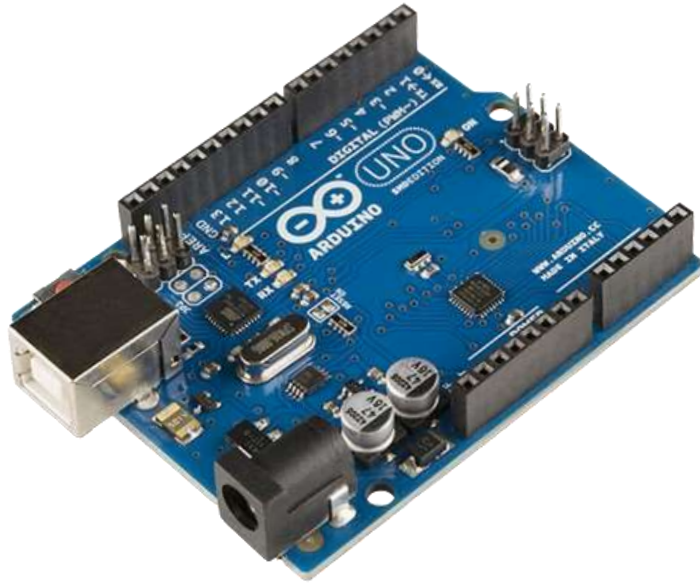


# اردوينو Arduino



Prepared by: Qossay Abu Rida  
IEEE Robotics and Automation Society

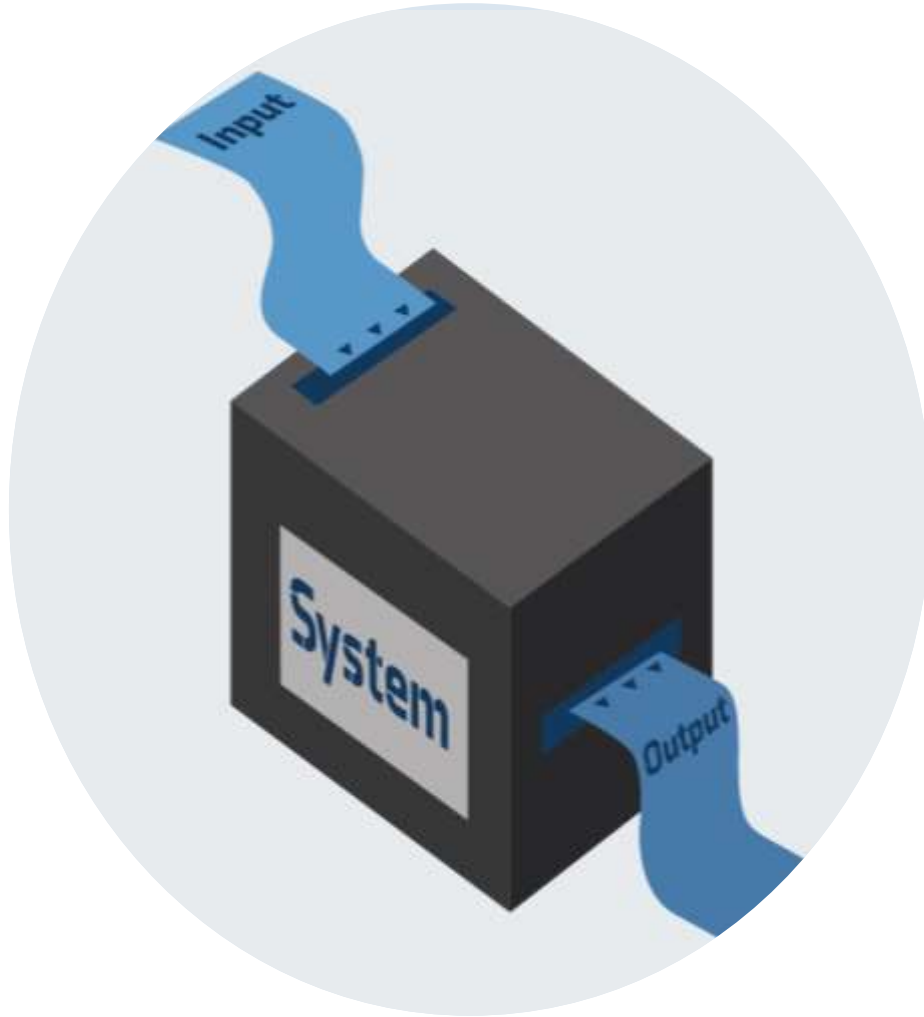
# من نحن؟! (Who Are We?!)

