

# Let's start with Arduino

Lec\_1

---

---

# Arduino

Arduino is an open-source microcontroller that enables programming and interaction; it is programmed in C/C++ with an Arduino library to allow it to access the hardware (sensors, indicators, motors, and etc...).

Arduino is composed of two major parts:

1. The Arduino board.
2. The Arduino IDE.

# Potential applications

- Real-world monitoring
  - Automated weather station
  - Lightning detection
  - Sun tracking for solar panels
  - Background radiation monitor
  - Automatic wildlife detector
  - Home or business security system
- Small-scale control
  - Small robots
  - Model rockets
  - Model aircrafts
  - Quadrotor UAVs
  - Simple CNCs for small machine tools
- Small-scale Automation
  - Automated greenhouse
  - Automated aquarium
  - Laboratory sample shuttle
  - Precision thermal chamber
  - Automated electronic test system
- Performance Art
  - Dynamic lighting control
  - Dynamic sound control
  - Kinematic structures
  - Audience responsive artwork

# Some Examples for Arduino Applications

**BEST ARDUINO PROJECTS FOR 2023**

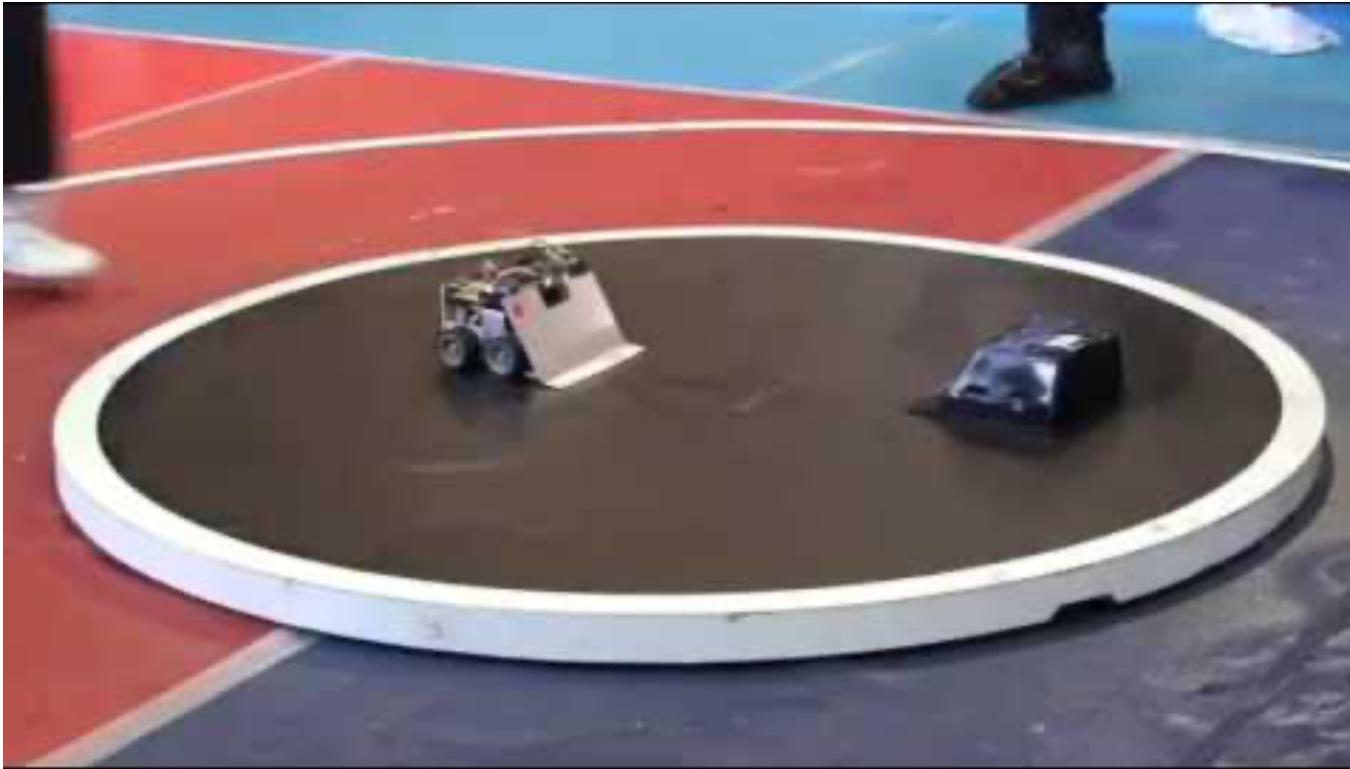
The image displays a grid of eight Arduino project examples, each with a small thumbnail image and a brief description:

