



# Arduino

# Micro Controllers vs Micro Processors

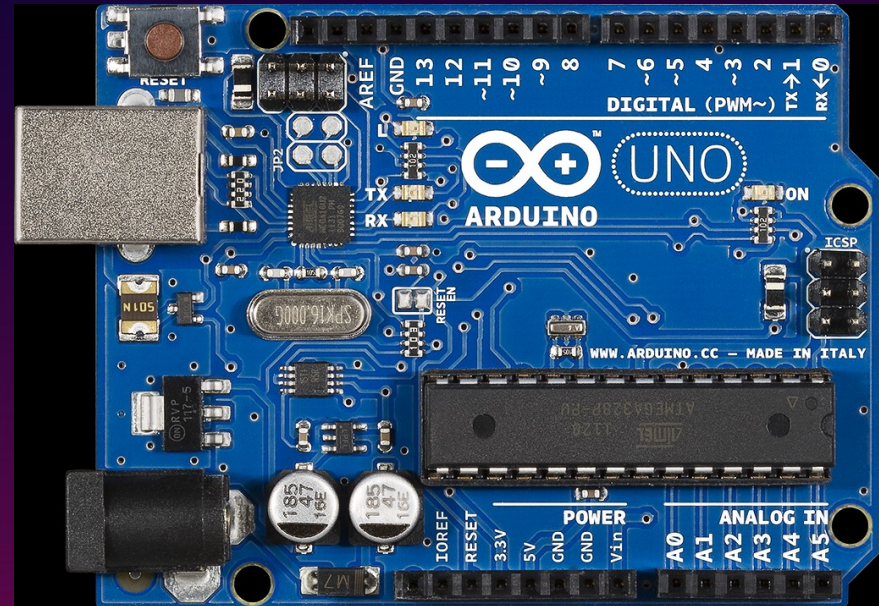
Parameters	Microprocessors	Microcontroller
Applications	Gaming, web browsing, document creation etc.	Dedicated for specific tasks (Camera, washing machine etc.)
Internal Structure	Memory and I/O devices connected externally	CPU, memory and I/O are present internally
Cost	High	Low
Power Consumption	High	Low
Memory (RAM)	512 MB to 32 GB	2KB to up to 256 KB
Storage	Hard Disk (128 GB to up to 2 TB)	Flash memory (32 KB to 2 MB)
Peripheral Interfaces	USB, UART, High speed Ethernet	UART, I2C, SPI

# Arduino vs Raspberry Pie

Specs	Arduino Uno 	Raspberry Pi 3 Model B 
CPU type	Microcontroller	Microprocessor
Operating System	None	Linux (usually Raspbian) or Win 10
Speed	16 Mhz	1.2Ghz
RAM	2KB	1GB
GPU/Display	None	VideoCore IV GPU
Disk	32KB	Depends on SD card
GPIO pins	14 digital pins (includes 6 analog)	26 digital pins
Other connectivity	None	USB, Ethernet, HDMI, audio
Power consumption	0.25W	6W

