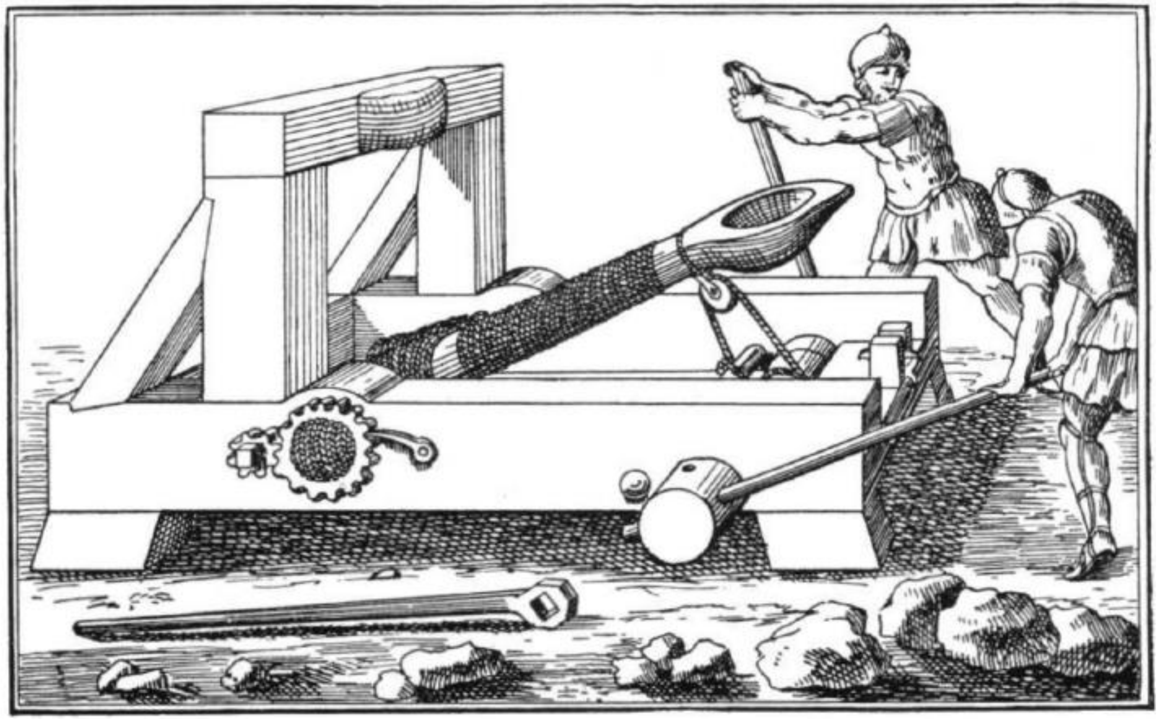
Engineering Design Project I
UTA013

Dr. Poonam Verma
Assistant Professor
ECED, TIET, Patiala

Engineering Design Project-I: Mangonel

- In Engineering Design Project-I, Mangonel (Roman catapult) is to be designed and implemented.



Engineering Design Project-I: Mangonel

- The Mangonel project allows students work together in small groups to learn the principles of mathematical modeling, software development, electronic hardware design, measurement using sensors and instrumentation, structural analysis, mechanical assembly and design.

- The Mangonel project fosters innovation and enables students to develop their creative skills in coherent and structured manner.

Mangonel

- The Mangonel project is interdisciplinary and requires knowledge of both electronics and mechanical engineering.
- **Mechanical part:**
    - Construction of Mangonel
    - Structural analysis of certain key components of Mangonel for static and dynamic stress using values of material properties.
- **Electronics part:**
    - Development of micro-electronic system
        - To determine the angular velocity of the throwing arm.
        - Angle of launch of the Mangonel

It contains infrared optical sensors, ultrasonic sensors, electronics components e.g. JK flip-flop, AND gate, 7-segment display, Arduino Uno kit, etc.

Mangonel: Electronics Part

The Electronic Part is divided into 4 sections:

- Programming of Arduino Digital I/O pins for various applications.
- Sensing any activity through Arduino and instructing accordingly. Also, data capturing through sensors.
- Interfacing of hardware and software to do a specific task (using 7-segment display)
- Develop a micro-electronic circuit to determine and display the angular velocity of the throwing arm.

Mangonel: Electronics Part

The Electronic Part is divided into 4 sections:

- Programming of Arduino Digital I/O pins for various applications.
- Sensing any activity through Arduino and instructing accordingly. Also, data capturing through sensors.
- Interfacing of hardware and software to do a specific task (using 7-segment display)
- Develop a micro-electronic circuit to determine and display the angular velocity of the throwing arm.

Electronic Components

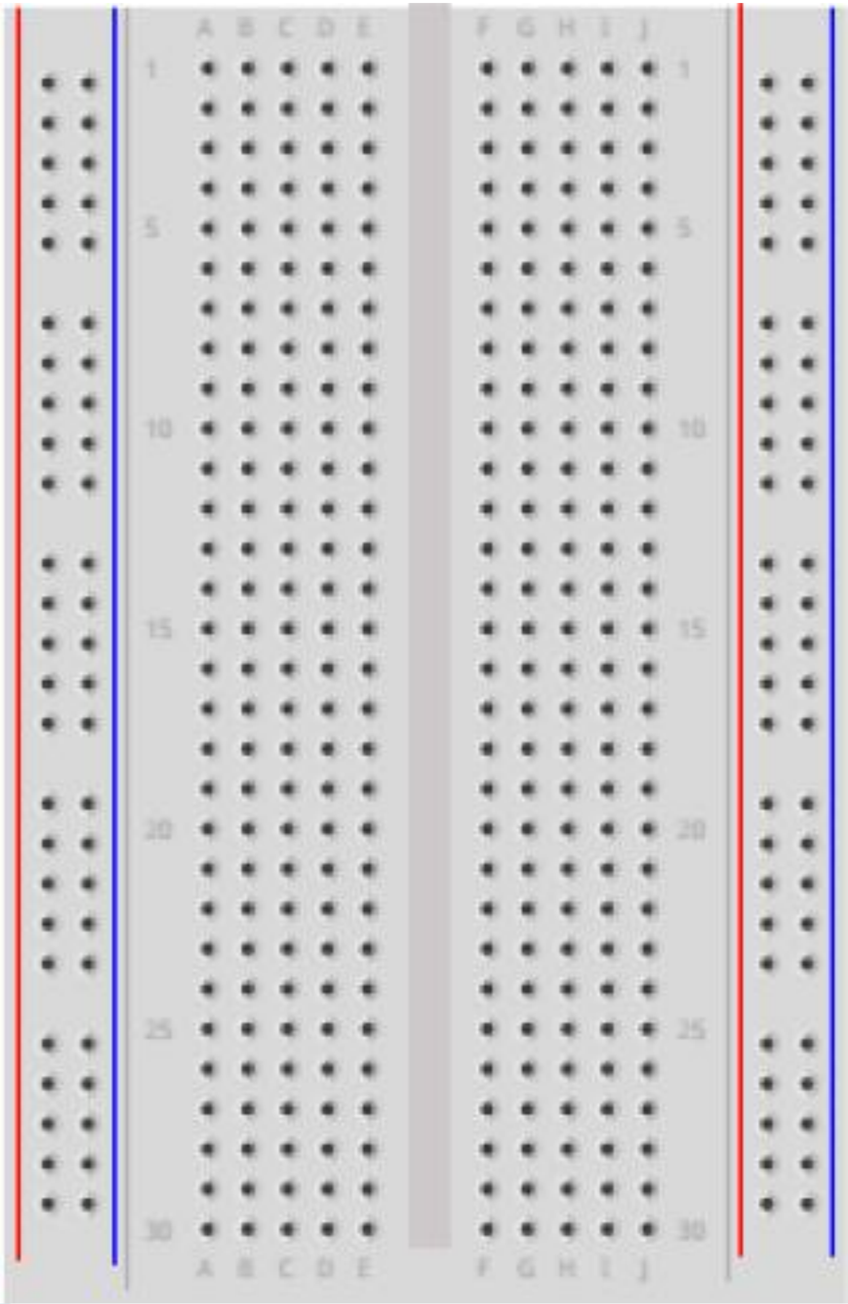
Electronic components

For this project, you should be familiar with

- Breadboard
- Switches
- Resistors, Light Emitting Diode (LED)
- Logic gates, flip-flops and ICs
- Seven segment display
- BCD to Seven segment decoder
- IR Sensors
- Arduino Board

Breadboard

- Breadboard contains holes to hold different electronic components.
- These holes are broken into two sets of rows and sets of columns.
- The columns are named as A, B, C, D, E, F, G, H, I, and J.
- The rows are numbered from 1 to 30/64.