- POV DISPLAY**: Shows a circular pattern of concentric lines on a screen, with a smartphone displaying a control interface.
- DIY Motorized Camera Slider**: Features a camera mounted on a mechanical slider mechanism.
- DIY Ultrasonic Mist Maker**: A blue cylindrical device with a red mist or spray effect.
- ESP32 Retro Game Console**: An ESP32 board connected to a screen showing a retro game.
- Track Capacitive PCB Light Panel**: Two glowing blue circular light panels with text on them.
- ARDUINO BLUETOOTH CAR**: A small remote-controlled car with a blue glow around it.

# Line Follower



# Sumo Robot



# Arduino Variants

- Standard: Atmega8/168/328 chip (UNO)
- Mega: Atmega1280/2560 chip (MEGA)

Additional boards can be found on [here](#)



UNO



MEGA

# Arduino UNO

Microcontroller: ATmega328P

14 Digital I/O ports

6 PWM Outputs

6 10-bit analog input ports

UART Interface

I2C Interface

SPI Interface

External Power Supply Range: 5 – 23 V

DC output: 5 V/3.3 V

## UNO Pinout

No Connection  
I/O Reference Voltage for shields

Reset Input

3.3V Output @ 50mA

5V Output or Input

Ground

Ground

7-12V Output or Input

Analog Pin 0 (**A0**)

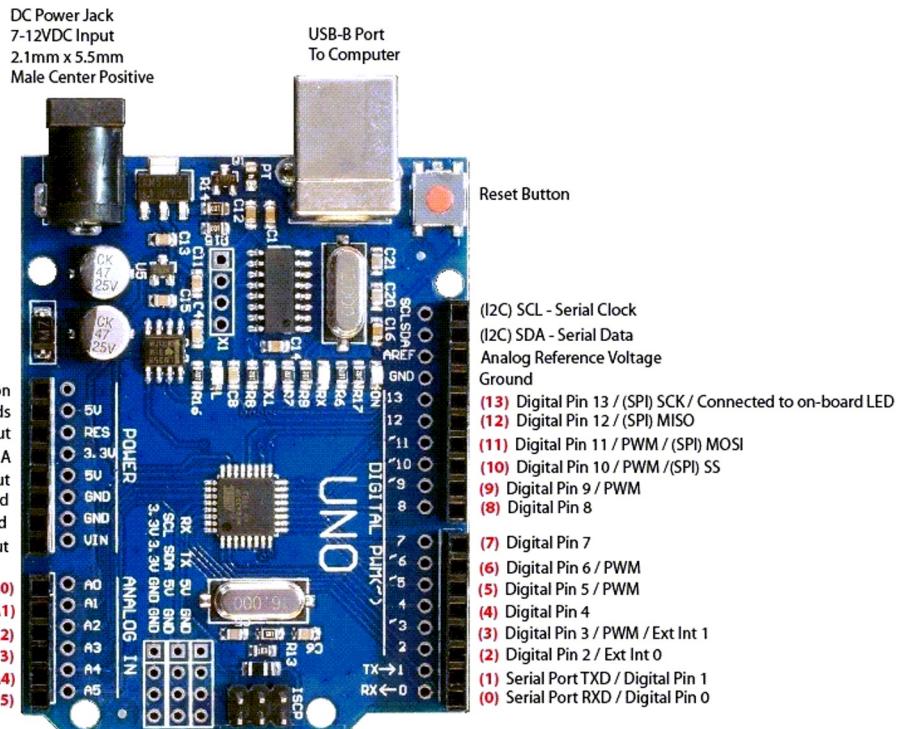
Analog Pin 1 (**A1**)

