ROBOTIC & COMMUNITY
(UKQA2242-01 / UKQF2442-01)

W3 – Micro-controllers, Micro-processors, Differences, Parts And Applications

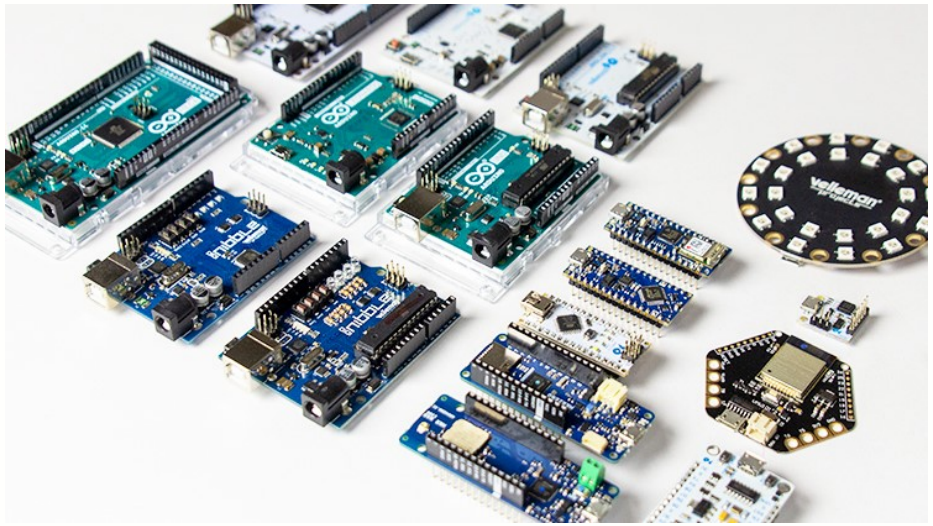
9 November 2021 |@ google meet

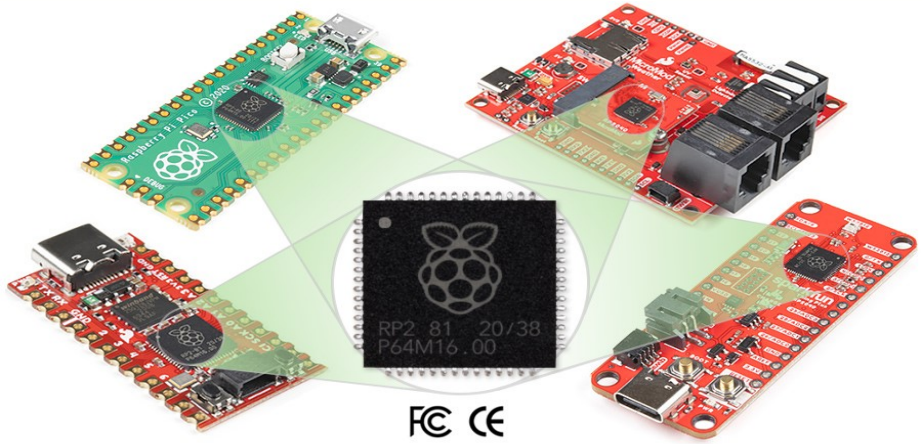
Muhammad Zulhelmi Bin Kamarudin

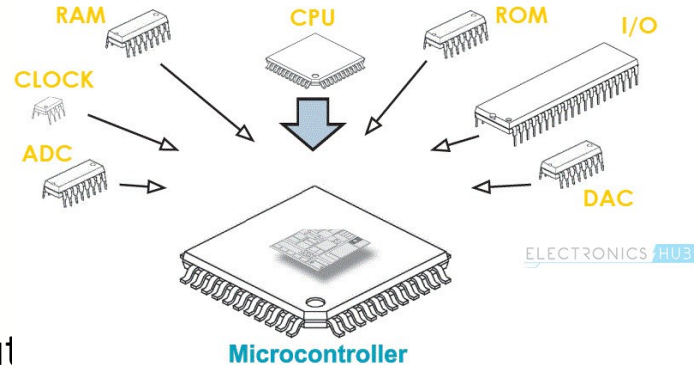
INTRODUCTION

Micro-controllers, Micro-processors, Differences, Parts And Applications

MICRO-CONTROLLERS







Control devices in one chip

Perform simple controls

Consists of CPU, RAM, ROM, Port
Works to receive and process signals and subsequently generate signals to be execut

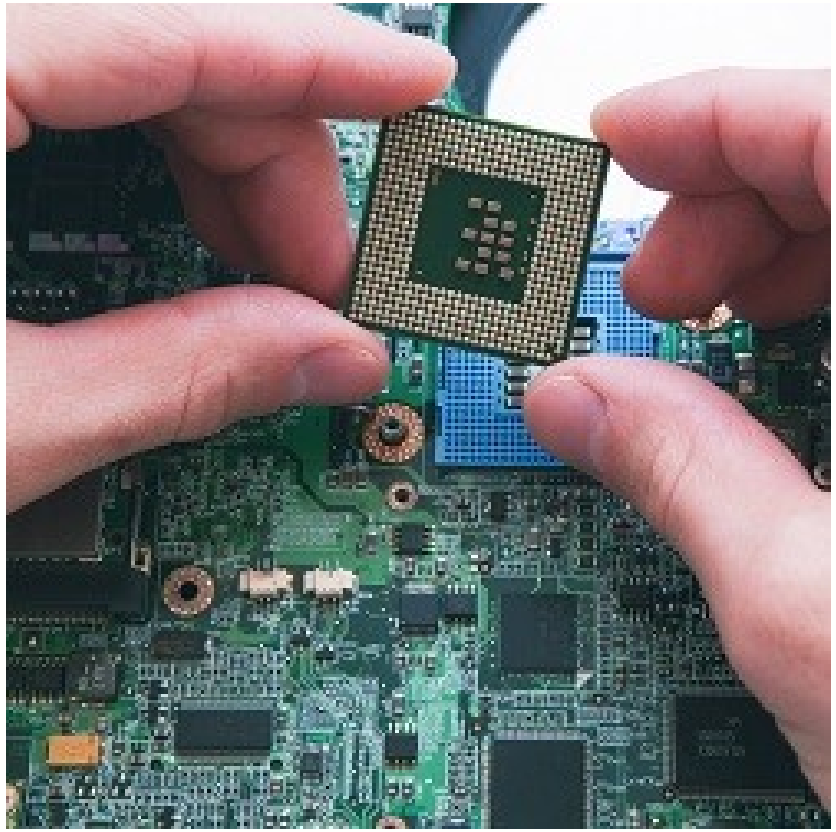
MICRO-PROCESSORS,

Micro processing devices in one chip

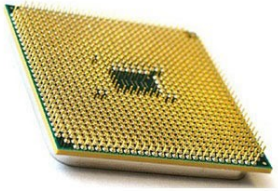
This device is used with support chips such as Output, Input, RAM, ROM

All support chips are outside the microprocessor and are connected via a system bus.

Used for large controls such as computers.



