



DEPARTAMENTO DE  
**INGENIERÍA  
MECÁNICA**  
UNIVERSIDAD DE SANTIAGO DE CHILE

# Tópico de la especialidad: Robótica

## Clase 4: Misceláneos de arduino

**René Torres**

Universidad de Santiago de Chile  
Departamento de Ingeniería Mecánica  
e-mail: [rene.torres.a@usach.cl](mailto:rene.torres.a@usach.cl)

26 de septiembre de 2023

# Función millis()

# Función millis()

## Descripción

Devuelve el número de milisegundos transcurridos desde que la placa Arduino comenzó a ejecutar el programa actual. Este número se desbordará (volverá a cero) después de aproximadamente 50 días.

# Función millis()

## Descripción

Devuelve el número de milisegundos transcurridos desde que la placa Arduino comenzó a ejecutar el programa actual. Este número se desbordará (volverá a cero) después de aproximadamente 50 días.

## Ejemplo

```
1  unsigned long myTime;
2  void setup() {
3    Serial.begin(9600);
4  }
5  void loop() {
6    Serial.print("Time:");
7    myTime = millis();
8    Serial.println(myTime);
9  }
```

## Hola Mundo.ino

```
1  void setup()  
2  {  
3  pinMode(LED_BUILTIN, OUTPUT);  
4  }  
5  
6  void loop()  
7  {  
8  digitalWrite(LED_BUILTIN, HIGH);  
9  delay(1000);  
10 digitalWrite(LED_BUILTIN, LOW);  
11 delay(1000);  
12 }
```

## Hola Mundo.ino

```
1  void setup()  
2  {  
3  pinMode(LED_BUILTIN, OUTPUT);  
4  }  
5  
6  void loop()  
7  {  
8  digitalWrite(LED_BUILTIN, HIGH);  
9  delay(1000);  
10 digitalWrite(LED_BUILTIN, LOW);  
11 delay(1000);  
12 }
```

## Tarea

Hacer lo mismo que el código anterior pero usando la función millis()

# Posible solución

## Sol.ino

```
1  unsigned long tiempo=0;
2  void setup() {
3      pinMode(10, OUTPUT);
4      Serial.begin(9600);
5  }
6  void loop() {
7      if (millis()-tiempo>=1000) {
8          tiempo=millis();
9          digitalWrite(10,!digitalRead(10));
10         }
11     }
```

## Tarea

Hacer que 3 leds parpadeen a las frecuencias de 1 Hz, 2Hz y 10Hz.



# Multitasking

## Tarea

Hacer que 3 leds parpadeen a las frecuencias de 1 Hz, 2Hz y 10Hz.

## Sol.

```
1
2  #define LED_Green 4
3  #define LED_Blue 8
4  #define LED_Red 12
5  void setup() {
6      Serial.begin(9600);
7      pinMode(LED_Green, OUTPUT);
8      pinMode(LED_Blue, OUTPUT);
9      pinMode(LED_Red, OUTPUT);
10 }
```

# “Multitasking”

Sol.

```
10     void loop() {  
11         led_verde();  
12         led_azul();  
13         led_rojo();  
14     }
```

# “Multitasking”

Sol.

```
15 void led_verde() {  
16     static long tiempo=0;  
17     if(millis()-tiempo >= 1000) {  
18         tiempo=millis();  
19         digitalWrite(LED_Green, !digitalRead(  
20             LED_Green));  
    } }
```

# “Multitasking”

Sol.

```
28 void led_rojo() {  
29     static long tiempo=0;  
30     if(millis()-tiempo >= 250) {  
31         tiempo=millis();  
32         digitalWrite(LED_Red, !digitalRead(LED_Red))  
33         ;  
34     }  
35 }
```

# Maquinas de estados finitos (FSM o Finite-State Machine)

## ¿Qué es?

Una máquina de estados finitos (FSM) es un modelo matemático de computación. Es una máquina abstracta que puede estar exactamente en un estado de un número finito de estados en cualquier momento dado. La FSM puede cambiar de un estado a otro en respuesta a algunas entradas; el cambio de un estado a otro se llama transición.