Analog Pin 2 (**A2**)

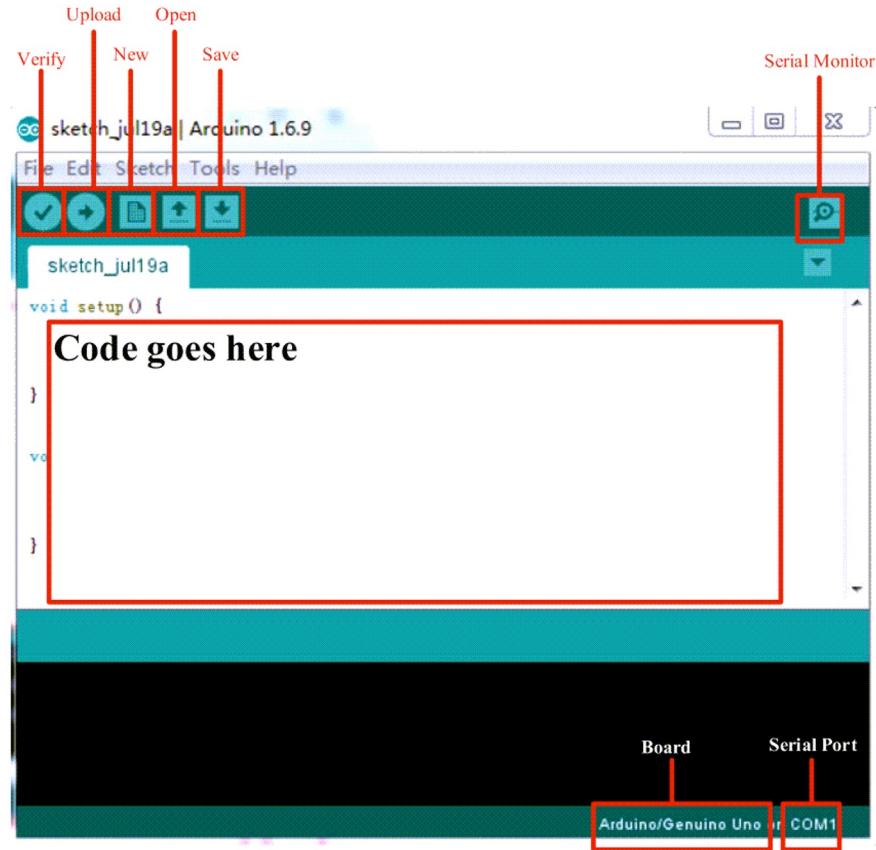
Analog Pin 3 (**A3**)

(I2C) SDA / Analog Pin 4 (**A4**)

(I2C) SCL / Analog Pin 5 (**A5**)

