

Microcontrollers and Interfacing

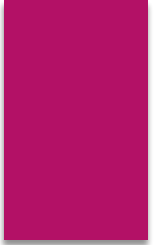
What are microcontrollers?

Microcontroller is a small computer on a single circuit, which may include one or more processors, RAM, ROM, Input/Output Peripherals, etc.

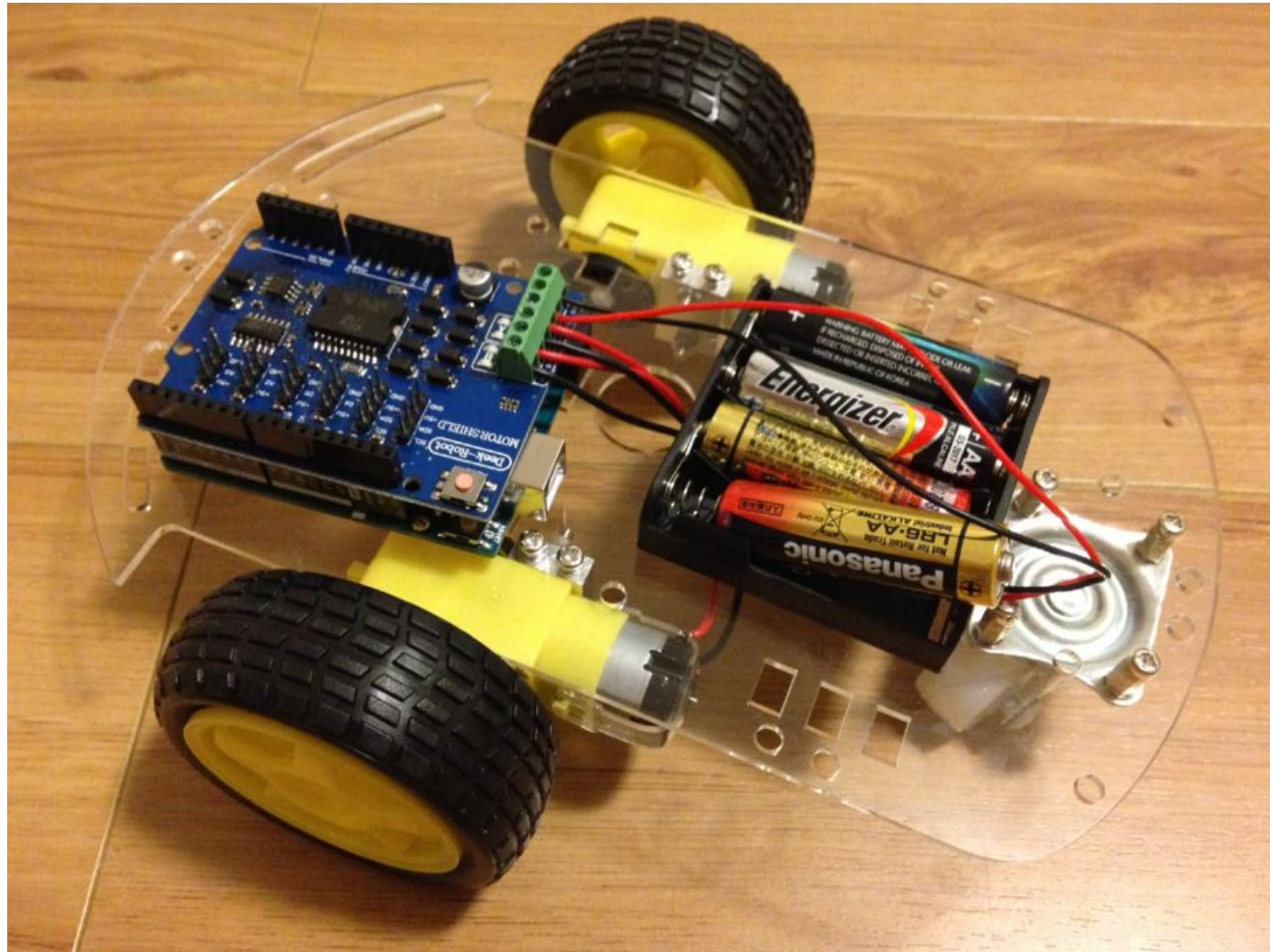
okay....



