية طبقًا لحقوق الملكية الفكرية المنصوص عليها في القانون.

Learn Arduino Step by Step

Prepared by
Eman Moumen



المقدمة:

بسم الله الرحمن الرحيم، والصلاة والسلام على سيدنا محمد أشرف الخلق أجمعين، ثم أما بعد...

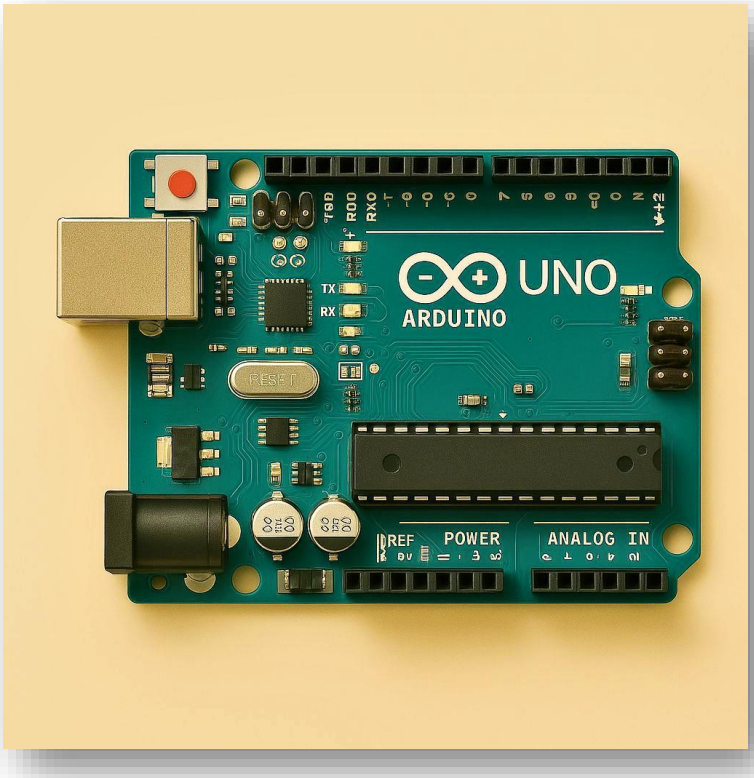
في عالمٍ يتسارع فيه التطور التكنولوجي، أصبحت الإلكترونيات والبرمجة جزءًا لا يتجزأ من حياتنا اليومية، من الهواتف الذكية إلى الأجهزة المنزلية الذكية، كلها تعتمد على أنظمة مدمجة تتحكم بها برمجيات دقيقة، وهنا يأتي دور الأردوينو، هذه اللوحة الصغيرة التي فتحت أبواب الابتكار للجميع، من الهواة إلى المحترفين.

هذا الكتاب هو دليلك العملي لتعلم الأردوينو "خطوة بخطوة" دون الحاجة إلى خلفية تقنية مسبقة، صُمِم ليأخذ بيدك من الأساسيات، مثل فهم المكونات الإلكترونية وكتابة أول سطر برمجي، وصولًا إلى بناء مشاريع حقيقية وممتعة.

سواء كنت طالبًا، هاويًا، أو حتى معلمًا يبحث عن وسيلة تعليمية مبسطة، ستجد في هذا الكتاب ما يلهمك ويمنحك الثقة لتبدأ رحلتك في عالم الإلكترونيات الذكية.

ما هو الأردوينو، ولماذا سُمِيَ بهذا الاسم؟

الأردوينو هو "لوحة إلكترونية قابلة للبرمجة" تُستخدم لتنفيذ مشاريع إلكترونية بسهولة، يعتمد على معالج دقيق يمكن برمجته ليؤدي مهام مختلفة، مثل: التحكم في الإضاءة، تشغيل المحركات، واستقبال البيانات من الحساسات.



💡ببساطة، الأردوينو هو العقل الذي يجعل الأجهزة الإلكترونية "تفكر وتنفذ الأوامر".

ليه اسمه "أردوينو" أصلًا؟

الاسم جاي من مقهى صغير في إيطاليا اسمه

"Bar di Re Arduino"

اللي اجتمع فيه المهندسين أول مرة لما بدأوا المشروع.

أنواع الأردوينو:

1. Arduino Uno:

- الأشهر والأكثر استخدامًا.
- يحتوي على ١٤ طرفًا رقميًا (Digital)، و٦ تناظري (Analog).

2. Arduino Mega:

- يحتوي على عدد كبير من الأطراف (٥٤ طرفًا رقميًا و١٦ تناظريًا).
- مناسب للمشاريع الكبيرة والمعقدة.

3. Arduino Nano:

- صغير الحجم ومناسب للمشاريع التي تتطلب مساحة أقل.

4. Arduino Leonardo:

- يدعم محاكاة لوحة مفاتيح أو فأرة عند توصيله بالكمبيوتر.

5. Arduino Due:

- يحتوي على معالج أقوى (ARM Cortex-M3).
- يُستخدم للمشاريع التي تحتاج معالجة أسرع.

6. Arduino Pro Mini:

- صغير وخفيف الوزن.
- مناسب للأجهزة القابلة للارتداء والمشاريع المحمولة.

7. Arduino MKR Series:

- مصمم لتطبيقات إنترنت الأشياء (IoT).
- يدعم تقنيات الاتصال مثل Wi-Fi وBluetooth.

كيف يعمل الأردوينو؟

يتم كتابة الأكواد البرمجية بلغة C/C++ عبر برنامج

Arduino IDE.

يتم تحميل الكود على اللوحة عبر كابل USB.

يقوم المعالج الموجود على اللوحة بتنفيذ الأوامر وفقًا للكود المكتوب.

أهم مكونات لوحة الأردوينو:

المعالج الدقيق: هو عقل الأردوينو، ويتحكم في تنفيذ الأكواد.

منافذ الإدخال والإخراج: تُستخدم لتوصيل مستشعرات، محركات، وأجهزة خارجية.

منافذ التغذية: تُستخدم لتغذية المكونات الموصلة بلوحة الأردوينو بالجهد اللازم لتشغيلها.

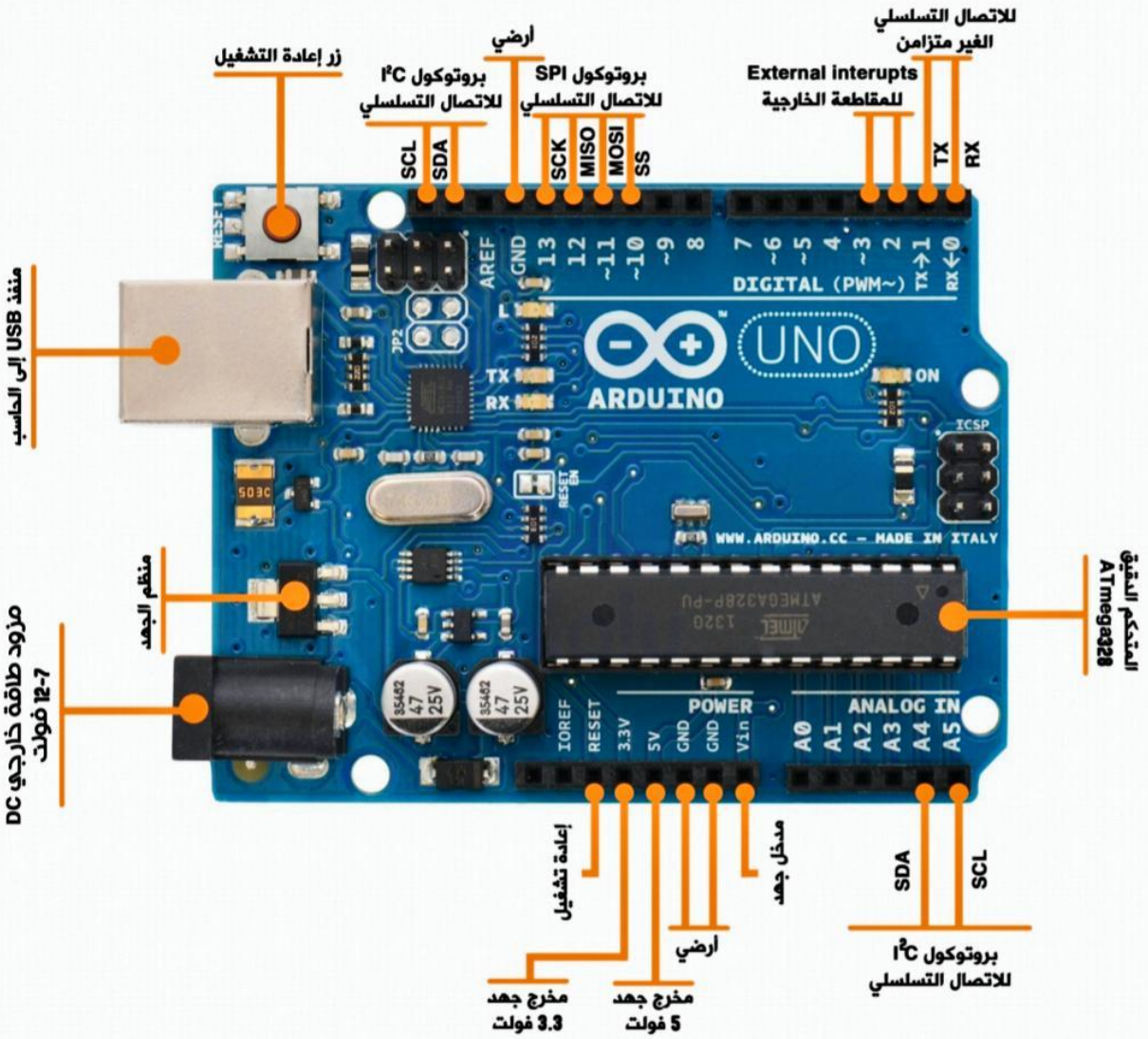
زر إعادة التشغيل Reset: لإعادة تشغيل البرنامج على اللوحة.

مزود الطاقة: يمكن تشغيل الأردوينو عبر USB

أو مصدر طاقة خارجي (بطارية).

هذه المكونات تجعل الأردوينو قادرًا على تنفيذ آلاف المشاريع الذكية!

مخطط أردوينو أونو ARDUINO UNO



المنافذ الرقمية من 2 إلى13 | المنافذ التناظرية من A0 إلى A5 | منافذ تعديل عرض الموجة PWM: 3 , 5 , 6 , 9, 10 , 11

شكل آخر من أردوينو أونو

ألواح أردوينو من صنع شركات أخرى

لتثبيت بيئة العمل: قم بتحميل Arduino IDE على جهازك (الكمبيوتر) من الموقع الرسمي (https://www.arduino.cc/en/software).

شرح بيئة عمل Arduino IDE:

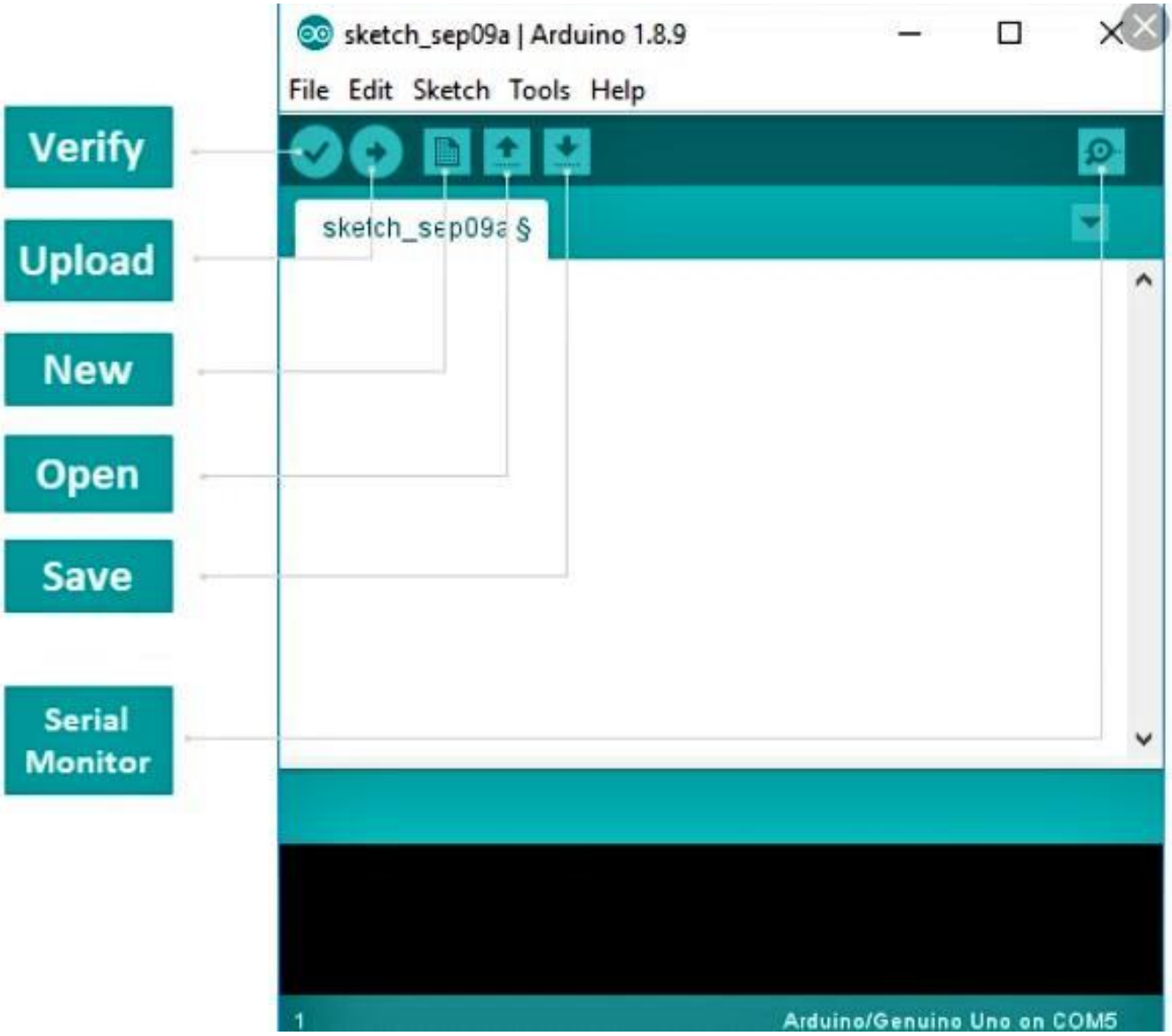
Arduino IDE هو البرنامج الرسمي المستخدم لكتابة الأكواد البرمجية وتحميلها على لوحات الأردوينو، يتميز بواجهة سهلة الاستخدام ويعمل على أنظمة Mac، Windows، و Linux.