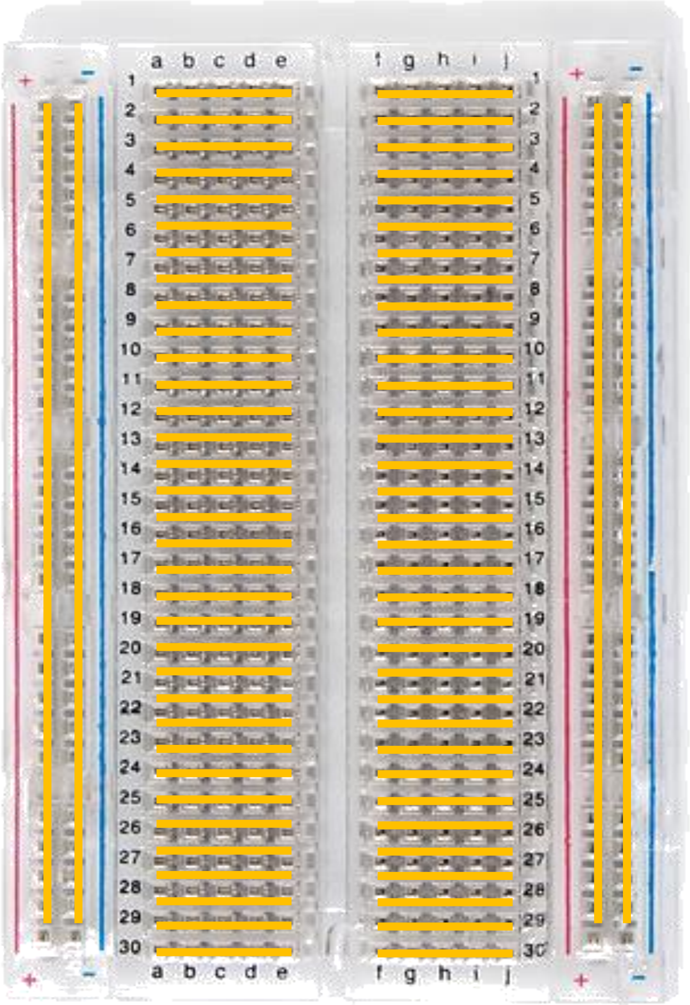


Breadboard

- Columns at the edges are connected from top to bottom inside the breadboard generally used for supply and ground.

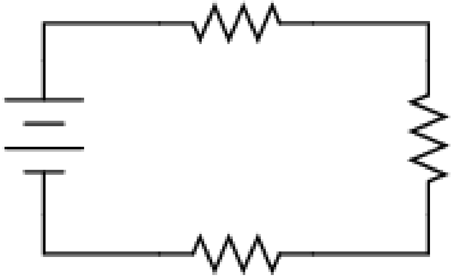
- Inside the breadboard, the holes in each row are connected up to the break in the middle of the board.

- For example, A1, B1, C1, D1, E1 all have a wire inside of the breadboard to connect them. Then F1, G1, H1, I1, and J1 are all connected but A1 is not connected to F1.

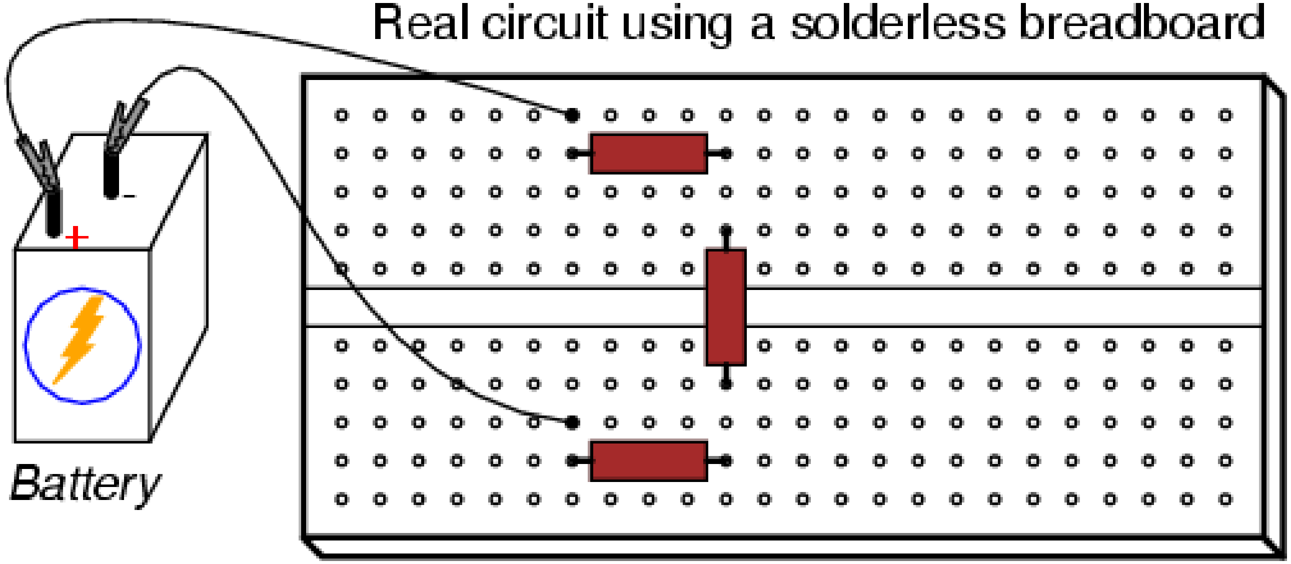


Circuit Implementation on Breadboard

Schematic diagram



Real circuit using a solderless breadboard



Resistor Color Bands

- In electronics, color codes are **a way of labeling** of passive electronic components such as **resistors.**

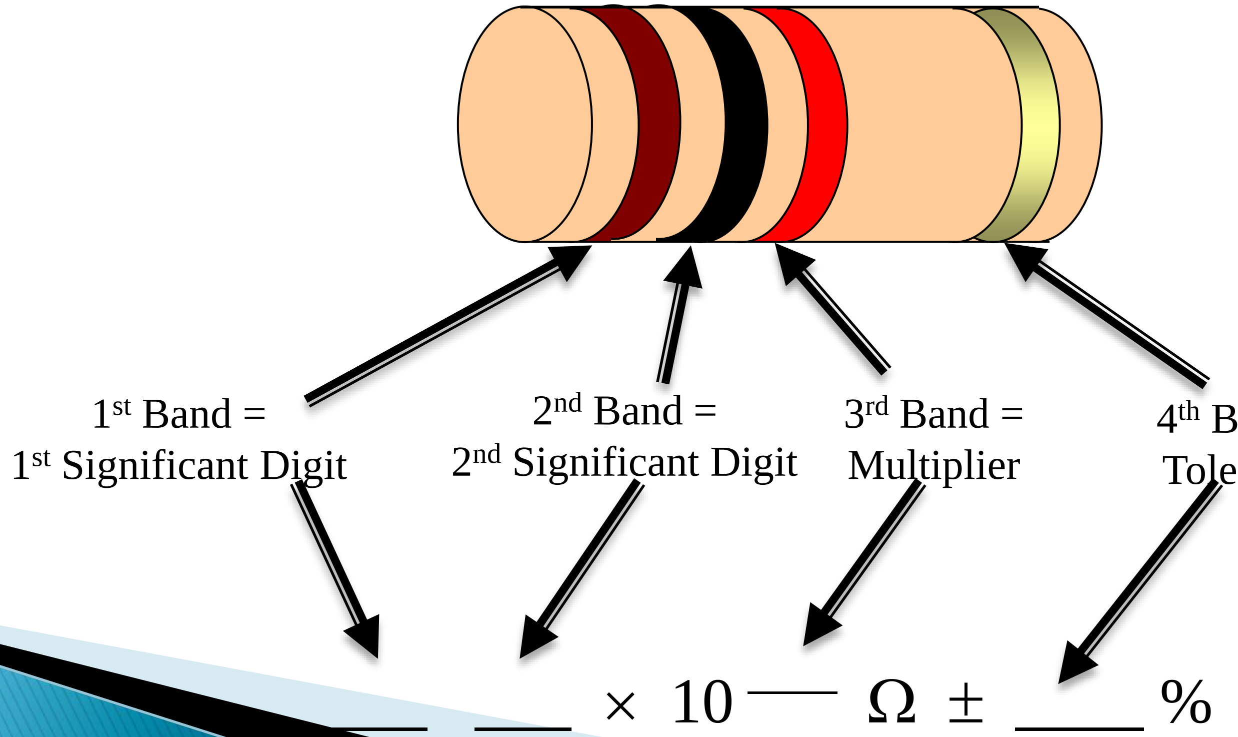
- It important to know, how to identify the nominal resistance and the tolerance of a resistor.