The difference
MicroControllers and MicroProcessors

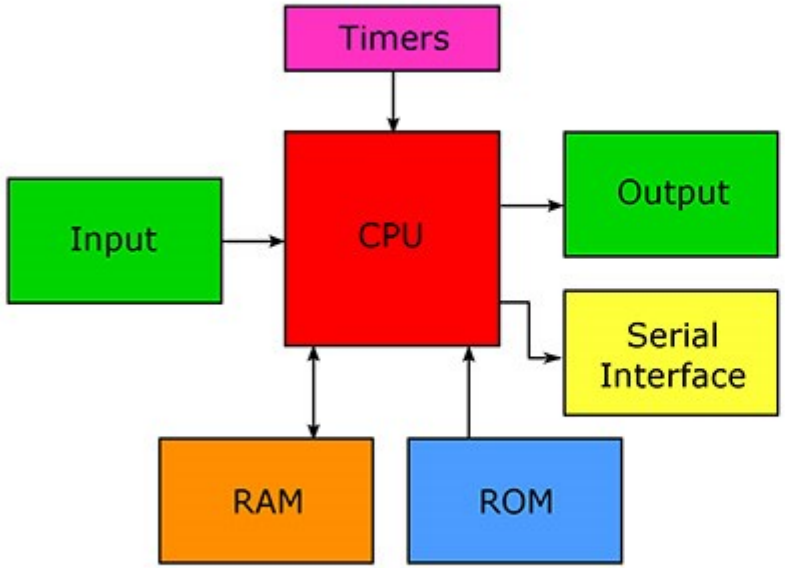


Microprocessor

VS

Microcontroller

Microprocesser: CPU
and several supporting chips.



Microcontroller: CPU
on a single chip.

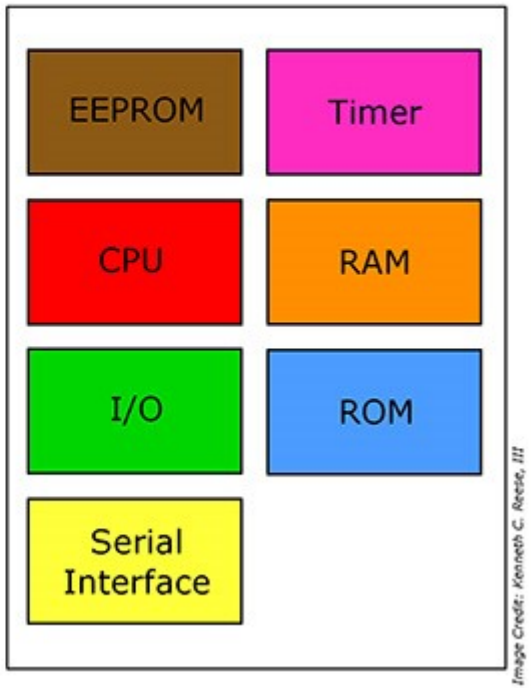


Image Credit: Kenneth C. Reese, III

List 5 devices around you that contain microcontrollers?

