Basic Engineering Course

Ex 01

Students meet

Introduction

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. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

source: www.arduino.cc/en/Guide/Introduction

Introduction

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. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

source: www.arduino.cc/en/Guide/Introduction

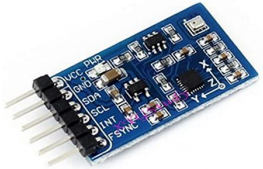
Arduino boards

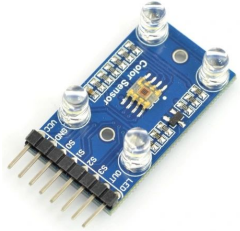
Where to use

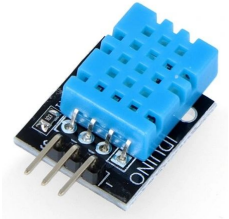
- Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics.
- Engineers, designers and architects build interactive prototypes.
- Musicians and artists use it for installations and to experiment with new musical instruments.
- ....

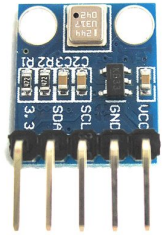
source: www.arduino.cc/en/Guide/Introduction

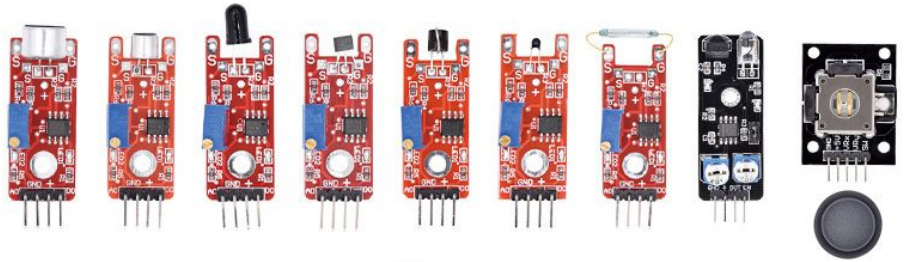
Arduino - sensors

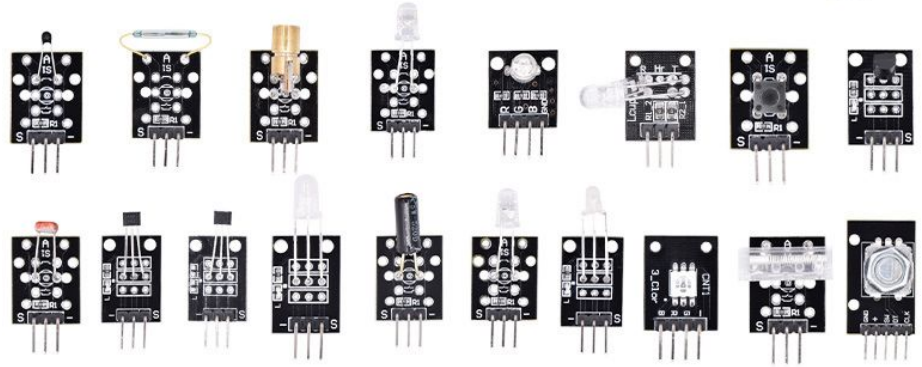


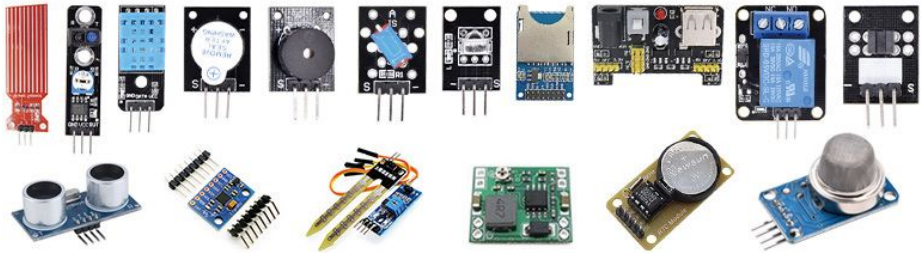




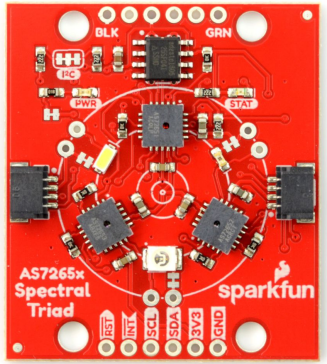


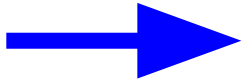


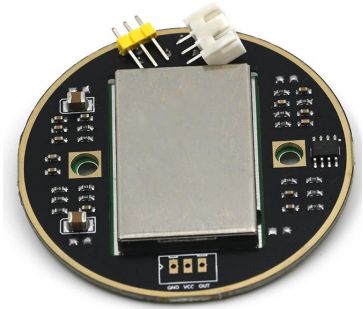


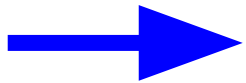


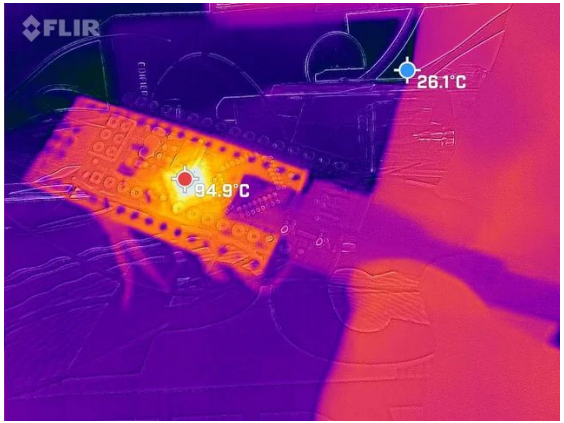
Arduino - advanced sensors







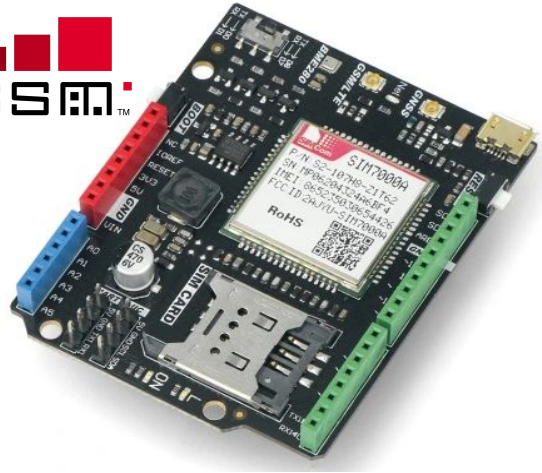




Arduino - communication

ARDUINO

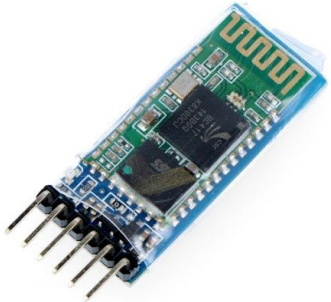
GSM



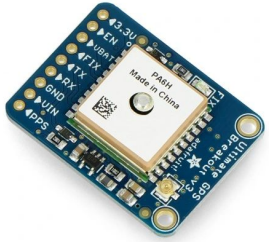
LoRa™

RS 485
RS 232

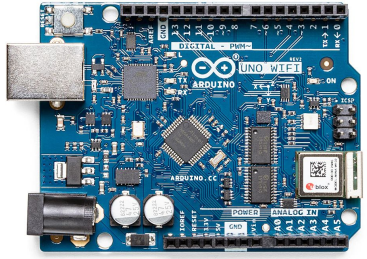
Bluetooth®



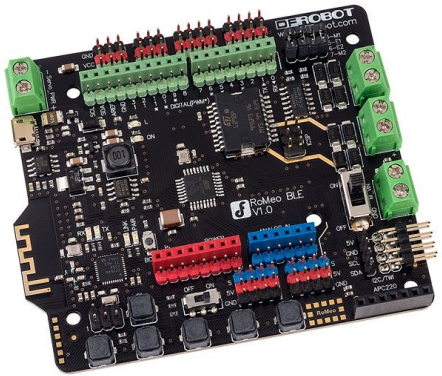