- For resistors with ±5% or ±10% tolerance, the color code consists of 4 color bands.

- For resistors with ±1% or ±2% tolerance, the color code consists of 5 bands.

4-Band Resistors

- The resistor nominal value is encoded in the color code in Powers of Ten Notation.
- The template for determining the nominal value and tolerance of a resistor with 4 color bands is given below:



1st Band =
1st Significant Digit

2nd Band =
2nd Significant Digit

3rd Band =
Multiplier

4th Band =
Tolerance

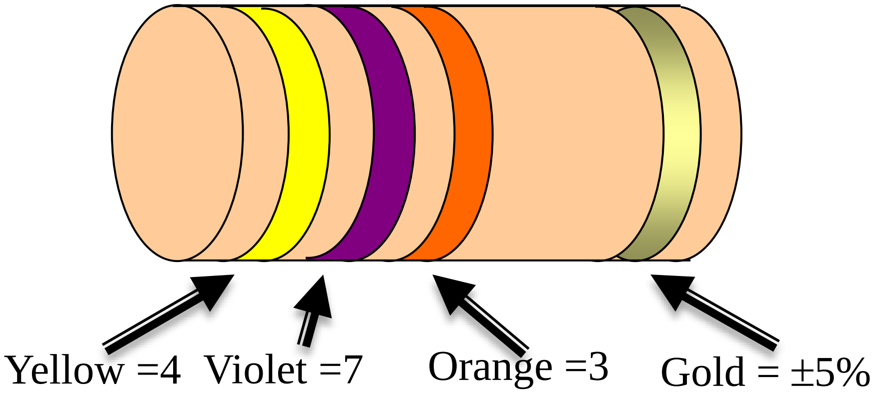
$\_\_\_\ \ \_\_\_ \times 10^{\_\_\_}\ \Omega \pm \_\_\_\_\ \%$

Color Code Table

	Color	Digit	Multiplier	Tolerance
⬛	Black	0	$10^0 = 1$	
🟫	Brown	1	$10^1 =10$	±1%
🟥	Red	2	$10^2 = 100$	±2%
🟧	Orange	3	$10^3 =1,000$	
🟨	Yellow	4	$10^4 = 10,000$	
🟩	Green	5	$10^5 =100,000$	
🟦	Blue	6	$10^6 = 1,000,000$	
🟪	Violet	7	$10^7 = 10,000,000$	
◼	Gray	8	$10^8 = 100,000,000$	
⬜	White	9	$10^9 = 1,000,000,000$	
▨	Silver		$10^{-2} = 0.01$	±10%
🟨	Gold		$10^{-1} = 0.1$	±5%
	No band	---	----------	±20%

Resistance Computation: Example-1

- Determine the nominal value and tolerance for the resistor below.
- What is the min. resistance value this resistor can actually have?
- What is the max resistance value this resistor can actually have?



$$\underline{\ 4\ }\quad \underline{\ 7\ } \times 10^{\underline{3}}\ \Omega \pm \underline{\ 5\ }\ \%$$

Resistor nominal value $= 47\times10^{3}\Omega$

$= 47{,}000\Omega$

$= 47\text{k}\Omega.$

Tolerance = ±5%

	Color	Digit	Multiplier	Tolerance
⬛	Black	0	$10^{0} = 1$	
🟫	Brown	1	$10^{1} = 10$	±1%
🟥	Red	2	$10^{2} = 100$	±2%
🟧	Orange	3	$10^{3} = 1{,}000$	
🟨	Yellow	4	$10^{4} = 10{,}000$	
🟩	Green	5	$10^{5} = 100{,}000$	
🟦	Blue	6	$10^{6} = 1{,}000{,}000$	
🟪	Violet	7	$10^{7} = 10{,}000{,}000$	
▨	Gray	8	$10^{8} = 100{,}000{,}000$	
⬜	White	9	$10^{9} = 1{,}000{,}000{,}000$	
▨	Silver		$10^{-2} = 0.01$	±10%
▨	Gold		$10^{-1} = 0.1$	±5%
	No band	---	---------	±20%

Resistance Computation

- **Minimum resistance value:**

  Multiply the nominal value by the tolerance and then *subtract* this from the nominal value:

$$= 47k\Omega - 47k\Omega * 0.05$$

$$= 47k\Omega - 2.35k\Omega$$

$$= 44.65k\Omega$$

- **Maximum resistance value:**

  Multiply the nominal value by the tolerance and then *add* this to the nominal value:

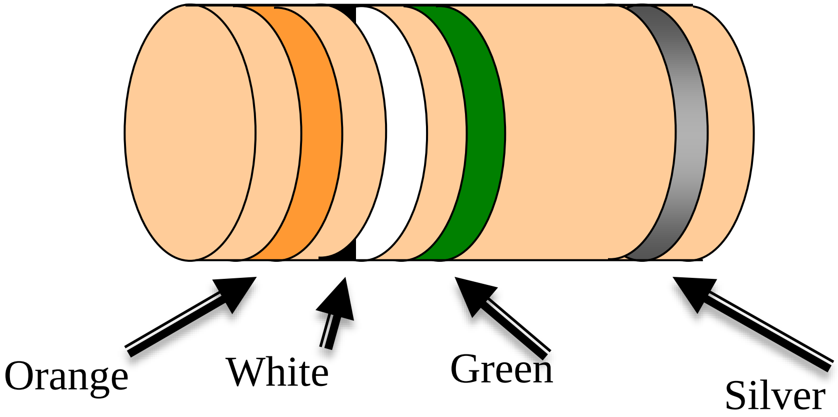
$$= 47k\Omega + 47k\Omega * 0.05$$

$$= 47k\Omega + 2.35k\Omega$$

$$= 49.35k\Omega$$

Example-2

- Determine the nominal value and tolerance for the resistor below.
- What is the min resistance value this resistor can actually have?
- What is the max resistance value this resistor can actually have?



$$\underline{3}\ \ \underline{9} \times 10^{\underline{5}}\ \Omega \pm \underline{10}\ \%$$

Resistor nominal value $= 39\times10^5\Omega$

$= 3{,}900{,}000\Omega$

$= 3.9\text{M}\Omega.$

Tolerance = ±10%

	Color	Digit	Multiplier	Tolerance
⬛	Black	0	$10^0 = 1$	
🟫	Brown	1	$10^1 = 10$	±1%
🟥	Red	2	$10^2 = 100$	±2%
🟧	Orange	3	$10^3 = 1{,}000$	
🟨	Yellow	4	$10^4 = 10{,}000$	
🟩	Green	5	$10^5 = 100{,}000$	
🟦	Blue	6	$10^6 = 1{,}000{,}000$	
🟪	Violet	7	$10^7 = 10{,}000{,}000$	
▨	Gray	8	$10^8 = 100{,}000{,}000$	
⬜	White	9	$10^9 = 1{,}000{,}000{,}000$	
▨	Silver		$10^{-2} = 0.01$	±10%
▨	Gold		$10^{-1} = 0.1$	±5%
	No band	---	---------	±20%