أهم مكونات Arduino IDE:

- **شريط الأدوات:** يحتوي على أزرار للتحقق من الكود، تحميله على الأردوينو، فتح المشاريع، وحفظها.
- **محرر الأكواد:** المكان الذي تكتب فيه التعليمات البرمجية بلغة C/C++.
- **الرسائل والإشعارات:** تعرض الأخطاء والملاحظات أثناء كتابة الكود.
- **المنفذ التسلسلي (Serial Monitor):** يُستخدم لرؤية البيانات القادمة من الأردوينو.
- **إدارة المكتبات:** يمكنك إضافة مكتبات جاهزة لتوسيع وظائف الأردوينو بسهولة.

كيف يعمل؟

- تكتب الكود في المحرر.
- تضغط على "Verify" للتحقق من الأخطاء.
- تضغط على "Upload" لتحميل الكود إلى الأردوينو عبر كابل USB.
- يمكنك استخدام Serial Monitor لمراقبة البيانات القادمة من الأردوينو (اختياري).



مخطط لبرنامج Arduino IDE من الداخل

يمكنك تحميل برنامج ArduinoDroid على هاتفك الأندرويد من [متجر Google Play]

https://play.google.com/store/apps/details?id=nam)
(e.antonsmirnov.android.arduinodroid2&hl=en-US

شرح بيئة عمل ArduinoDroid:

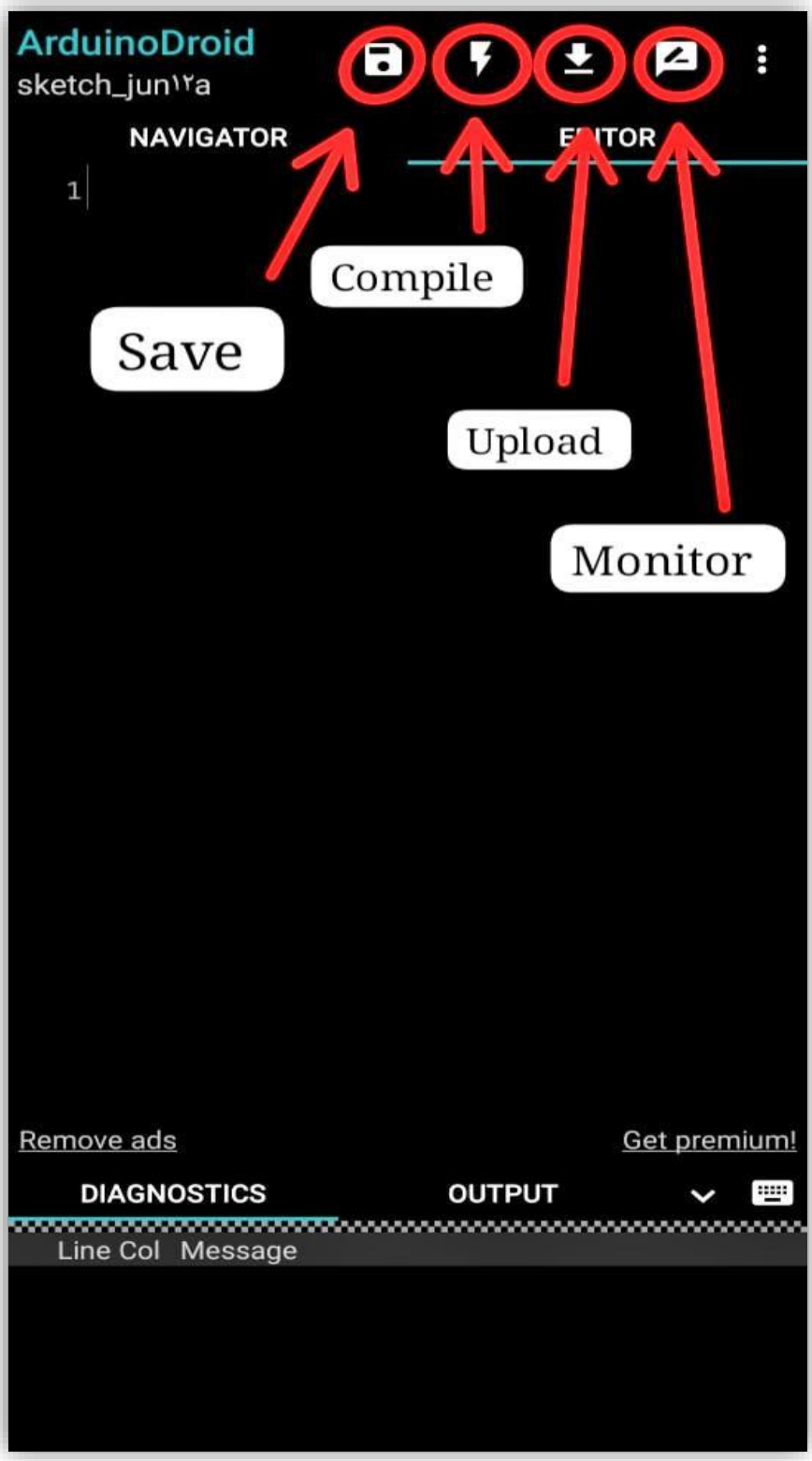
ArduinoDroid هو تطبيق مخصص للهواتف الذكية التي تعمل بنظام Android، يسمح لك ببرمجة الأردوينو مباشرةً من هاتفك دون الحاجة إلى كمبيوتر.

أهم مكونات ArduinoDroid:

- **محرر الأكواد:** يشبه محرر Arduino IDE لكنه مخصص للهواتف.
- **إدارة المكتبات:** يمكنك تحميل مكتبات إضافية مباشرةً من التطبيق.
- **المنفذ التسلسلي:** يسمح لك بمراقبة البيانات القادمة من الأردوينو.
- **دعم USB OTG:** تحتاج إلى وصلة OTG لتوصيل لوحة الأردوينو بالهاتف.
- **رفع الأكواد مباشرةً:** يمكنك تحميل الكود إلى لوحة الأردوينو دون الحاجة إلى كمبيوتر.

كيف يعمل؟

- تكتب الكود في التطبيق.
- تضغط على "Compile" للتحقق من الأخطاء.
- توصل لوحة الأردوينو بالهاتف عبر وصلة OTG.
- تضغط على "Upload" لتحميل الكود إلى لوحة الأردوينو.
- يمكنك استخدام Serial Monitor لرؤية البيانات القادمة من الأردوينو (اختياري).



مخطط لبرنامج ArduinoDroid من الداخل

كابل USB الخاص بلوحة الأردوينو

وصلة OTG

إذا أردت تجربة الأكواد ولم يكن لديك لوحة أردوينو، يمكنك تجربة الأكواد باستخدام برنامج Proteus، وهو محاكي قوي يسمح لك بتصميم وتشغيل الدوائر الإلكترونية افتراضيًا!

رابط تحميل برنامج الـ Proteus:

(/https://www.labcenter.com/downloads)

ملحوظة: هذا البرنامج يعمل على الكمبيوتر فقط.

شرح بيئة عمل برنامج البروتس:

Proteus

هو برنامج شامل يحتوي على عدة أدوات لمحاكاة الدوائر الإلكترونية، وأهم مكوناته:

Schematic Capture (محرر التصميم):

يسمح لك برسم الدوائر الإلكترونية باستخدام مكونات جاهزة.

Simulation (المحاكي):

يمكنك من خلاله تشغيل الدائرة ومراقبة كيفية عملها.

مكتبة المكونات:

تحتوي على آلاف العناصر مثل الأردوينو، المقاومات، الحساسات، المحركات، والمزيد...

تصميم لوحات PCB:

يمكنك تحويل الدائرة إلى تصميم جاهز للطباعة على لوحة إلكترونية.

كيف تستخدم Proteus لمحاكاة الأردوينو؟

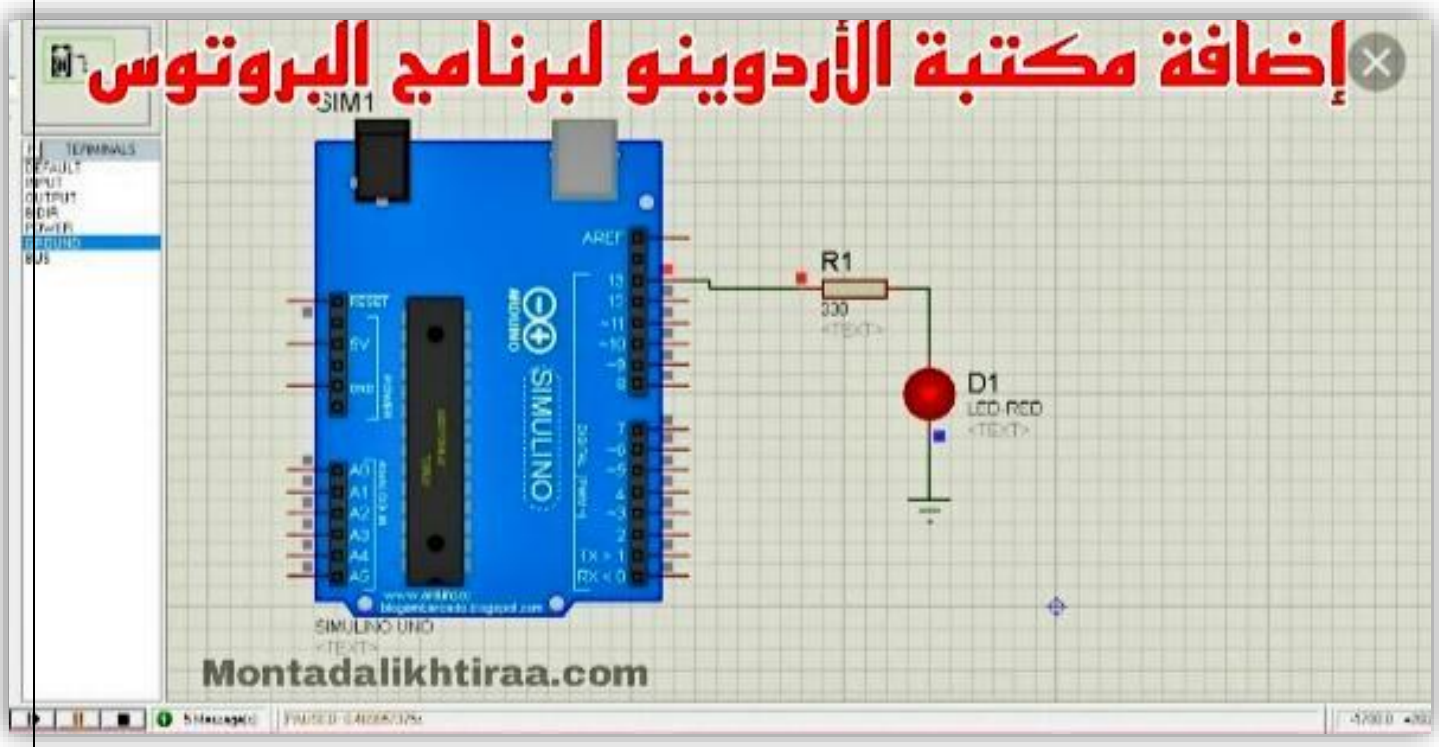
إضافة لوحة أردوينو من المكتبة داخل البرنامج.

توصيل المكونات مثل ليد ومقاومات وأسلاك.

كتابة الكود في Arduino IDE ثم تحويله إلى ملف Hex.

تحميل ملف Hex داخل الأردوينو الافتراضي في Proteus.

تشغيل المحاكاة ومشاهدة كيف يعمل الكود.



مخطط لبرنامج Proteus من الداخل

أساسيات البرمجة بلغة C/++C الخاصة بالأردوينو:

ما هي البرمجة ولماذا نستخدمها؟

البرمجة هي كتابة تعليمات وأوامر بلغة يفهمها الكمبيوتر أو الأردوينو لتنفيذ مهام معينة.

"تخيّل أنك تعطي جهازًا إلكترونيًا "خريطة طريق" ليقوم بمهمة مثل تشغيل لمبة، التحكم في محرك، أو قراءة بيانات من حساس".

ببساطة، البرمجة هي لغة التخاطب مع الأجهزة الذكية!

الهيكل الأساسي لأي كود أردوينو:

أي كود أردوينو يتكون من جزئين رئيسيين:

void setup()

يُنفذ مرة واحدة عند تشغيل الجهاز

void loop()

يُنفذ بشكل مستمر ومتكرر

لماذا نحتاج هذين الجزئين؟

Setup():

يحتوي على الإعدادات الأولية، مثل تحديد المنافذ.

loop():

يحتوي على التعليمات التي تتكرر باستمرار، مثل تشغيل لمبة أو قراءة مستشعر.

المتغيرات (Variables):

المتغيرات هي أماكن في ذاكرة الجهاز تُستخدم لتخزين البيانات مثل الأرقام والنصوص.

هناك أنواع متعددة منها، أهمها:

int: رقم صحيح مثل ٢٥

float: رقم عشري مثل ٢٣,٥

bool: منطقي (true/false)

char: حرف واحد مثل A

ما أهمية المتغيرات؟

تساعدنا على تخزين البيانات واستخدامها لاحقًا أثناء تنفيذ البرنامج.

التحكم في منافذ الأردوينو:

الأردوينو يحتوي على مخارج ومداخل تُستخدم للتحكم في الأجهزة المتصلة به.

تهيئة المنافذ قبل الاستخدام:

```
pinMode(13, OUTPUT);
// تحديد المنفذ ١٣ كمخرج
pinMode(2, INPUT);
// تحديد المنفذ ٢ كمدخل
```

تشغيل وإطفاء أي مخرج (مثل LED):

```
digitalWrite(13, HIGH);
// تشغيل
digitalWrite(13, LOW);
// إطفاء
```

قراءة حالة المدخلات (مثل زر تشغيل):

```
int buttonState = digitalRead(2);
```

لماذا نحتاج هذه الأوامر؟

لأننا نستخدم الأردوينو للتحكم في الأجهزة الإلكترونية مثل الحساسات، المحركات، والمصابيح.

التأخير (Delay):

أحيانًا نحتاج إلى إضافة فاصل زمني بين تنفيذ الأوامر:

```
delay(1000);
// انتظر لمدة ١٠٠٠ مللي ثانية (١ ثانية)
```

الجمل الشرطية (if statements):

تُستخدم لاتخاذ قرارات بناءً على شرط معين:

```
if (digitalRead(2) == HIGH) {
  digitalWrite(13, HIGH);
// تشغيل LED إذا كان الزر مضغوط
}
```

مثال عملي:

إذا ضغط المستخدم على الزر، سيضيئ المصباح.