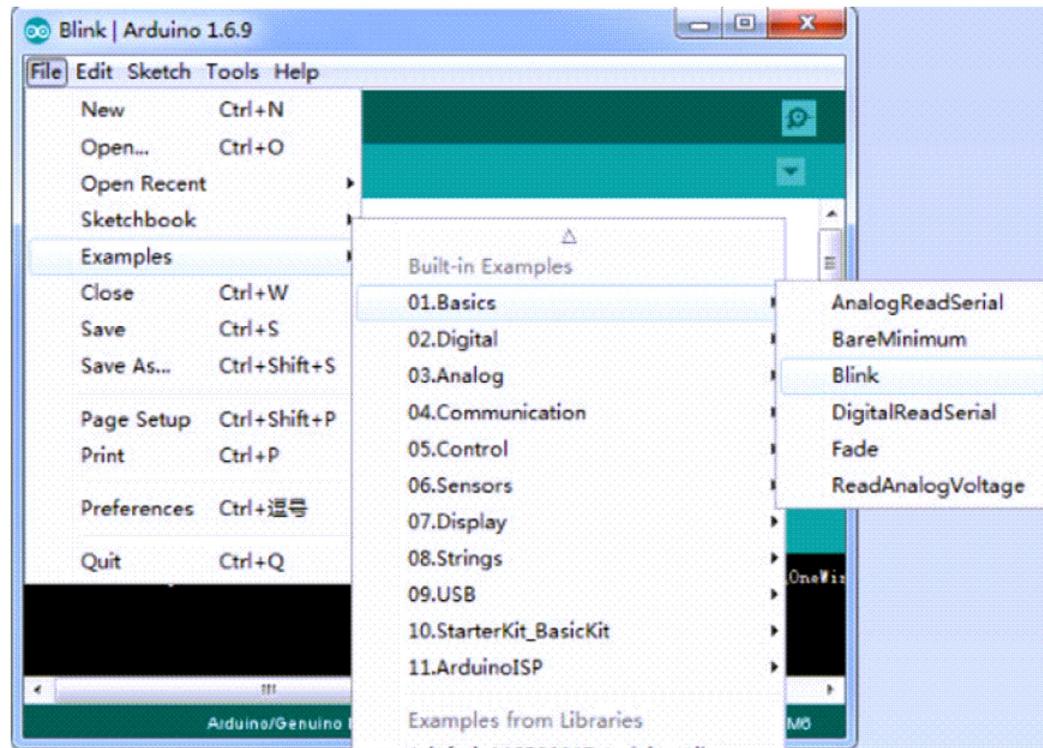


# Arduino IDE



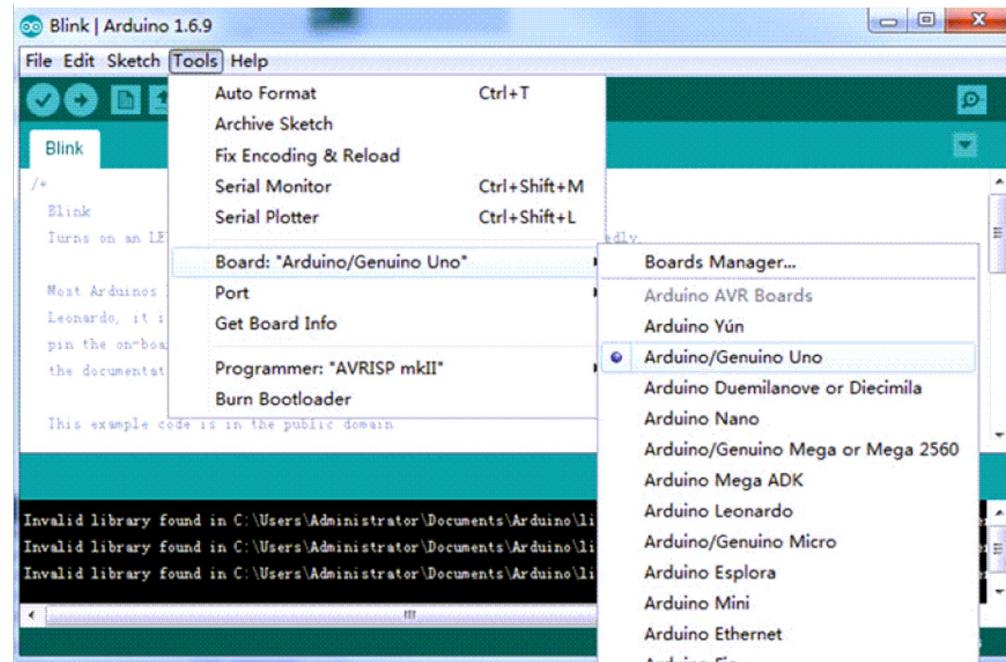
# Arduino IDE - Example

