

# Prototyping Interactive Systems

## DES 206



INDRAPRASTHA INSTITUTE *of*  
INFORMATION TECHNOLOGY  
DELHI

26-03-2024

**Richa Gupta**  
**Abhijeet Mishra**

# What is Arduino

- An **open-source** platform
- Based on an **easy-to-use** hardware and software
- **A programmable microcontroller**
- User friendly Development Environment (IDE)
- It was created for teaching electronics to artists and designers.



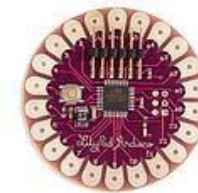
Arduino Uno



Arduino Leonardo



Arduino Mega 2560



Arduino LilyPad



Arduino Mega ADK



Arduino Fio



Arduino Ethernet



Arduino Pro



Arduino BT



Arduino Nano

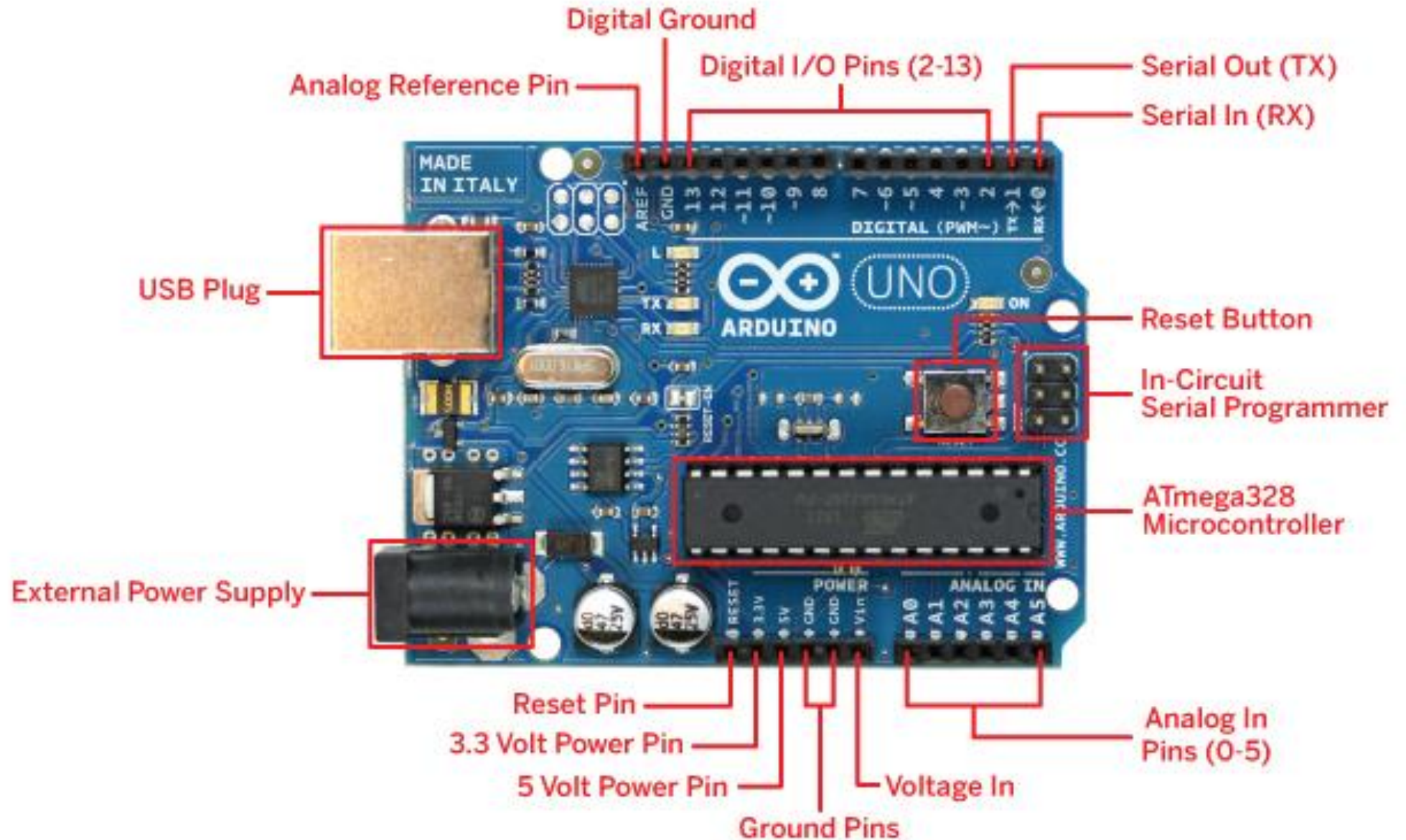


Arduino Mini

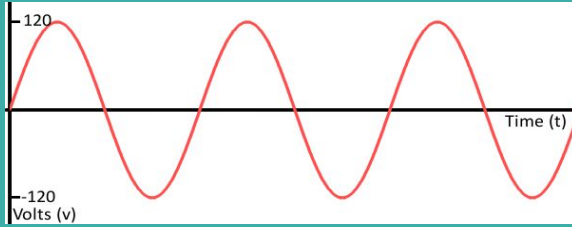


Arduino Pro Mini

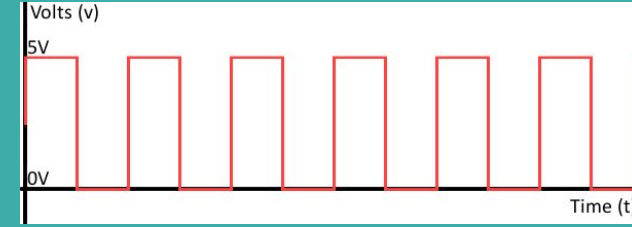
# Board



# ANALOG



# DIGITAL



**Signal** Analog signal is a continuous signal which represents physical measurements.

Digital signals are discrete time signals generated by digital modulation.