إذا لم يضغط، سيظل المصباح مُطفأ.

الشاشة التسلسلية (Serial Monitor):

تُتيح لك طباعة بيانات على شاشة الكمبيوتر لمعرفة القيم أثناء تشغيل الأردوينو.

تشغيل الاتصال التسلسلي:

```
Serial.begin(9600);
```

طباعة رسالة على الشاشة:

```
Serial.println("مرحبًا بك في عالم الأردوينو");
```

لماذا نستخدمها؟

لرؤية القيم القادمة من الحساسات، الأزرار، وغيرها أثناء تشغيل الأردوينو.

يلا بينا نشرح كود تشغيل وإطفاء الليد (LED) كل ثانية

هنخلي الأردوينو:

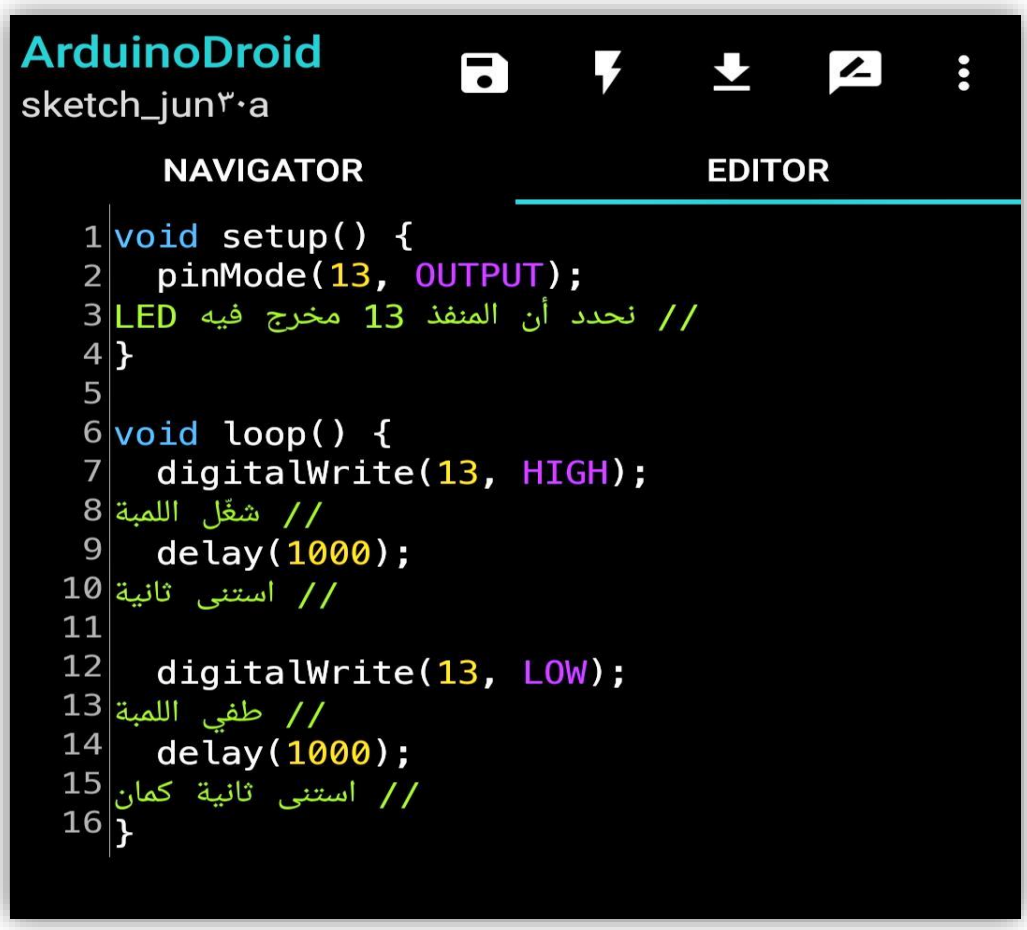
- يشغّل لمبة LED.
- ينتظر ثانية (١٠٠٠ مللي ثانية).

- يطفّيها.

- ينتظر ثانية تانية.

- ويكرر ده طول ما هو شغال.

الكود كاملًا:



شرح الكود سطر بسطر:

Void setup() {...}

دا الجزء اللي بيتنفّذ مرة واحدة بس لما تشغل الأردوينو.

pinMode(13, OUTPUT);

معناها إننا بنقول للأردوينو: "المنفذ رقم ١٣ هيكون مخرج" علشان نقدر نتحكم فيه.

بالمناسبة، أغلب لوحات الأردوينو فيها لمبة صغيرة داخليًا 💡 مربوطة بالمنفذ ١٣.

Void loop() {...}

دا الجزء اللي بيتكرّر بشكل لا نهائي.

digitalWrite(13, HIGH);

دي معناها: شغّل التيار في المنفذ ١٣ (شغّل اللمبة).

Delay(1000);

استنى ١٠٠٠ مللي ثانية (١ ثانية).

digitalWrite(13, LOW);

طفي التيار (طفي اللمبة).

delay(1000);

استنى كمان ثانية.

وهكذا... يفضل يكرّر نفس الخطوات طول ما الأردوينو شغال.

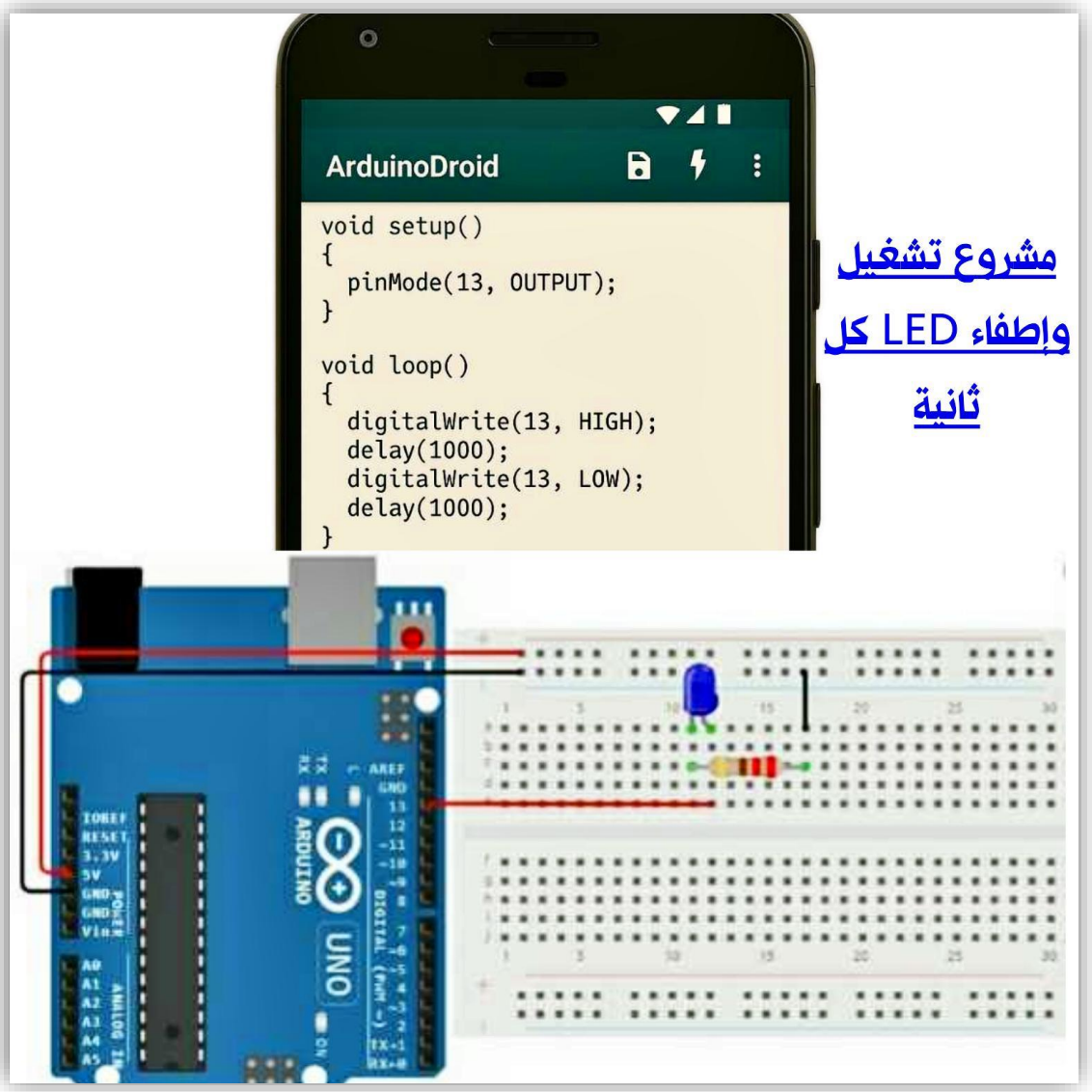
هنحتاج إيه علشان نطبّق الكود ده؟

- لوحة أردوينو.

- لمبة (LED).

- مقاومة (٢٢٠ أوم).

- أسلاك توصيل.



مشروع تشغيل وإطفاء LED كل ثانية

الحساسات... ما هي الحساسات؟

الحساسات (Sensors): هي أجهزة إلكترونية تستخدم لقياس وتسجيل البيانات مثل الحرارة، الحركة، والضوء، والأردوينو يستطيع قراءة هذه البيانات واتخاذ قرارات بناءً عليها.

هناك نوعان رئيسيان من الحساسات:

- **حساسات رقمية (Digital Sensors): تعطي قيمتين فقط (HIGH أو LOW)، مثل حساس الحركة PIR.**
- **حساسات تناظرية (Analog Sensors): تعطي قيمًا متعددة (٠ إلى ١٠٢٣)، مثل حساس الحرارة والضوء.**

حساس الحركة (Motion Sensor):

مثال: حساس PIR

هذا الحساس يستشعر حركة الأجسام أمامه، ويعطي خرج رقمي:

- HIGH (عندما يكتشف حركة)

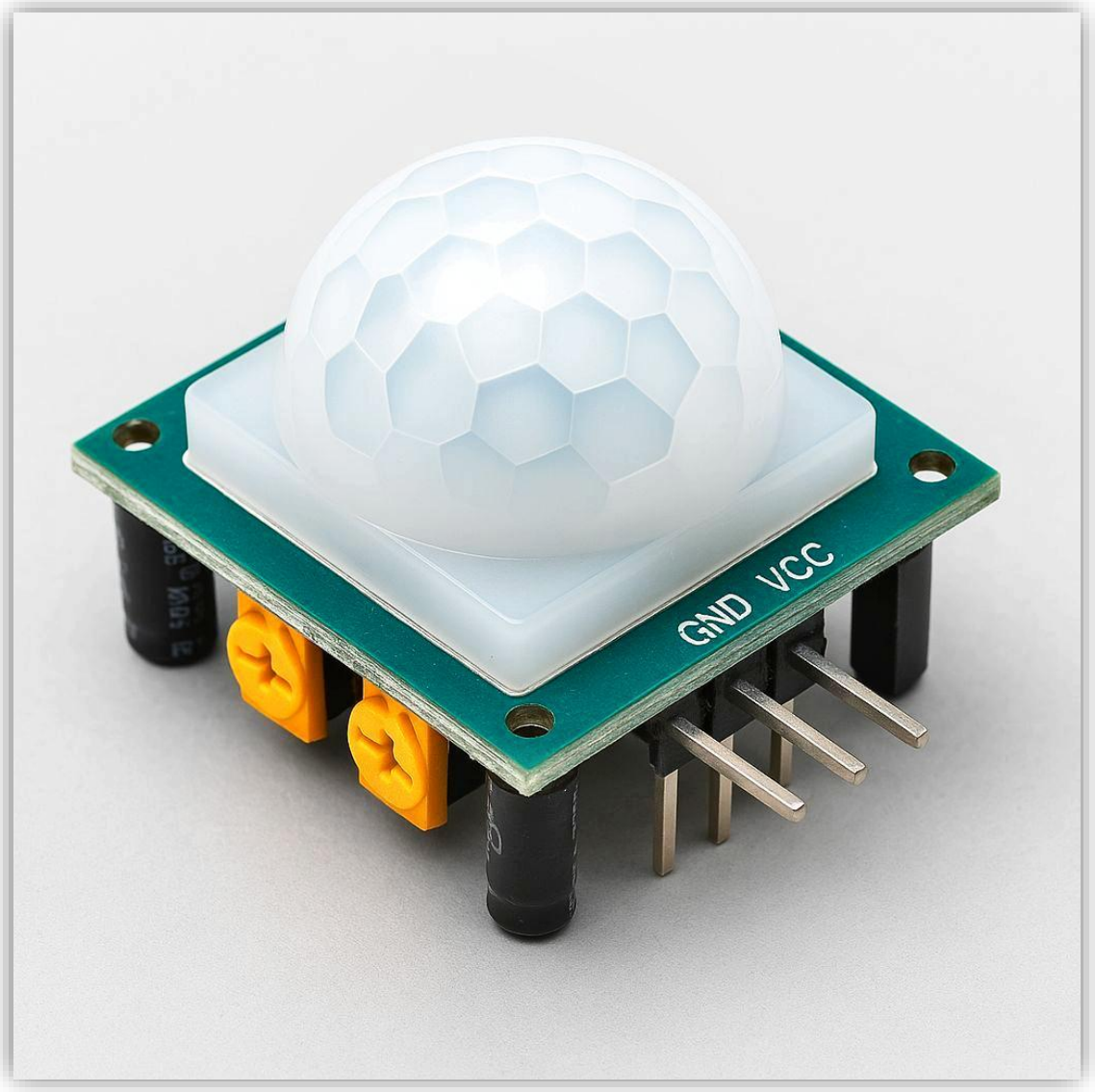
- LOW (عندما لا توجد حركة)