Resistance Computation

- **Minimum resistance value:**

  Multiply the nominal value by the tolerance and then *subtract* this from the nominal value:

$$= 3.9M\Omega - 3.9M\Omega * 0.1$$

$$= 3.9M\Omega - 0.39M\Omega$$

$$= 3.51M\Omega$$

- **Maximum resistance value:**

  Multiply the nominal value by the tolerance and then *add* this to the nominal value:

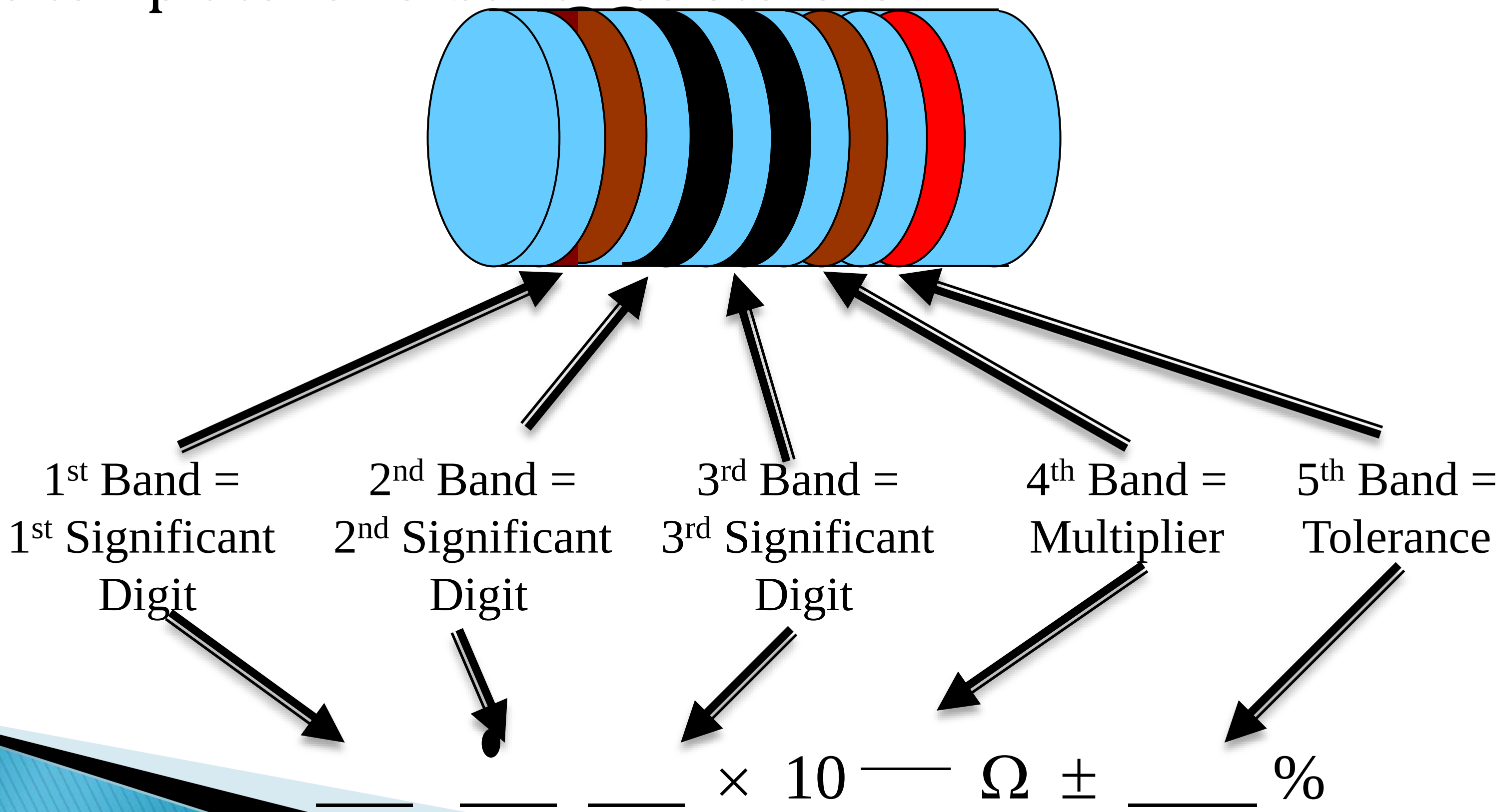
$$= 3.9M\Omega + 3.9M\Omega * 0.1$$

$$= 3.9M\Omega + 0.39M\Omega$$

$$= 4.29M\Omega$$

5-Band Resistor

- For resistors with ±1% or ±2% tolerance, the color code consists of 5 bands.
- The template for 5-band resistors is:



1st Band =
1st Significant
Digit

2nd Band =
2nd Significant
Digit

3rd Band =
3rd Significant
Digit

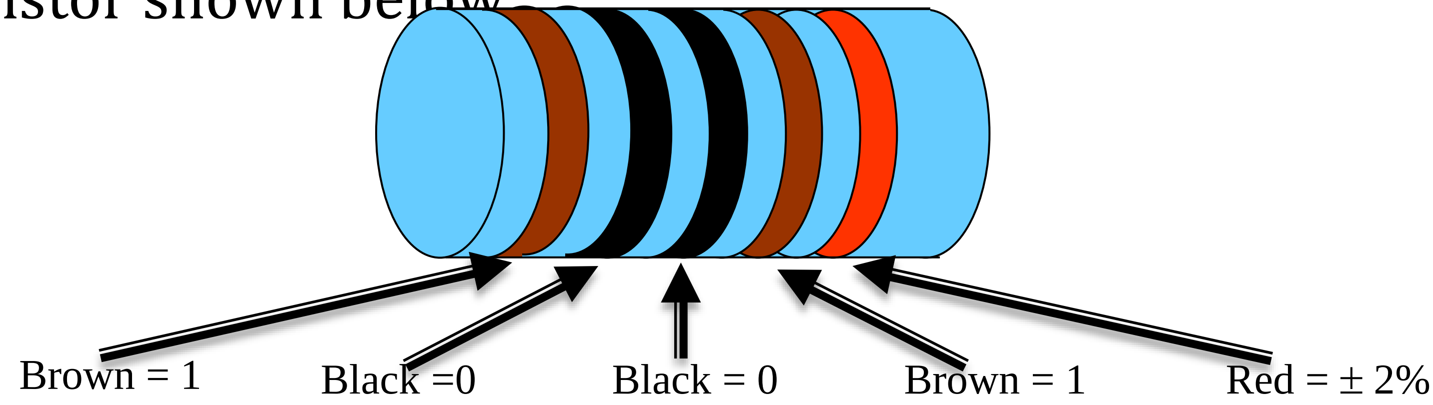
4th Band =
Multiplier

5th Band =
Tolerance

___ ___ ___ × 10 ___ Ω ± ____ %

Example

- Determine the nominal resistance and tolerance for the resistor shown below.