**نحن — IEEE Robotics and Automation Society (RAS)** مجتمع عالمي رائد تحت مظلة **IEEE**، نكرس جهودنا لدفع عجلة الابتكار في مجال الروبوتات والأتمتة. كرئيس لهذا المجتمع، أهدف مع زملائي الطلاب إلى تعزيز المعرفة، وتسهيل التعاون بين الطلاب والجامعات، وإلهام الجيل القادم من رواد هذا المجال.

## لماذا الأردوينو؟

كجزء من التزامنا بالتعليم العملي، نرى في الأردوينو أداة قوية لتبسيط مفاهيم الإلكترونيات، البرمجة، والأتمتة، مما يجعلها مثالية للمبتدئين والخبراء على حد سواء.





**تعريف بالنظام System**



# النظام (System)

النظام (System) هو مجموعة من الأجزاء تعمل معًا لتحقيق هدف.

كل نظام يحتاج إلى:

• مدخلات Inputs :

المعلومات أو الإشارات التي يستقبلها النظام.

• معالجة Processing :

ما يقوم به النظام بناءً على المدخلات.

• مخرجات Outputs :

النتيجة أو الاستجابة التي يعطيها النظام.

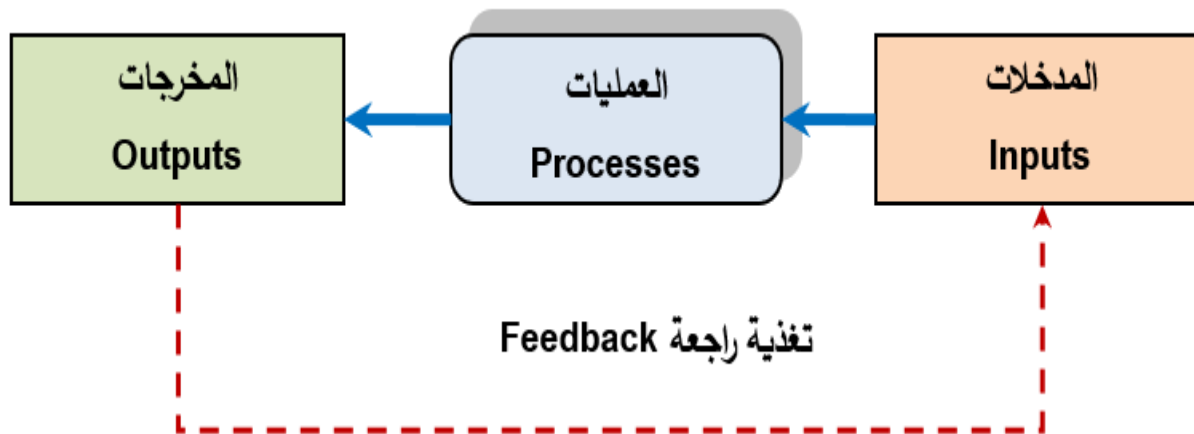
مثال:

• نظام الإشارة المرورية

• المدخل: مستشعرات حركة السيارات

• المعالجة: هل عدد السيارات كبير؟

• المخرج: تغيير لون الإشارة



# المدخلات (Inputs)

ما هي المدخلات؟

هي كل ما يستقبله النظام من معلومات أو إشارات من البيئة المحيطة.