كيف بنوصله بالأردوينو؟

VCC ☜ V5 في الأردوينو

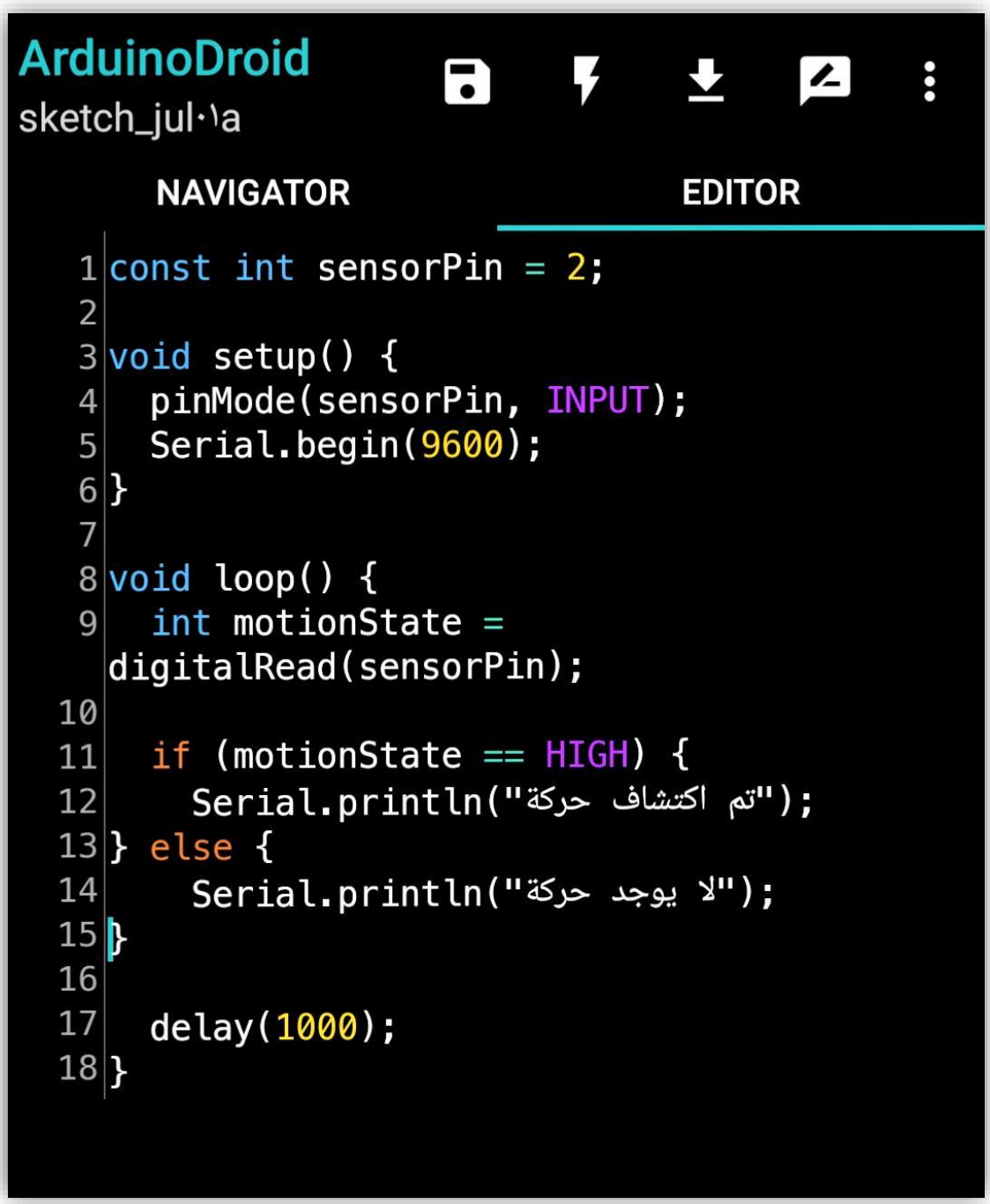
GND ☜GND في الأردوينو

OUT ☜ أي مدخل رقمي في الأردوينو (مثل D2)



حساس الحركة PIR

الكود البرمجي:


ArduinoDroid
sketch_jul٠١a
NAVIGATOR
EDITOR
1 const int sensorPin = 2;
2
3 void setup() {
4 pinMode(sensorPin, INPUT);
5 Serial.begin(9600);
6 }
7
8 void loop() {
9 int motionState =
digitalRead(sensorPin);
10
11 if (motionState == HIGH) {
12 Serial.println("تم اكتشاف حركة");
13 } else {
14 Serial.println("لا يوجد حركة");
15 }
16
17 delay(1000);
18 }

كيف يعمل؟

- عند اكتشاف حركة، يرسل الحساس إشارة HIGH للأردوينو.
- الأردوينو يعرض رسالة في Serial Monitor.

حساس الضوء (Light Sensor):

مثال: حساس LDR

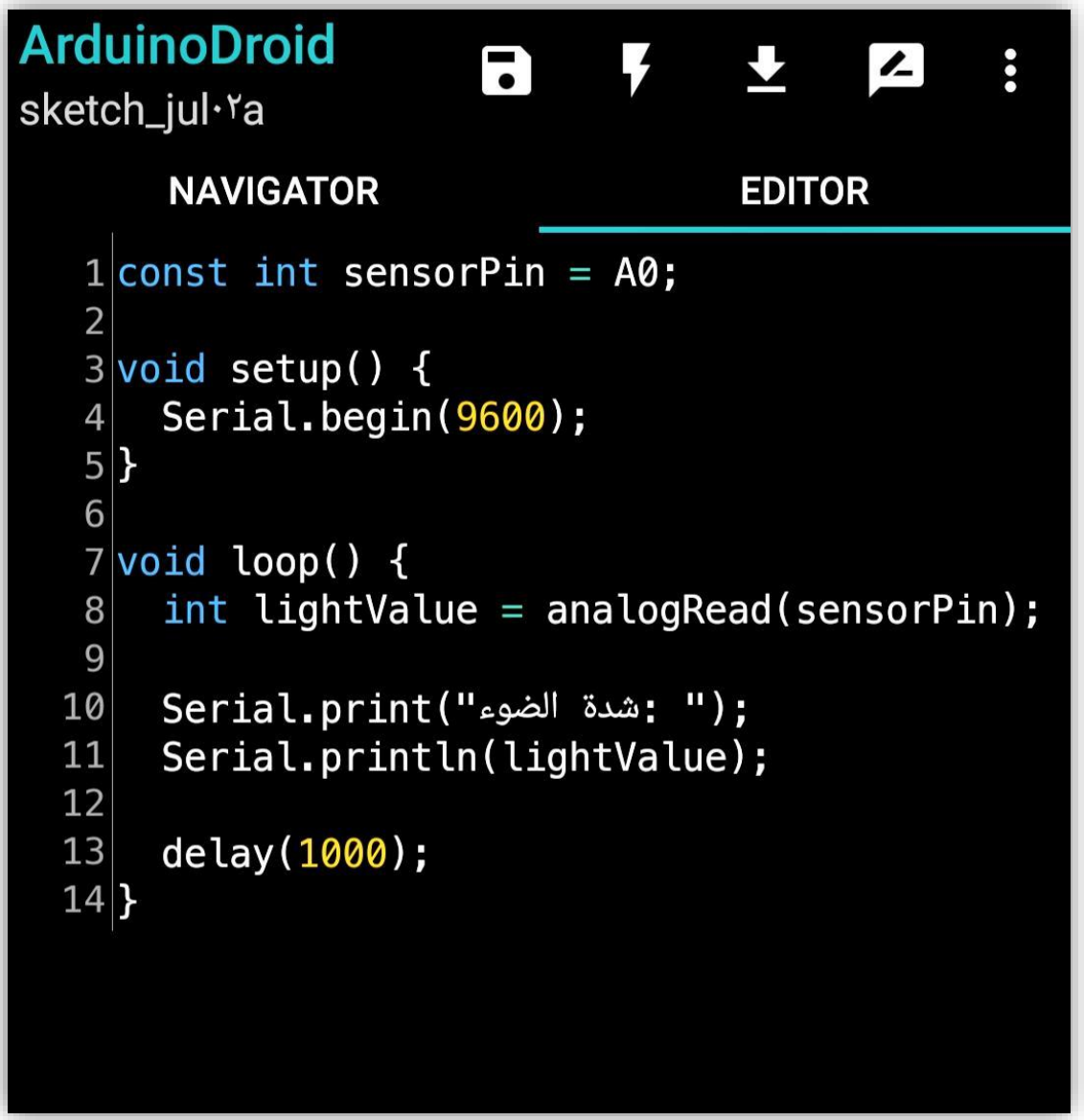
هذا الحساس يقيس شدة الضوء ويعطي خرج تناظري (٠ إلى ١٠٢٣)، يمكن استخدامه لتشغيل إضاءة تلقائية.

كيف بنوصله بالأردوينو؟

- إحدى أطراف LDR إلى V5 في الأردوينو.
- الطرف الآخر إلى أي منفذ تناظري (مثلًا A0) مع مقاومة K10 موصلة إلى الأرضي GND في الأردوينو.

حساس الضوء LDR

الكود البرمجي:

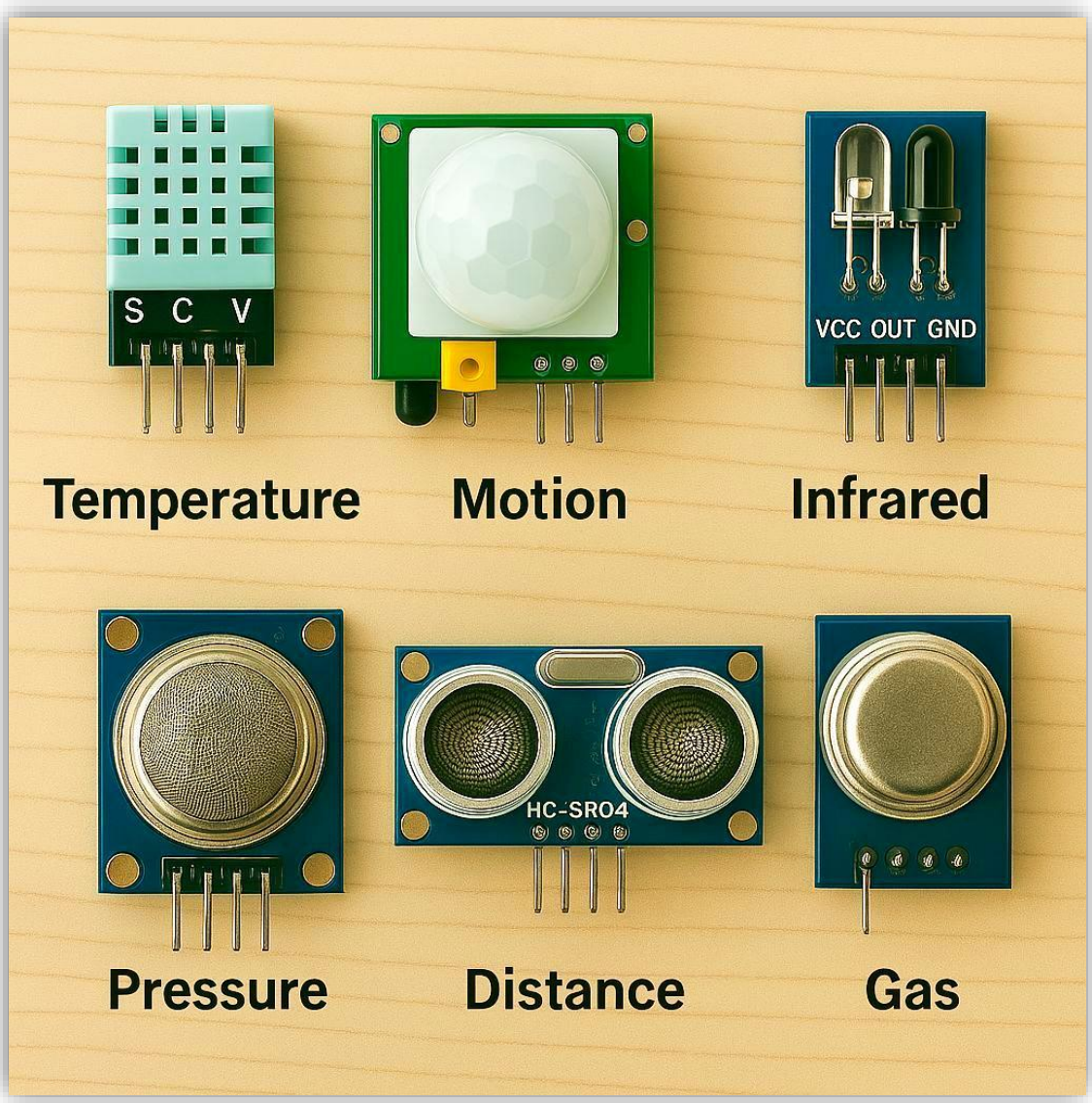


كيف يعمل؟

- الأردوينو يقرأ قيمة شدة الضوء من الحساس.
- يمكن برمجة الأردوينو لتشغيل LED تلقائيًا في الظلام.

من أنواع الحساسات:

- **حساس الغاز (MQ2):** يستشعر الدخان والغازات الخطرة.
- **حساس المسافة (Ultrasonic HC-SR04):** يقيس المسافة بينه وبين أي جسم.
- **حساس الرطوبة (DHT11):** يقيس الحرارة والرطوبة معًا.
- **حساس الحرارة (Temperature Sensor):** يقيس درجة الحرارة.



كيف يمكن استخدام هذه الحساسات فى المشاريع؟

مشروع إنذار ضد السرقة:

باستخدام حساس PIR، يشغّل الأردوينو صفارة إنذار عند اكتشاف حركة.

مشروع مزرعة ذكية:

باستخدام حساس الرطوبة والحرارة يُعدّل نظام الري تلقائيًا.

مشروع إضاءة تلقائية:

باستخدام حساس الضوء يتم تشغيل المصابيح في الليل (الظلام).