**Waves** Denoted by sine waves

Denoted by square waves

**Representation** Uses continuous range of values to represent information

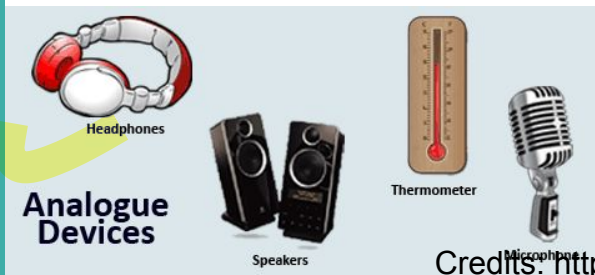
Uses discrete or discontinuous values to represent information

**Example** Human voice in air, analog electronic devices.

Computers, CDs, DVDs, and other digital electronic devices.

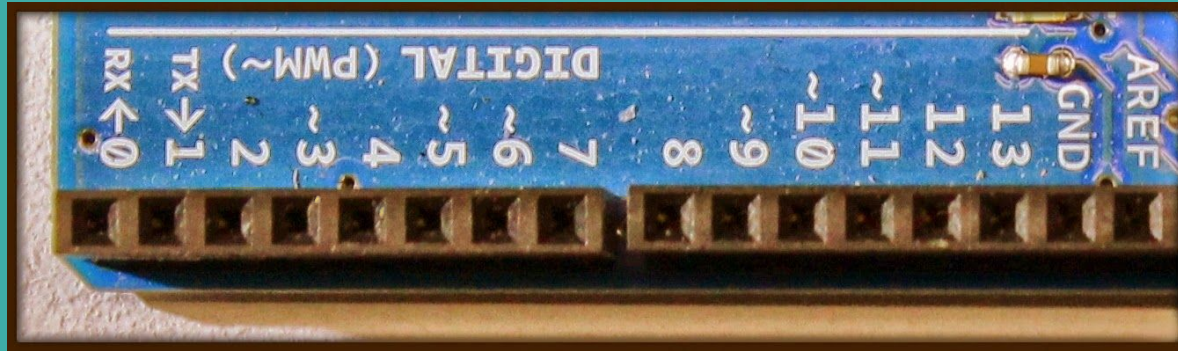
**Technology** Analog technology records waveforms as they are.

Samples analog waveforms into a limited set of numbers and records them.





# Digital pins



On board 13 Digital Pins are present

Generally, provides only two kinds of output i.e.

High = 5 Volt = 1

Low = 0 Volt = 0

Onboard 6 channel analog-to-digital (A/D) converter.

ARef: On this pin, user defined reference value of ADC is given (Not more than 5 Volt).



# Analog pins

# Arduino IDE

The Arduino Software (IDE) allows you to write programs and upload them to your board.