Blinking LED example



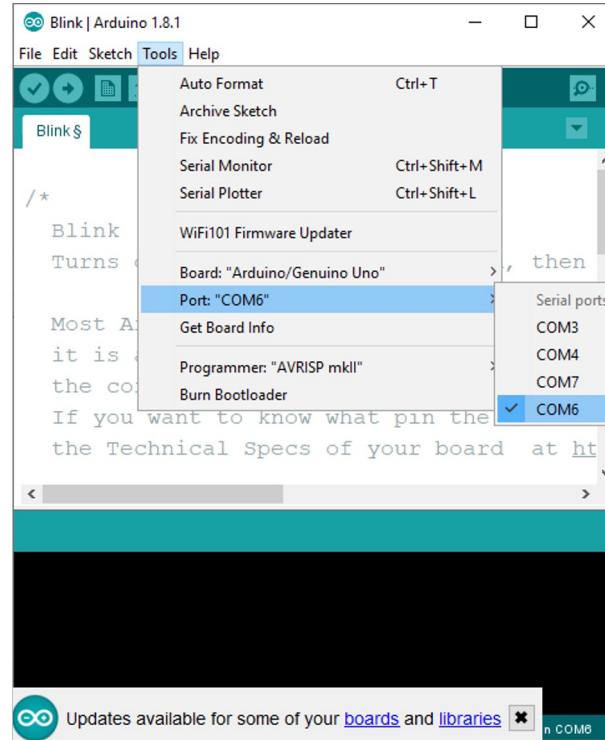
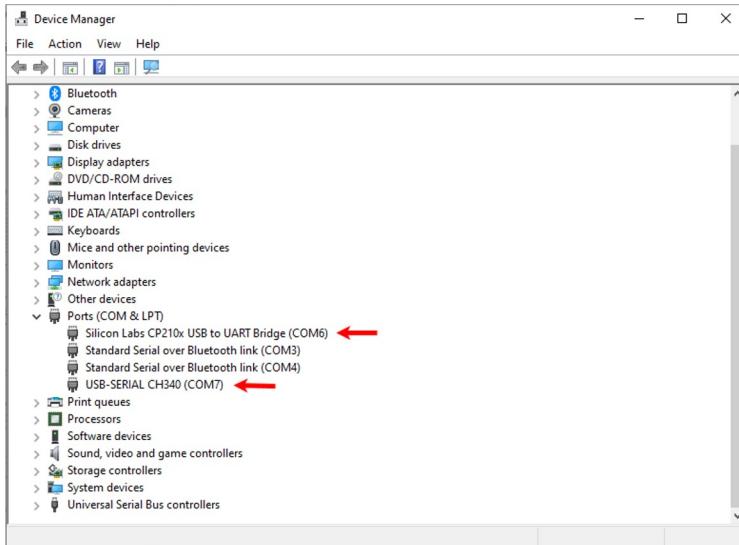
# Arduino IDE - Example