GPS

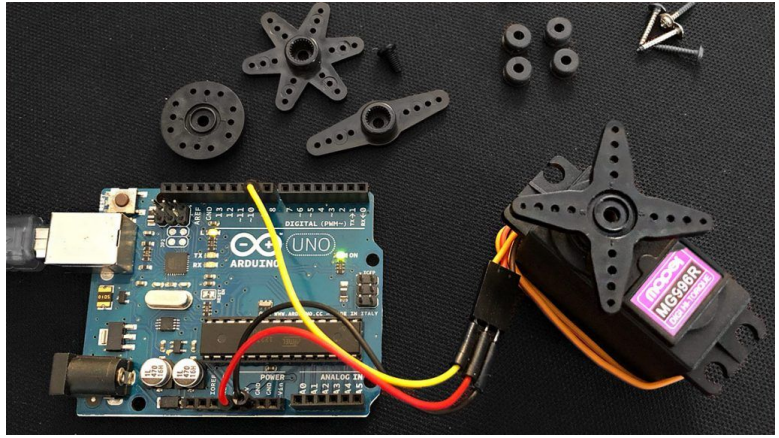


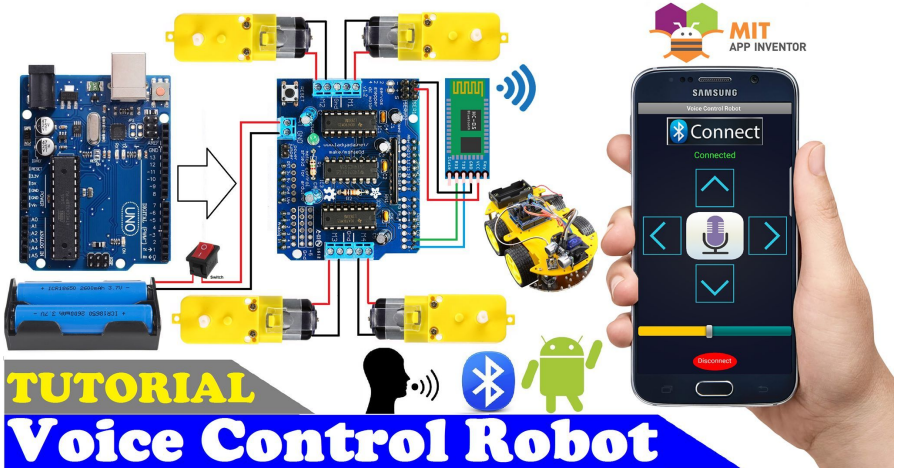
Wi Fi™

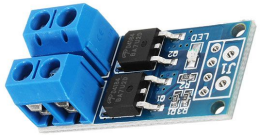


Arduino - control



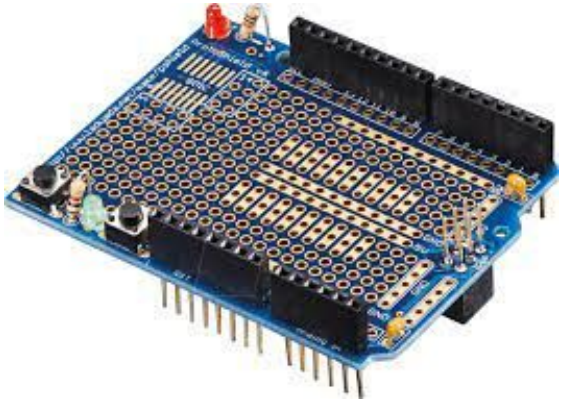






Arduino - shields

Shields are boards that can be plugged on top of the Arduino PCB extending its capabilities.

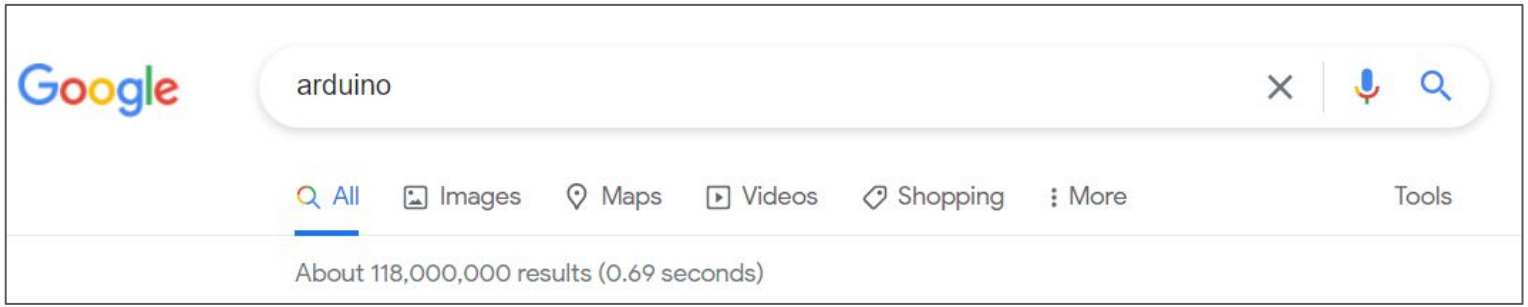


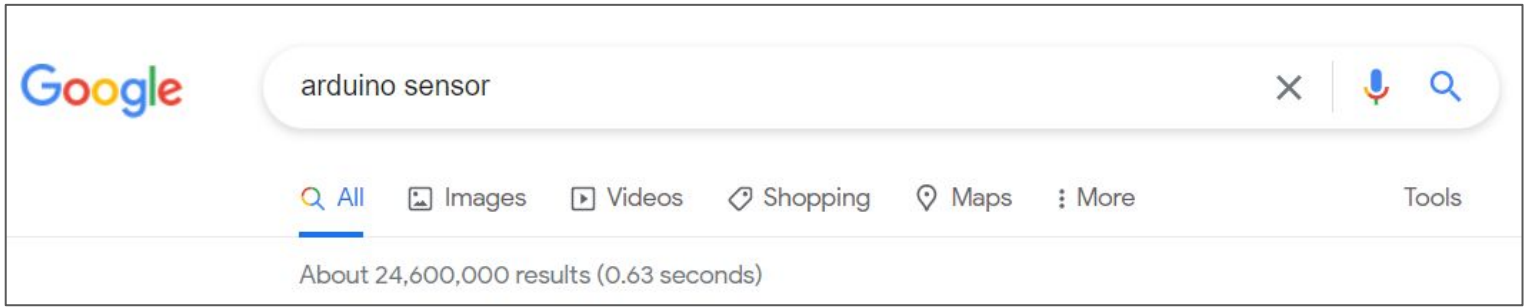
Why Arduino

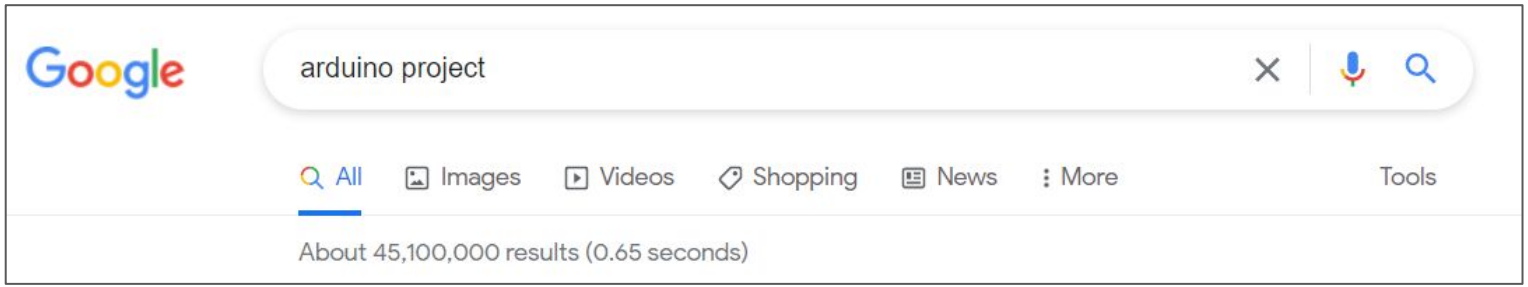
- Relatively inexpensive.
- Cross-platform IDE.
- Simple, clear programming environment.
- Open source and extensible software.
- Open source and extensible hardware .
- Large community.

source: www.arduino.cc/en/Guide/Introduction

Google


Google
arduino
All
Images
Maps
Videos
Shopping
More
Tools
About 118,000,000 results (0.69 seconds)


Google
arduino sensor
All
Images
Videos
Shopping
Maps
More
Tools
About 24,600,000 results (0.63 seconds)


Google
arduino project
All
Images
Videos
Shopping
News
More
Tools
About 45,100,000 results (0.65 seconds)

Arduino - links

- Language Reference

  www.arduino.cc/reference/en/

- Arduino and Processing

  playground.arduino.cc/Interfacing/Processing

- Random Nerd Tutorials

  randomnerdtutorials.com

Arduino - links

- Last Minute Engineers

  lastminuteengineers.com/electronics/arduino-projects/

- Open Electronics

  www.open-electronics.org/

- SparkFun

  learn.sparkfun.com

Getting started

1. Connect Arduino board to USB port of

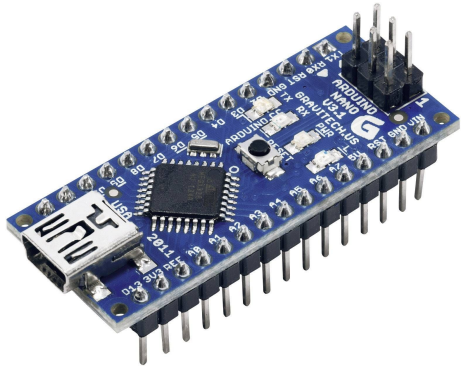
   Your computer.