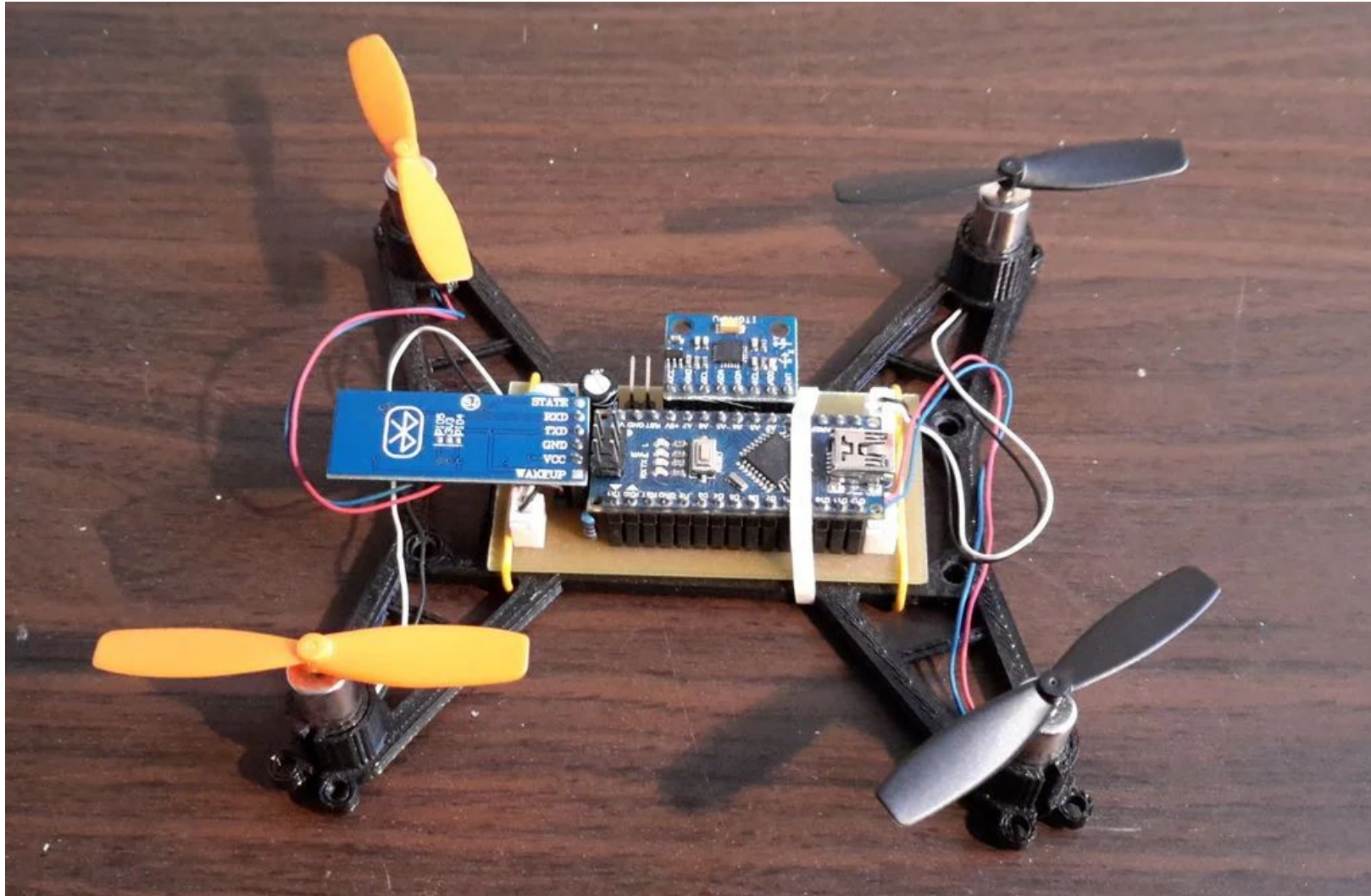
Why do need a
microcontroller?

- 
- Desktop computer you are using is a "general purpose computer" that can run any of thousands of programs. Microcontrollers are "special purpose computers." Microcontrollers do one thing well. Meaning, Microcontrollers are **dedicated to one task** and run one specific program. Program is stored in **ROM**.
 - Microcontrollers are often **low-power devices**. They consume very less power.
 - A microcontroller also takes input from the device it is controlling and controls the device by sending signals to different components in the device.