# Ejemplo

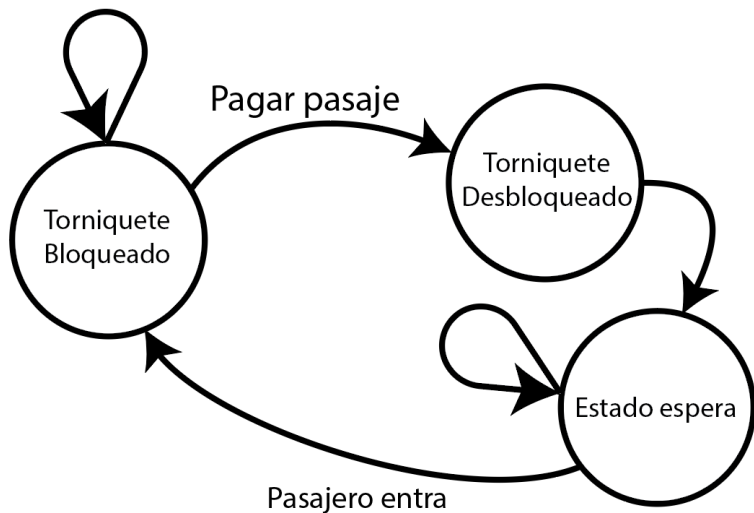
## Torniquite



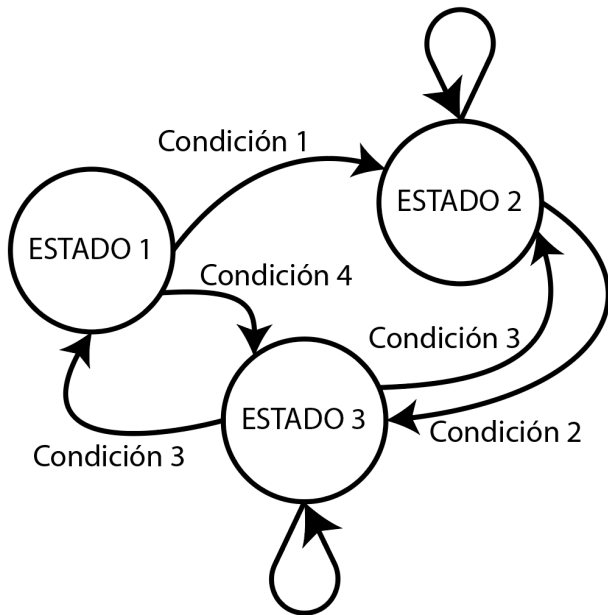
Link

FMS Torniquete

## Ejemplo: Torniquete FSM



# Maquinas de estados finitos (FSM o Finite-State Machine)





Sol.

```
1  #define ESTADO_INICIO 1
2  #define ESTADO_DOS 2
3  #define ESTADO_TRES 3
4  void setup()
5  {
6      estado=ESTADO_INICIO;
7  }
8  void loop()
9  {
10     switch(estado) {
11
12         case ESTADO_INICIO:
13             estado_inicio_function();
14             break;
```

Sol.

```
15     case ESTADO_DOS:
16         estado_dos_function();
17         break;
18     case ESTADO_TRES:
19         estado_tres_function();
20         break;
21     default: estado_DEFAULT;
22         estado=ESTADO_INICIO;
23         break;
24 }
25 }
```

# Estructura básica

Sol.

```
27 void estado_inicio_function() {
28     servoMotor.write(0);
29     digitalWrite(LED_RED, HIGH);
30     digitalWrite(LED_GREEN, LOW);
31     if (digitalRead(BTN) == 1) {
32         estado=ESTADO_DOS;
33     }
34 }
35 void estado_dos_function() {
36     digitalWrite(LED_RED, LOW);
37     digitalWrite(LED_GREEN, HIGH);
38     Serial.println("Pase_por_favor");
39     servoMotor.write(45);
40     estado=ESTADO_ESPERA;
41 }
```

Sol.

```
42 void estado_tres_function() {  
43     Serial.println("ESPERANDO....");  
44     if (digitalRead(BTNOUT)==1) {  
45         estado=ESTADO_INICIO;  
46     }  
47 }
```