أمثلة على المدخلات:

الضغط على زر

قراءة درجة حرارة من حساس

الإضاءة التي تراها العين الإلكترونية

صوت يلتقط بالميكروفون



في Arduino

نستخدم الحساسات Sensors

مثل حساس الحرارة أو الضوء أو المسافة كمدخلات.

# المعالجة (Processing)

ما هي المعالجة؟

هي ما يقوم به عقل النظام لتحليل المعلومات واتخاذ القرار المناسب.

أمثلة على المعالجة:

إذا كانت درجة الحرارة أعلى من 30 → شغل المروحة.

إذا كان الضوء منخفضاً → شغل المصباح.

في Arduino

المعالجة تتم داخل المتحكم Microcontroller

باستخدام البرمجة Code



# المخرجات (Outputs)

ما هي المخرجات؟  
هي الاستجابة أو النتيجة التي يعطيها النظام بعد المعالجة.

أمثلة على المخرجات:

تشغيل LED

إصدار صوت من جرس

تحريك محرك

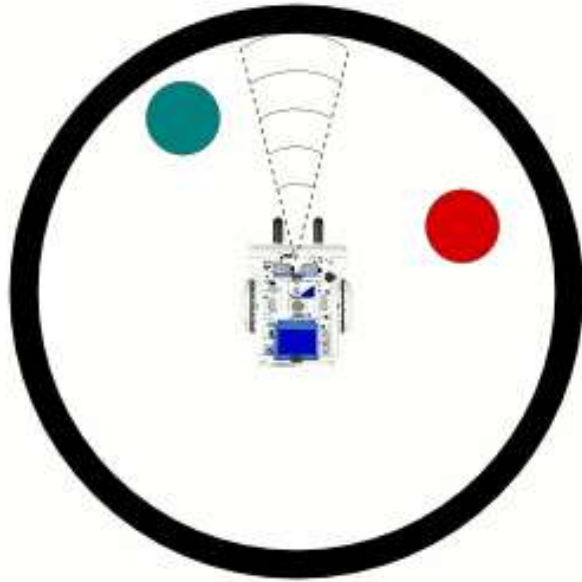
عرض نص على شاشة

في Arduino

نرسل أوامر لتشغيل أشياء مثل الأضواء، المحركات، أو الشاشات.



# التغذية الراجعة (Feedback)



ما هي التغذية الراجعة؟  
هي عندما يستخدم النظام المخرجات كمدخلات جديدة لتحسين الأداء أو اتخاذ قرارات جديدة.

مثال بسيط:  
روبوت يتحرك للأمام، ويستخدم حساس المسافة.  
إذا اقترب من جدار، يغير الاتجاه.  
هنا، المسافة (المدخل) تتغير حسب الحركة (المخرج).

في Arduino  
نستخدم التغذية الراجعة لجعل النظام يتفاعل باستمرار مع البيئة.