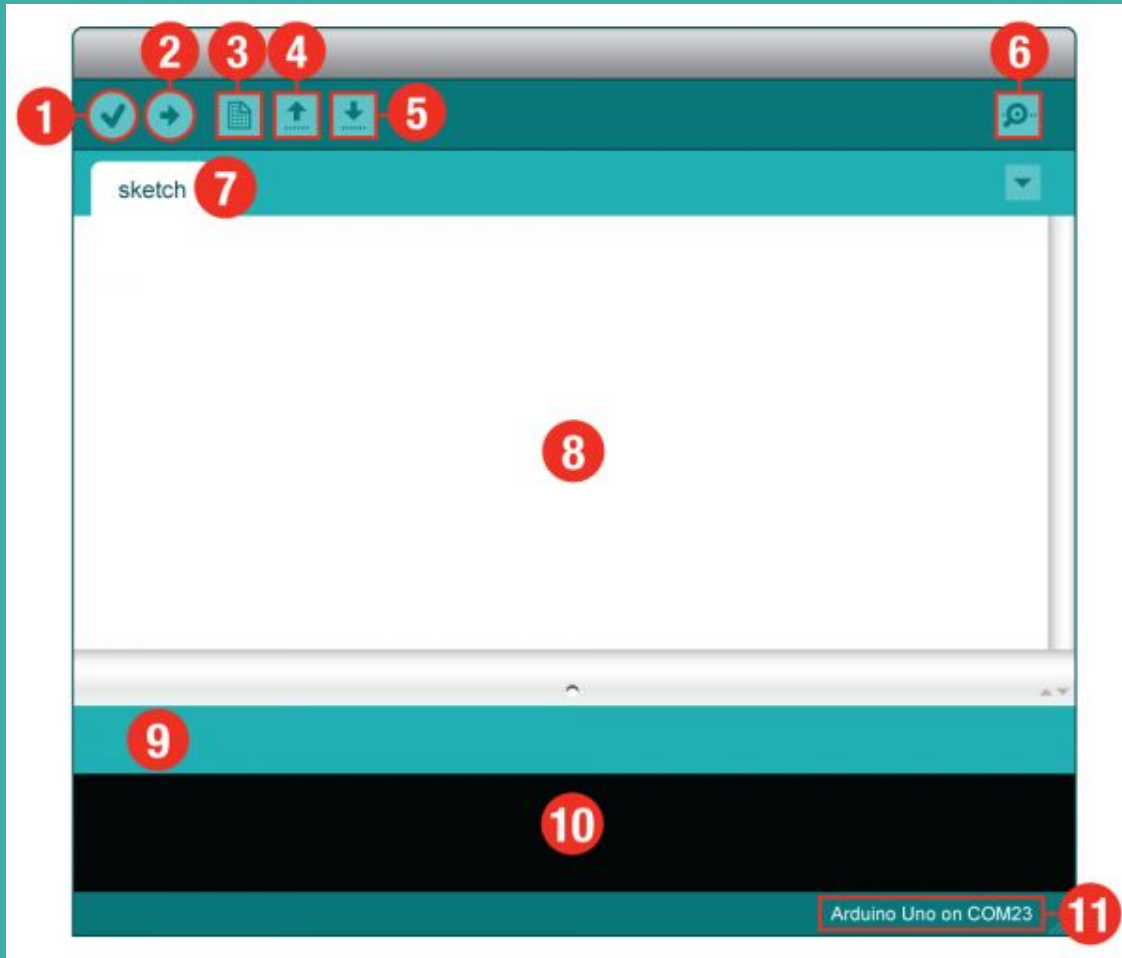
1. Provided by Arduino
2. Easy to write code and upload
3. Runs on various OS for ex. Windows, Mac, Linux, etc
4. Open-source software