# Arduino Uno

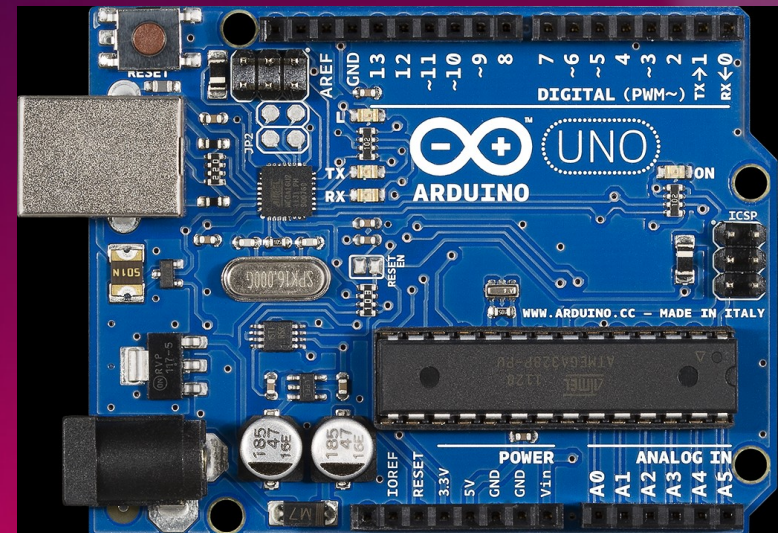
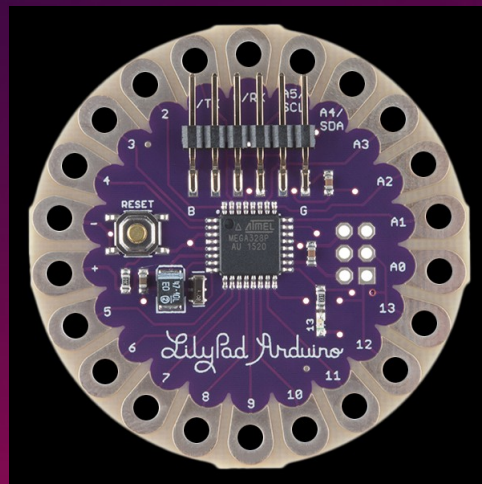
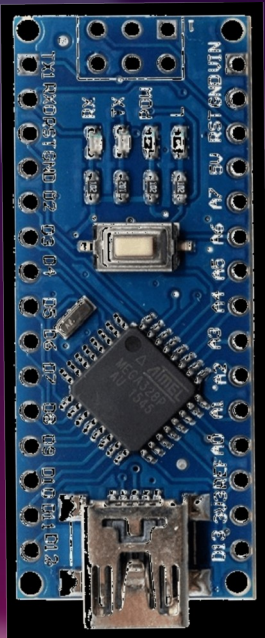
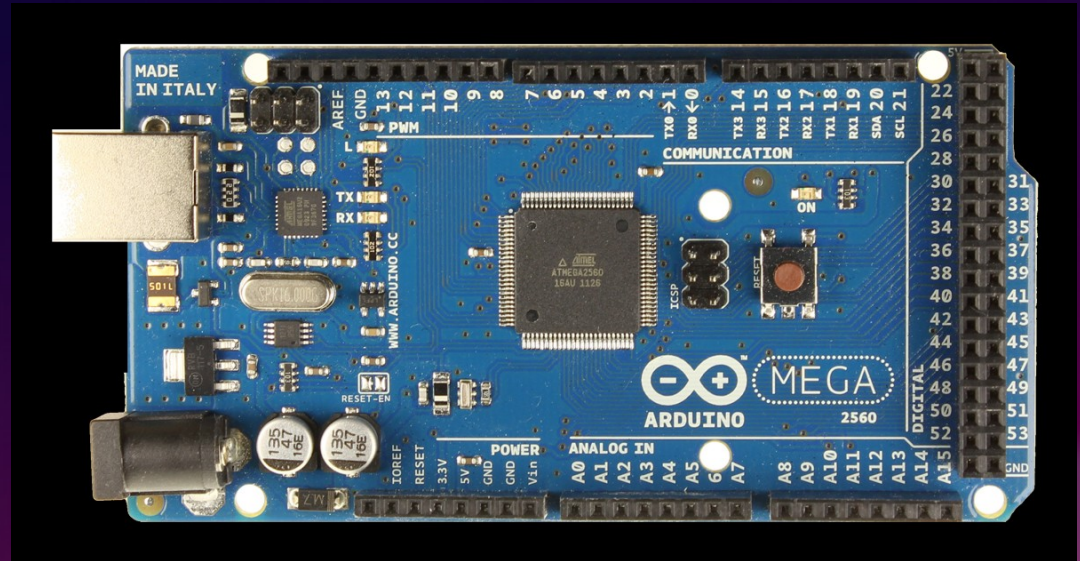
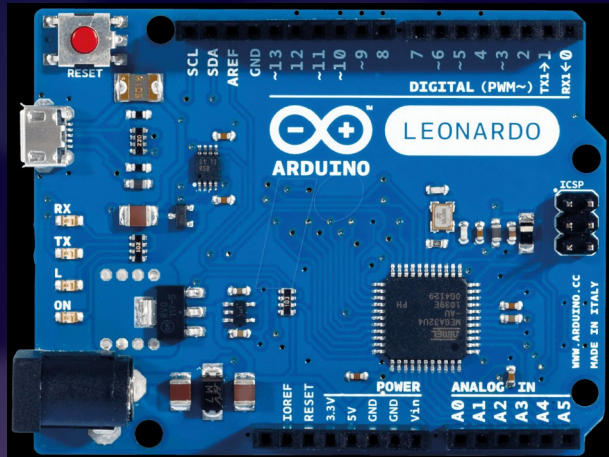
- Micro Controller
  - ATmega328
- Operating voltage
  - 5V and 3.3V
- Input Voltage
  - 7-12V
- Digital I/O pins
  - 14 pins
- Analog Input pins
  - 6 pins
- Flash Memory
  - 32 KB