2. Run ***Arduino IDE*** (*http://*arduino.cc).

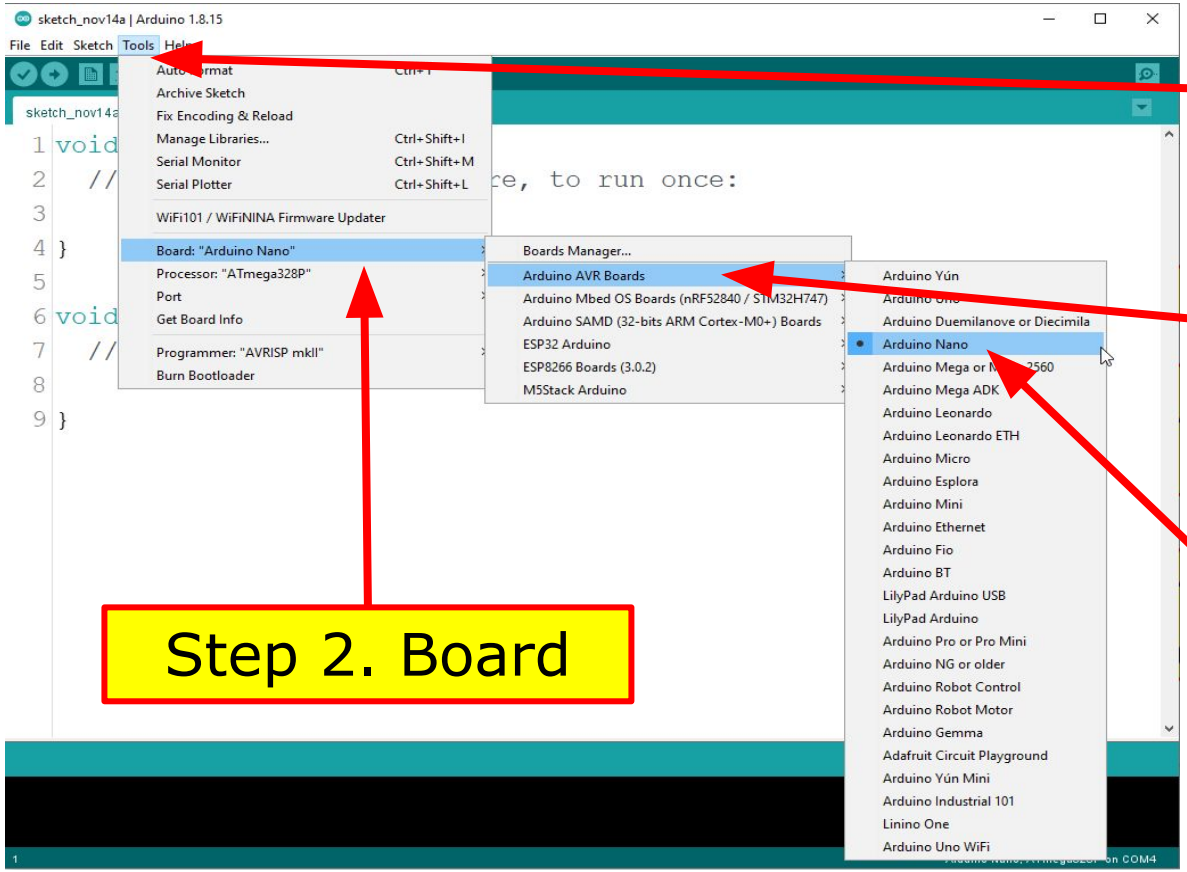
3. Create empty sketch: ***File -> New***.

4. Select proper board.

5. Select proper communication port.



Board selection



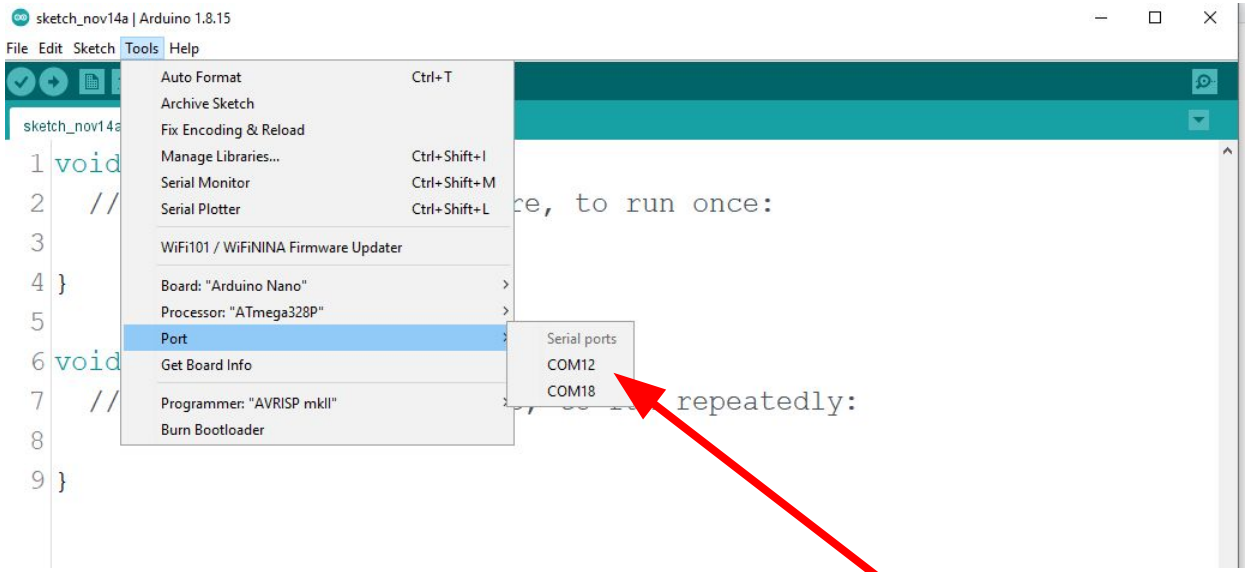
Step 1. Tools

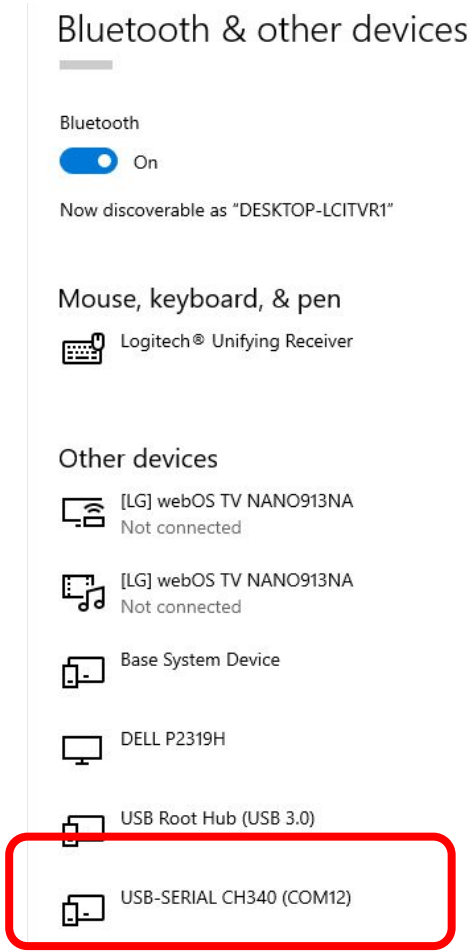
Step 2. Board

Step 3. Arduino AVR Boards

Step 4. Arduino Nano

Port selection





Select communication port

Ex. 01 Board programming

```
void setup() {
  // put your setup code here
  // to run once:

}

void loop() {
  // put your main code here
  // to run repeatedly:

}
```