Brown = 1

Black =0

Black = 0

Brown = 1

Red = ± 2%

$$\underline{\ 1\ }\ \ \underline{\ 0\ }\ \ \underline{\ 0\ } \times 10^{\underline{\ 1\ }}\ \Omega \pm \underline{\ 2\ }\ \%$$

Resistor nominal value $= 100\times10^{1}\Omega$

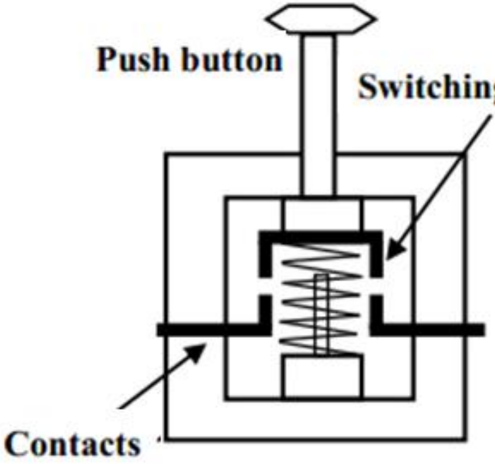
$= 1{,}000\Omega$

$= 1\text{k}\Omega.$

Tolerance = ±2%

Push Buttons

- A push button is switch used to close or open an electronic control circuit.
- Pushing a button causes wire under the button to be connected (closed circuit), allowing current to flow. When the button isn't pressed, no current can flow because the wires are not connected (open circuit).
- You can see that pushing down on the top causes there to be connection and spring causes it to not be connected when it is not being pushed down.

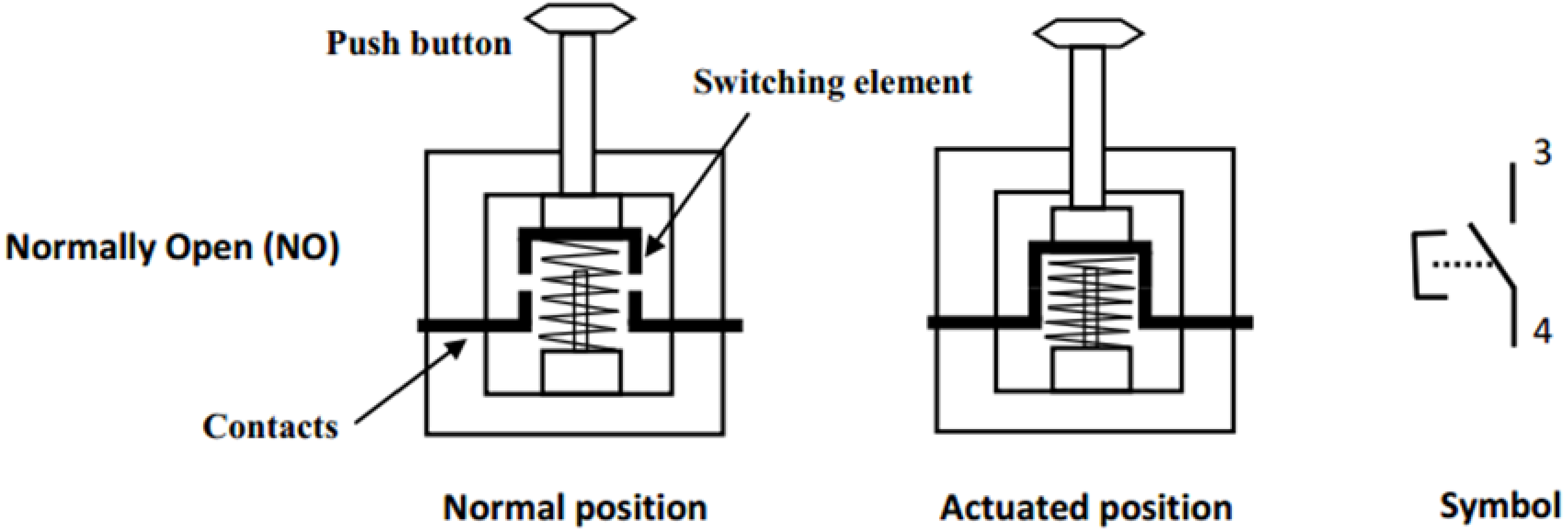


Push Buttons: Types

- Push buttons are of two types:
  - Momentary push button
  - Maintained contact or detent push button
- Momentary push buttons return to their unactuated position when they are released.
- Maintained (or mechanically latched) push buttons has a latching mechanism to hold it in the selected position.
- The contact of push buttons, distinguished according to their functions:
  - Normally Open (NO) type
  - Normally Closed (NC) type
  - Change Over (CO) type

NO-Type Push Button

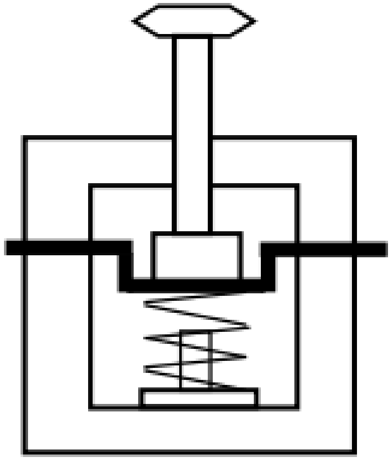
- In the NO type, the contacts are open in the normal position, inhibiting the energy flow through them. But in the actuated position, the contacts are closed, permitting the energy flow through them.



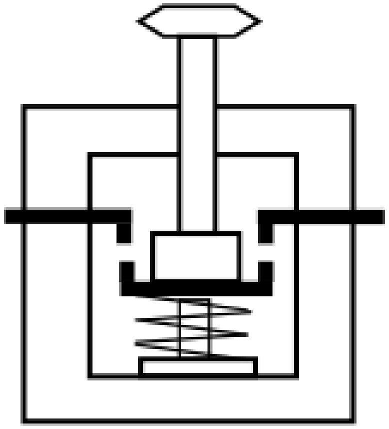
NC-Type Push Button

- In the NC type, the contacts are closed in the normal position, permitting the energy flow through them. And, the contacts are open in the actuated position, inhibiting the energy flow through them.

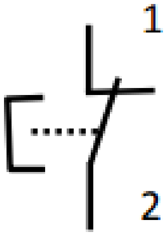
Normally closed (NC)



Normal position



Actuated position

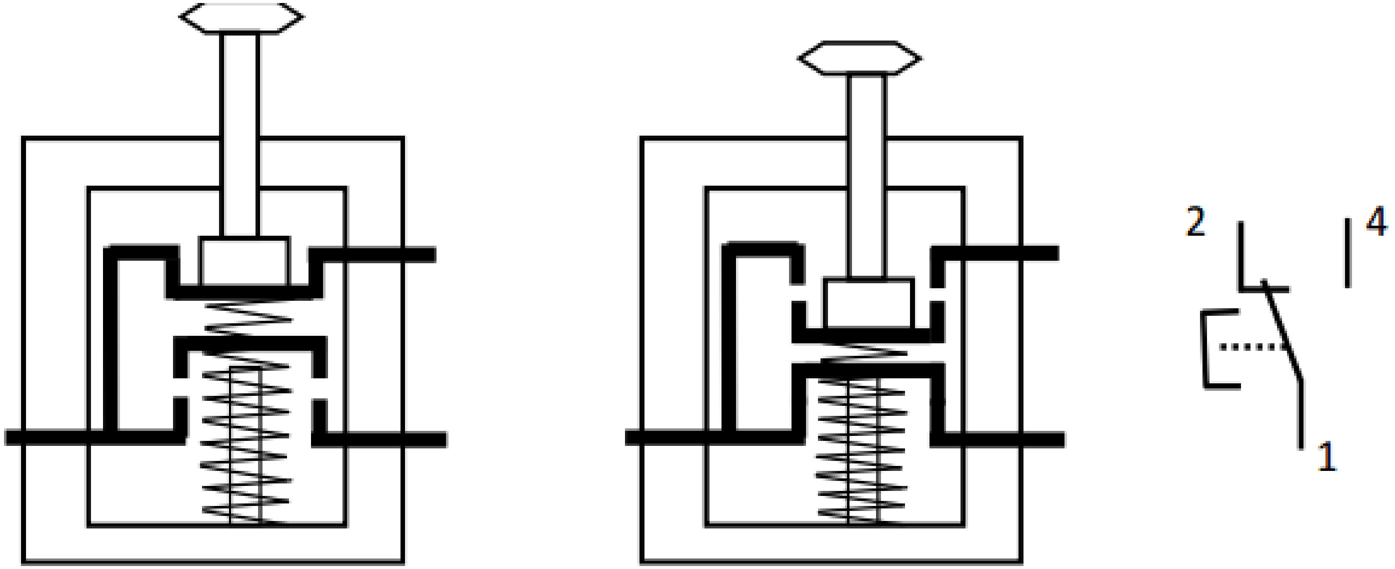


Symbol

CO-Type Push Button

- A **CO** contact is a combination of NO and NC contacts.