Options:-

1. Online IDE (Arduino Web Editor)
2. Desktop IDE



# Arduino IDE

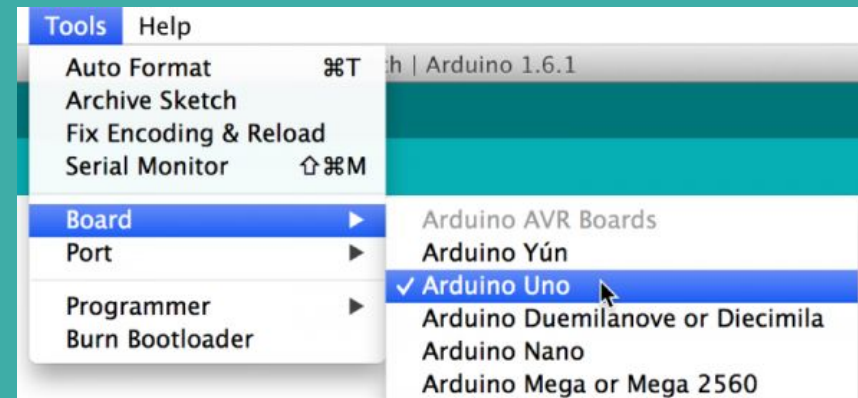


2. **Upload:** Sends your code to the Arduino board. When you click it, you should see the lights on your board blink rapidly.
3. **New:** This buttons opens up a new code window tab.
4. **Open:** This button will let you open up an existing sketch.
5. **Save:** This saves the currently active sketch.
6. **Serial Monitor:** This will open a window that displays any serial information your Arduino is transmitting. It is very useful for debugging.
7. **Sketch Name:** This shows the name of the sketch you are currently working on.
8. **Code Area:** This is the area where you compose the code for your sketch.
9. **Message Area:** This is where the IDE tells you if there were any errors in your code.
10. **Text Console:** The text console shows complete error messages. When debugging, the text console is very useful.