Remote TV

Printer

CCTV

Kipas

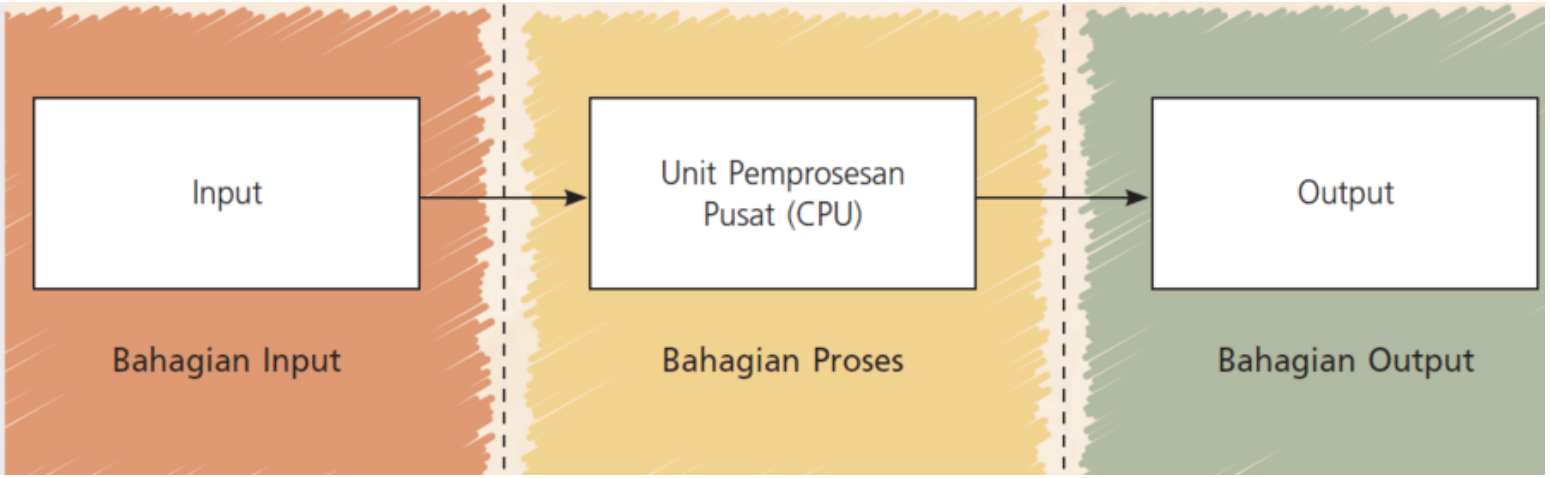
Aircond

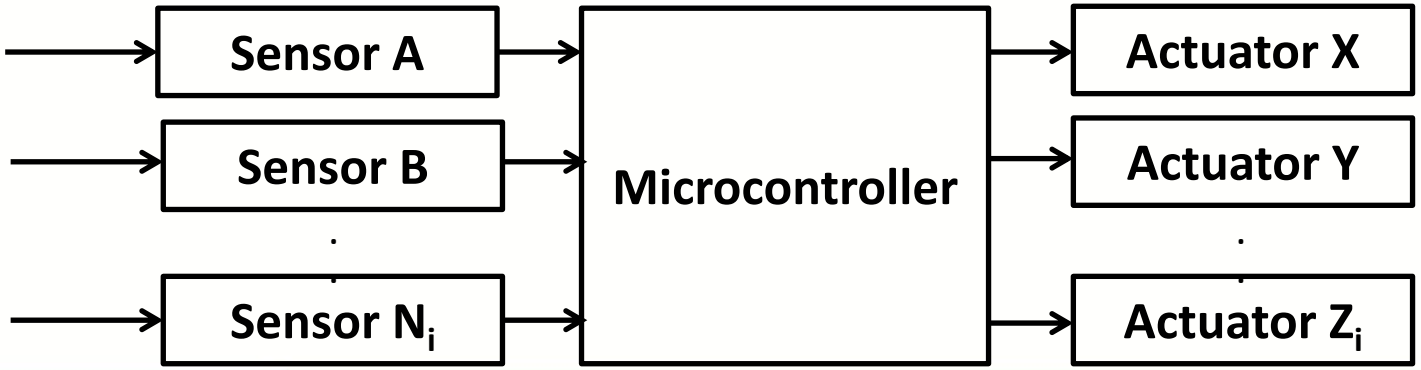
Auto Gate

Parts of Micro-Controller

Parts found on the microcontroller

- ❑ Input Section
- ❑ Process Section
- ❑ Output Section



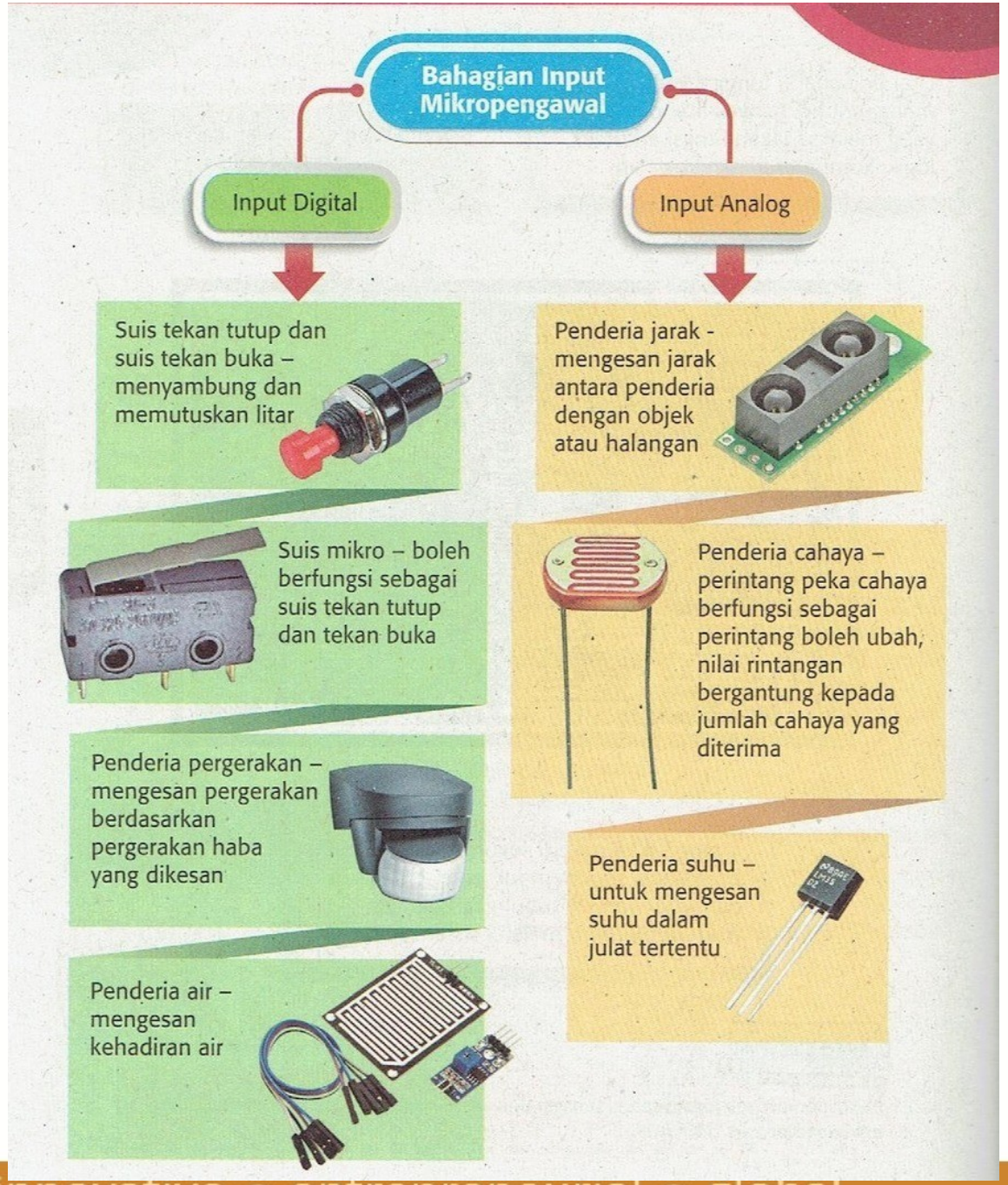


Input Section

One of the parts found in a microcontroller.

Task: Receive signals and send them to the Central Processing Unit (CPU).

Divided into two, namely digital input devices and analog input devices.



Process Section

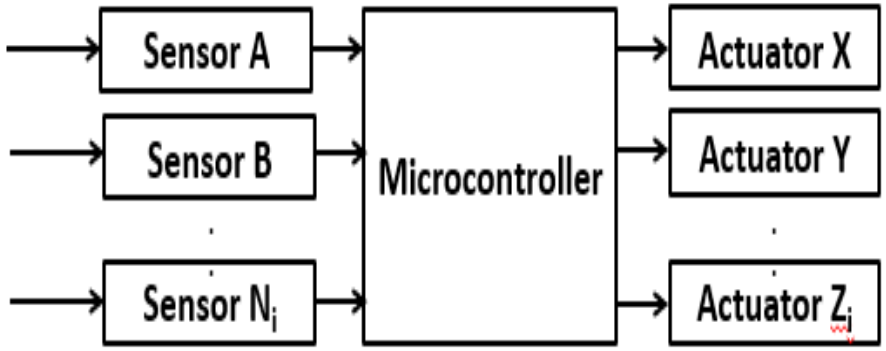
Performed by the Central Processing Unit (CPU).

The CPU will execute programs stored in ROM (Read Only Memory).

ROM - used to store programs and data permanently in the microcontroller.

RAM (Random Access Memory) -used to store data temporarily (will be erased when the power supply is turned off.

The signal received from the input will be processed by the CPU, then the CPU will generate a signal to the output.