Select the Arduino board



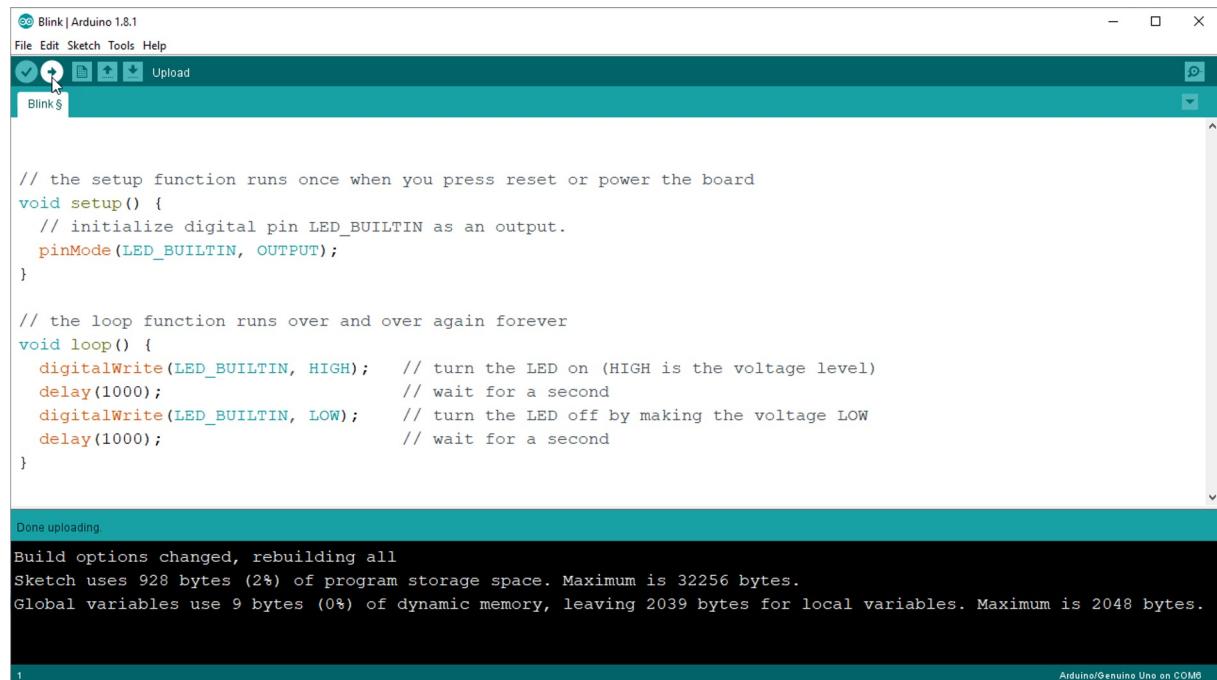
# Arduino IDE - Example

Select the COM port



# Arduino IDE - Example

## Upload to the board



The screenshot shows the Arduino IDE interface with the "Blink" sketch open. The code is as follows:

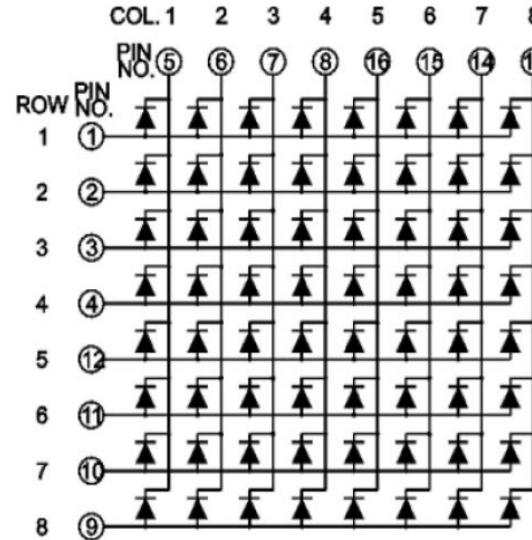
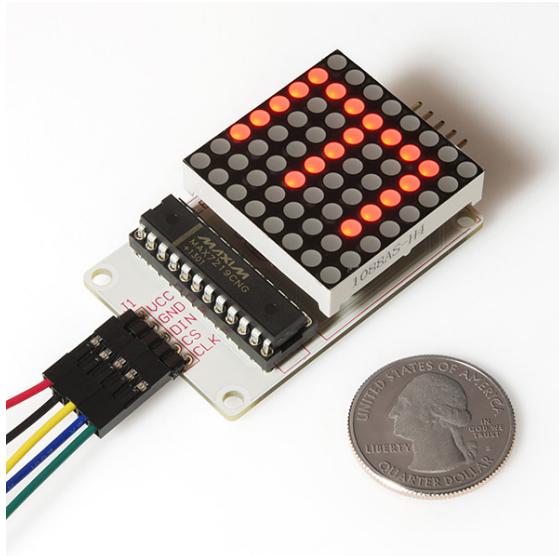
```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)
    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
}
```