- SRAM - 2 KB
- EEPROM - 1 KB
- Clock Speed - 16MHz

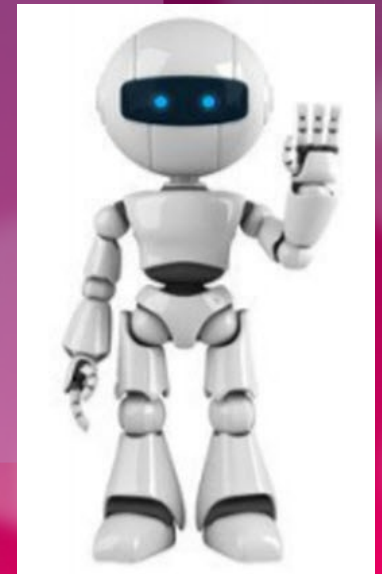
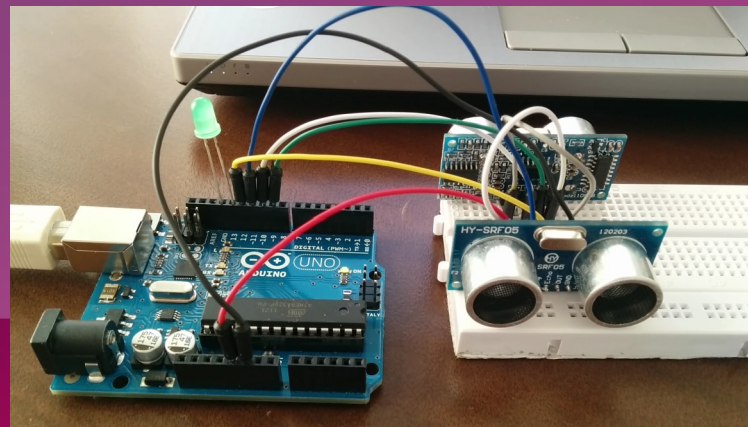


# Common Arduino Boards



# Applications of Arduino

- Home automation
- Sensor Prototyping
- Robotics
- ISP Programming
- Easy WiFi, GSM, Ethernet, Bluetooth connectivity

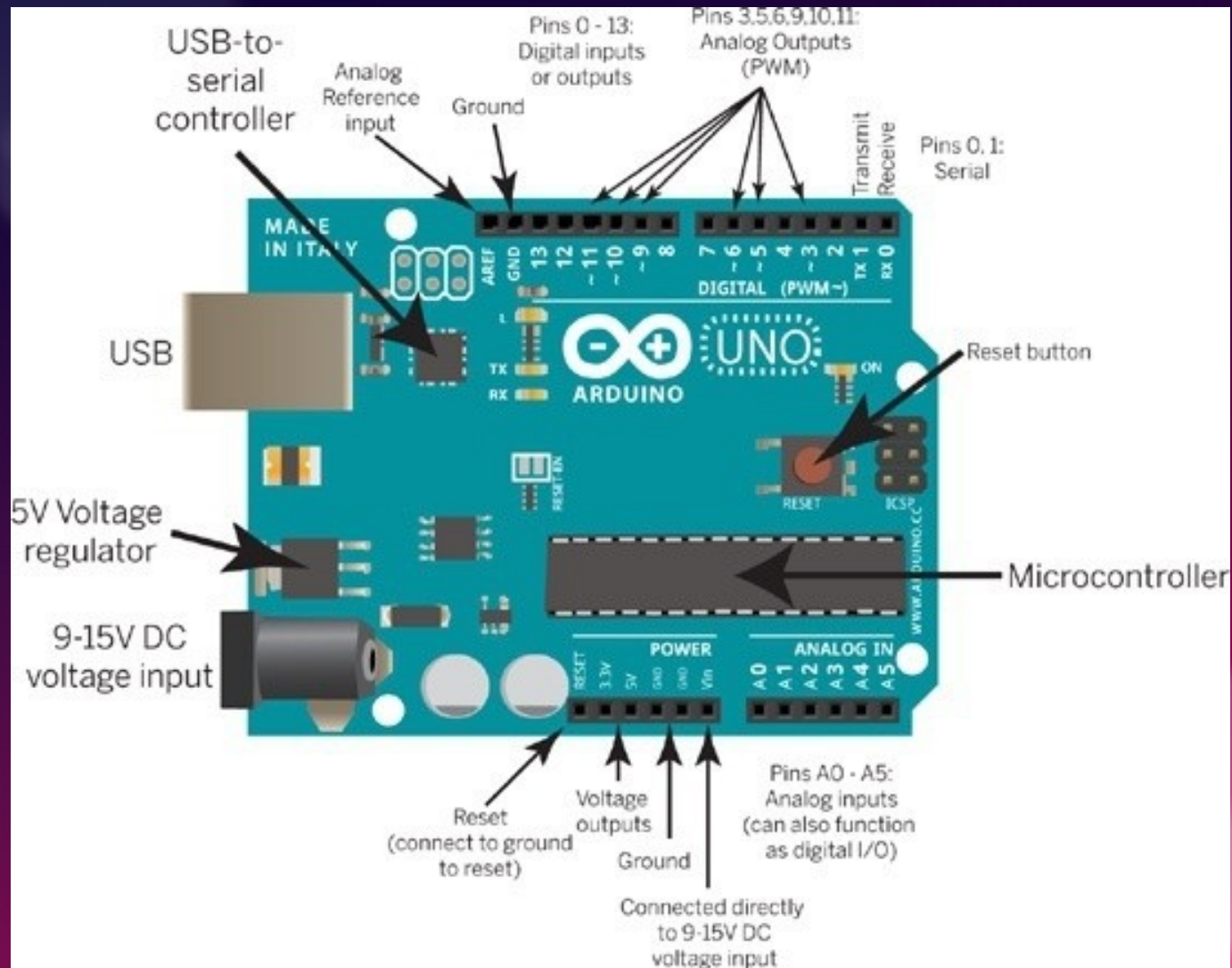




# Advantages of Arduino

- Open Source
- Simplified and user friendly
- No additional programmer or burner required.
- Portable
- Low power consumption