comments

- multiple lines comment

```
/*
comment
*/
```

- are ignored by the compiler
- inform about the way the program works

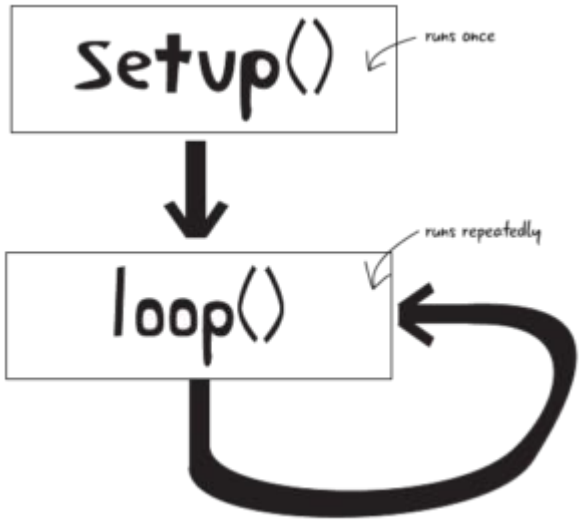
Ex. 01 Board programming

```
void setup() {
  // put your setup code here
  // to run once:

}

void loop() {
  // put your main code here
  // to run repeatedly:

}
```

source: arduinotogo.com

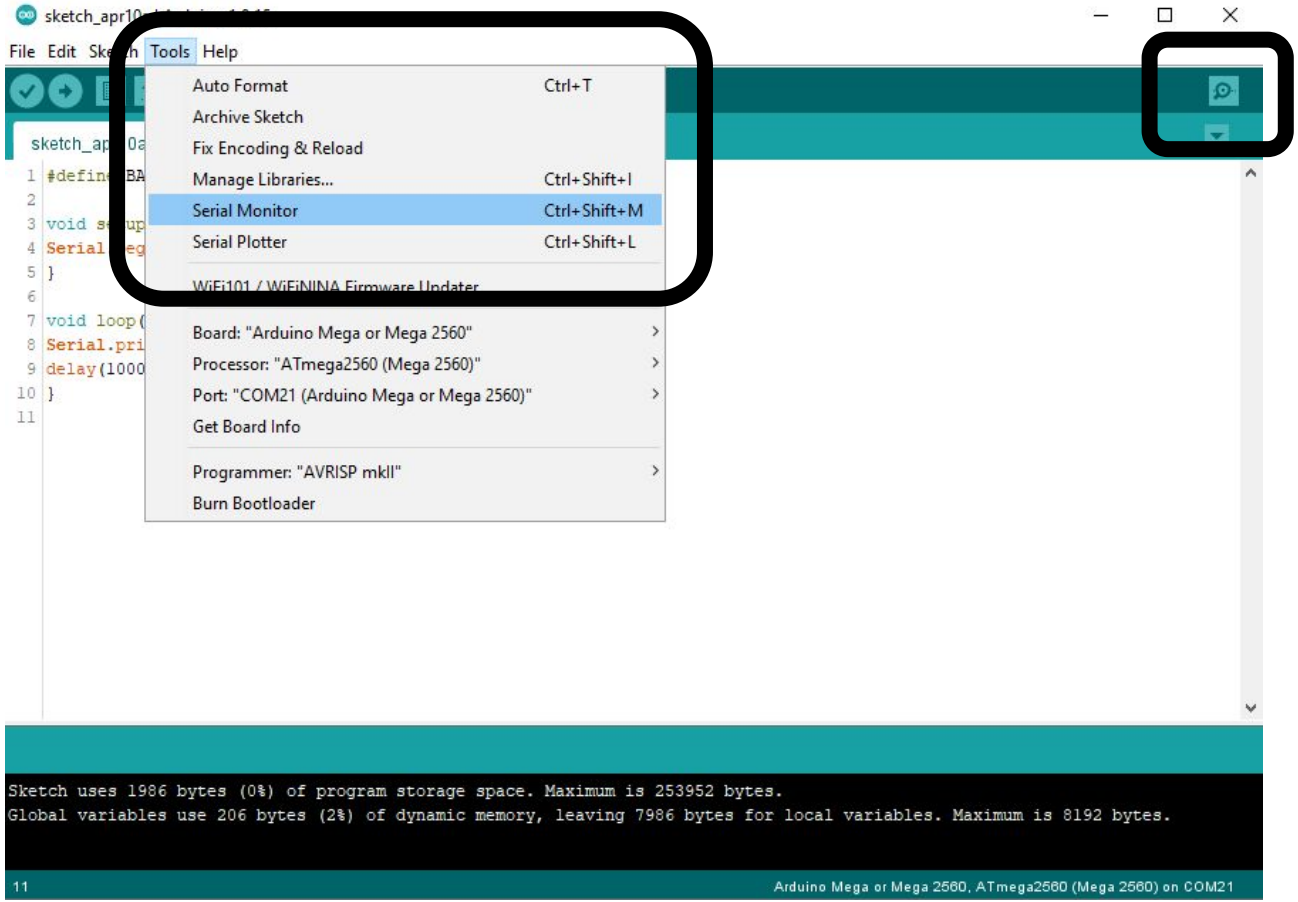
Ex. 02 “Hello Arduino IDE”

```
#define BAUDRATE 115200

void setup() {
Serial.begin(BAUDRATE);
}

void loop() {
Serial.println("Hello Arduino IDE!");
delay(1000);
}
```



end of a line of code

Ex. 02 “Hello Arduino IDE”



Ex. 02 “Hello Arduino IDE”

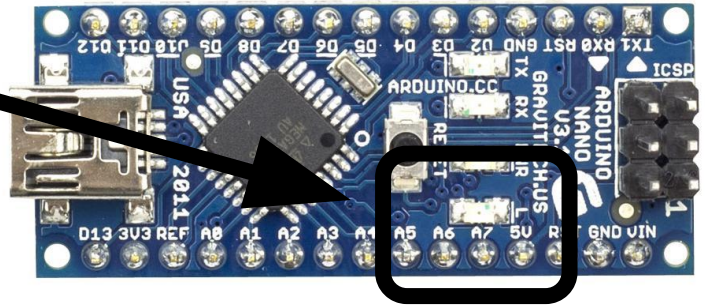
```
#define BAUDRATE 115200

void setup() {
Serial.begin(BAUDRATE);
}

void loop() {
Serial.println("Hello Arduino IDE!");
delay(1000);
}
```

Ex. 03 LED control

```
void setup() {
pinMode(LED_BUILTIN,OUTPUT);
}

void loop() {
digitalWrite(LED_BUILTIN,HIGH);
delay(500);
digitalWrite(LED_BUILTIN,LOW);
delay(500);
}
```

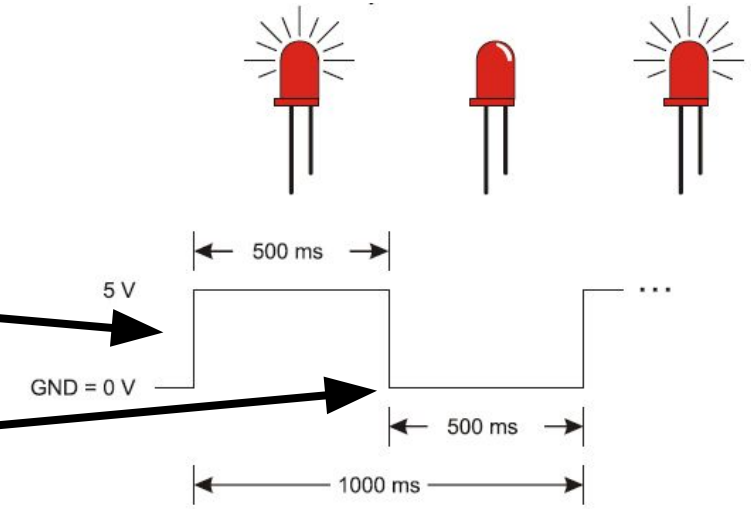
Ex. 03 LED control

```
void setup() {
pinMode(LED_BUILTIN,OUTPUT);
}

void loop() {
digitalWrite(LED_BUILTIN,HIGH);
delay(500);
digitalWrite(LED_BUILTIN,LOW);
delay(500);
}
```