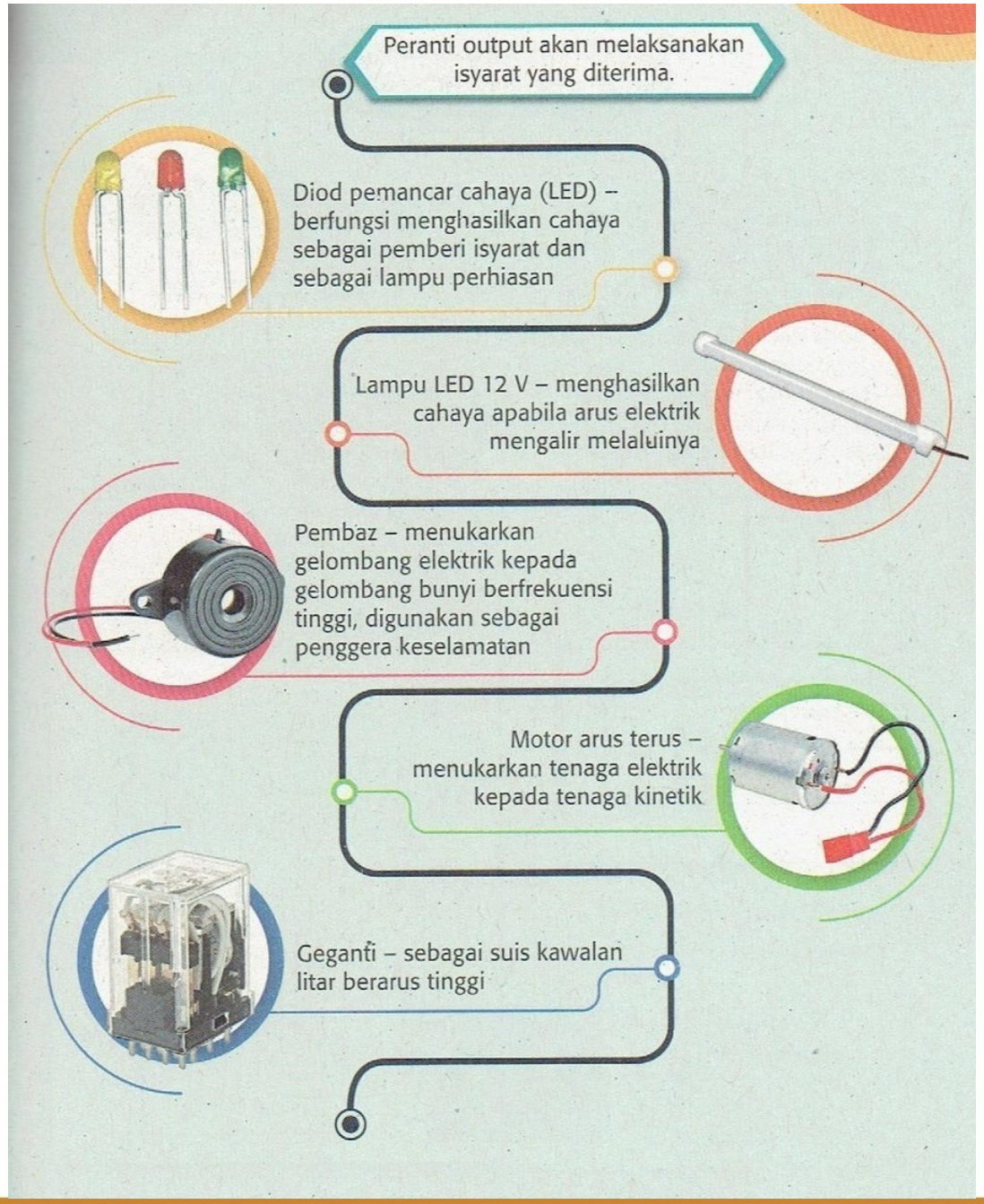


Output Section

Parts found in the microcontroller.

Task: Receive a signal from the Central Processing Unit (CPU) and send it to the output device.

The resulting output signal is in digital or analog form



Penderia Suhu

Push Button

Penderia Gerakan

Penderia Cahaya

Input devices are divided into two, namely digital input devices and analog input devices.

Choose two answers for digital input devices.

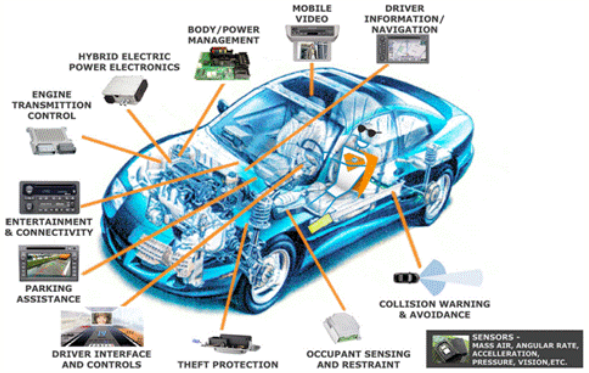
Penderia Suhu

Push Button

Penderia Gerakan

Penderia Cahaya

Applications



Automotive

Energy

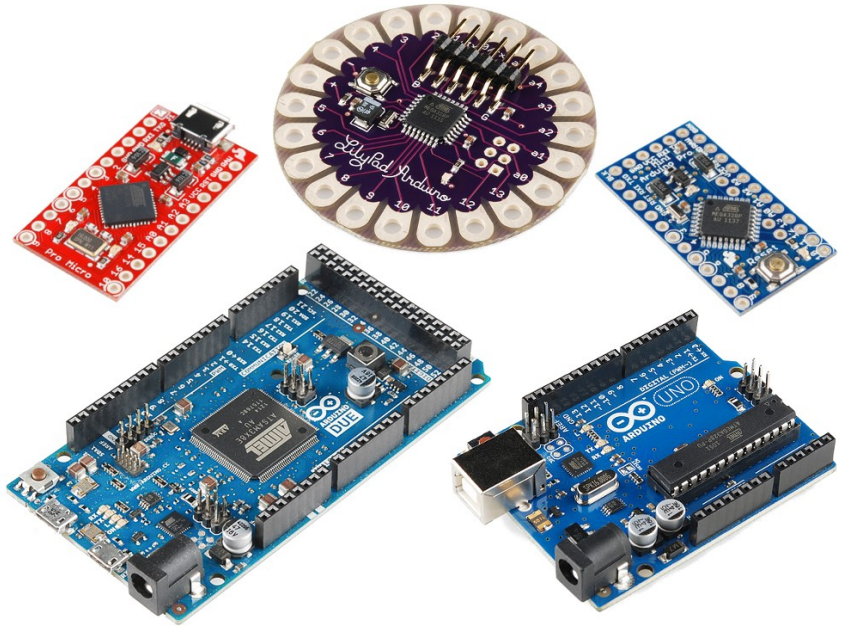
Robotics

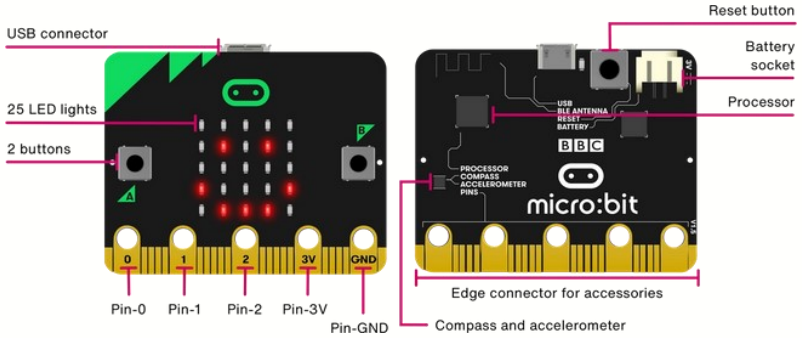
Home appliances

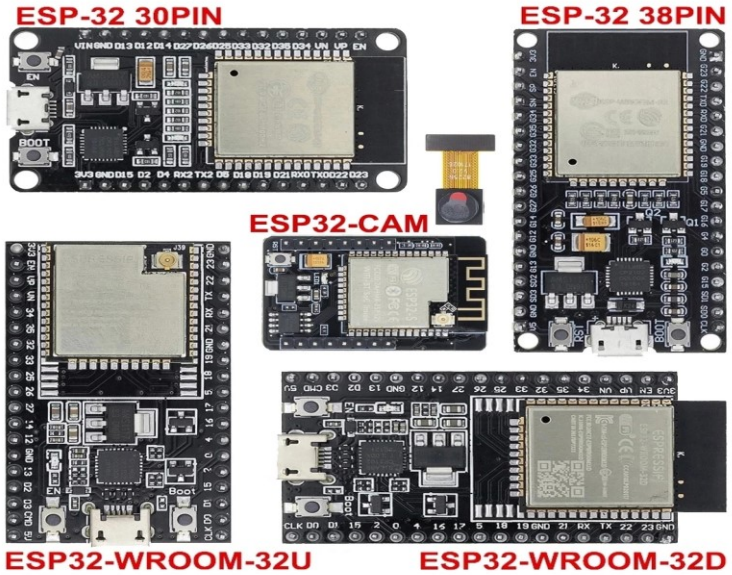
Medical devices

Internet of Thing (IOT)