# تعريف باء Arduino

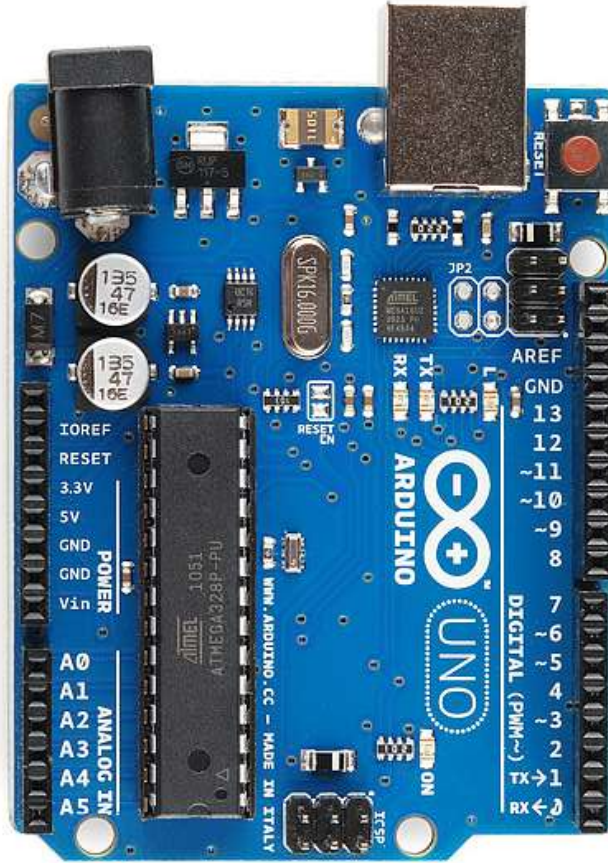


# Arduino

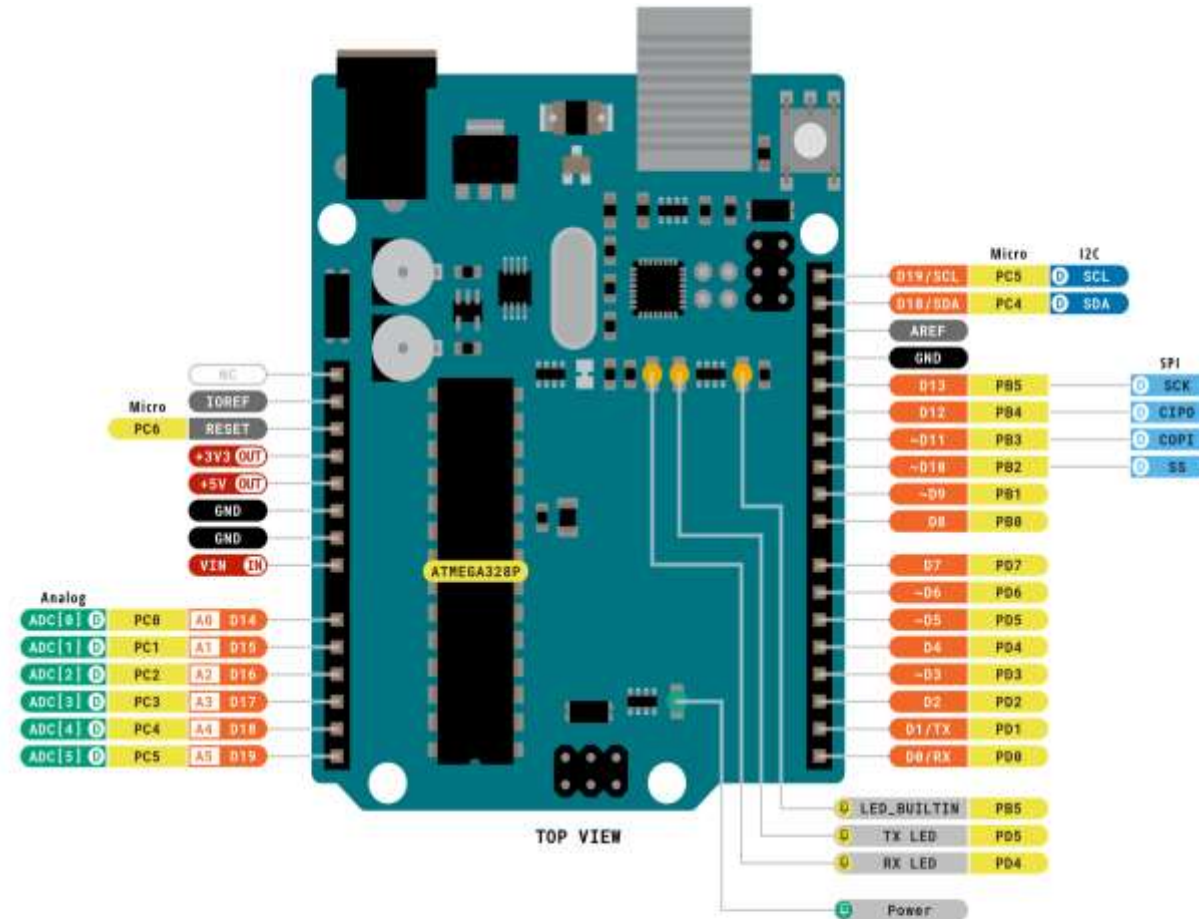
هو لوحة إلكترونية ذكية. تسمح لنا بربط العالم الحقيقي بالكمبيوتر عن طريق البرمجة. نستطيع استخدامها لقراءة معلومات من الحساسات، ثم تنفيذ أوامر مثل تشغيل مصباح أو تحريك محرك.