# Arduino Uno

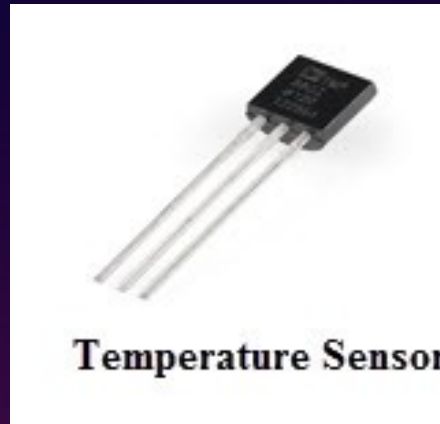




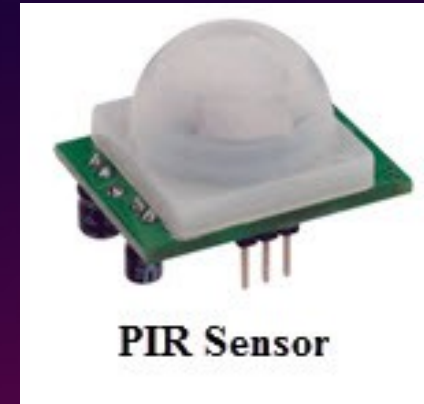
# Sensors



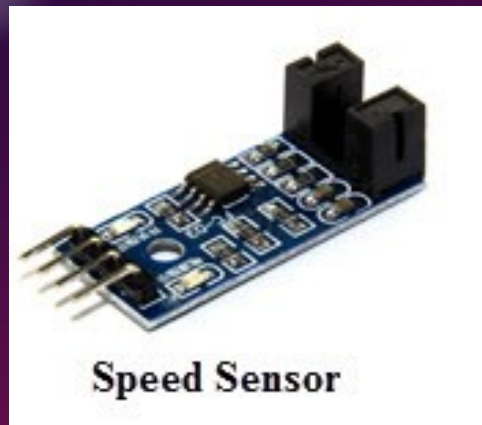