MicroControllers on the market







Carrot Microcontroller Board (Use Bluetooth)

Salad Microcontroller Board (Use USB OTG)

INTRODUCTION

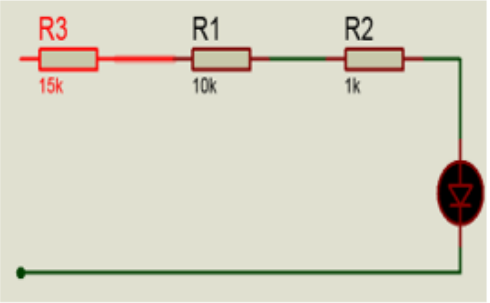
Electronics

ELECTRONICS COMPONENTS

Resistor

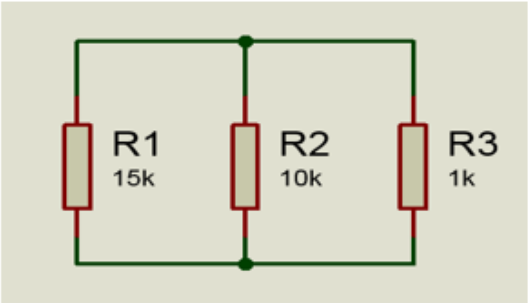
It controls the amount of current that passes through an electronic circuit.

Without these components, the voltage will be too large to be handled by the individual components and will result in excessive load or damage.



Series Connection

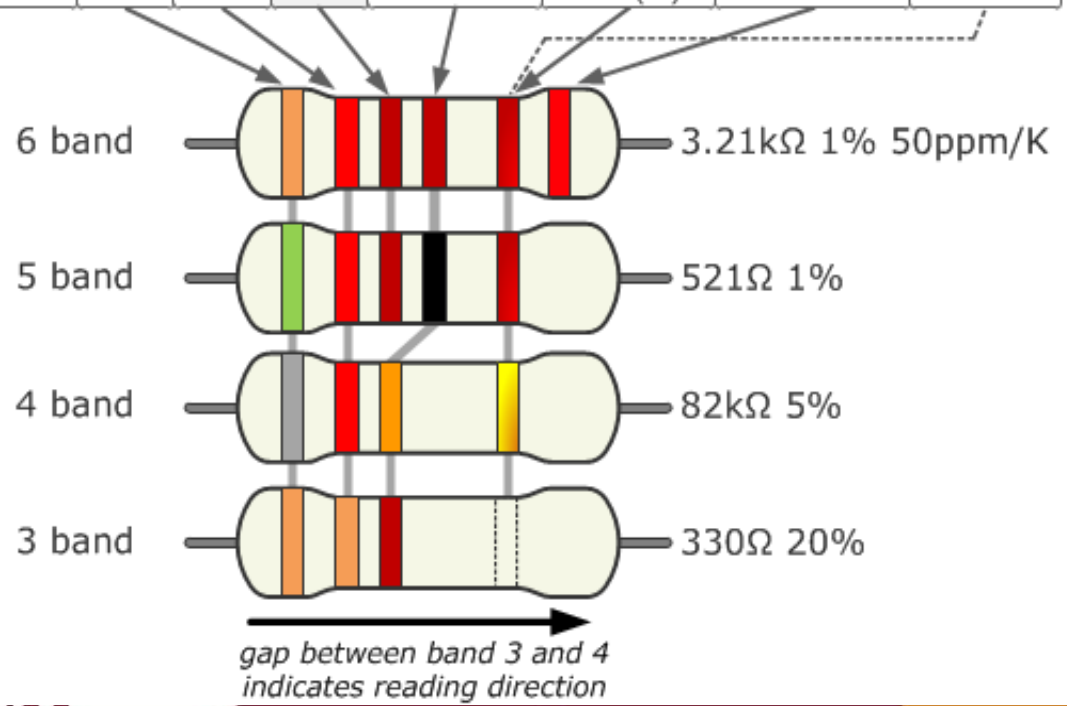
$$R_{total} = R_1 + R_2 + R_3$$



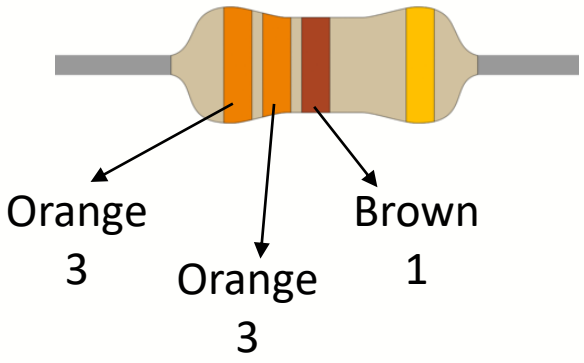
Parallel Connection

$$\frac{1}{R_{total}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		

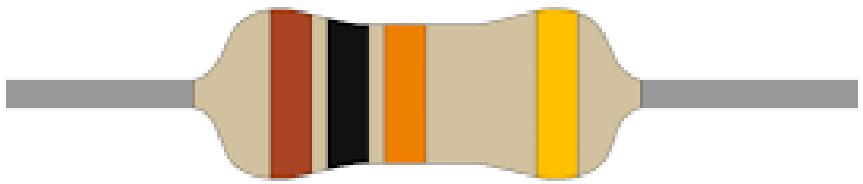


Contoh 1



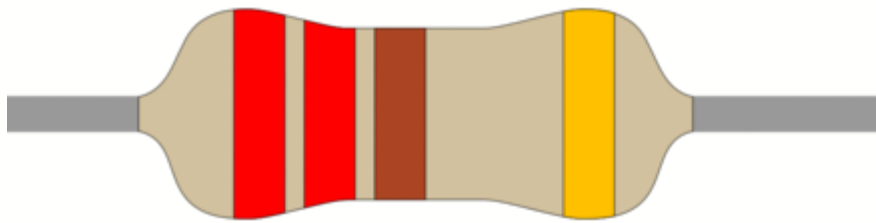
Jawapan: 3 3 x 10 = 330 ohm

	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		



_____ ohm

	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		



_____ ohm

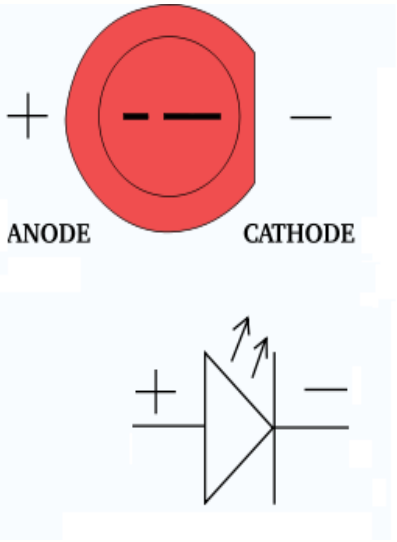
	Color	Signficant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		