Ground Based Robot



Quadcopter



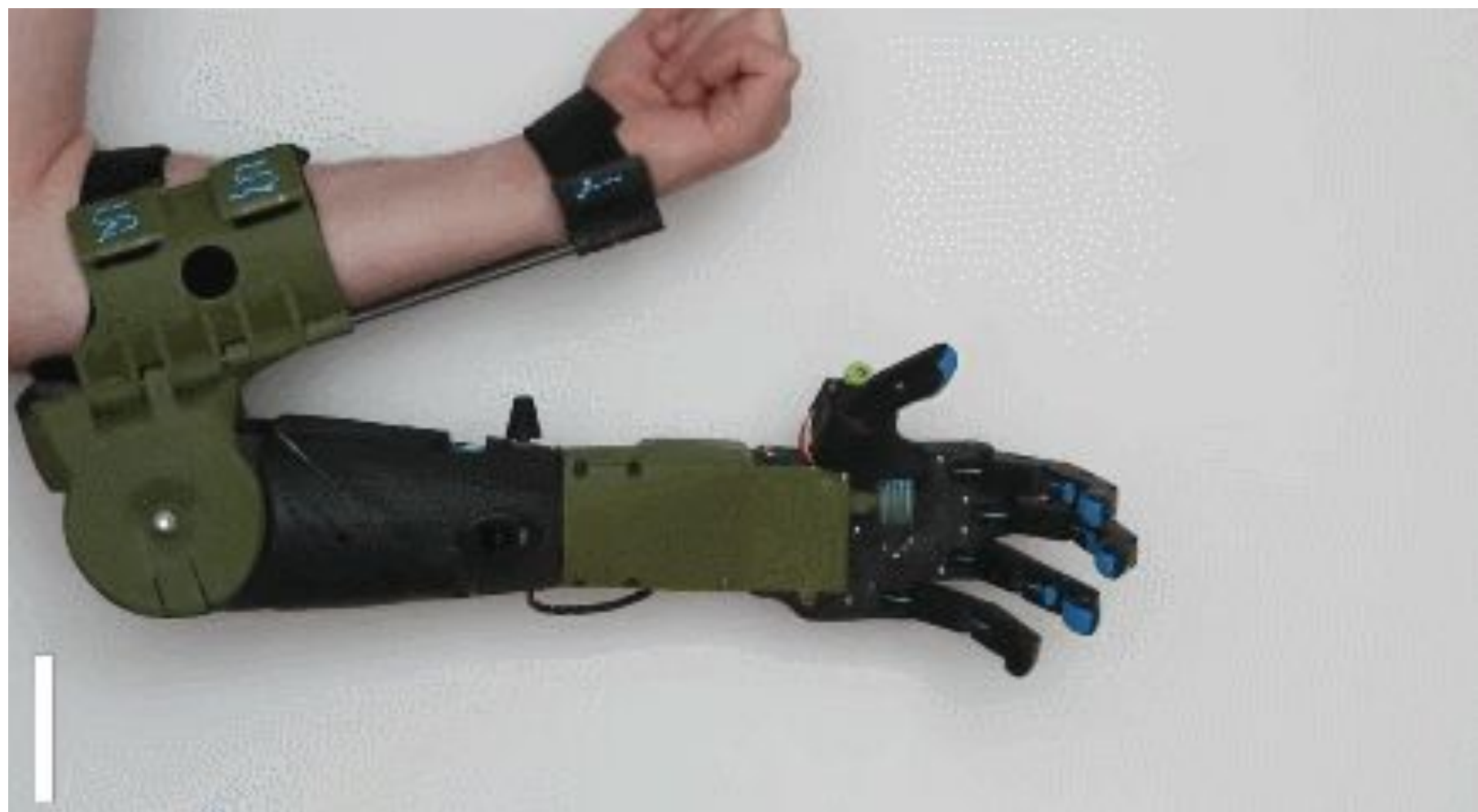
LED Wheel



LED Globe







Arduino

- ▶ **Arduino** is an **open-source** electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.
- ▶ It is a **microcontroller not a microprocessor**

Types of Arduino boards:



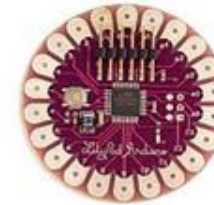
Arduino Uno



Arduino Leonardo



Arduino Mega 2560



?



Arduino Mega ADK



?



Arduino Ethernet



Arduino Pro



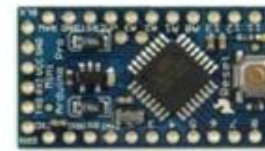
?



Arduino Nano



Arduino Mini



?

Why only Arduino?

- ▶ It's cheap
- ▶ It's open source
- ▶ It's is widely used by DIY hobbyists all over the globe
- ▶ It's easy to use for beginners
- ▶ Many variations that fit each and every demand

Arduino UNO:

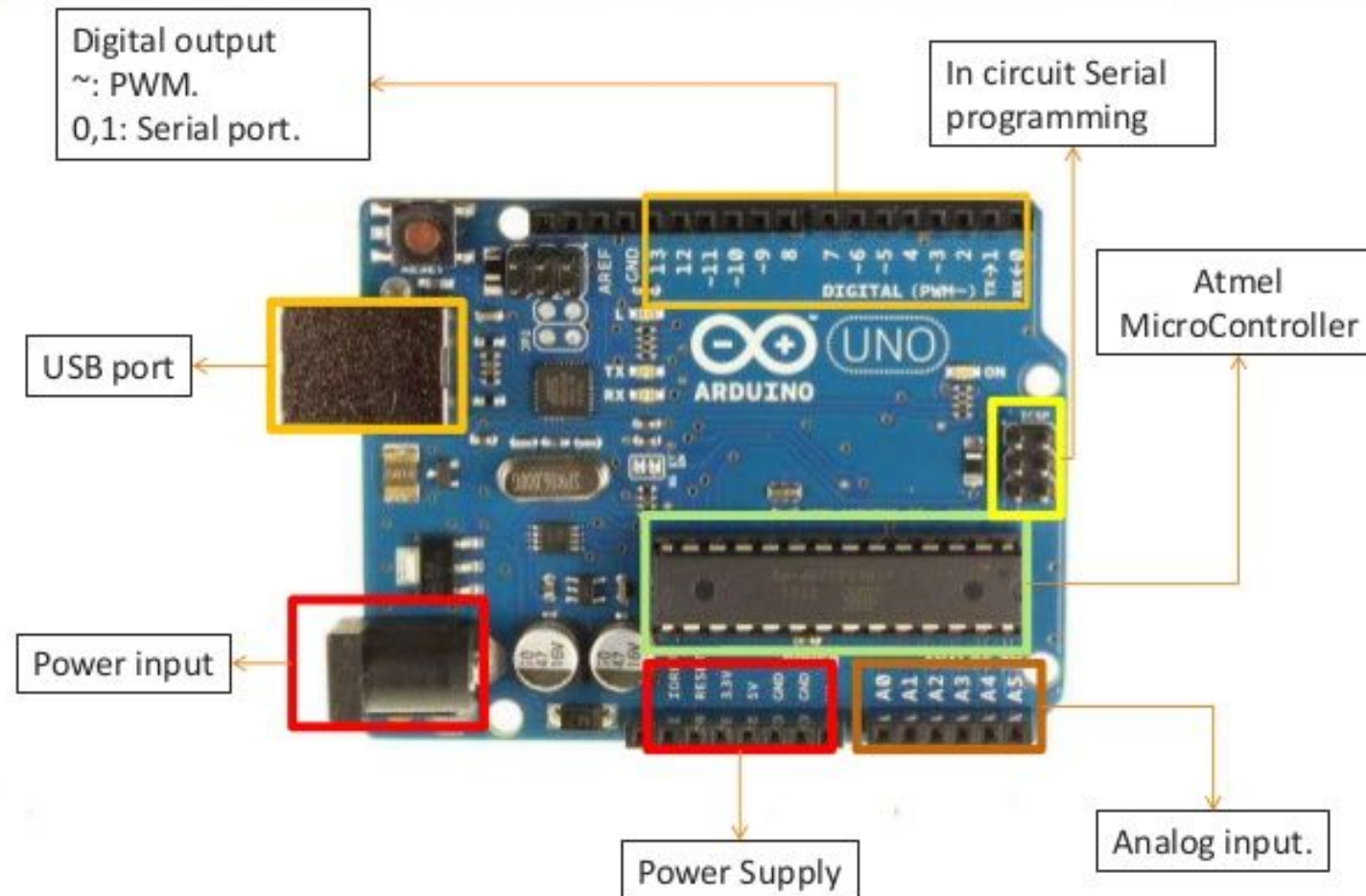
- ▶ The most widely used Arduino development board.
- ▶ No need to solder. Ready to use. Easy to handle and more robust.
- ▶ Uses **Atmega 328P** microcontroller.