**Touch Sensor**



**Temperature Sensor**



**PIR Sensor**

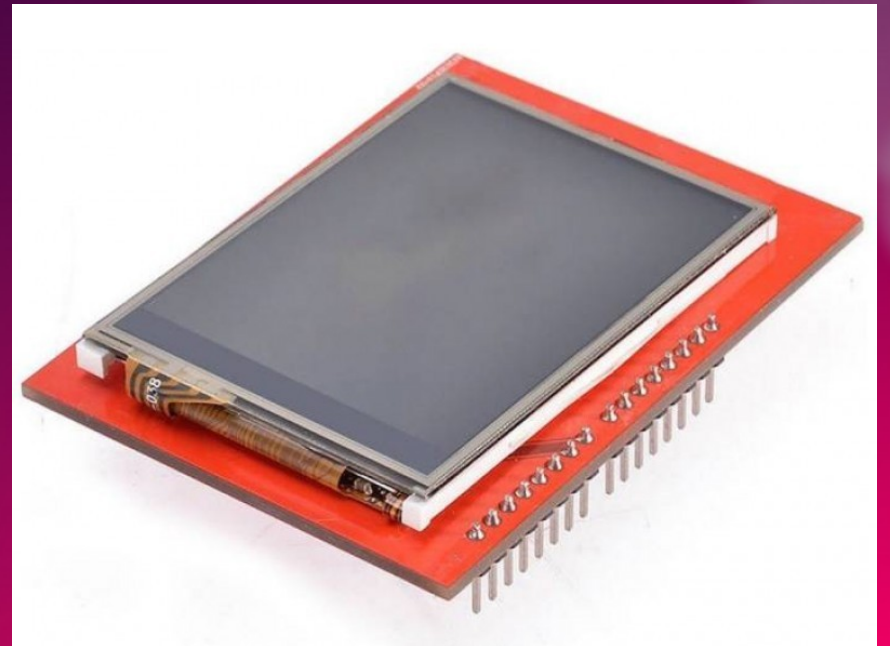
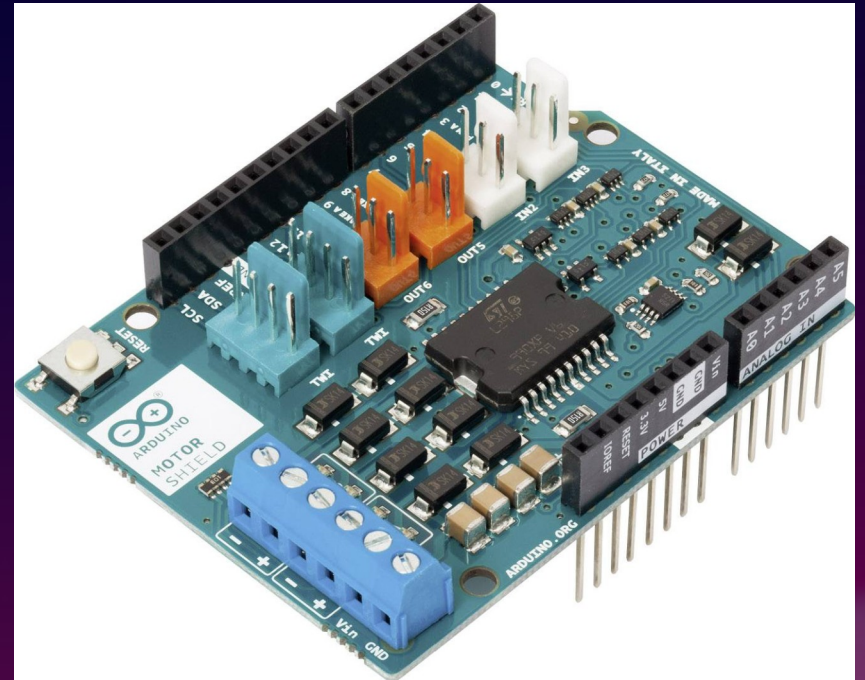
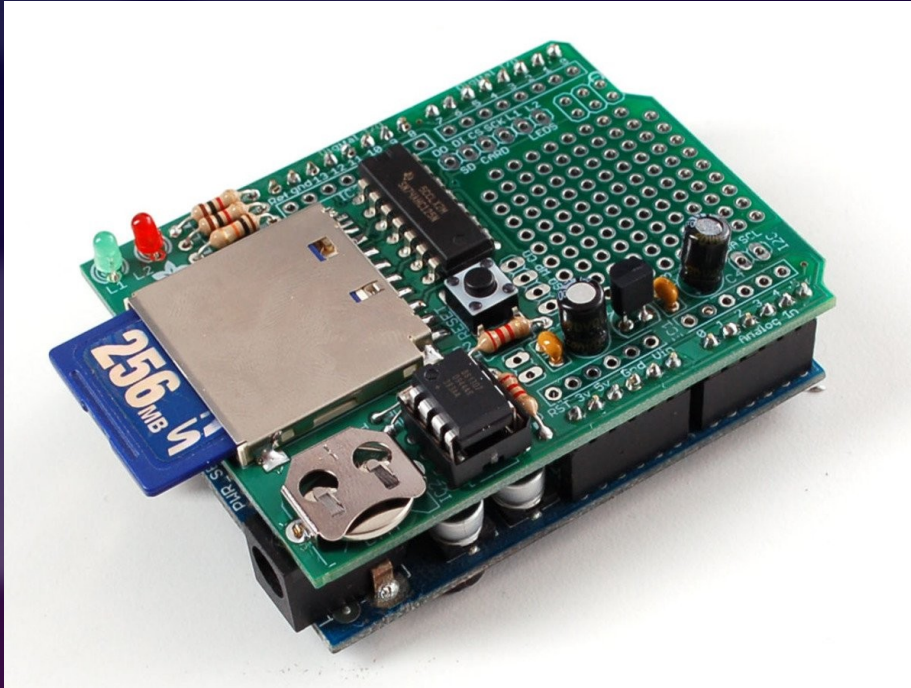