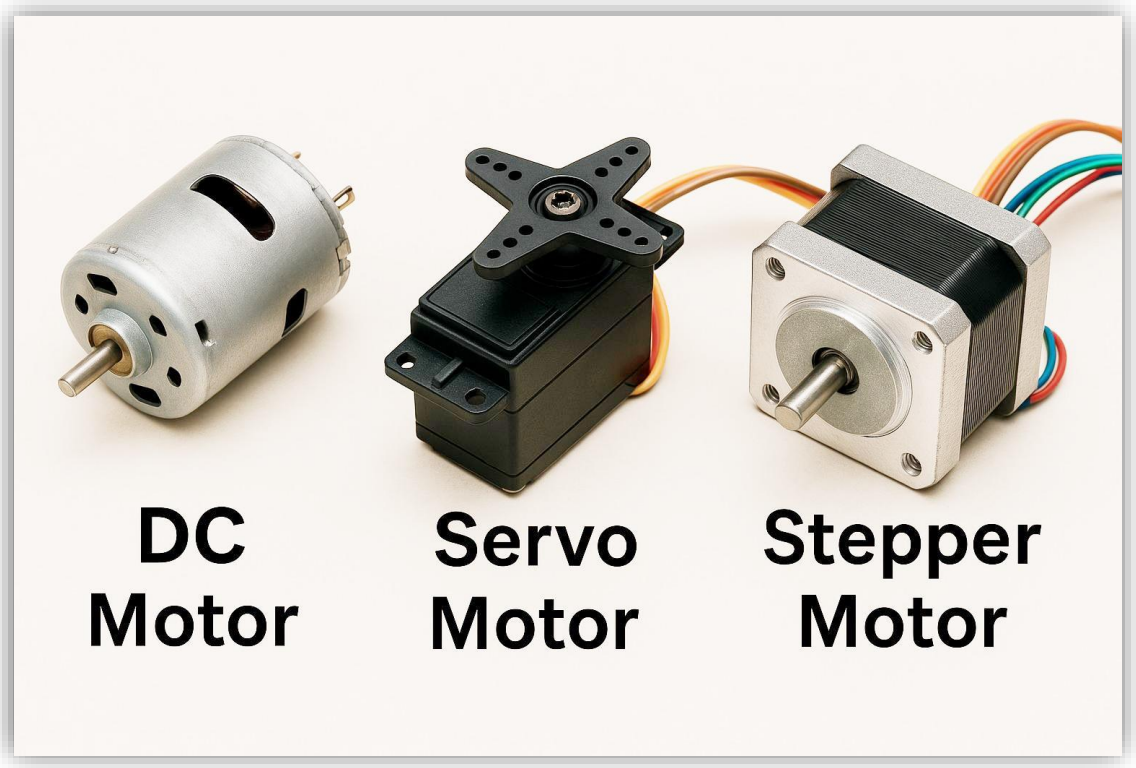
التحكم في المحركات والشاشات الرقمية:

أولًا: التحكم في المحركات

أنواع المحركات اللي بنتعامل معها في المشاريع:

- **محرك DC (التيار المستمر):** يدور في اتجاه واحد أو العكس.
- **محرك سيرفو (Servo):** يتحرك بزوايا محددة (٠° إلى ١٨٠°).

- **محرك خطوي (Stepper):** يتحرك بخطوات صغيرة ثابتة، مناسب للحركة الدقيقة.



من أنواع المحركات

مثال سهل: تشغيل محرك DC باستخدام ترانزستور

المكونات:

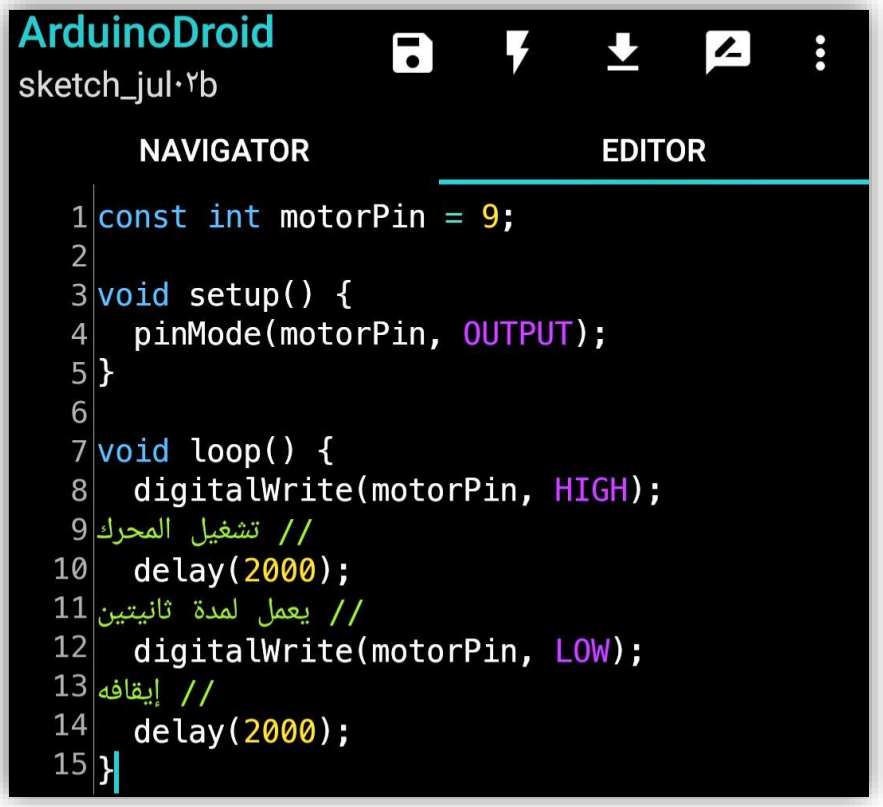
- محرك DC صغير
- ترانزستور NPN
- دايود N40071
- مقاومة ٢٢٠ أوم
- بطارية + أسلاك

التوصيل:

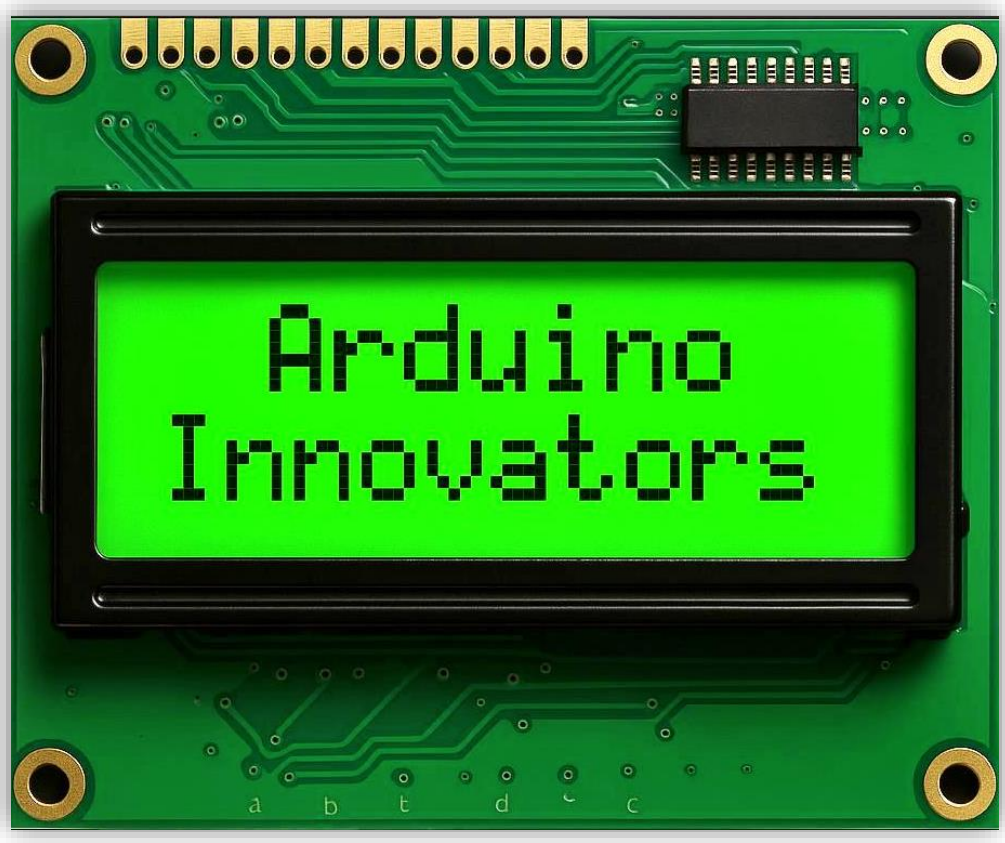
- الترانزستور يعمل كـ "مفتاح إلكتروني".

- **القاعدة (Base):** توصل بطرف المقاومة، ثم الطرف الثاني للمقاومة يوصل بخرج رقمي من الأردوينو (مثلًا D9).
- **المجمّع (Collector):** يوصل بأحد أطراف المحرك.
- **الباعث (Emitter):** يوصل بمنفذ GND.
- **الطرف الآخر من المحرك:** يوصل بمصدر طاقة خارجي (مثل بطارية V9).
- **الدايود:** يوصل بين طرفي المحرك لحمايته من الجهد العكسي.

الكود:



ثانيًا: الشاشات الرقمية



شاشة LCD 16x2

وظيفتها:

تعرض نصوص ورسائل بشكل واضح، ممتازة لعرض قراءات الحساسات أو الوقت أو أي بيانات.

التوصيل:

أسهل طريقة هي استخدام وحدة I2C مع الشاشة (توصيل بسيط بأربع أسلاك فقط):

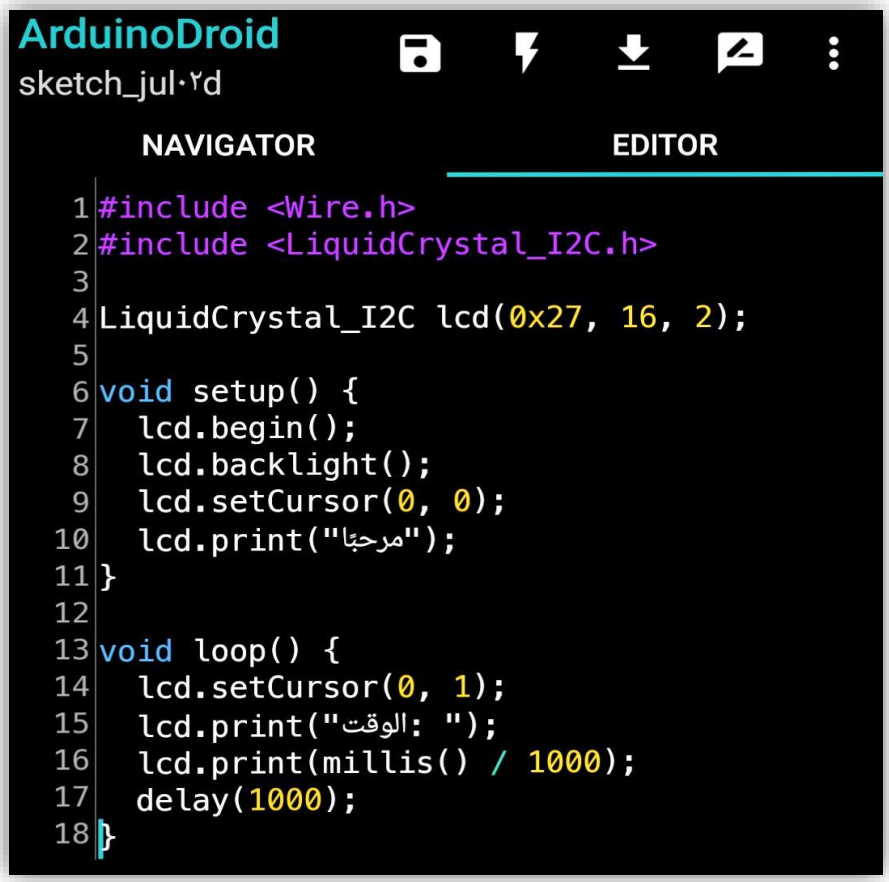
GND ☜GND في الأردوينو

V5 ☜VCC في الأردوينو

A4 ☜SDA في الأردوينو

A5 ☜SCL في الأردوينو

الكود (باستخدام مكتبة LiquidCrystal_I2C):




```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("مرحبًا");
}

void loop() {
  lcd.setCursor(0, 1);
  lcd.print("الوقت: ");
  lcd.print(millis() / 1000);
  delay(1000);
}
```

شرح الكود سطر بسطر

المكتبات المستخدمة:

```
#<include <Wire.h
#<include <LiquidCrystal_I2C.h
```

```
#<include <Wire.h
```

هذه المكتبة مسؤولة عن التواصل عبر بروتوكول I2C، وهو يسمح للأردوينو بالتواصل بسهولة مع الأجهزة مثل شاشة LCD I2C.

```
#<include <LiquidCrystal_I2C.h
```

مكتبة جاهزة للتحكم في شاشة LCD عبر I2C دون الحاجة إلى توصيل العديد من الأسلاك.

تعريف شاشة LCD:

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

هنا 👆 يتم إنشاء كائن lcd لتمثيل شاشة LCD.

٠x٢٧

هو عنوان الـ I2C الخاص بالشاشة، وهو يختلف من شاشة لأخرى (بعض الشاشات تستخدم ٠x٣F).

١٦, ٢

تعني أن الشاشة لديها ١٦ خانة عرض × ٢ صفوف.

ضبط الإعدادات في setup():

```
void setup() {
lcd.begin();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("مرحبًا");
}
```

```
lcd.begin();
```

تهيئة الشاشة حتى تبدأ في العمل.

```
lcd.backlight();
```

تشغيل الإضاءة الخلفية لجعل النص أكثر وضوحًا.

```
lcd.setCursor(0, 0);
```

تعيين المؤشر عند أول عمود وأول صف، بحيث يُكتب النص من البداية.

```
lcd.print("مرحبًا");
```

طباعة كلمة "مرحبًا" على الشاشة.

تنفيذ الحلقة loop():

```
void loop() {
lcd.setCursor(0, 1);
lcd.print("الوقت: ");
lcd.print(millis() / 1000);
delay(1000);
}
```

```
lcd.setCursor(0, 1);
```

وضع المؤشر عند أول عمود في الصف الثاني (أي أننا سنطبع النص في السطر السفلي من الشاشة).

```
lcd.print("الوقت: ");
```

طباعة كلمة "الوقت: " في الصف الثاني.

```
lcd.print(millis() / 1000);
```

استخدام دالة millis() لعرض الزمن الذي مضى منذ تشغيل الأردوينو، مقسومًا على ١٠٠٠ لتحويله من ميللي ثانية إلى ثوانٍ.

```
delay(1000);
```

تأخير لمدة ثانية واحدة، ثم تكرار الحلقة loop() مرة أخرى، وبالتالي يتم تحديث الوقت كل ثانية.

النتيجة المتوقعة على الشاشة:

- **الصف الأول: مرحبًا**
- **الصف الثاني: الوقت: ثم يتغير الرقم كل ثانية.**

شاشة Segment-7

وظيفتها:

تُستخدم لعرض أرقام مثل العداد، ويمكن التحكم بها باستخدام Arduino مع مقاومات أو IC مثل ٧٤HC٥٩٥ لتوفير المخرجات الرقمية.