Arduino UNO



Arduino hardware

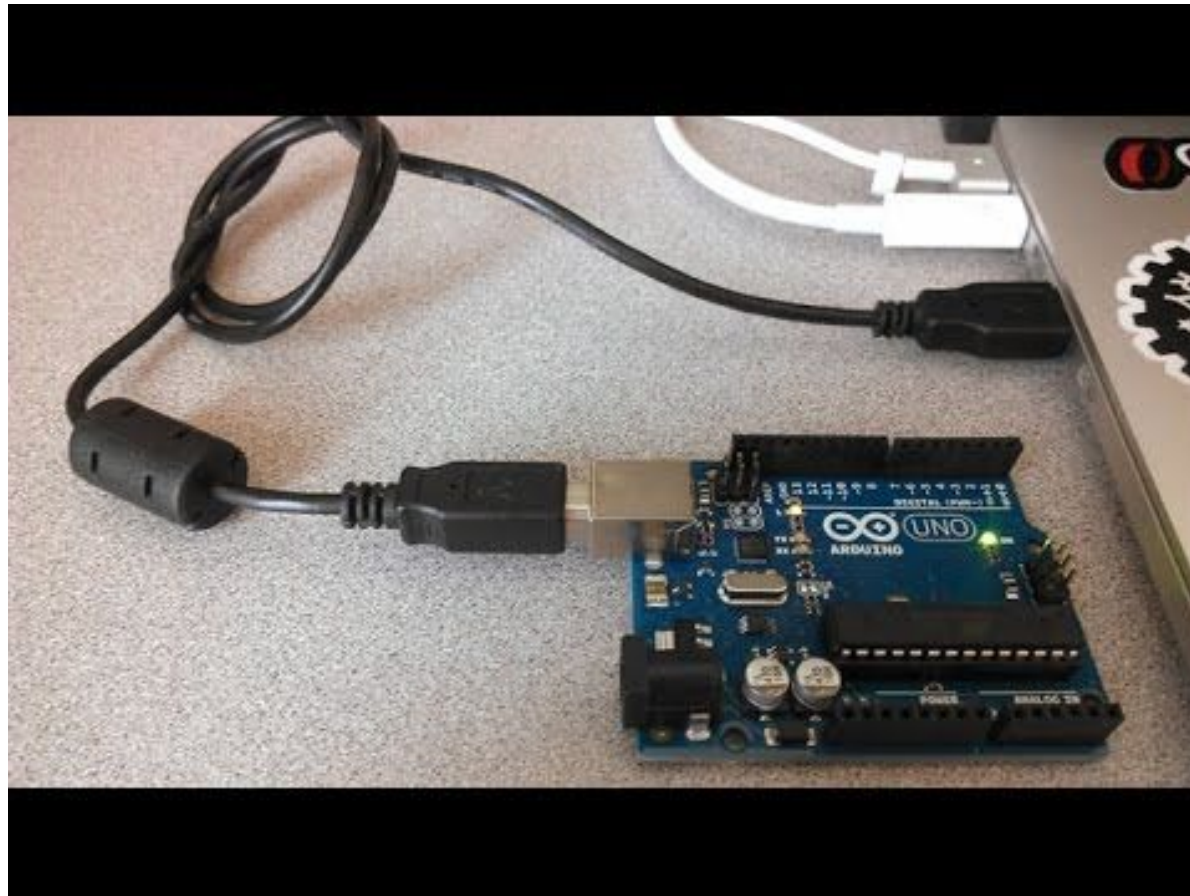
Arduino UNO:



Powering up the Arduino

Three methods to power up the Arduino are :