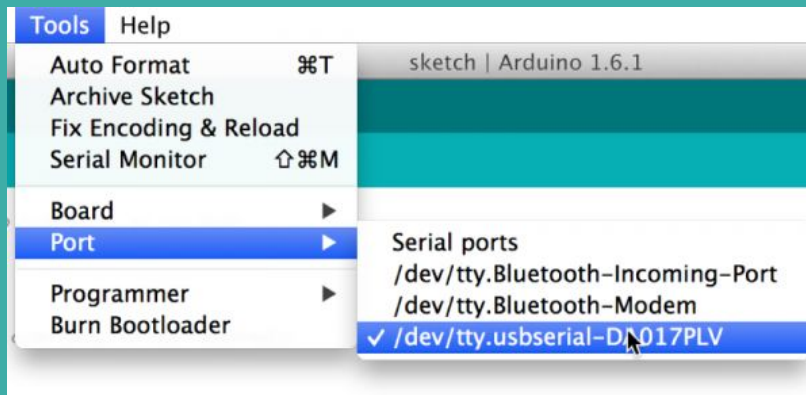
# Board and Port Selection

1. Connect your board to PC using USB

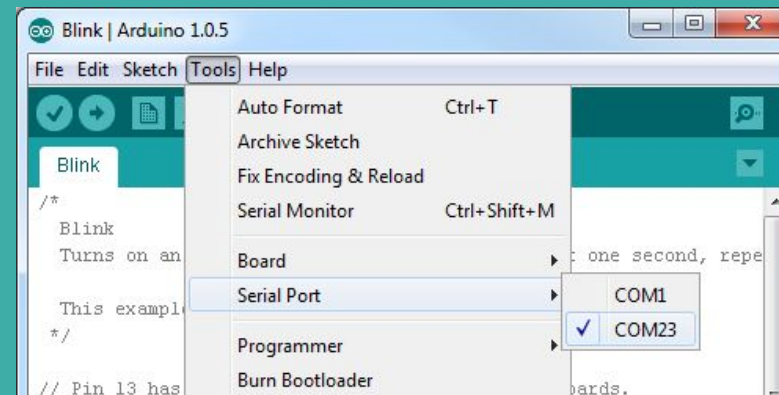
2. Select the board



3. (MAC) Select the port

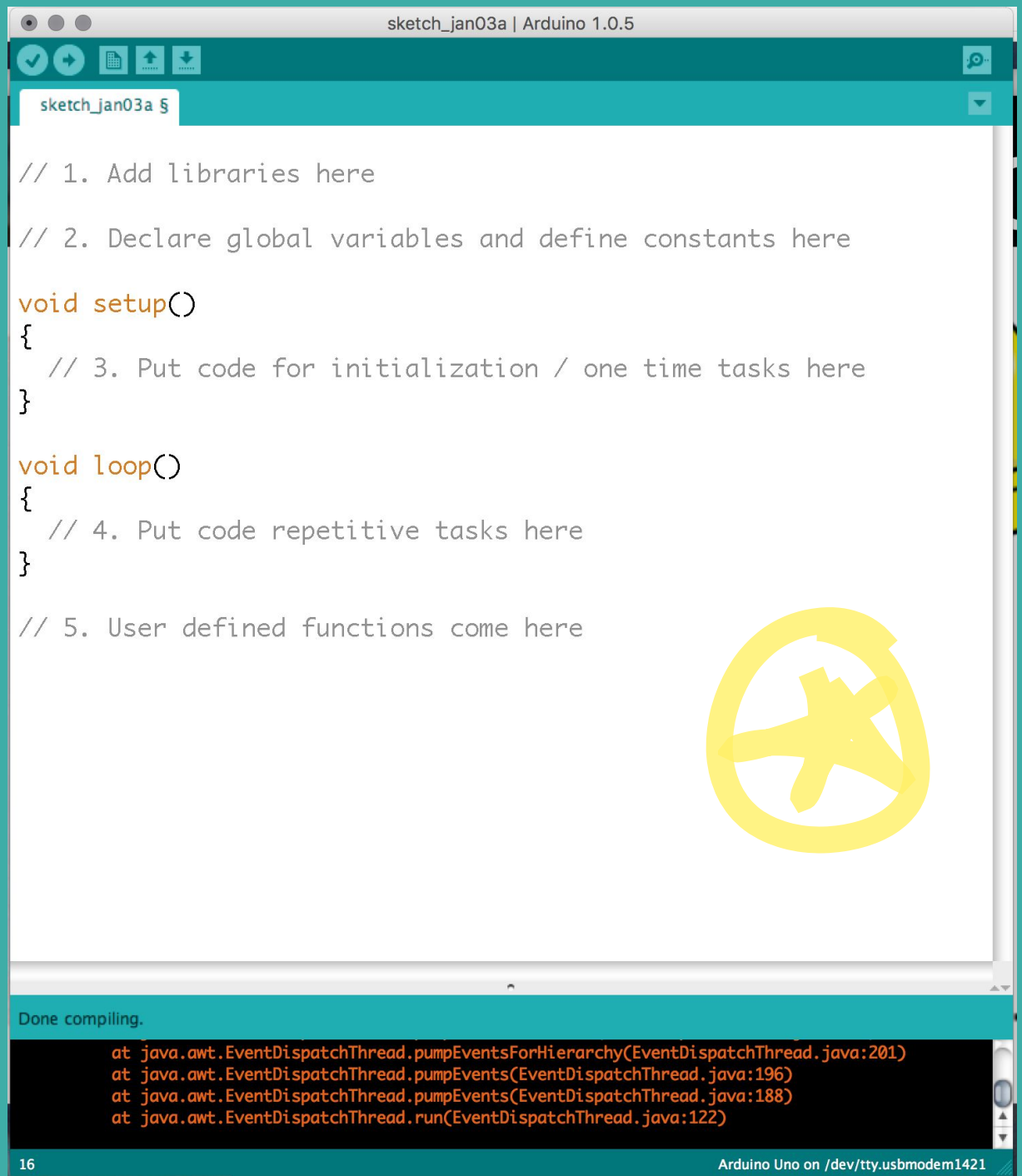


3. (Windows) Select the port





# Code Basics



The screenshot shows the Arduino IDE interface. The title bar at the top reads "sketch\_jan03a | Arduino 1.0.5". The main text area contains the following code:

```
// 1. Add libraries here

// 2. Declare global variables and define constants here

void setup()
{
  // 3. Put code for initialization / one time tasks here
}

void loop()
{
  // 4. Put code repetitive tasks here
}

// 5. User defined functions come here
```

A yellow star icon is drawn on the right side of the code area. The bottom status bar shows "Done compiling." and a stack trace:

```
at java.awt.EventQueueThread.pumpEventsForHierarchy(EventDispatchThread.java:201)
at java.awt.EventQueueThread.pumpEvents(EventDispatchThread.java:196)
at java.awt.EventQueueThread.pumpEvents(EventDispatchThread.java:188)
at java.awt.EventQueueThread.run(EventDispatchThread.java:122)
```

The bottom left corner of the IDE window displays the number "16". The bottom right corner shows "Arduino Uno on /dev/tty.usbmodem1421".