كيفية التوصيل بالأردوينو:

التوصيل الكهربائي

حدد نوع الشاشة:

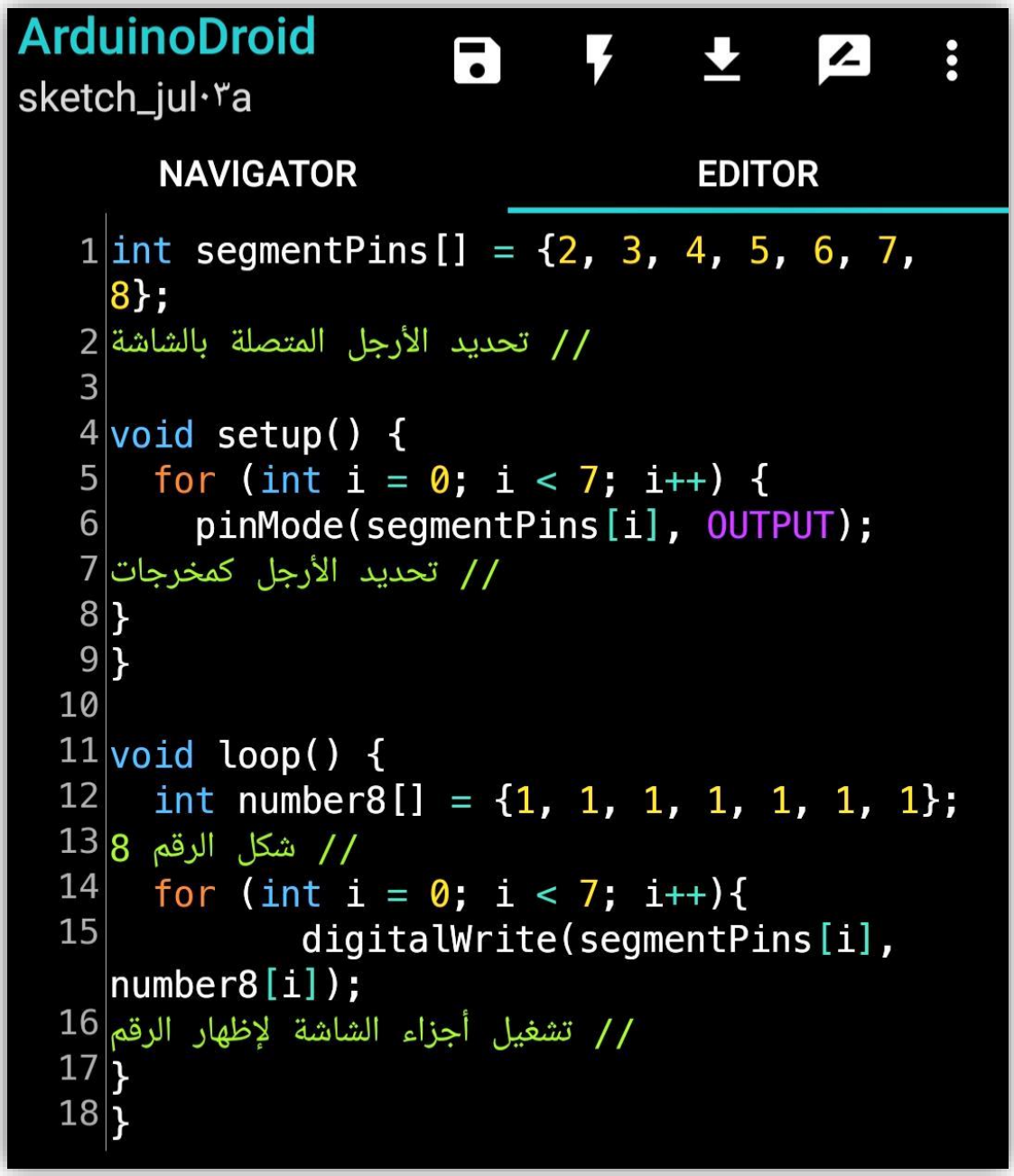
- إذا كانت أنود مشترك وصل الطرف المشترك إلى VCC (5V).
- إذا كانت كاثود مشترك وصل الطرف المشترك إلى GND.

وصل الأرجل السبع الخاصة بالإضاءة إلى الأردوينو عبر مقاومات لتجنب تلفها.

التوصيل النموذجي:

- A → Pin 2
- B → Pin 3
- C → Pin 4
- D → Pin 5
- E → Pin 6
- F → Pin 7
- G → Pin 8

كود أردوينو بسيط لتشغيل الرقم "٨":



يمكنك استخدام IC 74HC595 لتوفير عدد الأرجل.

أو الاستفادة من وحدات MAX7219 لإدارة أكثر من شاشة بسهولة.

كيفية التواصل بين الأردوينو والكمبيوتر باستخدام Serial Monitor:

أولًا: ما هو الـ Serial Monitor؟

هي أداة داخل برنامج Arduino IDE تسمح لك بمشاهدة البيانات القادمة من الأردوينو، أو إرسال أوامر له من الكمبيوتر.

تُستخدم بشكل أساسي في اختبار الكود، مراقبة القيم، أو إرسال بيانات (مثل نصوص وأرقام).

كيف يتم التواصل ؟

التواصل يتم عبر USB باستخدام بروتوكول اسمه Serial Communication (الاتصال التسلسلي)، ويكون بين:

- **الكمبيوتر (المرسل/المستقبل)**
- **ولوحة الأردوينو (المرسل/المستقبل)**

خطوات التواصل عمليًا:

- **وصل الأردوينو بالكمبيوتر باستخدام كابل USB.**
- **افتح Arduino IDE.**
- **في الكود استخدم:**

```
Serial.begin(9600);
```

لتفعيل الاتصال بسرعة ٩٦٠٠ بت في الثانية.

```
Serial.print()
```

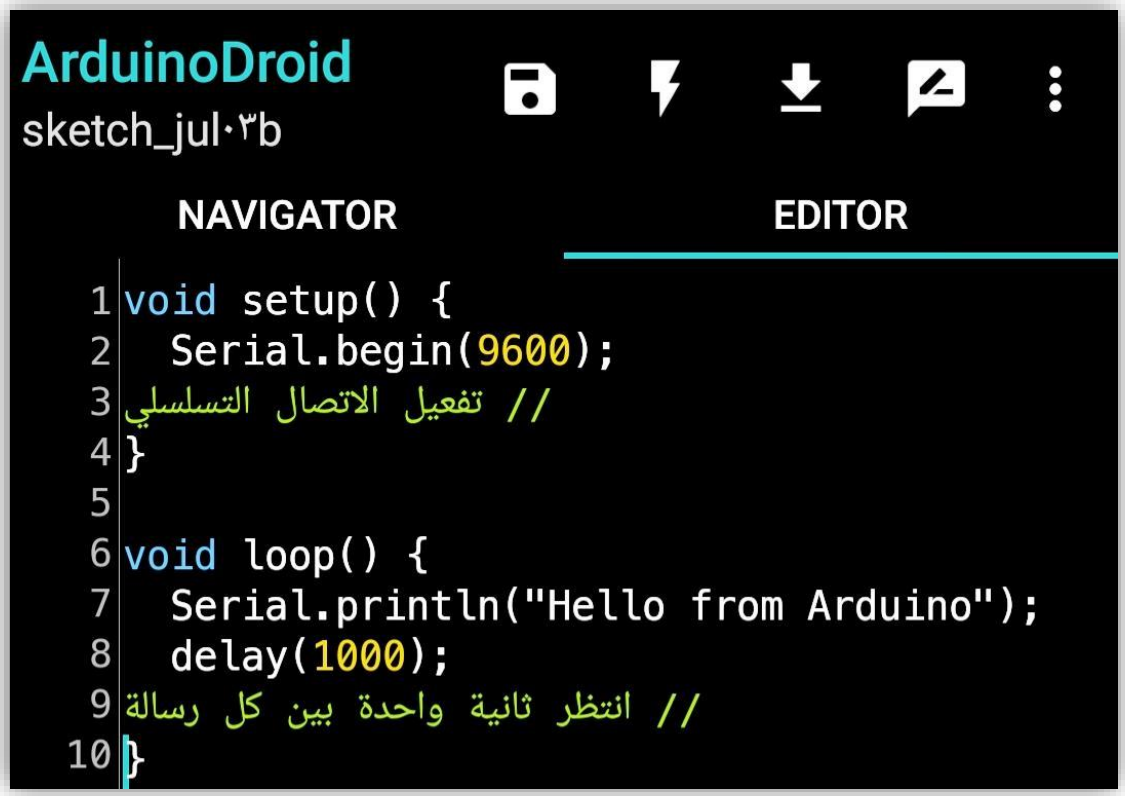
أو 👇

```
Serial.println()
```

لإرسال البيانات إلى شاشة الكمبيوتر.

- بعد رفع الكود اضغط على أيقونة المكبر (العدسة) في الزاوية اليمنى بالأعلى لفتح Serial Monitor.
- سترى البيانات التي يرسلها الأردوينو تظهر على الشاشة.

مثال عملي : إرسال "Hello" إلى الكمبيوتر



عند تشغيل هذا الكود وفتح Serial Monitor سترى الرسالة تتكرر كل ثانية.

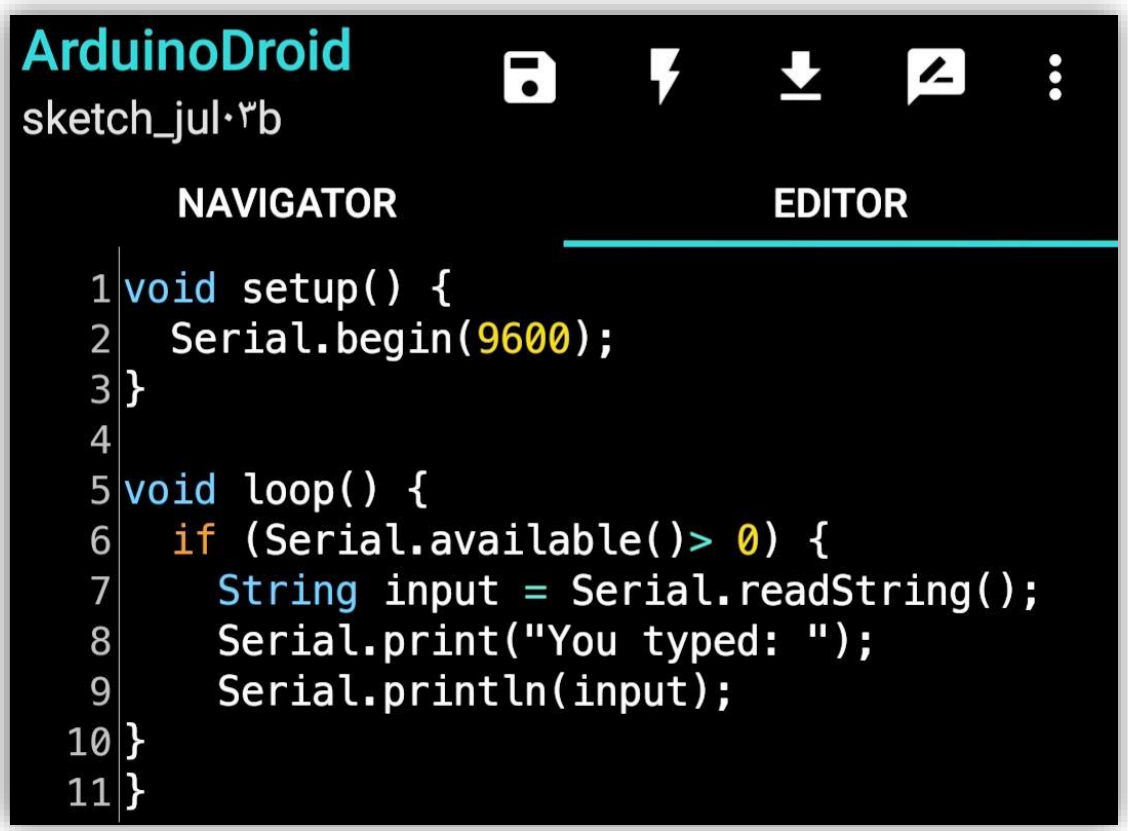
إرسال بيانات من المستخدم إلى الأردوينو:

يمكنك استخدام 👇

Serial.read()

و Serial.readString()

لاستقبال بيانات من المستخدم.



هنا الأردوينو ينتظر إدخال من المستخدم عن طريق الشاشة، ثم يعيد إظهاره.

استخدامات الـ Serial Monitor:

- قراءة القيم من أجهزة الاستشعار (مثل حرارة أو ضوء).
- إرسال أوامر للتحكم في محرّك أو إضاءة.
- اختبار الكود ومعرفة الأخطاء.

مشروع: "دائرة إنذار للكشف عن الحركة باستخدام الأردوينو وحساس الحركة PIR"

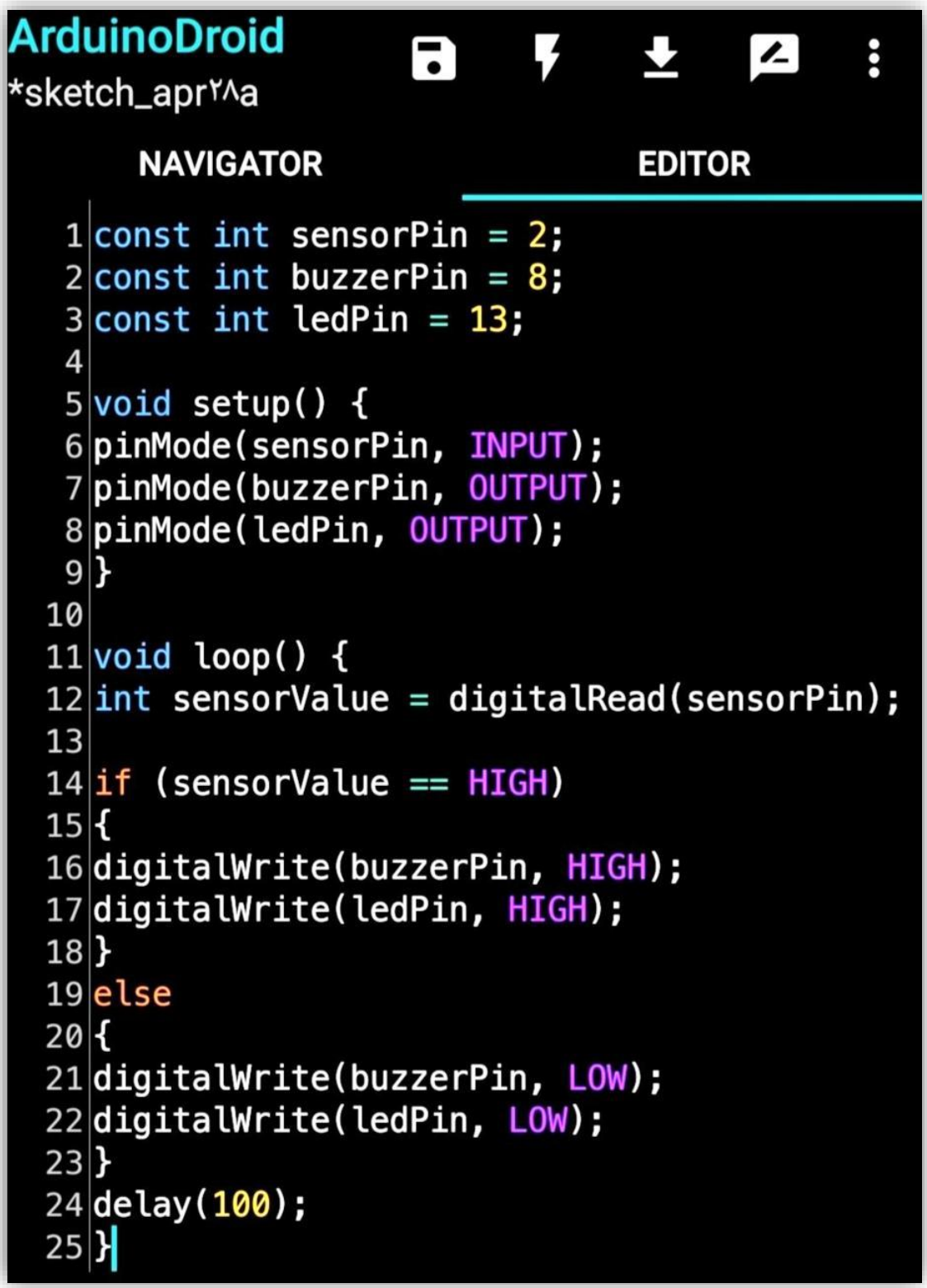
الهدف من المشروع: يهدف هذا المشروع إلى تصميم وتنفيذ دائرة إنذار تستخدم مستشعر الحركة للكشف عن وجود أي حركة غير متوقعة وإصدار تنبيه صوتي ومرئي.