Through the USB cable



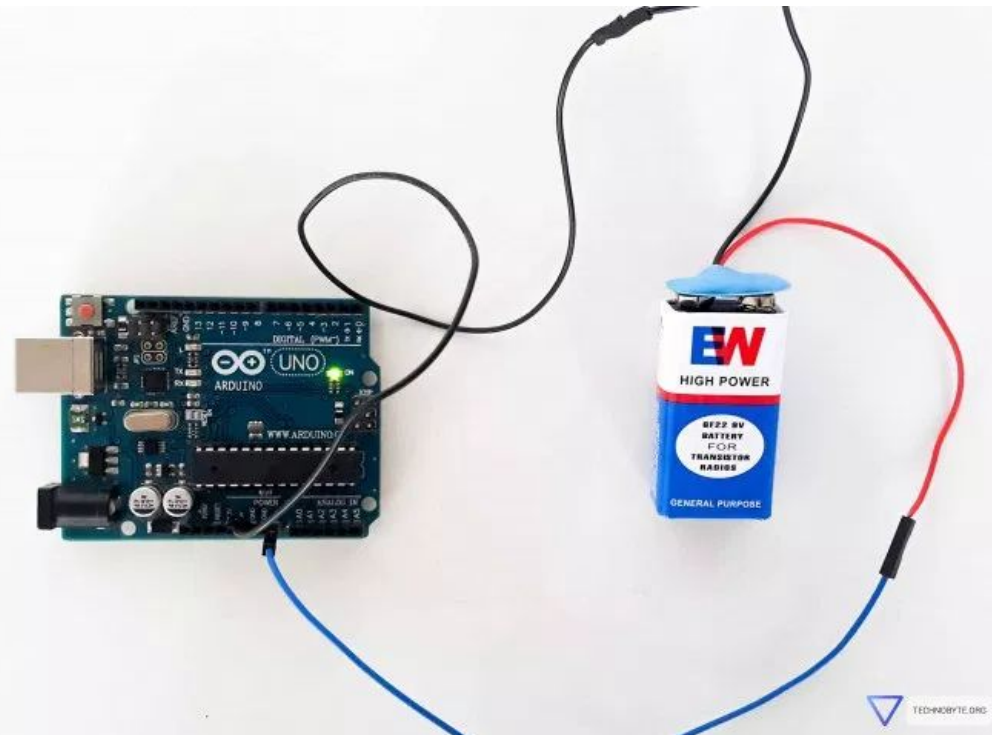
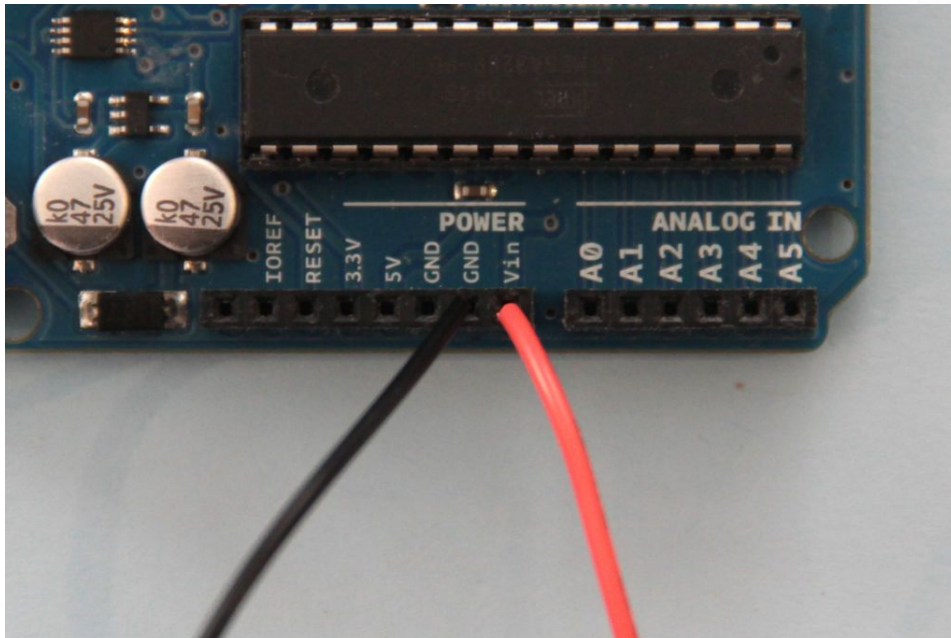
Power up using the barrel connector.

- ▶ The input voltage can be from 7 volts to 20 volts through this barrel connector.



Use the Vin pin

- ▶ To this pin also you can directly provide 7 to 20 volts via the battery or any other power supply.