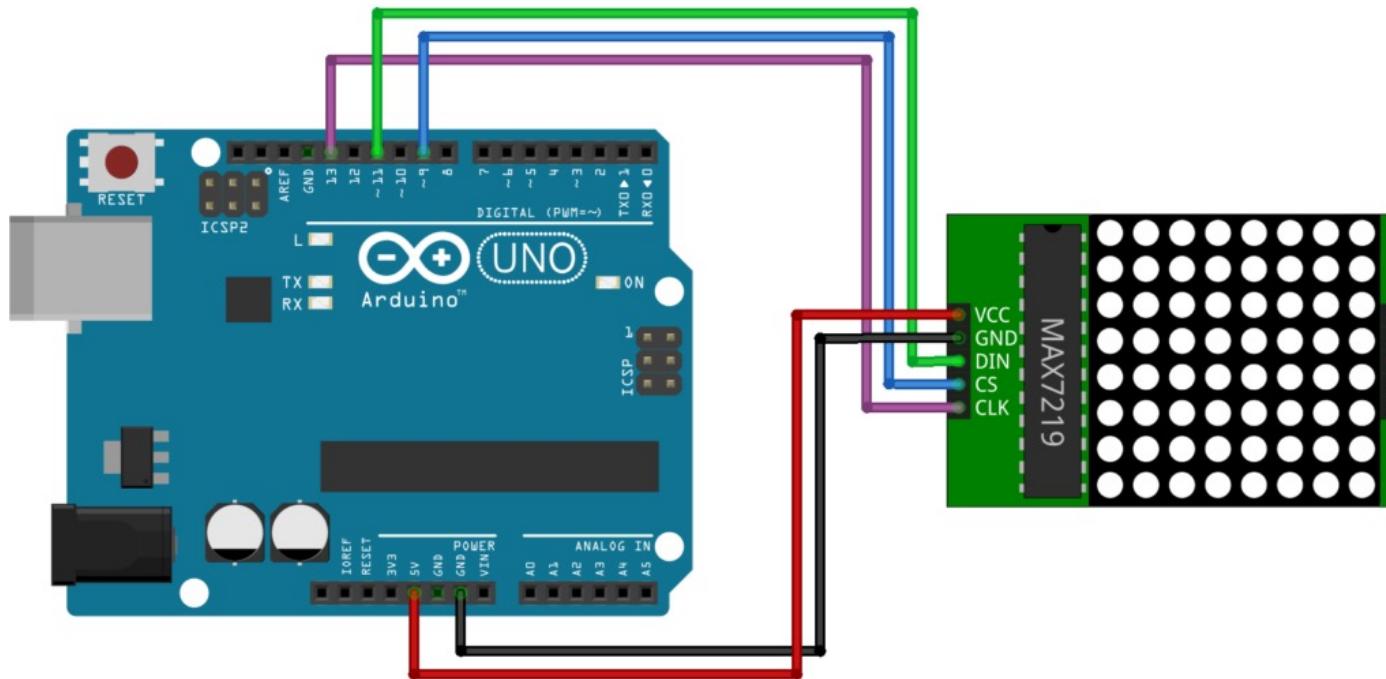
The status bar at the bottom displays the message "Done uploading." followed by build statistics: "Build options changed, rebuilding all", "Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.", and "Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes." The bottom right corner also shows "Arduino/Genuine Uno on COM6".

# Connect 8x8 Led Matrix Arduino Module

- Let's connect and program MAX7219CNG to Arduino and try to program it.



# Wiring Diagram



# Coding



SCAN ME

# Practice

- Divide yourselves into four groups and each group try to display IROS on the 8x8 led matrix
- I -> delay(1000) -> R -> delay(1000) -> O -> delay(1000) -> S -> delay(1000)

**That's All for Today**