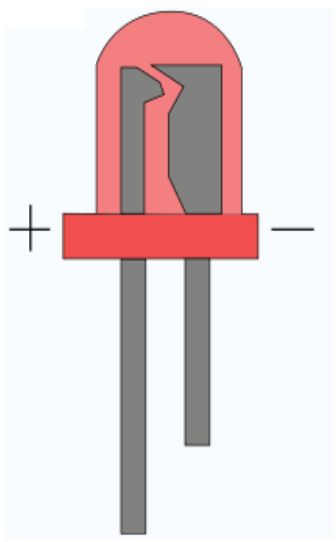
_____ ohm

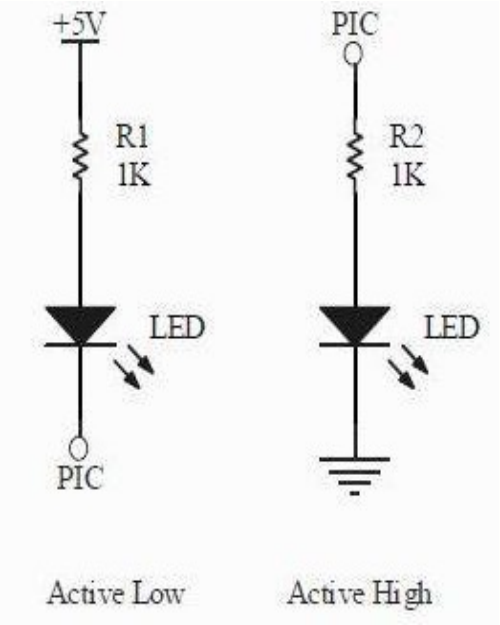
Light-Emitting Diode (LED)

Usually used as an indicator light for electronic devices.

Too much current flowing into it will damage it.





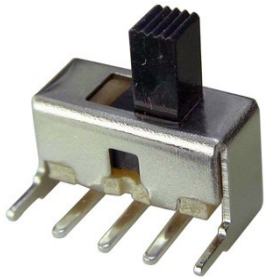


Push Button & Switch

This is a simple mechanism for controlling some aspect of a machine or process.

Compact in size and can be operated with just a finger.

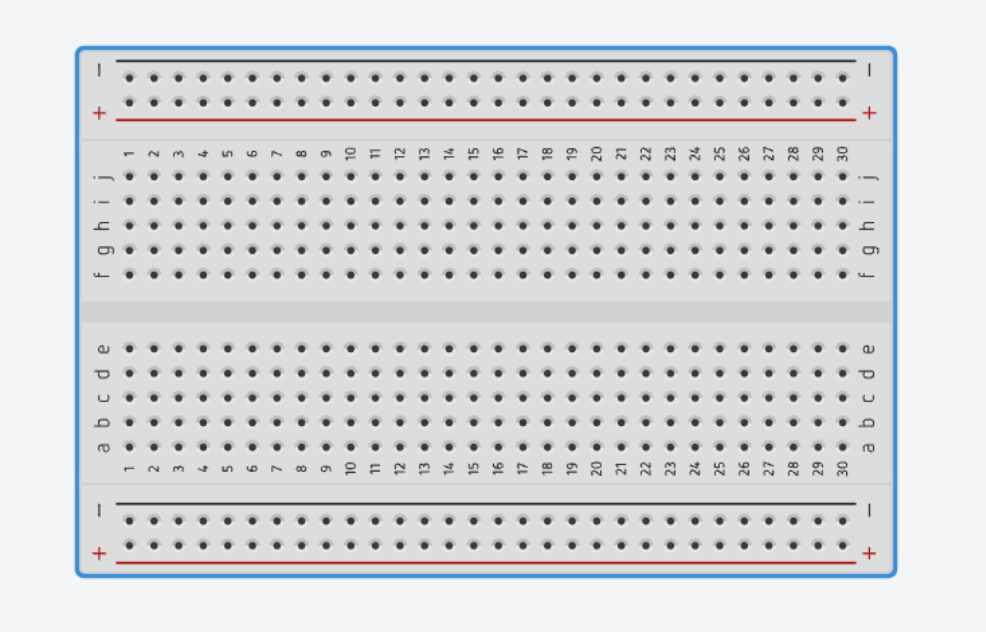
Ability to turn on or off without having to pull the main cord.

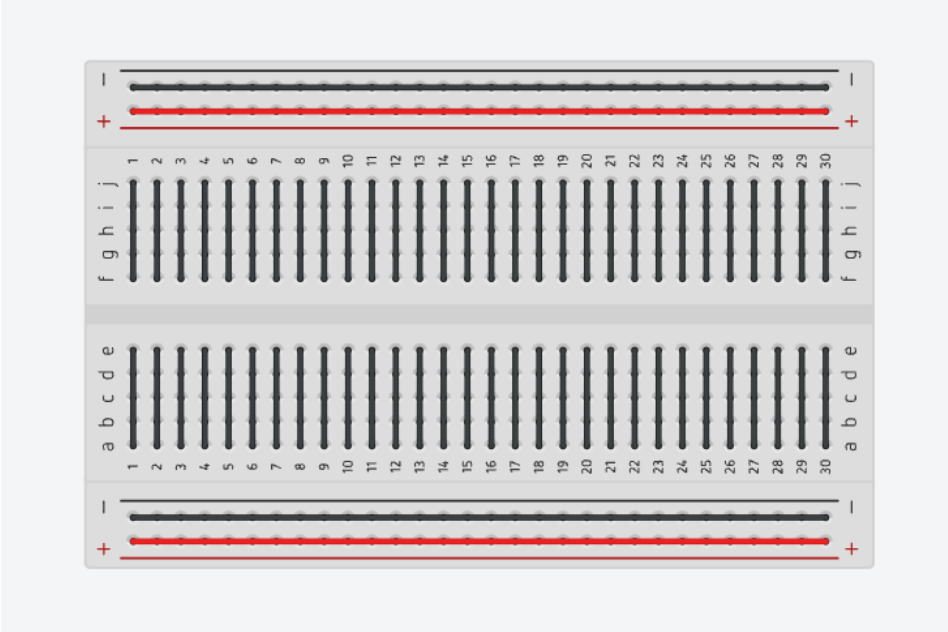


Breadboard

The breadboard, or protoboard, is the basis of the construction of an electronic prototype.