## مكونات لوحة Arduino

- Microcontroller** = العقل الذي يعالج الأوامر.
- Pins** = مداخل ومخارج لتوصيل الأجهزة.
- USB Port** = لتوصيل Arduino بالكمبيوتر.
- Power Port** = لتوصيل الطاقة الكهربائية.



# Arduino Uno board



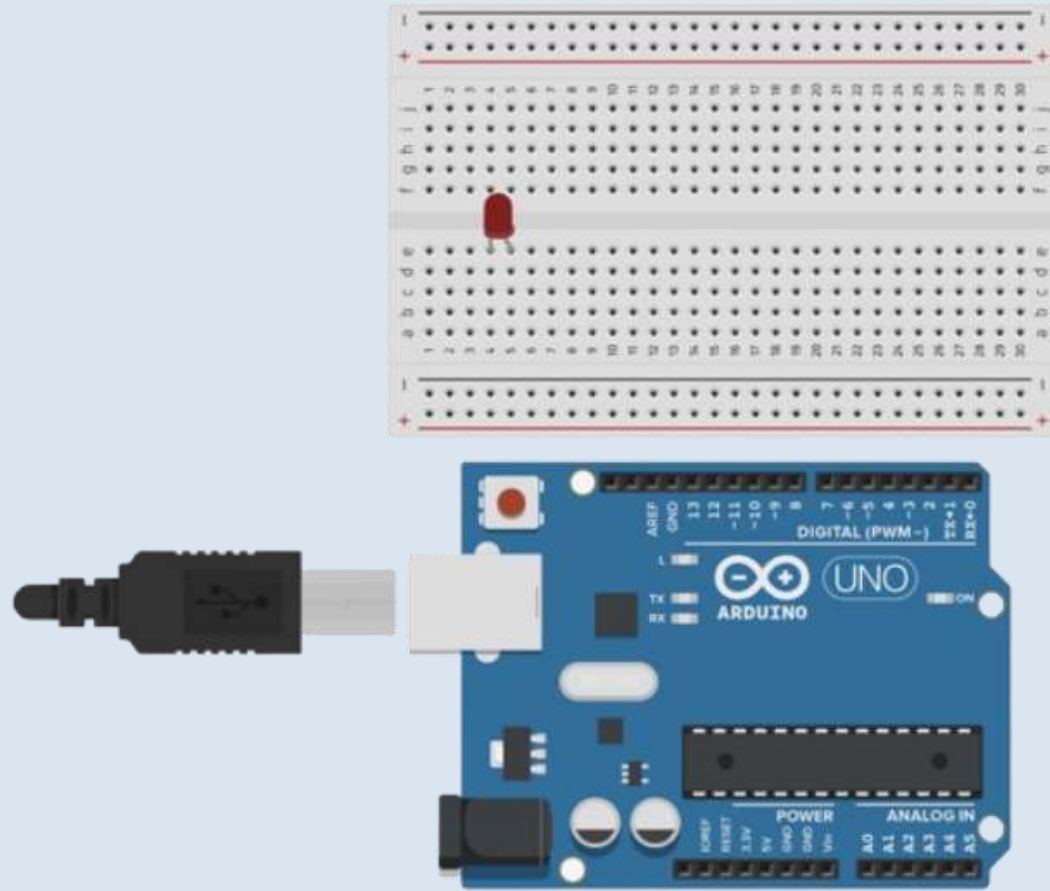
Legend:	Digital	I2C
Power	Analog	SPI
Ground	Main Part	Analog