Change Over (CO)

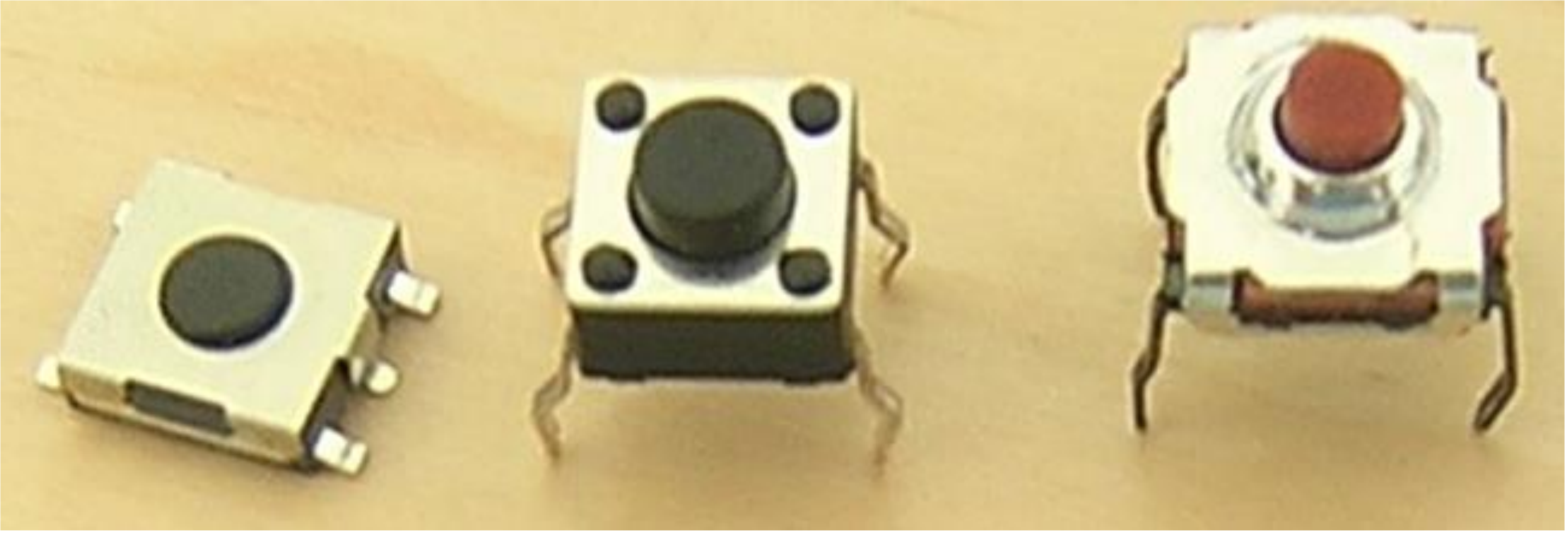


Normal position

Actuated position

Symbol

Real Switches



Arduino : Introduction & Programming

What is an Arduino ?

- **Open Source** electronic prototyping **platform** based on flexible **easy to use** hardware and software.

- Single-board microcontroller, intended to make the application of interactive objects or environments more accessible.

- Designed to make the process of using electronic multidisciplinary projects more accessible.

Arduino Features

Arduino can

- Sense the environment by receiving input from variety of sensors.
- Controls its surrounding lights, motors, and other actuators.

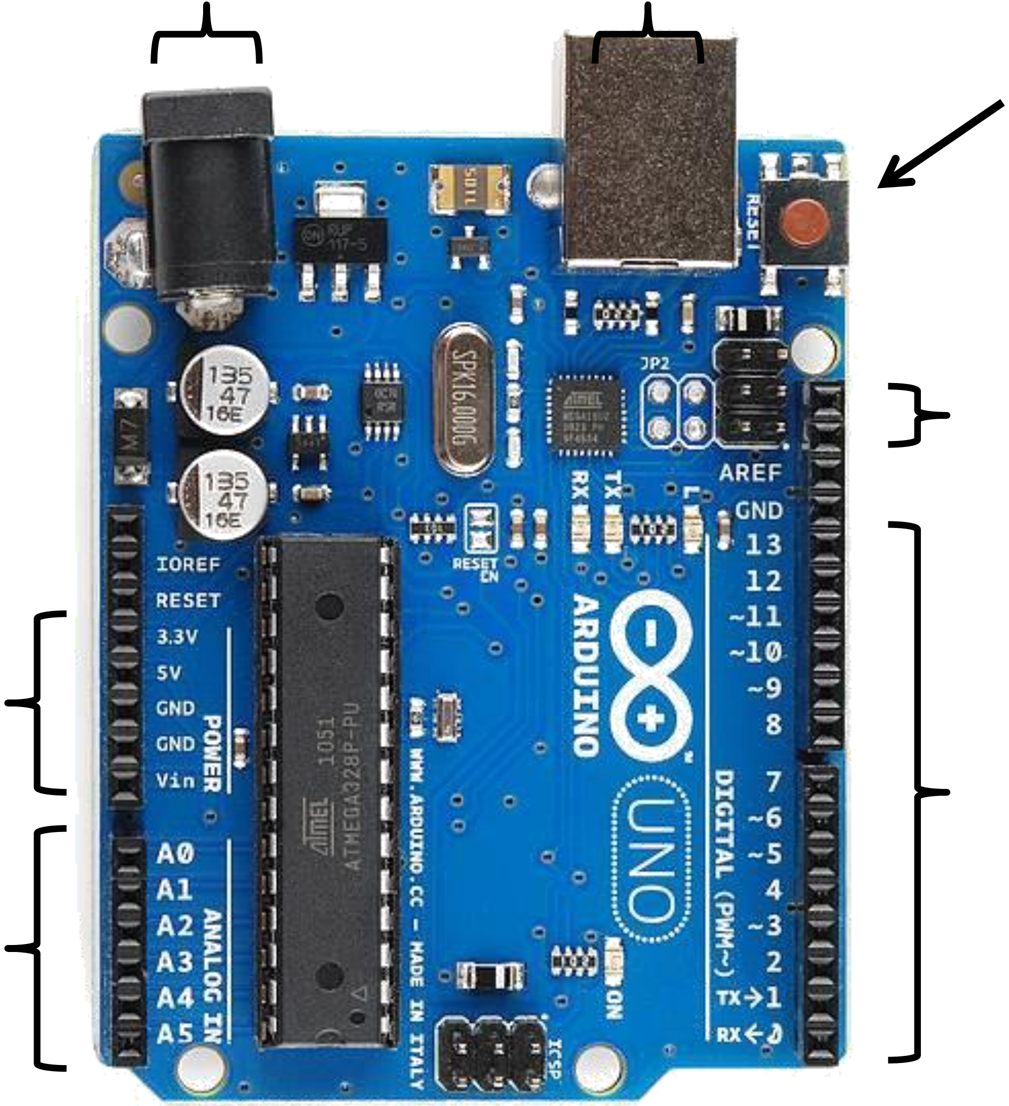
Advantages

- Portable and low power consumption
- Simplified and user-friendly programming language
- Open source
- No additional programmer hardware is required for programming board

PWR IN

USB
(to Computer)

RESET



SCL\SDA
(I2C Bus)