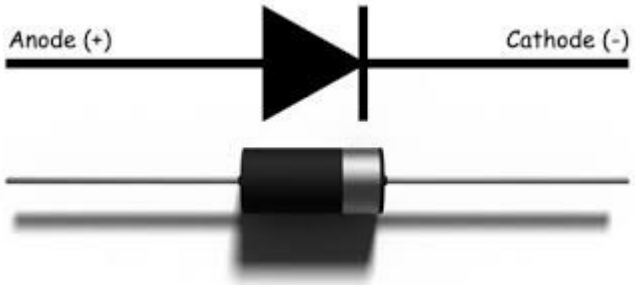
Used to develop prototypes of electronic circuits, breadboards can be reused for future work.
They can be used to make a system but rarely become commercial products.





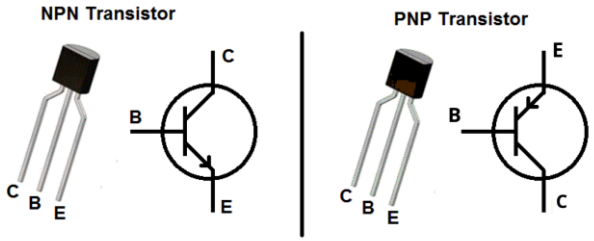
Other Components

Potentiometer

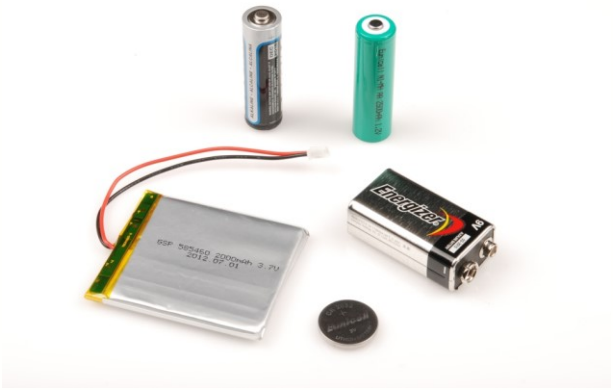


diode

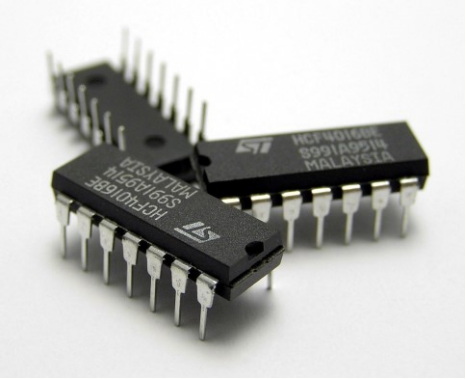
capacitor



Transistor



Battery



IC

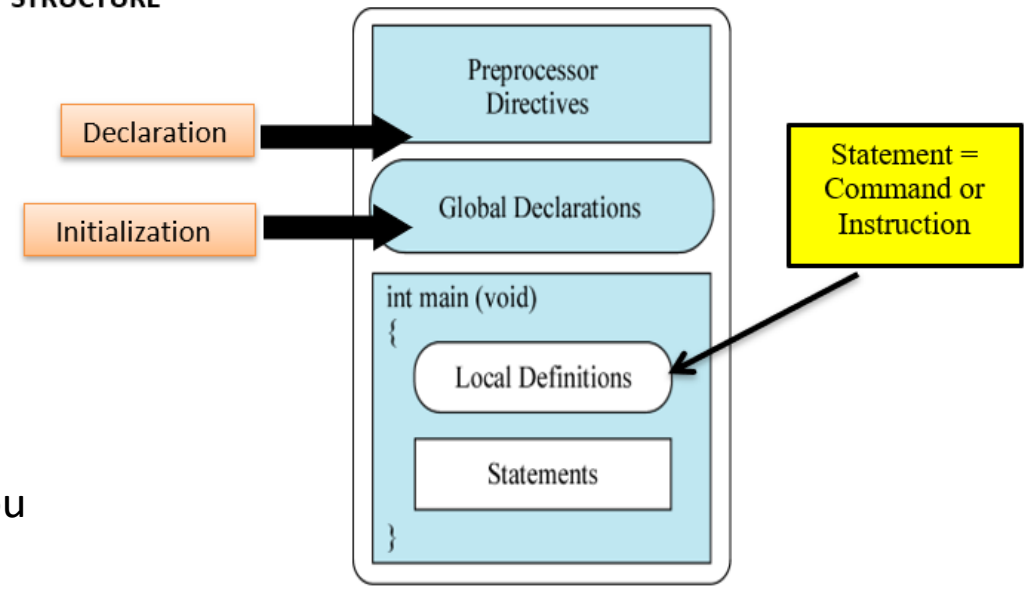
INTRODUCTION

PROGRAMMING

INTRODUCTION OF PROGRAMMING

Computer programming is a process that leads from an original formulation of a computing problem to executable computer programs. Programming involves activities such as analysis, developing understanding, and generating algorithms. The purpose of programming is to find a sequence of instructions that will automate performing a specific task or solving a given problem.

STRUCTURE



- ✓ While doing programming using any programming language, you need to use various variables to store various information.
- ✓ Variables are nothing but reserved memory locations to store values.
- ✓ This means that when you create a variable, you reserve some space in memory.

Local Variable

Variables that are declared inside a function or block.
They can be used only by statement that is inside that function or block of code.

sketch_aug20a §

```
int gPWMval;  // any function will see this variable

void setup()
{
  // ...
}

void loop()
{
  int i;    // "i" is only "visible" inside of "loop"
  float f;  // "f" is only "visible" inside of "loop"
  // ...

  for (int j = 0; j <100; j++){
  // variable j can only be accessed inside the for-loop brackets
  }

}
```

Local Variable

Global Variable / Global Declaration

Variables that defined outside of all the functions, usually on top of the program.
The global variables will hold their value throughout the lifetime of your program.
A global variable can be accessed by any function.

project_CODING_Trial §

```
#include<stdio.h>
#include<SoftwareSerial.h>
SoftwareSerial bt(2,3);
int pwm1 = 10;
int in1 = 9;
int in2 = 8;
int pwm2 = 5;
int in3 = 7;
int in4 = 6;
char data='S';
char prevdata='A';
void front(), left(), rear(), right(), brake();
int manmode(char data);
void setup()
{
 //left wheel
 pinMode(pwm1,OUTPUT);
```

Global Variable

OPERATION

Arithmetic Operators

Operations of addition, subtraction, multiplication and division correspond literally to their respective mathematical operators.
The last one, module operator, represented by a percentage sign (%), gives the remainder of a division of two values.

operator	description
+	addition
-	subtraction
*	multiplication
/	division
%	modulo

Compound Assignment

Compound assignment operators modify the current value of a variable by performing an operation on it.
They are equivalent to assigning the result of an operation to the first operand.

expression	equivalent to...
`y += x;`	`y = y + x;`
`x -= 5;`	`x = x - 5;`
`x /= y;`	`x = x / y;`
`price *= units + 1;`	`price = price * (units+1);`

Increment & Decrement

```
1 ++x;
2 x+=1;
3 x=x+1;
```

Relational & Comparison Operators

Two expressions can be compared using relational and equality operators. For example, to know if two values are equal or if one is greater than the other.