**Speed Sensor**

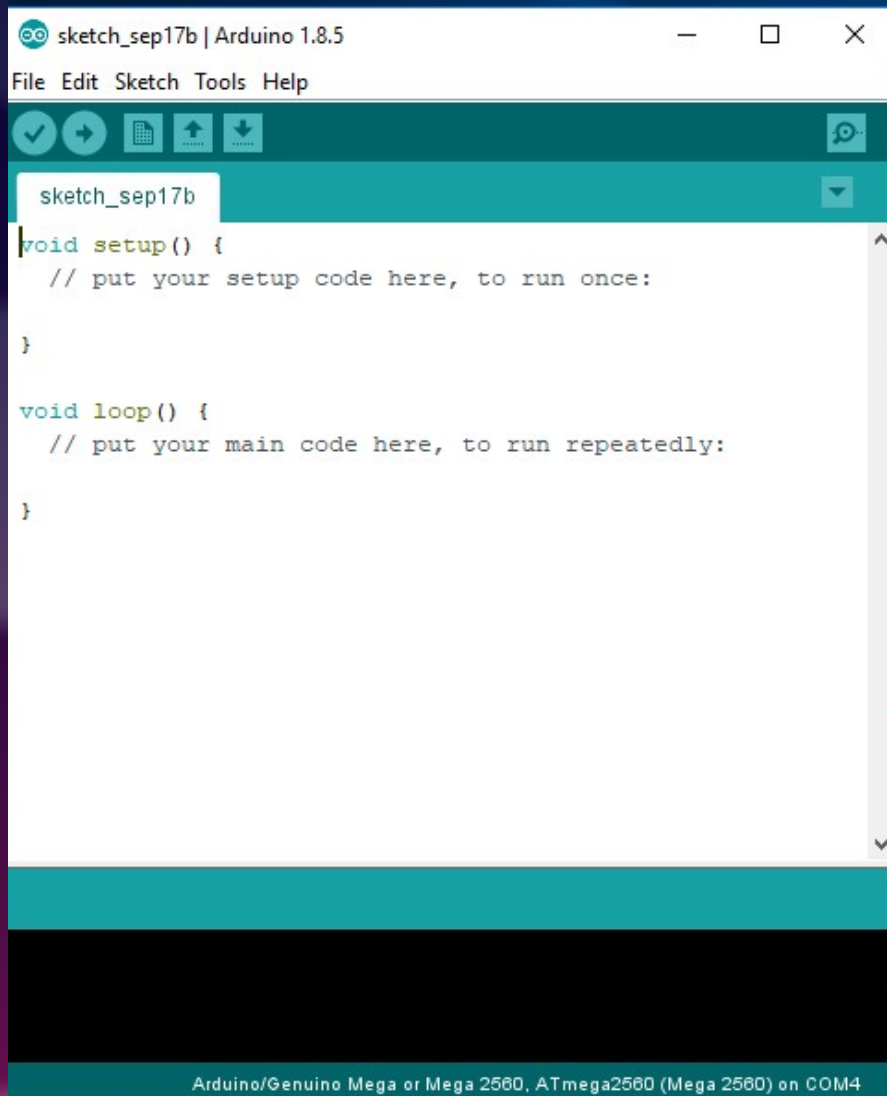


**Ultrasonic Sensor**

# Arduino Shields



# Arduino IDE



- Setup
  - It's called only when arduino is powered on or reset. Used to initialize variables and pin modes.
- Loop
  - Loop function runs continuously till the device is powered off.

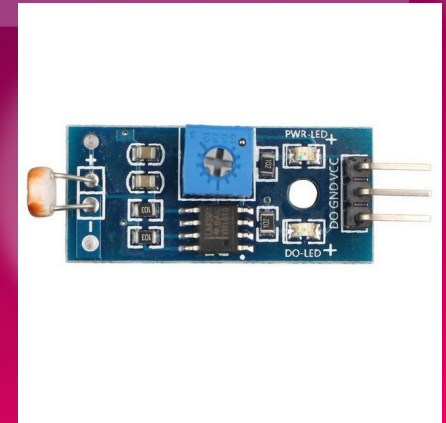
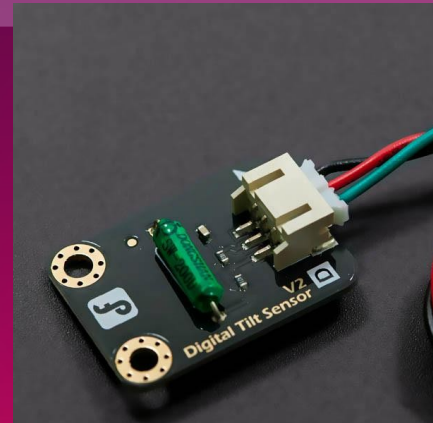
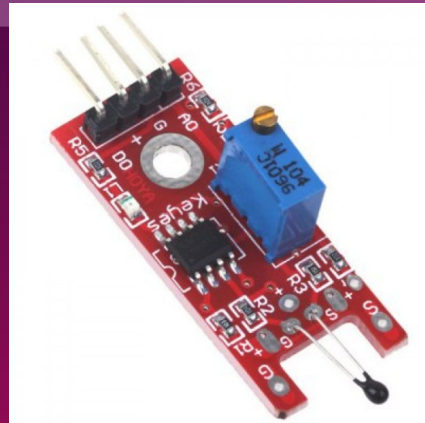
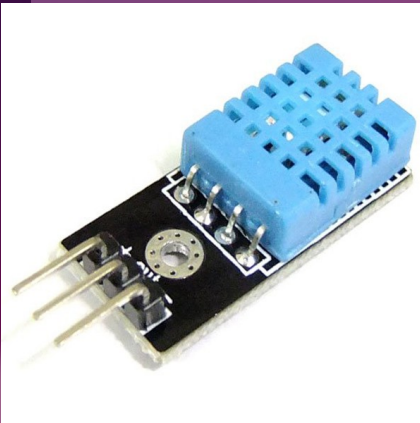