وصف المشروع: تعتمد دائرة الإنذار على حساس PIR الذي يعمل على استشعار وجود الأجسام المتحركة في نطاقه وإرسال إشارة إلى وحدة التحكم (Arduino)، وعند اكتشاف الحركة يقوم النظام بتفعيل صفارة الإنذار (Buzzer) وإضاءة LED لتحذير المستخدم من النشاط غير المتوقع.

المكونات ووظائفهم:
Arduino Uno: هو العقل المسئول عن تشغيل الكود البرمجي ومعالجة البيانات من المستشعر وتفعيل التنبيه.
مستشعر الحركة (PIR Sensor): يستخدم لكشف وتتبع حركة الأجسام في نطاق معين.

صفارة إنذار (Piezo Buzzer): تستخدم لإصدار صوت تنبيه عند اكتشاف الحركة.
LED: إضاءة كمؤشر بصري عند تشغيل الإنذار.
مقاومة ٢٢٠ أوم: لحماية الليد من التلف.
أسلاك توصيل: لتوصيل المكونات معًا.
لوحة تجارب (Breadboard): لتجربة وتنفيذ الدائرة.

الكود البرمجي:


ArduinoDroid
*sketch_apr٢٨a
NAVIGATOR
EDITOR
1 const int sensorPin = 2;
2 const int buzzerPin = 8;
3 const int ledPin = 13;
4
5 void setup() {
6 pinMode(sensorPin, INPUT);
7 pinMode(buzzerPin, OUTPUT);
8 pinMode(ledPin, OUTPUT);
9 }
10
11 void loop() {
12 int sensorValue = digitalRead(sensorPin);
13
14 if (sensorValue == HIGH)
15 {
16 digitalWrite(buzzerPin, HIGH);
17 digitalWrite(ledPin, HIGH);
18 }
19 else
20 {
21 digitalWrite(buzzerPin, LOW);
22 digitalWrite(ledPin, LOW);
23 }
24 delay(100);
25 }

شرح الكود:

يتم تعريف المتغيرات والثوابت:

المستشعر يوصل بالمدخل الرقمي D2.

الصفارة توصل بالمخرج الرقمي D8.

الليد يوصل بالمخرج الرقمي D13.

الإعداد (Setup):

يتم إعداد الحساس كمدخل رقمي، والصفارة والـ LED كمخارج رقمية لإصدار تنبيه مرئي وصوتي عند الكشف عن الحركة.

الدورة الرئيسية (Loop):

يتم قراءة حالة الحساس باستخدام digitalRead(sensorPin)، وإذا كان هناك حركة (HIGH) يتم تشغيل التنبيه، وإذا لم يكن هناك حركة (LOW) يتم إيقاف التنبيه.

التأخير (Delay):

يعطي فترة قصيرة (١٠٠ مللي ثانية) لتحسين الأداء عن طريق منع التكرار السريع للقراءات، وتجنب الضوضاء الناتجة عن اهتزاز الإشارات.

خطوات التنفيذ:

طريقة توصيل المكونات الأساسية:

مستشعر الحركة (PIR Sensor):

توصيل الطرف الموجب VCC بمنفذ V5 في الأردوينو.

توصيل الطرف الأرضي GND بمنفذ GND في الأردوينو.

توصيل الطرف OUT (إشارة الإخراج) بالمنفذ الرقمي D2 في الأردوينو.

صفارة الإنذار (Piezo Buzzer):

توصيل الطرف الموجب للصفارة بالمنفذ الرقمي D8 في الأردوينو.

توصيل الطرف السالب بمنفذ GND في الأردوينو.

LED:

توصيل الطرف الموجب (Anode) بالمنفذ الرقمي D13 في الأردوينو.

توصيل الطرف السالب (Cathode) بمقاومة (٢٢٠ أوم) ثم بمنفذ GND في الأردوينو.

كيفية تحميل الكود وتجربة المشروع:

بعد كتابة الكود البرمجي الخاص بالمشروع على برنامج Arduino IDE تم التحقق أولًا من صحة الكود باستخدام ميزة Verify لضمان عدم وجود أخطاء برمجية، بعد ذلك تم تحميل الكود على لوحة Arduino عبر كابل

USB، ثم تم تشغيل الدائرة ومراقبة أدائها عمليًا، حيث عَمَل النظام بنجاحٍ تام.

فوائد المشروع:

تعزيز الأمان والحماية: يوفر إنذارًا فوريًا عند اكتشاف أي حركة غير متوقعة.

يمكن استخدامه في تطبيقات متنوعة في مختلف المجالات.

استخدام منزلي: حماية الأبواب والنوافذ وكشف الحركة في المناطق الحساسة.

استخدام تجاري: تنبيه الموظفين عند دخول العملاء إلى المتجر أو المكتب.

استخدام في المدارس والمستودعات: حماية المناطق المهمة من الوصول غير المصرح به.

يمنع حالات السرقة أو الدخول غير المصرح به عبر الاستجابة السريعة للإنذار.

يساعد في تقليل الحاجة إلى المراقبة المستمرة، حيث يعمل تلقائيًا دون تدخل بشري.

"التحكم عن بعد باستخدام وحدة البلوتوث (Bluetooth Module HC-٠٥)"

الهدف: نستخدم البلوتوث كحلقة وصل بين هاتف المستخدم

(أو أي جهاز بلوتوث) وبين لوحة الأردوينو، عشان نقدر نتحكم في أجهزة زي LED أو مروحة عن بُعد.

أولًا المكونات المطلوبة:

١. أردوينو (Uno أو أي إصدار)

٢. وحدة بلوتوث HC-05

٣. مقاومة ١ كيلو أوم

٤. مقاومة ٢ كيلو أوم

٥. أسلاك توصيل (jumper wires)

٦. مصدر تغذية (USB أو بطارية ٩ فولت)

٧. تطبيق بلوتوث على الموبايل (زي: "Bluetooth Terminal" أو "Arduino Bluetooth Controller")

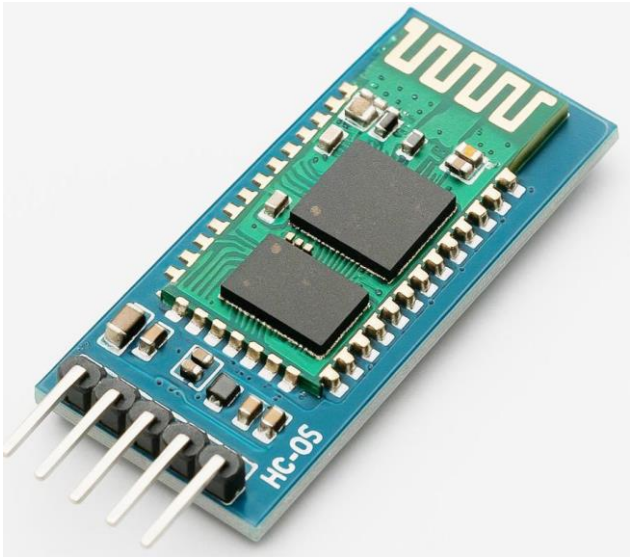
طريقة التوصيل:

وحدة البلوتوث HC-05 فيها ٤ أرجل رئيسية:

- V5: (VCC) لتوصيل الكهرباء
- GND: (GND) الأرضي

- TX: (TXD) إرسال بيانات (Transmit)

- RX: (RXD) استقبال بيانات (Receive)



وحدة بلوتوث HC-05

مهم: الأردوينو يرسل V5، لكن HC-05 يفضل V3.3 على مدخل RXD، عشان كده بنعمل مجزئ جهد (Voltage Divider) باستخدام مقاومتين:

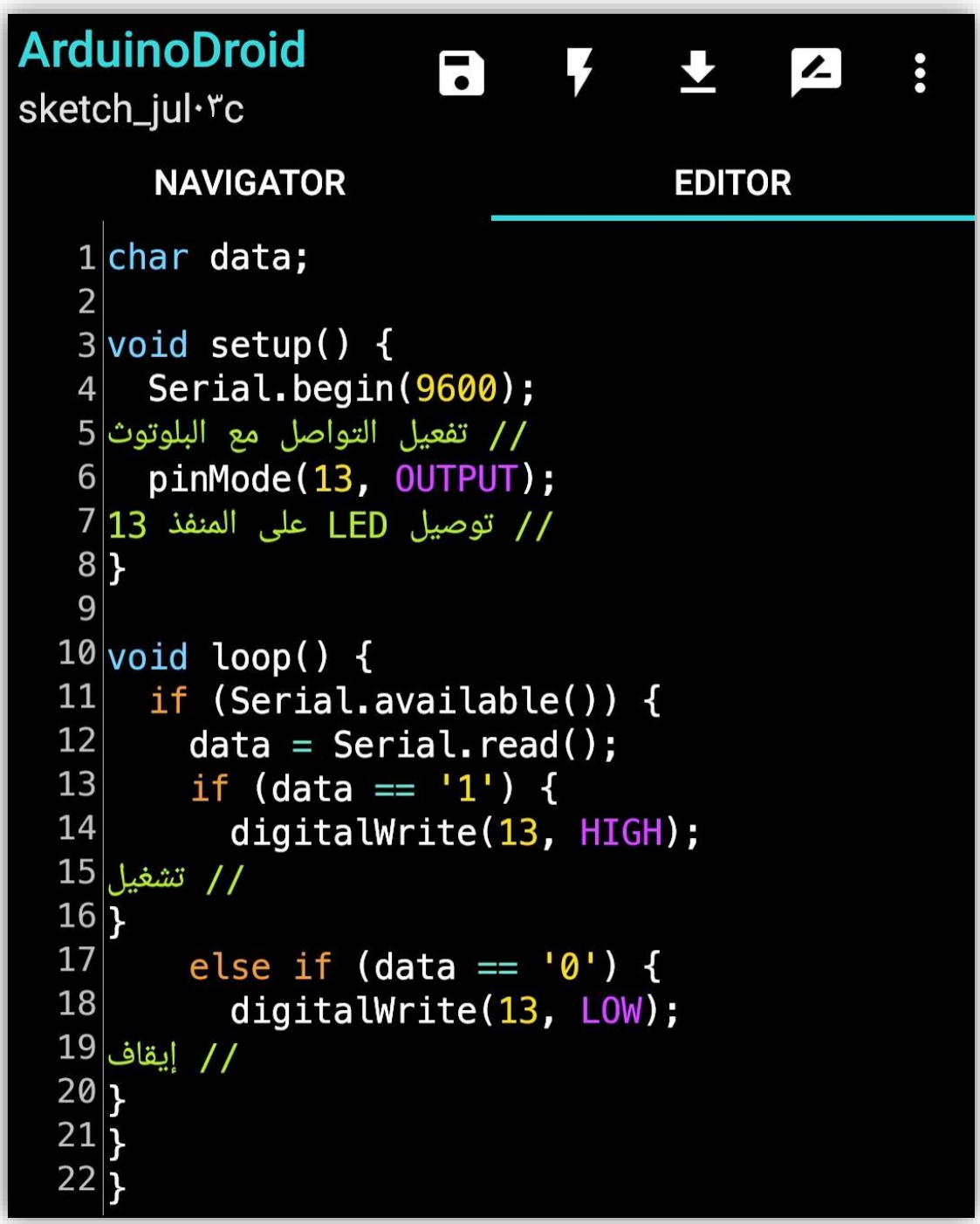
١ كيلو أوم و٢ كيلو أوم:

- صِل الـTX من الأردوينو إلى الطرف الأول للمقاومة K1

- بعدين صِل المقاومة K2 بين نقطة المنتصف وGND

- ومن نقطة المنتصف نروح إلى RXD في البلوتوث

الكود الأساسي (مثال بسيط للتشغيل والإيقاف):



على تطبيق البلوتوث بالموبايل:

- اختار "connect" واختر وحدة HC-05 (الرمز عادةً ١٢٣٤ أو ٠٠٠٠).
- أرسل "١" لتشغيل اللمبة، و"٠" لإطفائها.

تقدر تحمّل تطبيق Arduino Bluetooth Controller من متجر Google Play من خلال الرابط التالي :👇

(https://play.google.com/store/apps/details?id=com.giristudio.hc05.bluetooth.arduino.control&hl=ar)[1]

التطبيق بيسمحلك تتحكم في مشاريع الأردوينو عن بُعد باستخدام البلوتوث، وبيوفر أوضاع متعددة زي:

- Gamepad للتحكم في الروبوتات.
- Terminal لإرسال واستقبال الأوامر.
- Voice Control للتحكم بالصوت.
- وأزرار قابلة للتخصيص لتشغيل/إيقاف الأجهزة.

"التعامل مع الإنترنت باستخدام الأردوينو والواي فاي"

أولًا: إيه الفكرة؟

الفكرة إننا نوصّل لوحة أردوينو بالإنترنت (Wi-Fi) عشان:

- نستقبل بيانات من الإنترنت (زي حالة الطقس).

- نرسل بيانات (زي قراءة حساس) إلى موقع أو تطبيق.

- نتحكم في أجهزة عن بعد (زي لمبة أو مروحة).