ARDUINO

ARDUINO UNO REV3  
SKU code: A000066  
Pinout  
Last update: 6 Oct, 2022

# Example:





# تعريف بالـ Programming



# Arduino Programming

```
void setup() {  
  
}  
  
void loop() {  
  
}
```

◆ البرمجة هي كتابة أوامر بلغة يفهمها الكمبيوتر أو Arduino  
◆ نقول له ماذا يفعل خطوة بخطوة.  
◆ في Arduino نستخدم لغة قريبة من C++

• كل مشروع يبدأ بدالتين رئيسيتين:  
◆ تجهيز الأشياء قبل البدء `setup()`  
◆ الأوامر التي تتكرر دائماً `loop()`

# أنواع البيانات (Data Types)



```
int age = 15;  
float temp = 23.5;  
bool lightOn = true;  
char grade = 'A';  
String name = "Ali";
```

◆ (int): عدد صحيح (مثال: 5 أو -2)

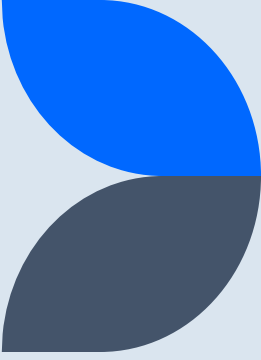
◆ (float): عدد عشري (مثال: 3.14)

◆ (bool): صحيح أو خطأ (true/false)

◆ (char): حرف واحد مثال: 'A'

◆ (String): نص طويل مثال: "Hello"

# المتغيرات (Variables)



```
int speed = 100;
```

```
speed = 120; // تغيير القيمة لاحقًا
```

◆ المتغير هو مكان نخزن فيه معلومة مؤقتة.

◆ يتكون من:

- 1- نوع البيانات
- 2- اسم المتغير
- 3- القيمة



# الجمل الشرطية (if statements)

◆ نستخدم **if** لاتخاذ قرار بناءً على شرط.

```
if (temp > 30) {  
    // حار  
} else if (temp < 10) {  
    // بارد  
} else {  
    // معتدل  
}
```

```
if (temperature > 30) {  
    // تشغيل المروحة  
}
```

# الحلقات (Loops)

◆ الحلقات تساعدنا على تكرار الأوامر.



```
for (int i = 0; i < 5; i++) {  
    // action  
}
```



```
while (sensorValue < 100) {  
    // استمر في القياس  
}
```

# نصائح مهمة عند البرمجة

✓ تأكد من كتابة الكود بدون أخطاء لغوية.

✓ تأكد من فتح وإغلاق الأقواس { } بشكل صحيح.

✓ تأكد من كتابة الفواصل المنقوطة ; بعد كل أمر.

✓ راجع الكود بهدوء إذا ظهرت أخطاء.

# دوال مهمة لـ Arduino



# pinMode



```
pinMode(12, OUTPUT); // اجعل المخرج 12 يرسل إشارات  
pinMode(13, INPUT); // اجعل المدخل 13 يستقبل إشارات
```

**pinMode(pin, mode) ⚡**

⚡ لتحديد وظيفة المخرج أو المدخل

# digitalWrite



```
digitalWrite(13, HIGH); // شغل الـ LED
```

```
digitalWrite(13, LOW);  // أطفئ الـ LED
```

**digitalWrite(pin, value)** ⚡

⚡ لإرسال إشارة كهربائية (تشغيل أو إيقاف).

# digitalRead



```
int buttonState = digitalRead(2); // قراءة حالة الزر
```

**digitalRead(pin)** ⚡

⚡ لقراءة قيمة مدخل (زر مثلاً).

# delay



```
delay(1000); // انتظر ثانية واحدة
```

**delay(milliseconds)** ⚡

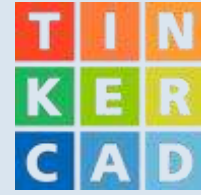
⚡ لإيقاف تنفيذ الأوامر مؤقتًا لعدد معين من الميلي ثانية.



# محاكاة الـ Arduino



# Tinkercad



<https://www.tinkercad.com/>