# Pin Mode

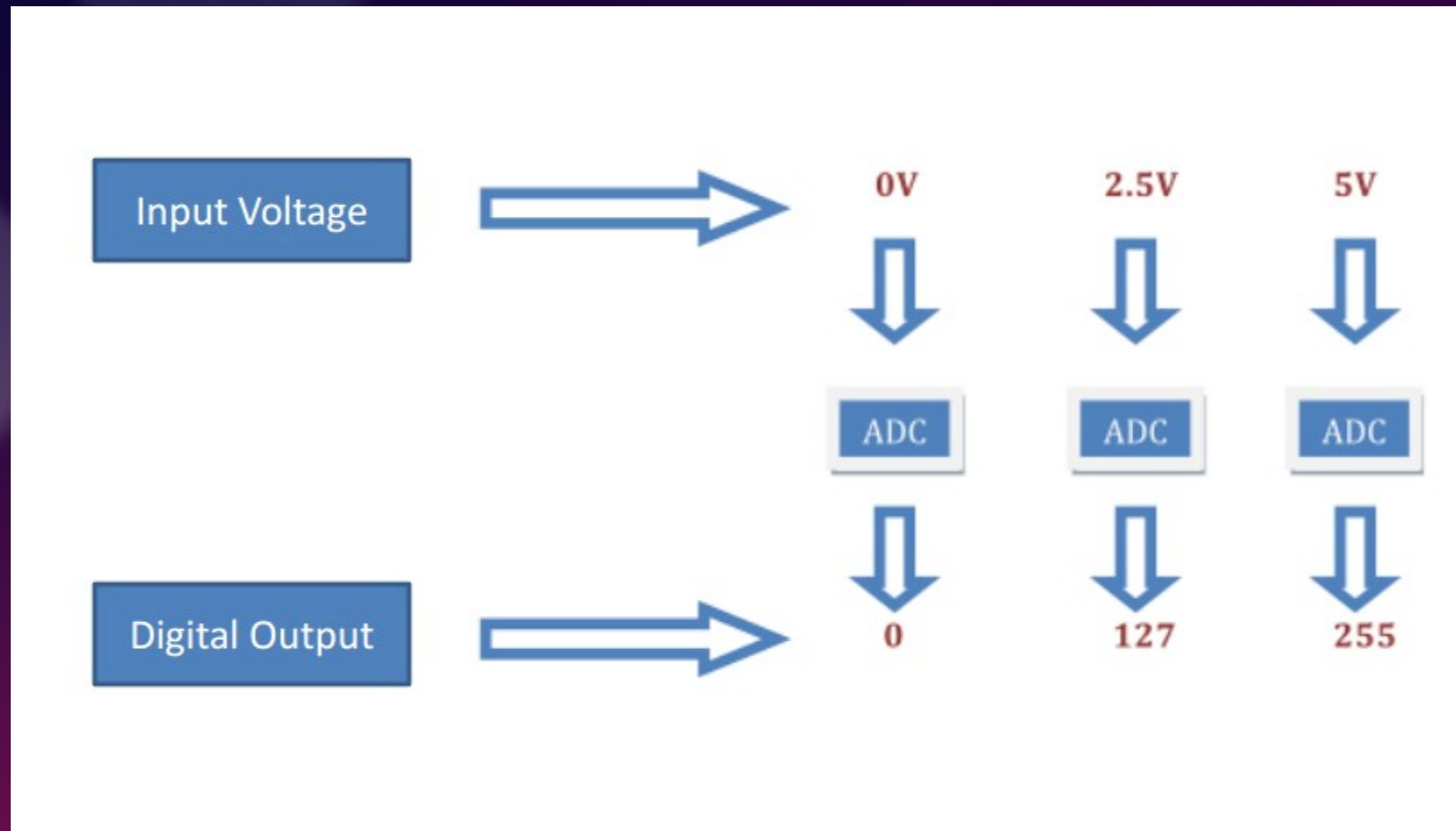
- **pinMode** function is used to set a pin in arduino as input or output.
  - `pinMode(13, Output);`      `// Sets pin 13 as output pin`
  - `pinMode(13, Input);`      `// Sets pin 13 as input pin`

# Reading / Writing digital values

- `digitalWrite(13, LOW);`
  - Makes the output voltage on pin 13, 0V
- `digitalWrite(13, HIGH);`
  - Makes the output voltage on pin 13, 5V
- `int buttonState = digitalRead(2);`
  - Reads the value of pin 2 in button state