ثانيًا: إيه اللي هنحتاجه؟

لوحة أردوينو عادية (Uno مثلًا) ما فيهاش واي فاي مدمج، فبنحتاج وحدة إضافية:

المكونات المطلوبة:

١. لوحة Arduino Uno

٢. وحدة Wi-Fi ESP8266 أو ESP01

(أو نستخدم لوحة فيها Wi-Fi مدمج زي NodeMCU

أو ESP32)

٣. أسلاك (Jumpers)

٤. حساس (اختياري)

٥. موبايل أو كمبيوتر لاتصال الإنترنت

ثالثًا أشهر سيناريو: استخدام ESP8266 (كعميل إنترنت)

توصيل الوحدة:

لو بتستخدم ESP8266 كقطعة مستقلة:

- **VCC → 3.3V**
- **GND → GND**
- **TX → RX**

(عبر محول جهد أو على NodeMCU)

- **RX → TX**

أو لو بتستخدم NodeMCU/ESP32 فمش محتاج لوحة أردوينو خالص، تبرمجها مباشرةً.

تحميل المكتبة:

من Arduino IDE، روح لـ 👇

Tools → Manage Libraries

واكتب:

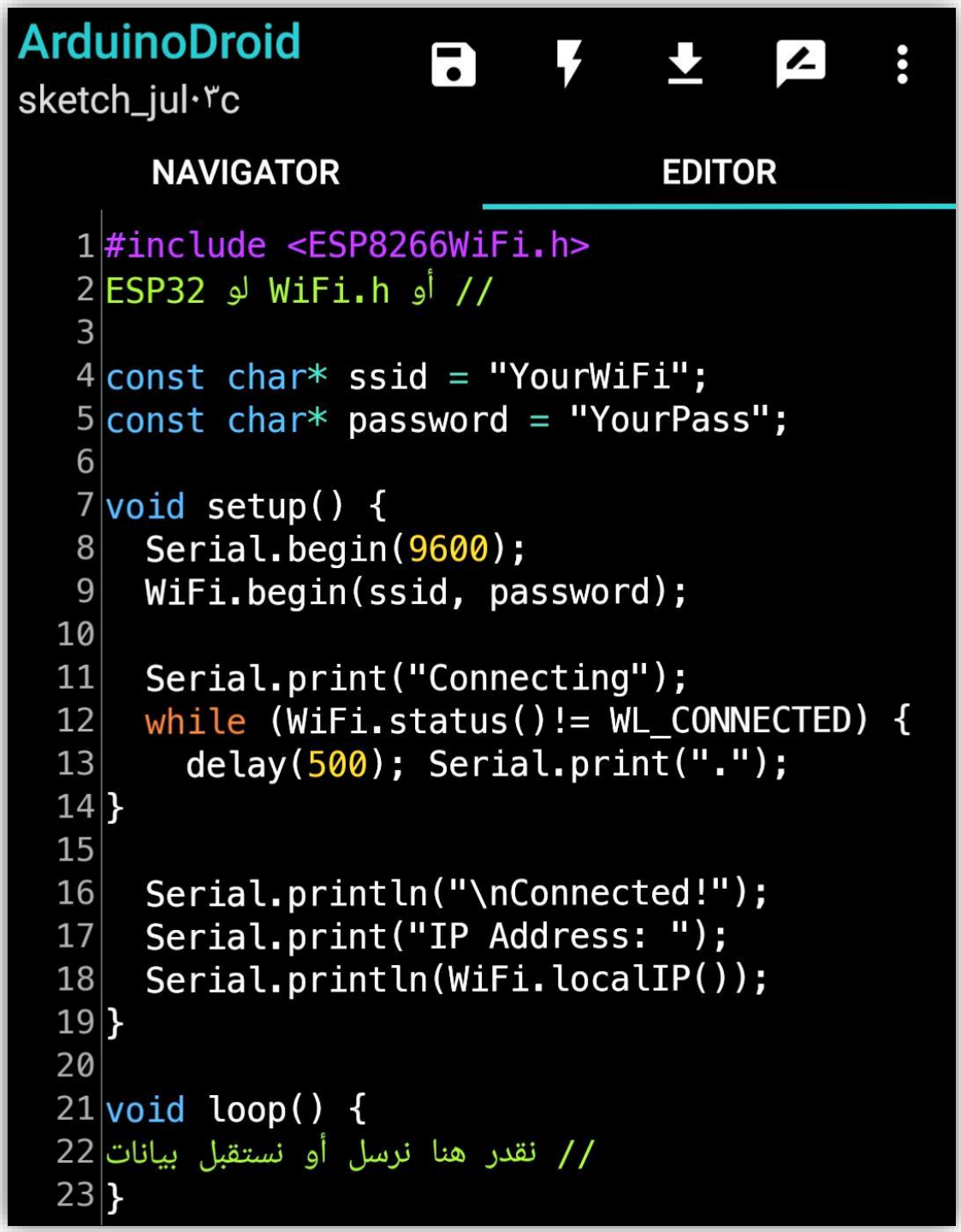
- **ESP8266WiFi**

علشان توصل بالشبكة

- **WiFiClient**

علشان تتعامل مع مواقع أو APIs

مثال برمجي: الاتصال بالواي فاي وطباعة IP



الخطوة الجاية إرسال بيانات من الأردوينو إلى موقع:

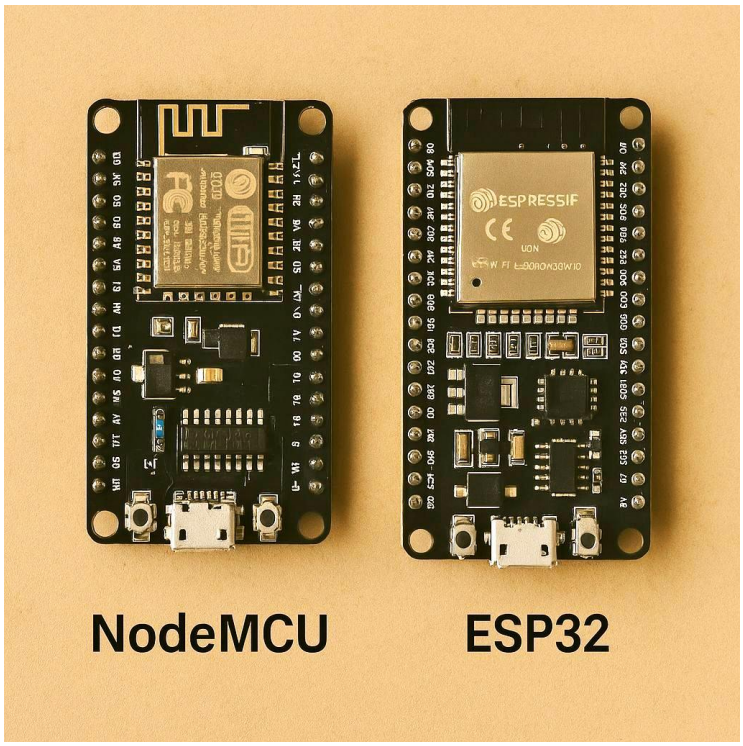
ممكن نستخدم:

- خدمة ThingSpeak (لرفع بيانات الحساسات).
- أو أي API (زي telegram bot أو server خاص).

توضيحات مهمة:

- الإنترنت ما بيشتغلش إلا بعد الاتصال الناجح بالشبكة.
- لازم ننتبه لفولت وحدة الواي فاي (غالبًا V3.3).
- التعامل مع الإنترنت بيحتاج شوية صبر للتجربة لأن الشبكة أحيانًا بتأثر.

وحدة ESP-01



لوحات بها واي فاي مدمج

مشروع المنزل الذكى – التحكم فى الإضاءة والمراوح

أولًا: فكرة المشروع

الهدف إننا نتحكم عن بُعد في:

- إضاءة المنزل (LEDS)

- مراوح أو أجهزة كهربائية

باستخدام تطبيق على الموبايل + وحدة بلوتوث HC-05 + أردوينو.

ثانيًا: المكونات المطلوبة

١. لوحة Arduino Uno

٢. وحدة Bluetooth HC-05

٣. ريليه Relay Module 5V لتشغيل المراوح/مصابيح V220 بأمان

٤. LED (لو هنجرب على صغير قبل التوصيل العملي)

٥. ترانزستور (اختياري) لو هتتحكم في أكثر من مخرج

٦. مقاومات (لـ LED فقط)

٧. أسلاك توصيل

٨. مصدر طاقة مناسب (USB أو بطارية V9)

٩. تطبيق بلوتوث على الموبايل زي 👇

(Arduino Bluetooth Controller)

ثالثًا فكرة التوصيل:

التوصيل الأساسي:

وحدة بلوتوث HC-05 يوصل كالآتي:

- **VCC → 5V**
- **GND → GND**
- **TX → RX**
- **RX → TX**

(في الأردوينو مع مجزئ جهد)

- ريليه:

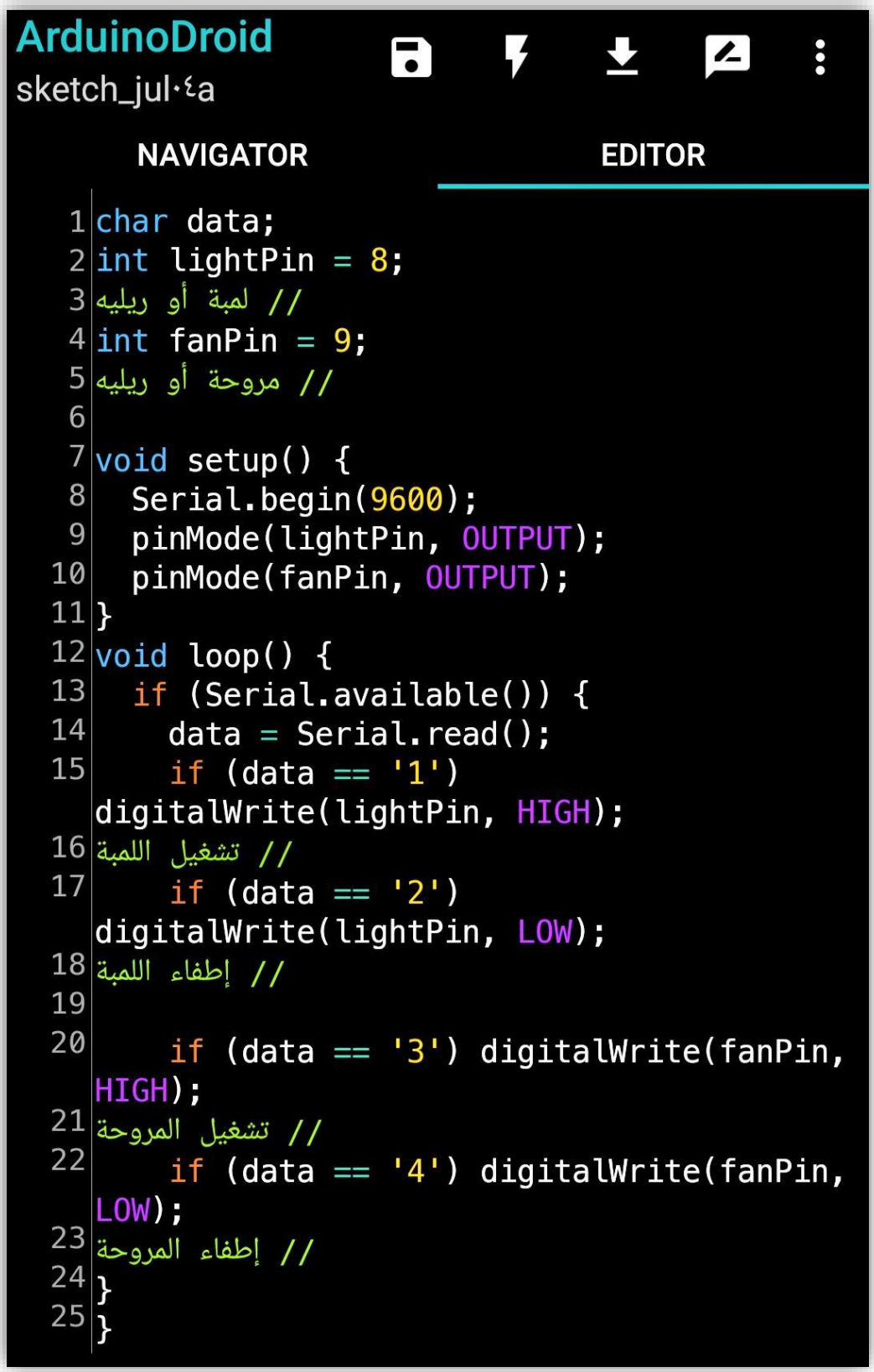
IN1 من ريليه

منفذ رقمي في أردوينو (مثلًا D8)

VCC, GND → 5V, GND

أطراف الريليه الأخرى تتوصل بمفتاح كهربائي أو مروحة

رابعًا: الكود الأساسي للتحكم في ضوء ومروحة




```
char data;
int lightPin = 8;
لمبة أو ريليه //
int fanPin = 9;
مروحة أو ريليه //

void setup() {
  Serial.begin(9600);
  pinMode(lightPin, OUTPUT);
  pinMode(fanPin, OUTPUT);
}
void loop() {
  if (Serial.available()) {
    data = Serial.read();
    if (data == '1')
digitalWrite(lightPin, HIGH);
تشغيل اللمبة //
    if (data == '2')
digitalWrite(lightPin, LOW);
إطفاء اللمبة //

    if (data == '3') digitalWrite(fanPin,
HIGH);
تشغيل المروحة //
    if (data == '4') digitalWrite(fanPin,
LOW);
إطفاء المروحة //
}
}
```

خامسًا استخدام الموبايل:

من تطبيق البلوتوث:

- اربط مع وحدة HC-05 (الكود: ١٢٣٤ أو ٠٠٠٠)
- أرسل:
  - "١" ← تشغيل الضوء
  - "٢" ← إطفاء الضوء
  - "٣" ← تشغيل المروحة
  - "٤" ← إطفاء المروحة

نصيحة:

- ابدأ بتجربة LED بدل ريليه لتجربة آمنة.

ريليه

"أتمنى أن يكون هذا الشرح البسيط وافيًا لكم، وأن يكون بداية مشوّقة لاكتشاف عالم الإلكترونيات والبرمجة، لا تترددوا في التجربة، فالإبداع يبدأ بخطوة."

ARDUINO

The Arduino platform is an open-source electronics ecosystem based on easy-to-use hardware and software. Arduino boards are able to read inputs – a light on a sensor, a finger on a button, or a Twitter message – and turn it into an output – activating a mtor, turning on an LED, publishing something on-line. You can tell your board what to do by sending a set of instructions to the micro-controller ona the board.