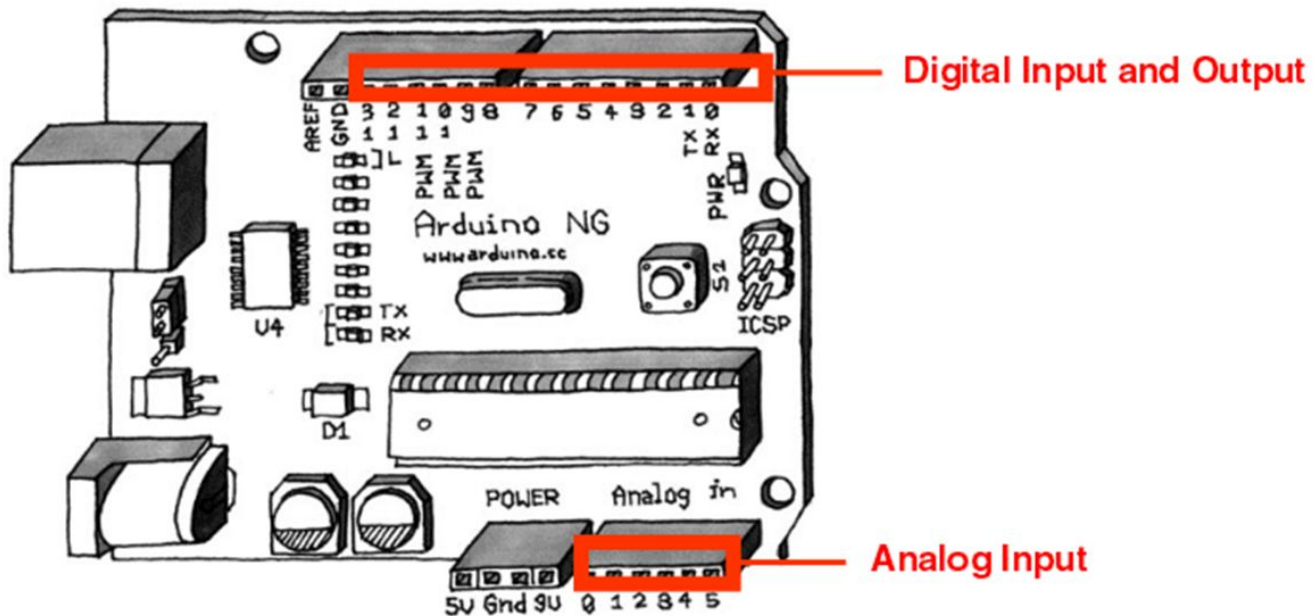


Digital Pins:

- ▶ Can read digital (1 or 0) inputs and even output (1 and 0) values.
- ▶ 1 -> 5 volts DC output
- ▶ 0 -> 0 volts output

Digital and analog pins

Two states (binary signal) vs. multiple states (continuous signal)



PWM pins

- ▶ **Pulse Width Modulation** pins
- ▶ They can output different voltages from 0 to 5 v in **256** steps
- ▶ So $0 \rightarrow 0$ volts
- ▶ And $255 \rightarrow 5$ volts
- ▶ So if you write `Output = 127`, it will output 2.5 volts.
- ▶ There are 6 digital pins that can be used as PWM pins
 - ▶ Pin number $\rightarrow 3, 5, 6, 9, 10, 11$.

Analog Pins

- ▶ They are only input pins
- ▶ The input taken is in 1024 steps.
- ▶ i.e. $0 \rightarrow 0$ volts
- ▶ And $5 \rightarrow 1023$ volts
- ▶ So output read as 511 is actually 2.5 volts.