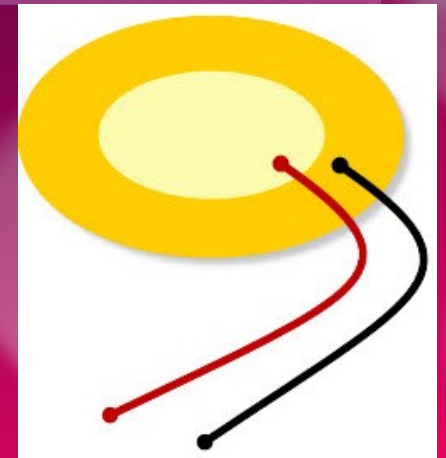
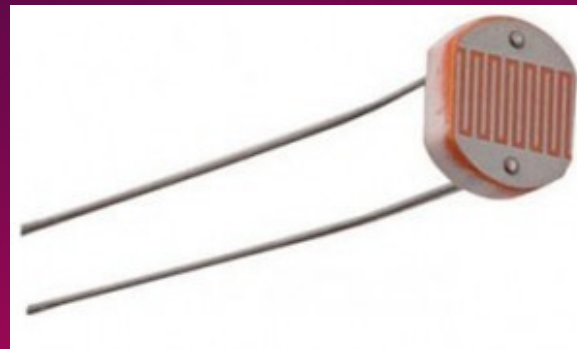
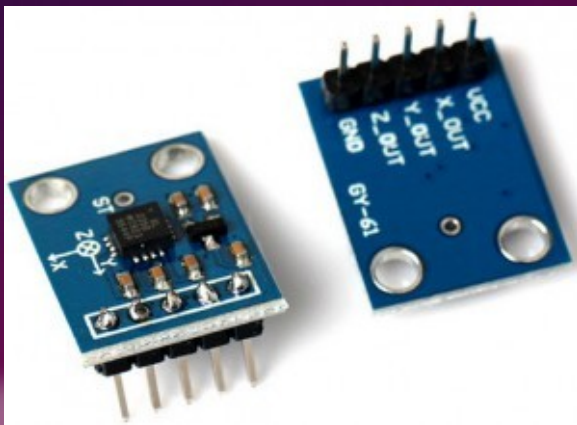
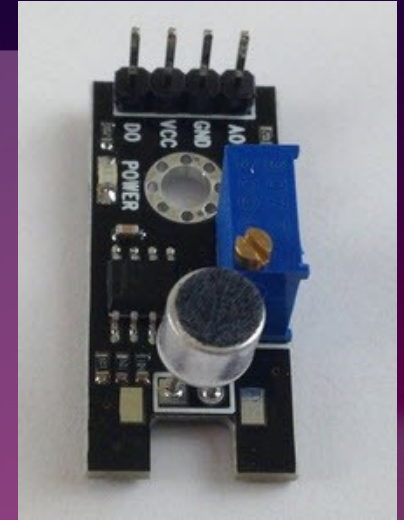
# Converting Analog values to Digital





# Reading / Writing analog values

- `analogRead(A0);`
  - Used to read the analog value from the pin A0
- `analogWrite(2, 128);`
  - Used to write the analog value of 128 to pin 2



# Blinking inbuilt LED

// 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)
```

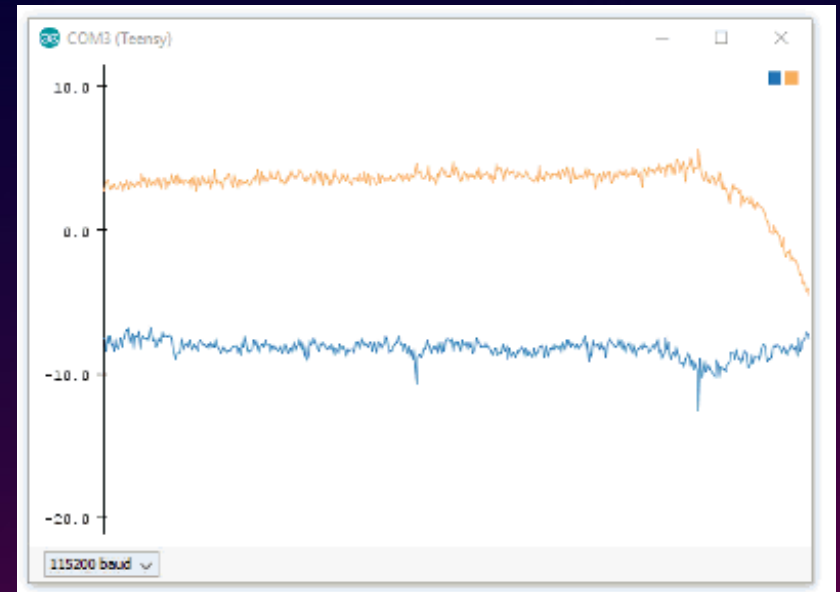
```
  delay(1000);                      // wait for a second
```

```
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage  
                                   LOW
```

```
  delay(1000);                      // wait for a second
```

```
}
```

# Serial Plotter



# Serial Monitor