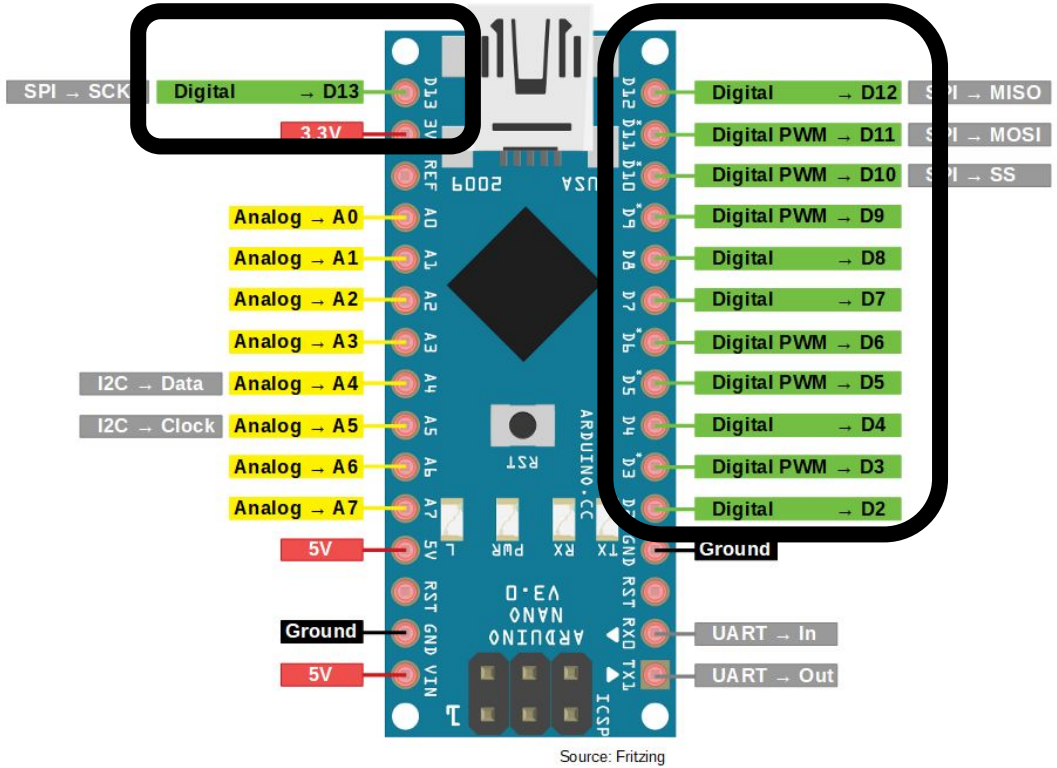


Ex. 03 LED control

```
void setup() {
pinMode(LED_BUILTIN,OUTPUT);
}

void loop() {
digitalWrite(LED_BUILTIN,HIGH);
delay(500);
digitalWrite(LED_BUILTIN,LOW);
delay(500);
}
```



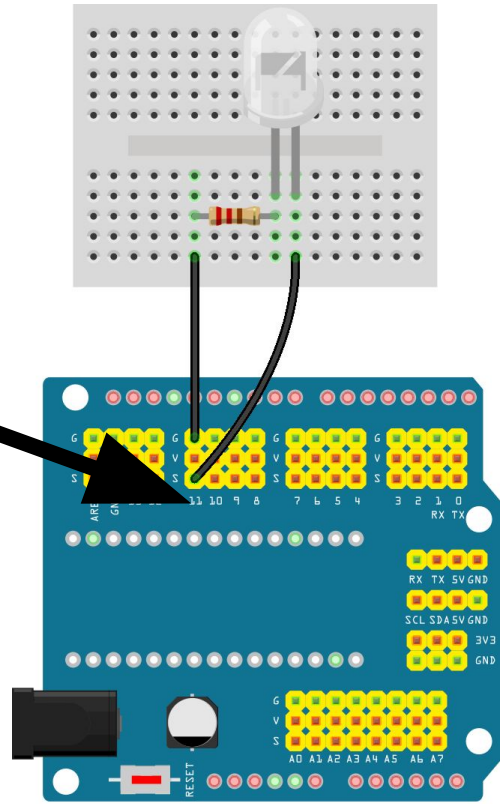
General Purpose Input/Output



LED_BUILTIN is connected to the GPIO no 13

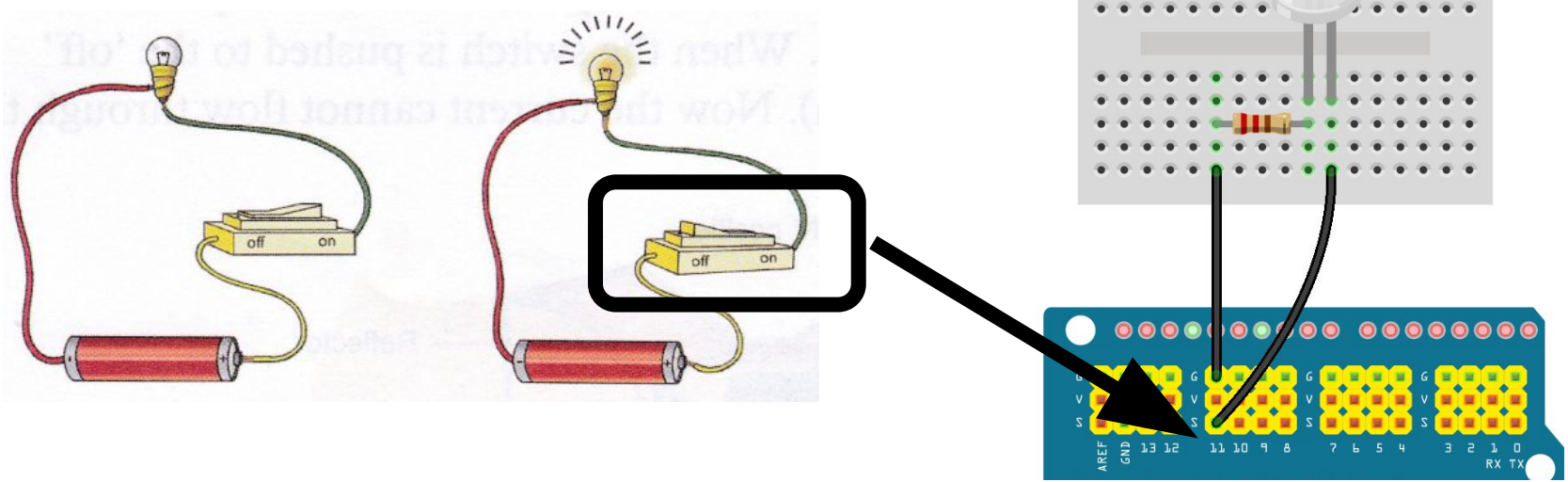
Ex. 04 External LED control

```
void setup() {
pinMode(11,OUTPUT);
}

void loop() {
digitalWrite(11,HIGH);
delay(500);
digitalWrite(11,LOW);
delay(500);
}
```



General Purpose Input/Output

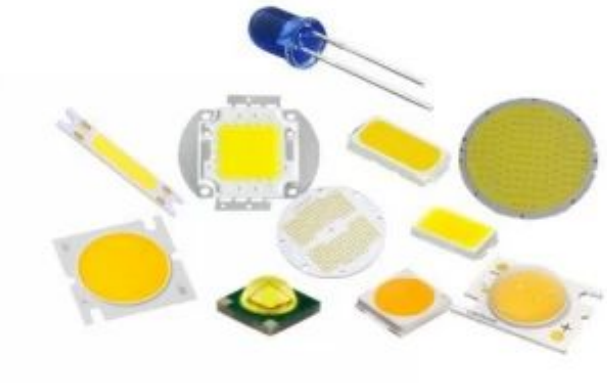
"0"	0V	LOW	OFF
"1"	5V	HIGH	ON

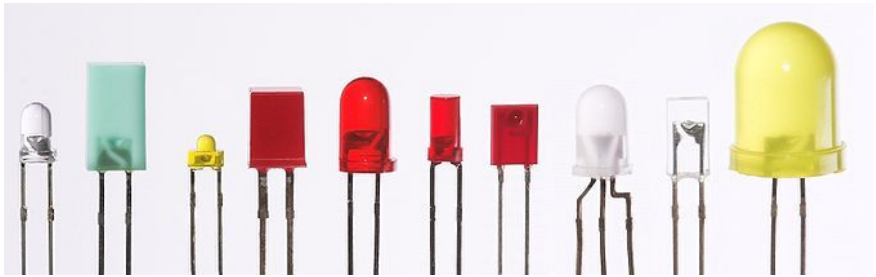


Light Emitting Diodes

LED is an electronic component that converts electrical energy into light.

LED stands for "**L**ight **E**mitting **D**iode."