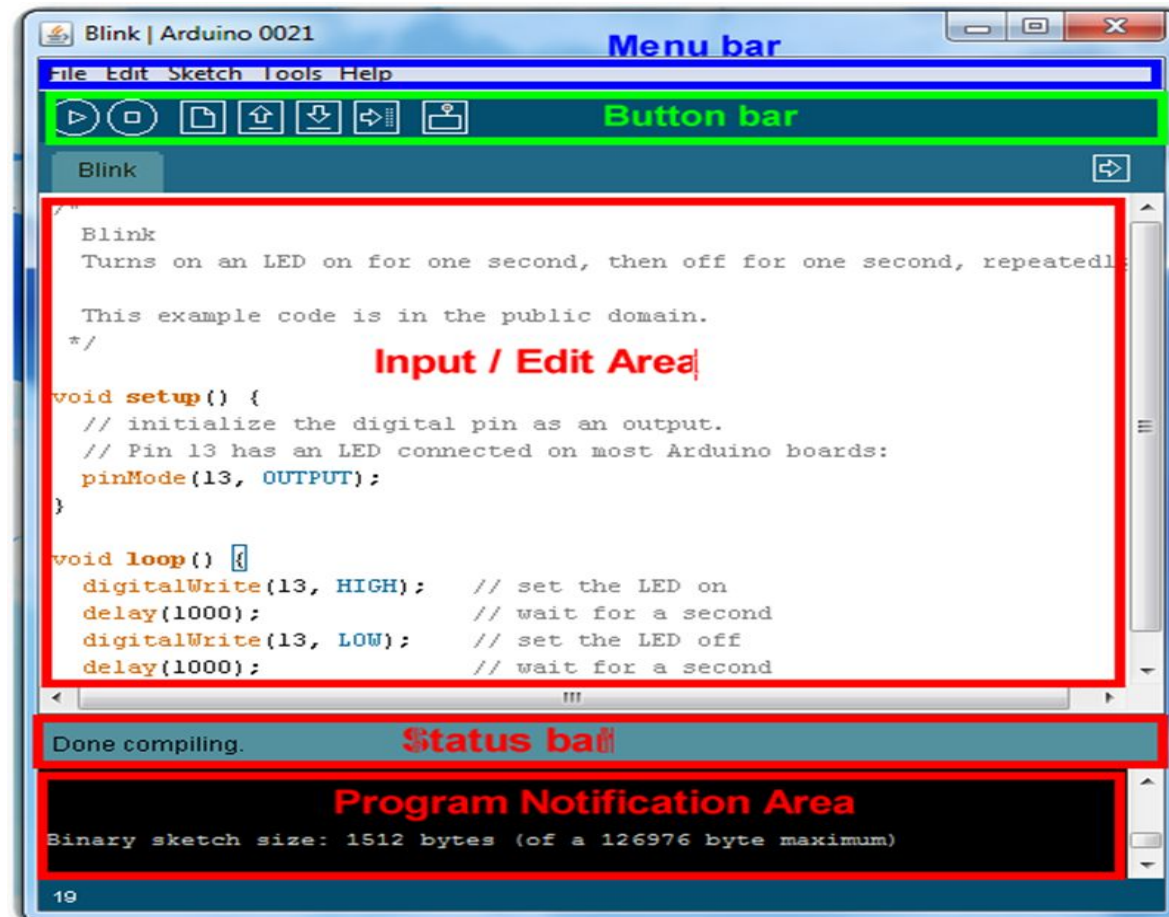
Questions

- ▶ Which pin will you use if you have to get output for an LED and control its brightness
- ▶ Which pin will you use if you have to switch an LED on and off.
- ▶ Which pin will you use if you have to take light sensor's output (it gives output in the range of $0 \rightarrow 5$ volts that linearly depends on the light intensity)

How to program Arduino?

- ▶ It can be programmed via a software called the Arduino IDE.
- ▶ It converts all the code written in c language in the IDE to binary that can be understood by the Arduino's Atmega 328p.
- ▶ It then uploads the code in the Arduino board so that it can run the code every time when the board powers up.

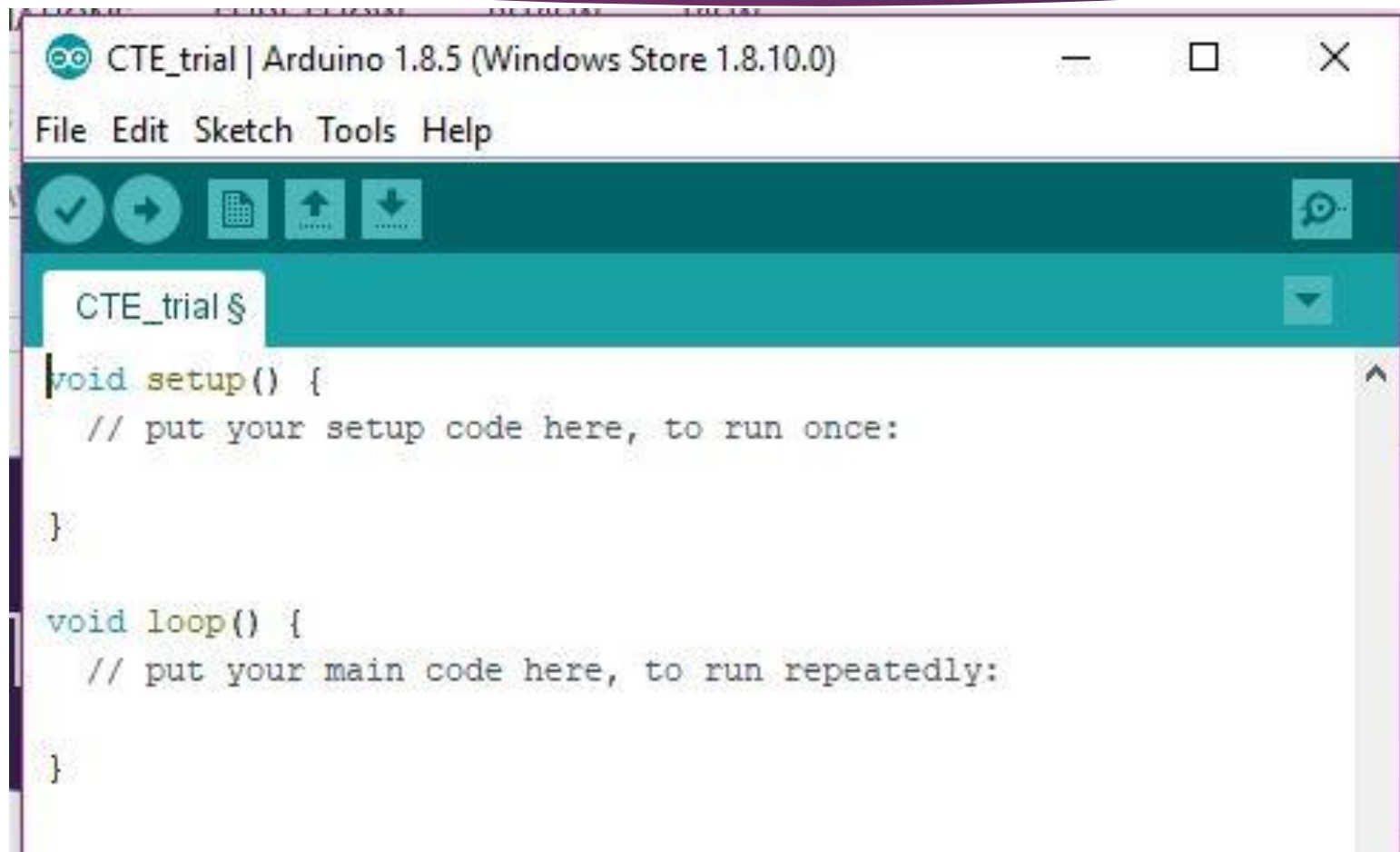
How does the software look like?