operator	description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Logical Operators

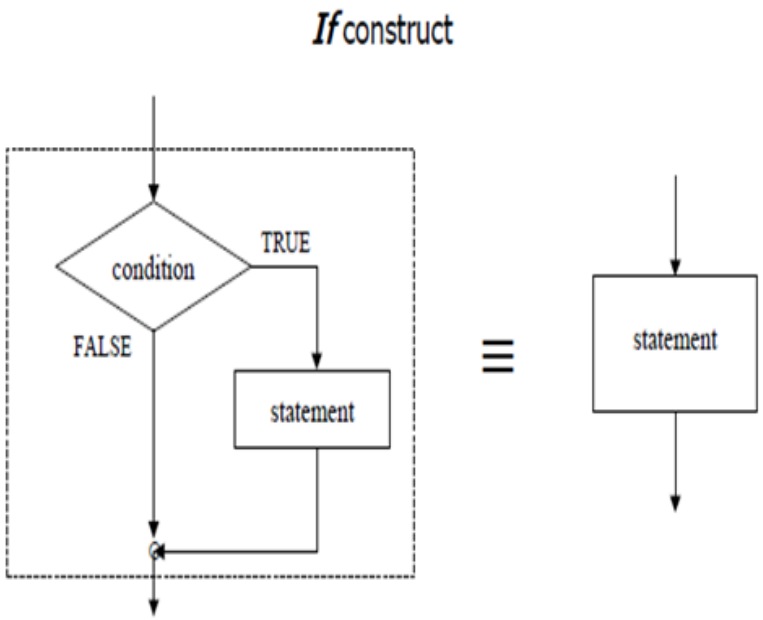
The operator && corresponds to the Boolean logical operation AND, which yields true if both operands are true, and false otherwise.
The operator || corresponds to the Boolean logical operation OR, which yields true if either of its operands is true, thus being false only when both operands are false.

&& OPERATOR (and)		
a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

|| OPERATOR (or)		
a	b	a || b
true	true	true
true	false	true
false	true	true
false	false	false

DECISIONS

If Statement



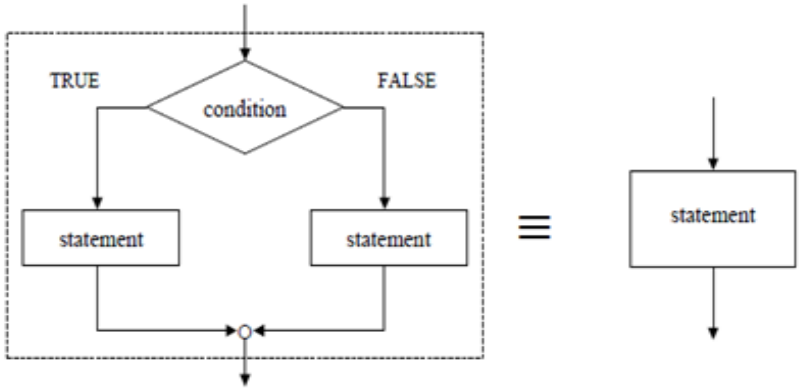
If set condition is true execute statement, else execute nothing

sketch_aug21a §

```
void loop() {
if (x > 120)
{
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}
}
```

If-Else Statement

If-else construct

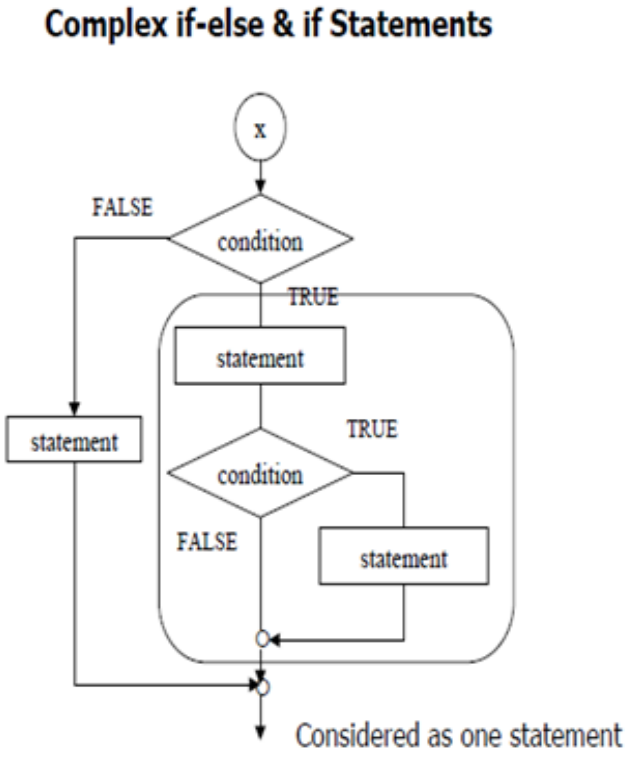


If set condition is true execute first statement, else execute second statement

sketch_aug21a §

```
void loop() {
if (x > 120)
{
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, HIGH);
}
else
{
  digitalWrite(LEDpin1, LOW);
  digitalWrite(LEDpin2, LOW);
}
}
```

If Else & If Statement

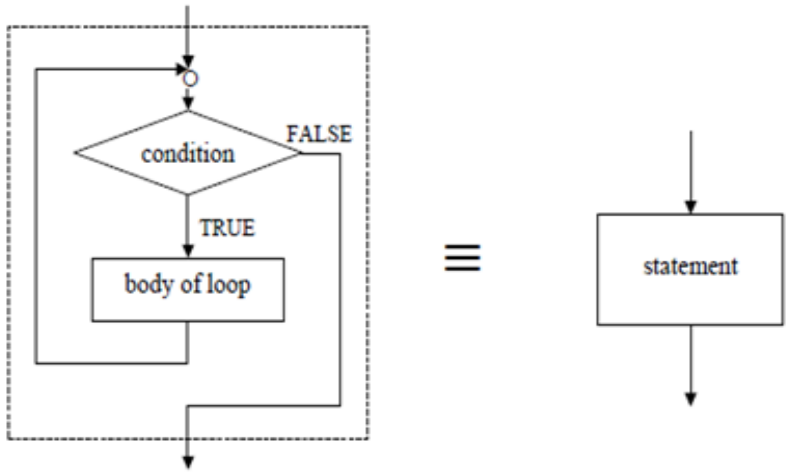


sketch_aug21a §

```
void loop() {
if (x > 120)
{
  digitalWrite(LEDpin1, HIGH);
  digitalWrite(LEDpin2, LOW);
}
else if (x < 120)
{
  digitalWrite(LEDpin1, LOW);
  digitalWrite(LEDpin2, HIGH);
}
else
{
  digitalWrite(LEDpin1, LOW);
  digitalWrite(LEDpin2, LOW);
}
}
```

While Loop

while Loop



While a set condition is true, repeat statement (body of loop)

sketch_aug14a §

```
var = 0;
while(var < 200){
  // do something repetitive 200 times
  var++;
}
```