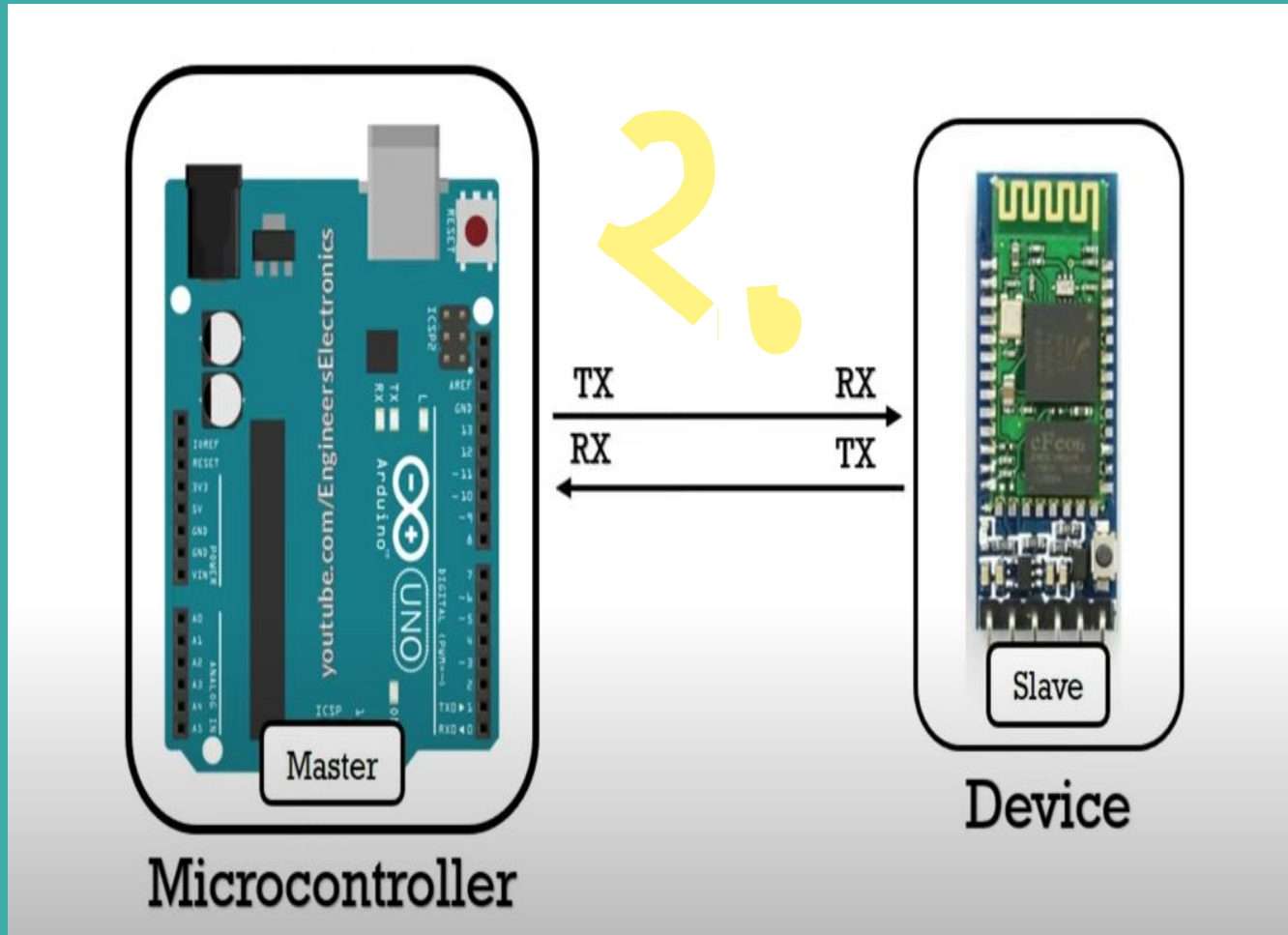
# **RULES:**

- 1. Every Arduino Program has void setup() and void loop()**
- 2. Program runs in linear fashion**
- 3. Remember the { } Curly brackets**
- 4. Program is Case sensitive**
- 5. Use // to add comments in Program**

# **NOTES:**

- 1. Variables are used to store information to be referenced & manipulated in a computer program.**
- 2. Three things are required to declare a variable**
  - > Data type**
  - > Variable name**
  - > Value**
- 3. Global variables can be accessed (used) on any function in the program**
- 4. Local variables are declared inside a function, and can be used only inside that function**