Basic terms of programming

- ▶ Syntax
- ▶ Variable
- ▶ Control structure → if
- ▶ Data structure → lists and arrays
- ▶ Compiler
- ▶ Include
- ▶ Functions
- ▶ Libraries

Two main functions



The screenshot shows the Arduino IDE window titled "CTE_trial | Arduino 1.8.5 (Windows Store 1.8.10.0)". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for checking, running, saving, uploading, and downloading. The sketch editor displays the following code:

```
CTE_trial $  
  
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```




```
void setup(){.....}
```

- ▶ It is executed first when the code is run
- ▶ It is executed only once
- ▶ Generally write all the initialization code here. Like making pin number 5 as output pin etc.



```
void loop() {.....}
```

- ▶ After void setup() runs for one time the loop() runs continuously runs for all time.
- ▶ It repeats itself infinite times.
- ▶ Example: if you want to blink LED forever then put the code in this block.

Some Components:

- ▶ Resistor
- ▶ Diode
- ▶ LED
- ▶ Photoresistor

What is a Potentiometer?

- ▶ It is a variable resistor with three terminals.
- ▶ When the extreme 2 terminals are provided power the middle terminal gives output voltage that linearly changes with the rotation of the wheel.



Some basic inbuilt functions

- ▶ There are many basic inbuilt functions in the IDE that help us to do simple functions like lighting up an led or reading the sensor data.
- ▶ We don't have to write a separate function for that we can directly use the inbuilt functions

pinMode()

- ▶ pinMode(pin number, INPUT/OUTPUT);
- ▶ Ex: pinMode(13, OUTPUT);

digitalWrite()

- ▶ digitalWrite(pin number, PUTPUT/INPUT);
- ▶ Ex: digitalWrite(13, OUTPUT);

digitalRead()

- ▶ `digitalRead(pin number);`
- ▶ Ex:
 - ▶ `Int a = 0;`
 - ▶ `a = digitalRead(3);`
 - ▶ `// now a= 1 if HIGH voltage is detected on the pin number 3`
`and LOW if low voltage detected.`

analogRead()

- ▶ `analogRead(pin number);`
- ▶ Ex:
 - ▶ `Int x = 0;`
 - ▶ `X = analogRead(3);`
 - ▶ `// the value read by the A3 pin(the analog pin) will be stored in integer x, and the value can be within 0 to 1023`

analogWrite()

- ▶ `analogWrite(pin number, value);`
- ▶ Can be used only on the PWM pins
- ▶ Ex:
 - ▶ `Int out_val = 210;`
 - ▶ `analogWrite(9, out_val);`
 - ▶ `// gives 4.1176 volts to the 9th pin of the board.`

delay()

- ▶ delay(value in milliseconds)
- ▶ Ex:
 - ▶ digitalWrite(13,1);
 - ▶ Delay(1000);
 - ▶ digitalWrite(13,0);

Question:

What is the maximum delay you can give?

The 'hello world' of Arduino : "blink"

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.5 (Windows Store 1.8.10.0)". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for saving, running, uploading, and downloading. The main text area shows the "Blink" sketch with the following code:

```
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

Task : Blinking 3 LEDs simultaneously

- ▶ You have to blink 3 LEDs connected on the 3 different pins of Arduino.
- ▶ Each led should blink with a time period of 1 second simultaneously.
- ▶ You can choose any 3 pins on the Arduino Uno board to connect your 3 LEDs.

Task : blinking 3 LEDs sequentially.

- ▶ Connect 3 LEDs to suitable 3 pins of Arduino Uno.
- ▶ The first LED should blink for 1 second while the other 2 LEDs stay off. Then for the next second the 2nd Led should stay on and the other 2 LEDs stay off.
- ▶ Similarly the sequence goes on and each led blinks for one second at a time.

Task: blinking LEDs with different time periods.

- ▶ Connect 3 LEDs to suitable 3 pins of Arduino Uno.
- ▶ Each led should blink independently of the other led for respective time periods of 1,2,3 seconds.

Serial Processing

- ▶ Two functions
 - ▶ `Serial.begin(9600);` or `Serial.begin(115200);` or `Serial.begin(baud rate);`
 - ▶ `Serial.print("hello");`
 - ▶ `Serial.print(var);` `// where var is an integer`
 - ▶ `Serial.println("hello");`
 - ▶ `Serial.println(var);`

Ref: <https://www.arduino.cc/reference/en/language/functions/communication/serial/>

Task : LED fading.

- ▶ Connect the led to a terminal of Arduino that can give analog voltage as the output.
- ▶ The write a code that changes the brightness of the led with time.
- ▶ The brightness should first increase form zero to max and then decrease tot zero.
- ▶ This process should repeat infinitely.

Task : Controlling LED using potentiometer.

- ▶ Connect potentiometer such that the output pin gives voltage values of 0 to 5 volts.
- ▶ Connect the output of the potentiometer to a suitable pin on Arduino that can read its output.
- ▶ Based on the input voltage from the potentiometer, give the same voltage as output to an led connected to suitable pin.

Task : Measuring voltage

- ▶ Take input of a potentiometer in any analog pin
- ▶ Read the value on the voltage on that pin
- ▶ Display that voltage on the Serial monitor in volts. Using `Serial.begin()` and `Serial.println()` functions.

References:

<https://electronics.howstuffworks.com/microcontroller1.htm>

<https://create.arduino.cc/projecthub>

<https://learn.adafruit.com/search?q=arduino%20projects>

<https://www.arduino.cc/reference/en/>