Light Emitting Diodes

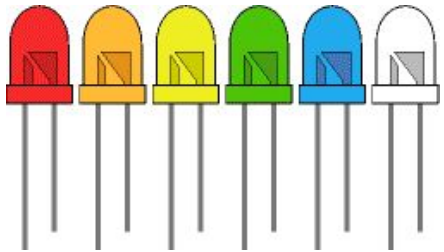
LEDs are like tiny light bulbs, but by comparison to conventional light bulbs:

- are DC components;
- require a lot less power to light up;
- are more energy efficient.

Light Emitting Diodes

Color	Wavelength [nm]	Semiconductor material
Infrared	$\lambda > 760$	Gallium arsenide (GaAs) Aluminium gallium arsenide (AlGaAs)
Red	$610 < \lambda < 760$	Aluminium gallium arsenide (AlGaAs) Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
Orange	$590 < \lambda < 610$	Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
Yellow	$570 < \lambda < 590$	Gallium arsenide phosphide (GaAsP) Aluminium gallium indium phosphide (AlGaInP) Gallium(III) phosphide (GaP)
Green	$500 < \lambda < 570$	**Traditional green:** Gallium(III) phosphide (GaP) Aluminium gallium indium phosphide (AlGaInP) Aluminium gallium phosphide (AlGaP) **Pure green:** Indium gallium nitride (InGaN) / Gallium(III) nitride (GaN)

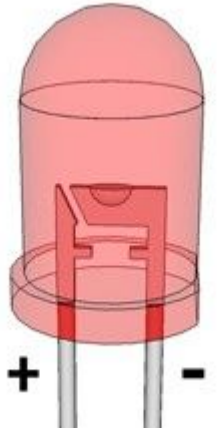
Color	Wavelength [nm]	Semiconductor material
Blue	$450 < \lambda < 500$	Zinc selenide (ZnSe) Indium gallium nitride (InGaN) Silicon carbide (SiC) as substrate Silicon (Si) as substrate—under development
Violet	$400 < \lambda < 450$	Indium gallium nitride (InGaN)
Purple	multiple types	Dual blue/red LEDs, blue with red phosphor, or white with purple plastic
Ultraviolet	$\lambda < 400$	Diamond (235 nm) Boron nitride (215 nm) Aluminium nitride (AlN) (210 nm) Aluminium gallium nitride (AlGaN) Aluminium gallium indium nitride (AlGaInN)—down to 210 nm
Pink	multiple types	Blue with one or two phosphor layers: yellow with red, orange or pink phosphor added afterwards, or white with pink pigment or dye.
White	Broad spectrum	Blue/UV diode with yellow phosphor

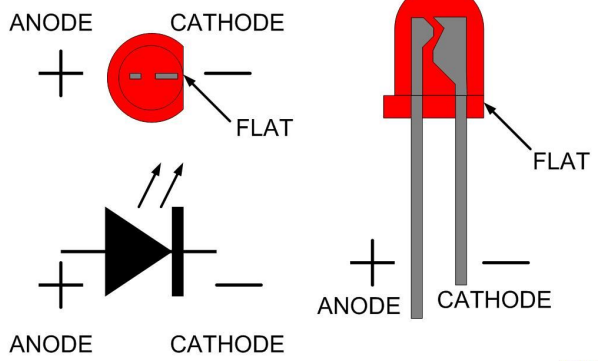


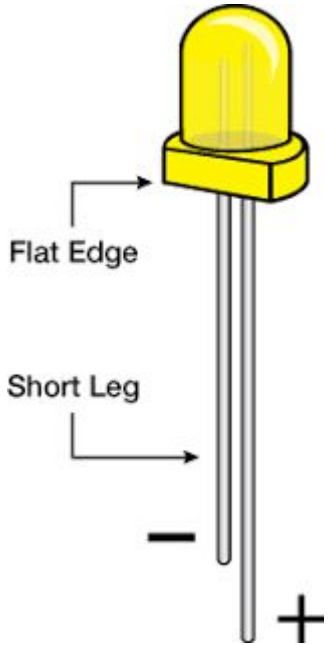
Light Emitting Diodes

Polarity matters.

LEDs, being diodes, will only allow current to flow in one direction







Light Emitting Diodes

ARDUINO

More current equals more light.

- The brightness of an LED is directly dependent on how much current it draws.
- Super bright LEDs drain batteries more quickly, because the extra brightness comes from the extra power being used.
- You can control the brightness of an LED by controlling the amount of current through it.

Light Emitting Diodes

ARDUINO

There is such a thing as too much power.

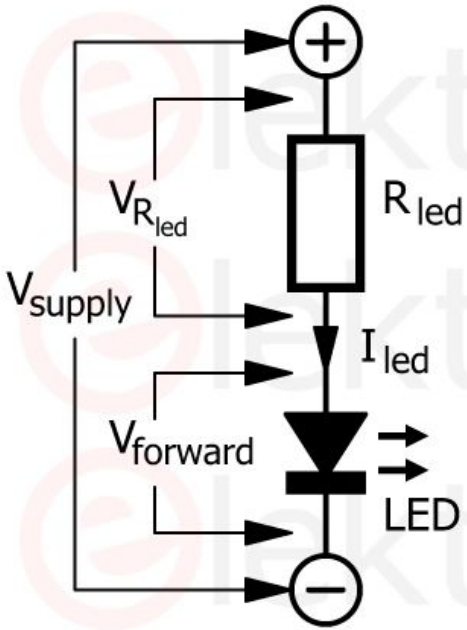
If you connect an LED directly to a current source it will try to dissipate as much power as it's allowed to draw, and, like the tragic heroes of olde, it will destroy itself. That's why it's important to limit the amount of current flowing across the LED.

Light Emitting Diodes

ITEMS	Symbol	Absolute Maximum Rating	Unit
Forward Current	I_F	20	mA
Peak Forward Current	I_{FP}	30	mA
Suggestion Using Current	I_{su}	16-18	mA
Reverse Voltage (V_R=5V)	I_R	10	uA
Power Dissipation	P_D	105	mW
Operation Temperature	T_{OPR}	-40 ~ 85	℃
Storage Temperature	T_{STG}	-40 ~ 100	℃
Lead Soldering Temperature	T_{SOL}	Max. 260℃ for 3 Sec. Max. (3mm from the base of the expoxy bulb)	

ITEMS	Symbol	Test condition	Min.	Typ.	Max.	Unit
Forward Voltage	V_F	I_F=20mA	1.8	---	2.2	V
Wavelenength (nm) or TC(k)	$\Delta\lambda$	I_F=20mA	620	---	625	nm
*Luminous intensity	I_V	I_F=20mA	150	---	200	mcd

Light Emitting Diodes

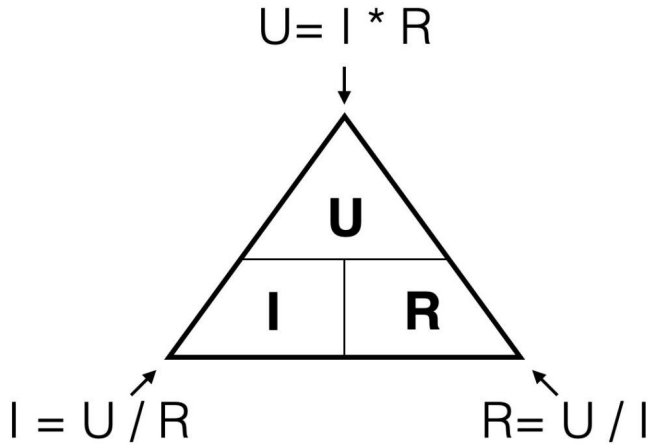


$$V_{R_{led}} = V_{supply} - V_{forward}$$

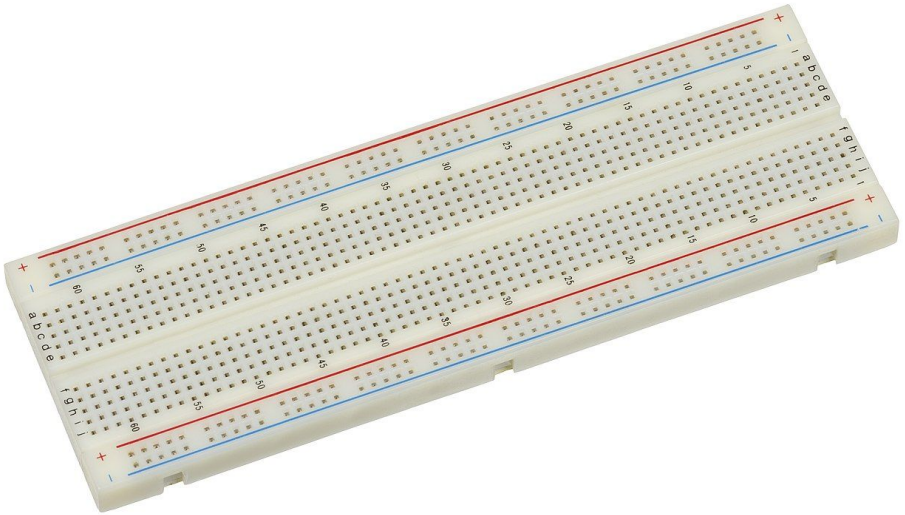
$$R_{led} = \frac{V_{R_{led}}}{I_{led}}$$