# About Serial and PWM Command



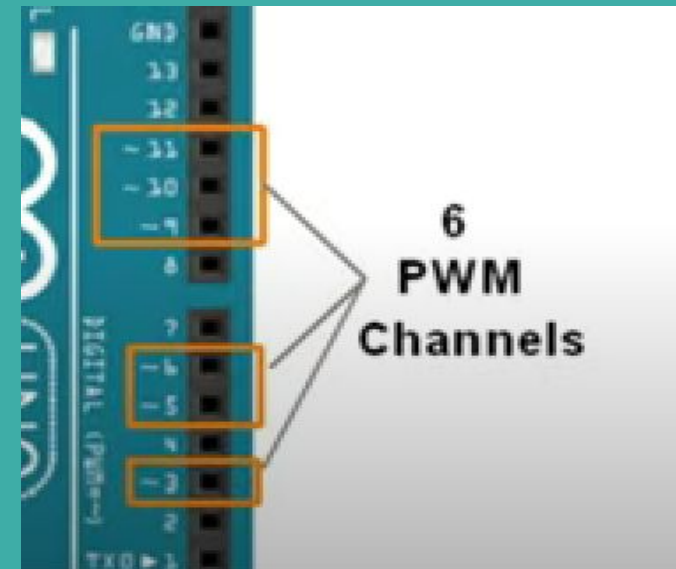
```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    if(Serial.available() == 1)  
    {  
        char val = Serial.read();  
        Serial.print(val);  
    }  
}
```



# About AnalogRead/Write and PWM Command

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  int val = analogRead(A0);  
  Serial.println(val);  
  delay(1000);  
}
```

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  analogWrite(6, 100);  
}
```



## ■ Comparison Operators:

`x == y` (x is equal to y)

`x != y` (x is not equal to y)

`x < y` (x is less than y)

`x > y` (x is greater than y)

`x <= y` (x is less than or equal to y)

`x >= y` (x is greater than or equal to y)

```
void setup()
{
  pinMode(2, INPUT_PULLUP);
  pinMode(6, OUTPUT);
}
void loop()
{
  int val = digitalRead(2);
  if(val == 0)
  {
    digitalWrite(6, HIGH);
  }
  else
  {
    digitalWrite(6, LOW);
  }
}
```

```
int val[6];
```

```
int val[] = {2, 3, 4, 5, 6, 7};
```

```
int val[6] = {2, 3, 4, 5, 6, 7};
```

```
char val[] = {'a', 'b', 'c', 'd'};
```

```
char val[4] = {'a', 'b', 'c', 'd'};
```

```
char val[] = "abcd";
```

```
char val[4] = "abcd";
```

```
String val[] = {"apple" , "mango" , "banana"};
```

```
void setup() {
```

```
  // put your setup code here, to run once:
```

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

```
  Serial.print(val[1]);
```

```
}
```



# Interrupt

A

```
#define RED_LED 12
#define YELLOW_LED 13
#define BUTTON 2

void setup() {
  // put your setup code here, to run once:
  pinMode(RED_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(BUTTON, INPUT_PULLUP);
}

void loop() {
  // put your main code here, to run repeatedly:
  int button_state = digitalRead(BUTTON);

  if (button_state == LOW)
  {
    digitalWrite(RED_LED, HIGH);
  }
  else
  {
    digitalWrite(RED_LED, LOW);
  }

  digitalWrite(YELLOW_LED, HIGH);
  delay(1000);
  digitalWrite(YELLOW_LED, LOW);
  delay(1000);
}
```

## Syntax:

- `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);`

pin: the Arduino pin number.

ISR: the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an interrupt service routine.

mode: defines when the interrupt should be triggered. Four constants are predefined as valid values:

LOW to trigger the interrupt whenever the pin is low,

CHANGE to trigger the interrupt whenever the pin changes value

RISING to trigger when the pin goes from low to high,

FALLING for when the pin goes from high to low.

Only digital pin 2 and 3  
can be used for  
interrupt in Arduino

B

C

```
void setup() {
  // put your setup code here, to run once:
  pinMode(RED_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(BUTTON, INPUT_PULLUP);

  attachInterrupt(digitalPinToInterrupt(BUTTON), control_led, CHANGE);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(YELLOW_LED, HIGH);
  delay(1000);
  digitalWrite(YELLOW_LED, LOW);
  delay(1000);
}

void control_led()
{
  int button_state = digitalRead(BUTTON);

  if (button_state == LOW)
  {
    digitalWrite(RED_LED, HIGH);
  }
  else
  {
    digitalWrite(RED_LED, LOW);
  }
}
```

**Thank You**