POWER
5V / 3.3V / GND

Analog INPUTS

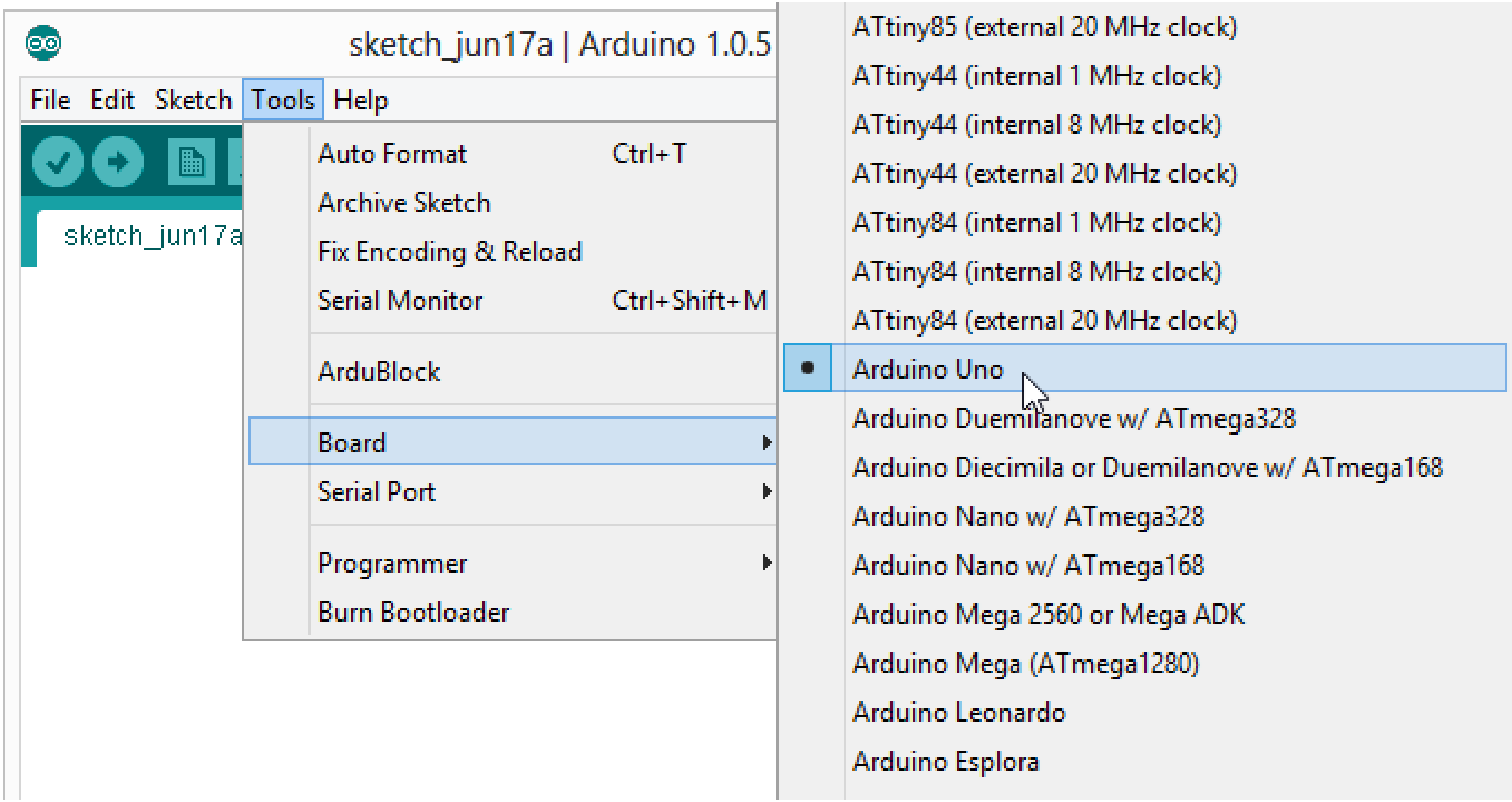
Digital I\O
PWM(3, 5, 6, 9, 10, 11)

Specification of Arduino Uno Board

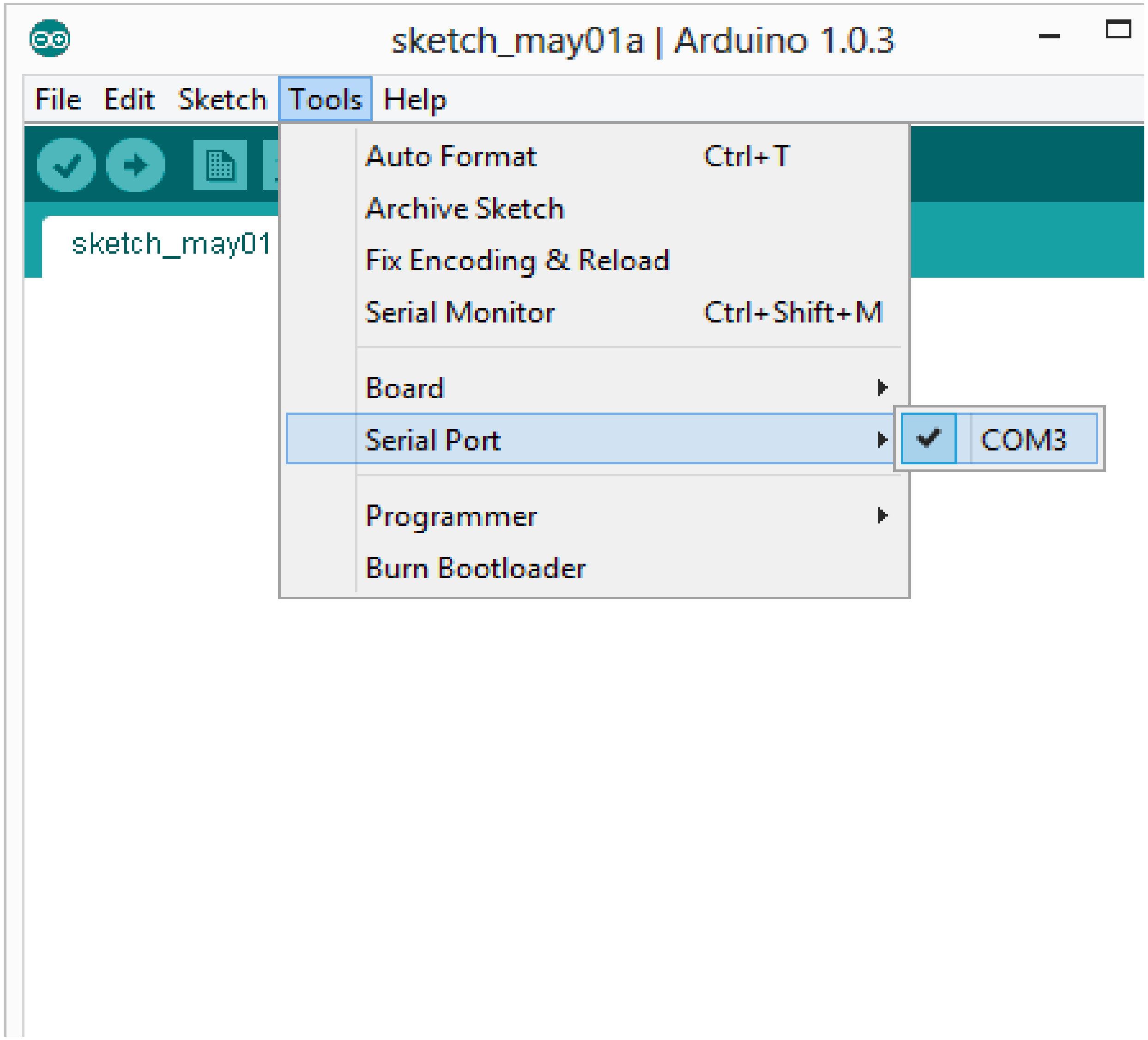
- Microcontroller ATmega328P
- Operating Voltage 5V and 3.3 V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by Bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz

Getting started with Programming

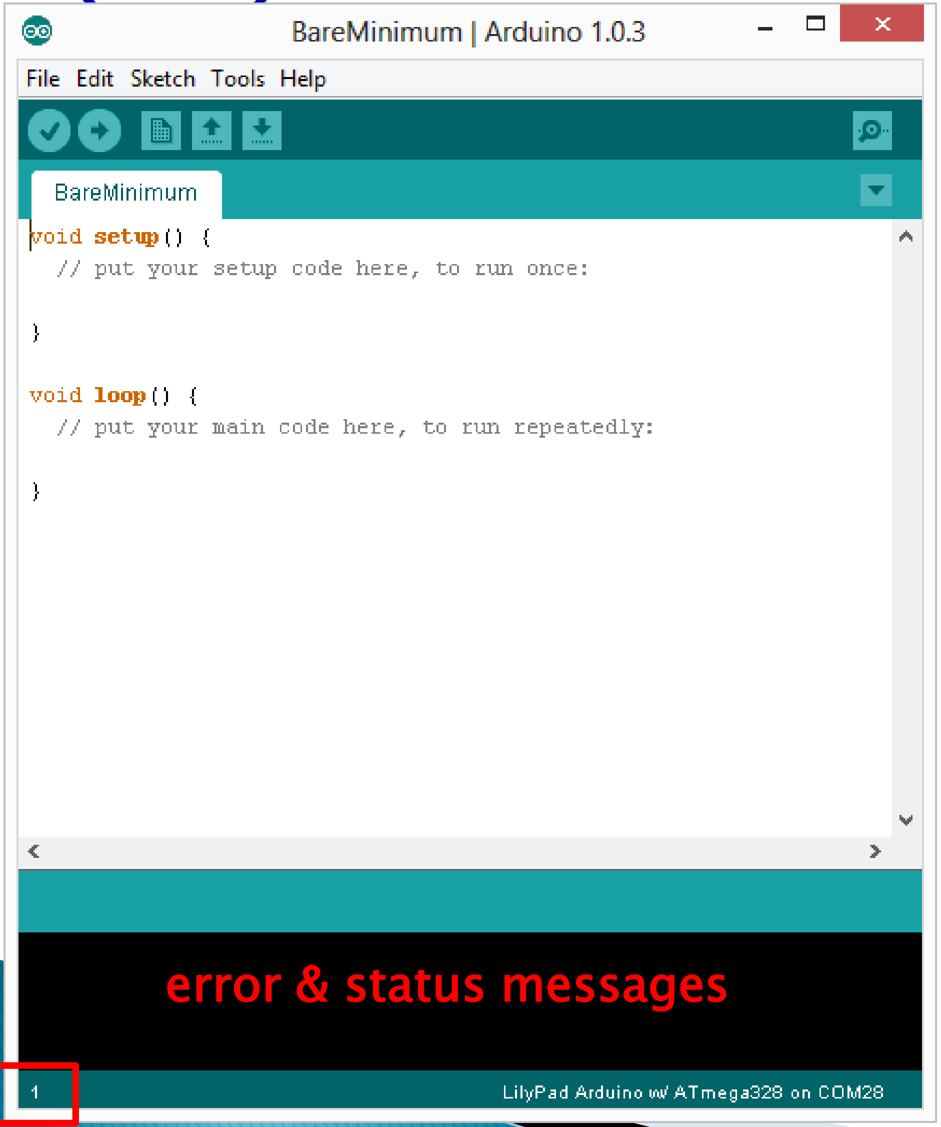
Settings: Tools → Board


sketch_jun17a | Arduino 1.0.5
File
Edit
Sketch
Tools
Help
sketch_jun17a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
ArduBlock
Board
Serial Port
Programmer
Burn Bootloader
ATtiny85 (external 20 MHz clock)
ATtiny44 (internal 1 MHz clock)
ATtiny44 (internal 8 MHz clock)
ATtiny44 (external 20 MHz clock)
ATtiny84 (internal 1 MHz clock)
ATtiny84 (internal 8 MHz clock)
ATtiny84 (external 20 MHz clock)
● Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora

Settings: Tools → Serial Port



Integrated Development Environment (IDE)



BareMinimum | Arduino 1.0.3
File Edit Sketch Tools Help
BareMinimum

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

}

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

}
```

error & status messages

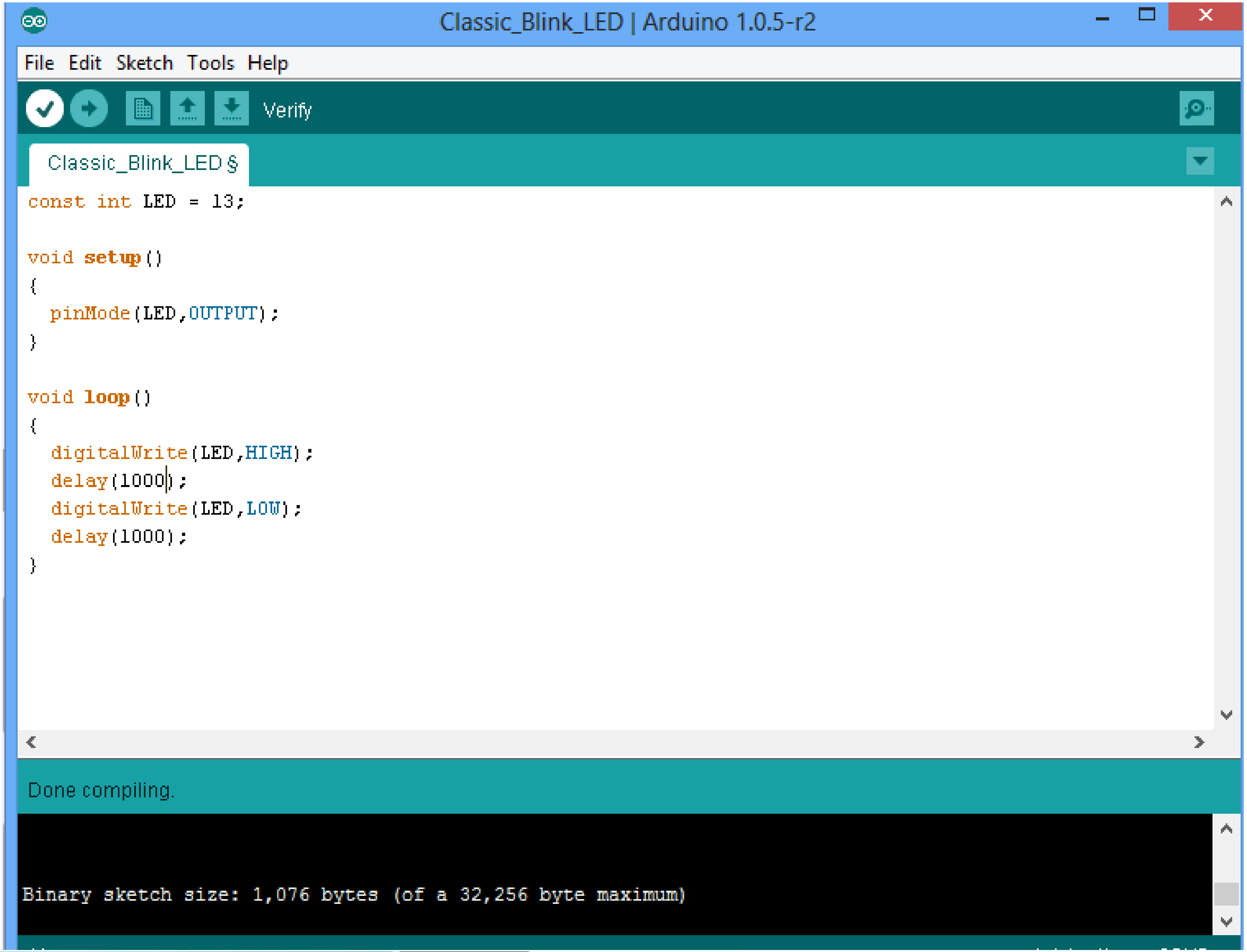
1
LilyPad Arduino w/ ATmega328 on COM28

Two required functions / methods / routines:

```
void setup()
{
        // runs once
}

void loop()
{
        // repeats
}
```

LED Blinking Example


Classic_Blink_LED | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Verify
Classic_Blink_LED §

```
const int LED = 13;

void setup()
{
  pinMode(LED,OUTPUT);
}

void loop()
{
  digitalWrite(LED,HIGH);
  delay(1000);
  digitalWrite(LED,LOW);
  delay(1000);
}
```

Done compiling.

```
Binary sketch size: 1,076 bytes (of a 32,256 byte maximum)
```

Thanks