$$P_{R_{led}} = I_{led} \times V_{R_{led}}$$

$$P_{LED} = I_{led} \times V_{forward}$$



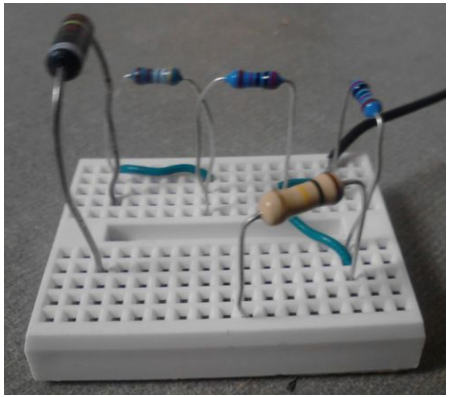
Breadboard

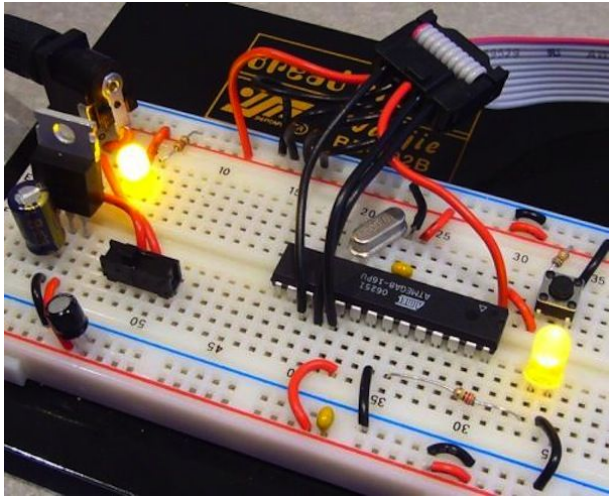


A **breadboard**, or protoboard, is a construction base for prototyping of electronics.

An electronics breadboard is actually referring to a **solderless breadboard**. These are great units for making temporary circuits and prototyping, and they require no soldering.

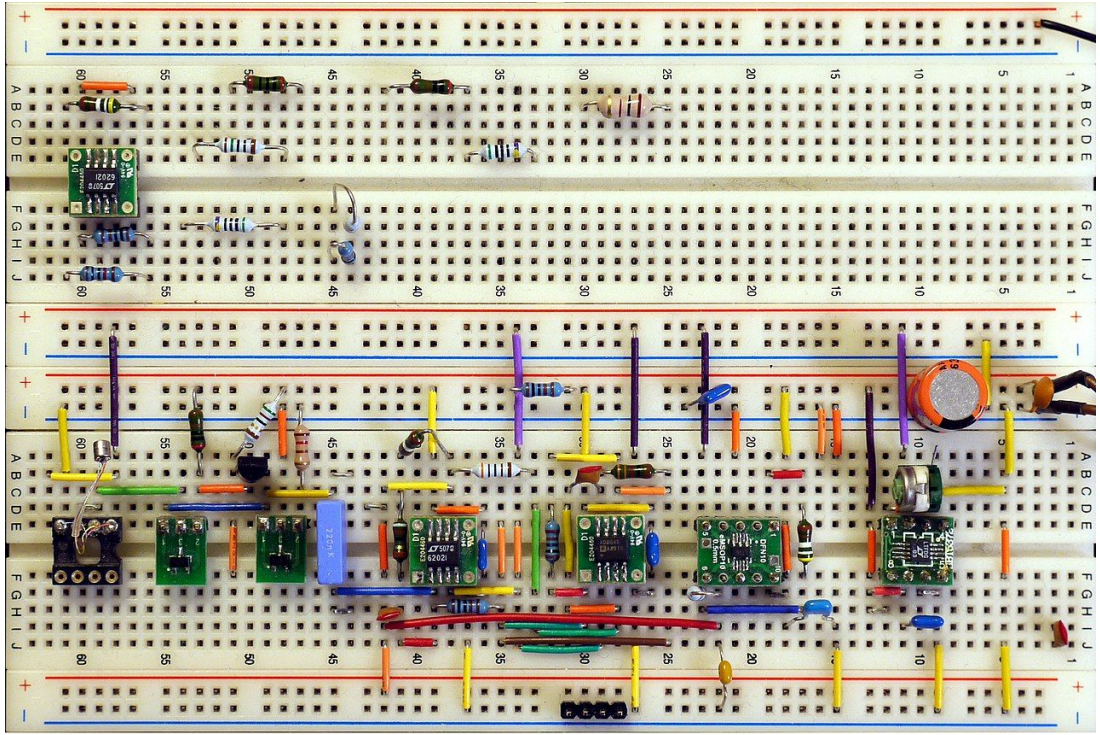
Breadboard

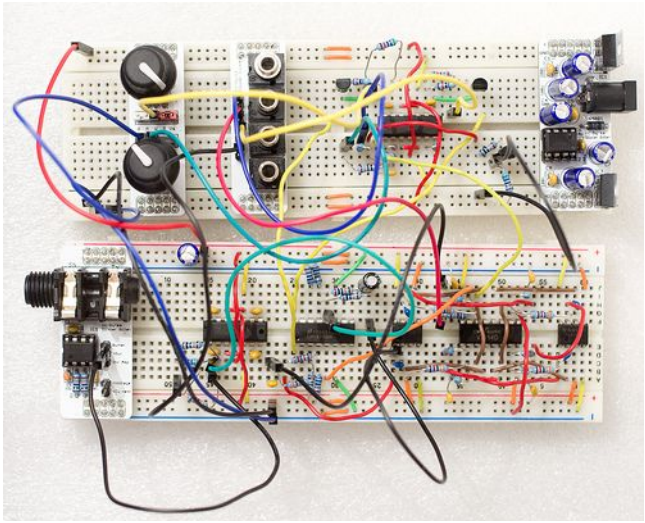


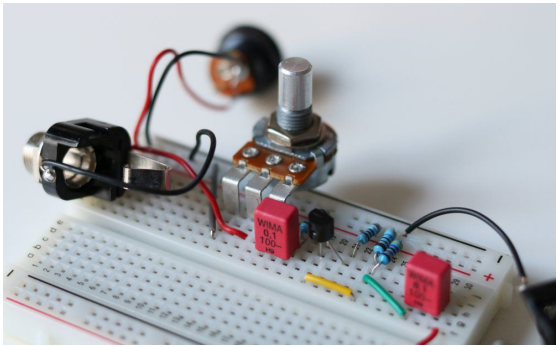


Prototyping is the process of testing out an idea by creating a preliminary model from which other forms are developed or copied. If you aren't sure how a circuit will react under a given set of parameters, it's best to build a prototype and test it out.

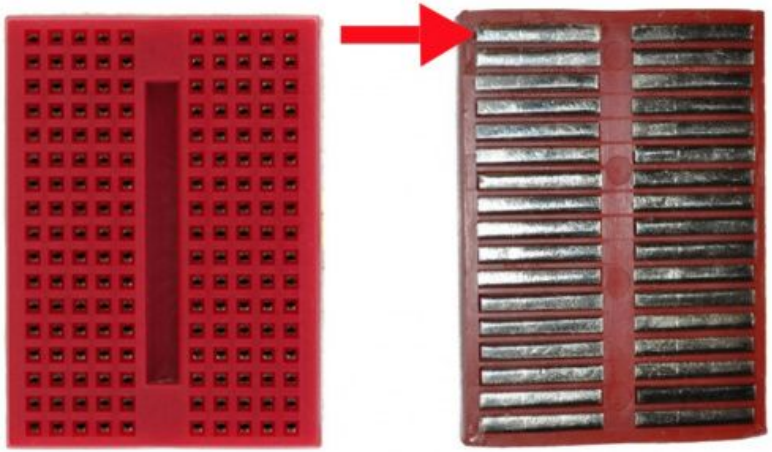
Breadboard

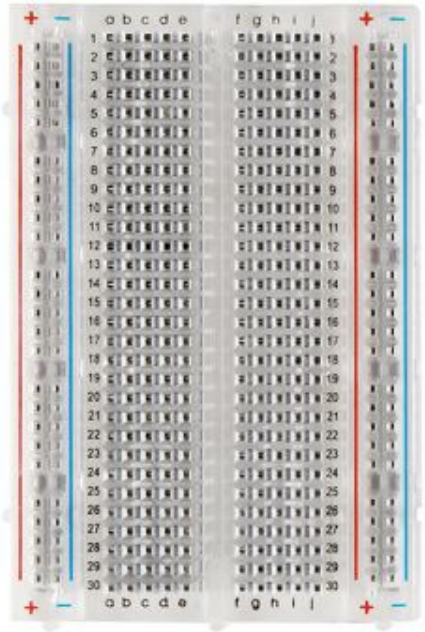






Breadboard





Ex. 04 External LED control

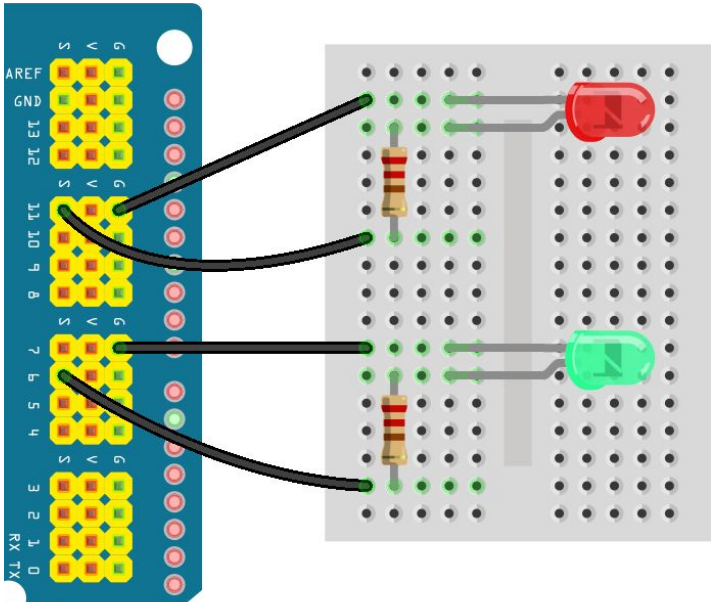
```
#define MY_LED 11

#define MY_DELAY 1000



void setup() {

pinMode(MY_LED,OUTPUT);

}
```

```
void loop() {

digitalWrite(MY_LED,HIGH);

delay(MY_DELAY/2);

digitalWrite(MY_LED,LOW);

delay(MY_DELAY/2);

}
```

Ex. 05 LEDs control

Blink the LEDs alternately. Send information about the LEDs' state to Your computer. Blinking period equals 1s.

```
#define LED_RED 11
#define LED_GREEN 6
#define MY_DELAY 1000

void setup() {
pinMode(LED_RED,OUTPUT);
pinMode(LED_GREEN,OUTPUT);
}
```

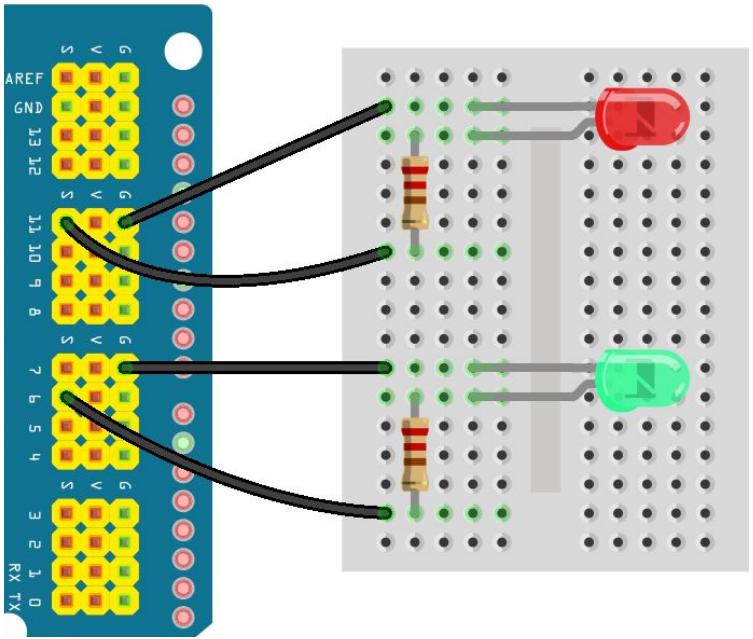


Ex. 05 LEDs control

```
void loop() {

    digitalWrite(LED_RED,HIGH);

    digitalWrite(LED_GREEN,LOW);

    delay(MY_DELAY/2);

    digitalWrite(LED_RED,LOW);

    digitalWrite(LED_GREEN,HIGH);

    delay(MY_DELAY/2);

}
```

Ex. 06 LEDs control - sequence

Create a sequence of lights according to the table. Send information about current event to Your PC.



event	LEDs	time
1	LED_R is on LED_G is off	1s
2	LED_R is off LED_G is on	1s
3	LED_R is off LED_G is off	1s

Ex. 06 LEDs control - sequence

```
#define LED_R 11
#define LED_G 6
#define MY_DELAY 1000
#define BAUDRATE 115200

void setup() {
pinMode(LED_R,OUTPUT);
pinMode(LED_G,OUTPUT);
Serial.begin(BAUDRATE);
}
```

Ex. 06 LEDs control - sequence

```
void loop() {
// event 1
digitalWrite(LED_R,HIGH);
digitalWrite(LED_G,LOW);
Serial.println("Event 1");
delay(MY_DELAY);

// event 2
digitalWrite(LED_R,LOW);
digitalWrite(LED_G,HIGH);
Serial.println("Event 2");
delay(MY_DELAY);
```

```
// event 3
digitalWrite(LED_R,LOW);
digitalWrite(LED_G,LOW);
Serial.println("Event 3");
delay(MY_DELAY);
}
```

Ex. 07 LEDs control - sequence

Use Task 06 circuit. Create a sequence of lights according to the tables.

event	LED	time
1	LED_G is on	0.025s
2	LED_G is off	0.001s

event	LED	time
1	LED_R is off	0.025s
2	LED_R is on	0.001s

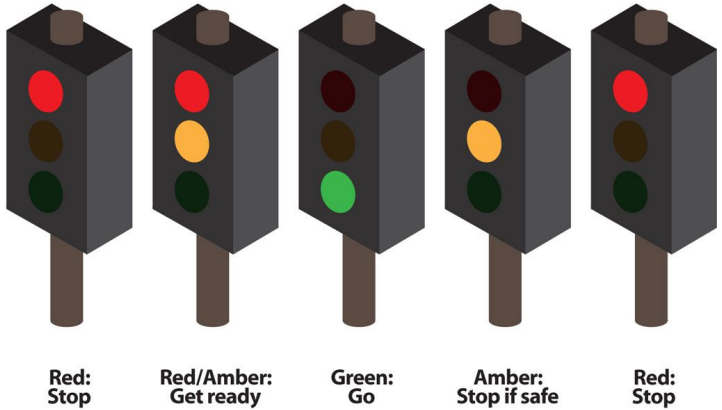
Ex. 07 LEDs control - sequence

```
#define LED_R 11
#define LED_G 6
#define MY_DELAY 25

void setup() {
pinMode(LED_R,OUTPUT);
pinMode(LED_G,OUTPUT);
}
```

```
// event 1
digitalWrite(LED_G,HIGH);
digitalWrite(LED_R,LOW);
delay(MY_DELAY);

// event 2
digitalWrite(LED_G,LOW);
digitalWrite(LED_R,HIGH);
delay(MY_DELAY/25);
}
```

Additional tasks

1. Create a traffic lights simulator.
2. Send information about current signal to Your PC.
3. Improve Your project by adding pedestrian signals.



Summary communication

Sets the data rate in bits per second (baud) for serial data transmission:

```
Serial.begin(BAUDRATE);
```

Prints data to the serial port as human-readable text followed by a new line character

```
Serial.println("My message");
```

Summary GPIOs

Configure the specified pin to behave as an output:

```
pinMode(pin,OUTPUT);
```

Write a HIGH or a LOW value to a digital pin:

```
digitalWrite(pin,HIGH);
```

or

```
digitalWrite(pin,LOW);
```

For those interested

Arduino Mini Course

randomnerdtutorials.com/arduino-mini-course-access/

How to Use a Breadboard

learn.sparkfun.com/tutorials/how-to-use-a-breadboard