

# Curso avanzado Arduino: Midiendo tiempos

[elCacharreo.com](http://elCacharreo.com)



[elCacharreo.com](http://elCacharreo.com)



José Antonio Vacas

# Tiempo: millis

- `long millis()` : número de milisegundos desde que se encendió la placa. Se vuelve a poner a cero cada 50 días aproximadamente.

```
long ini=millis();  
//hacemos algo  
long segundos=(millis()-ini)/1000;
```

- `micros()`
- `delay()`
- `delayMicroseconds()`



# Tiempos: micros

- `long micros()` : número de microsegundos desde que se encendió la placa. Se vuelve a poner a cero cada 70 minutos aproximadamente. Tiene una resolución de 4 microsegundos.

```
long ini=micros();
```

```
//hacemos algo
```

```
long segundos=(micros()-ini)/1000000;
```

- `delay()`
- `delayMicroseconds()`



# Tiempos: delay delayMicroseconds

- `delay(milisegundos)`: espera los milisegundos indicado
- `delayMicroseconds(microsegundos)`: espera los microsegundos indicado. El número máximo que soporta es 16383

Se recomienda no usar ninguna de las funciones de delay puesto que ocupan la cpu mientras esperan



# Tiempos: tiempos de pulsado 3.2.1

Medir tiempos de pulsaciones



# Tiempos: formatos humanos

Podemos trabajar en un formato más humano con Time

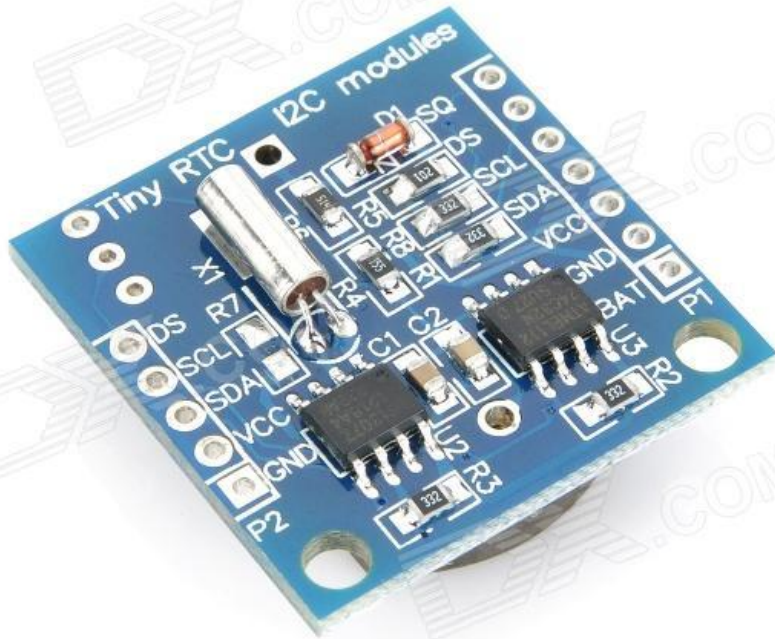
<http://playground.arduino.cc/Code/time>

```
hour();      // The hour now (0-23)
minute();    // The minute now (0-59)
second();    // The second now (0-59)
day();       // The day now (1-31)
weekday();   // Day of the week, Sunday is day 1
month();     // The month now (1-12)
year();      // The full four digit year: (2009, // 2010 etc)
```

Necesita una fuente de tiempo Real



# Tiempos: RTC

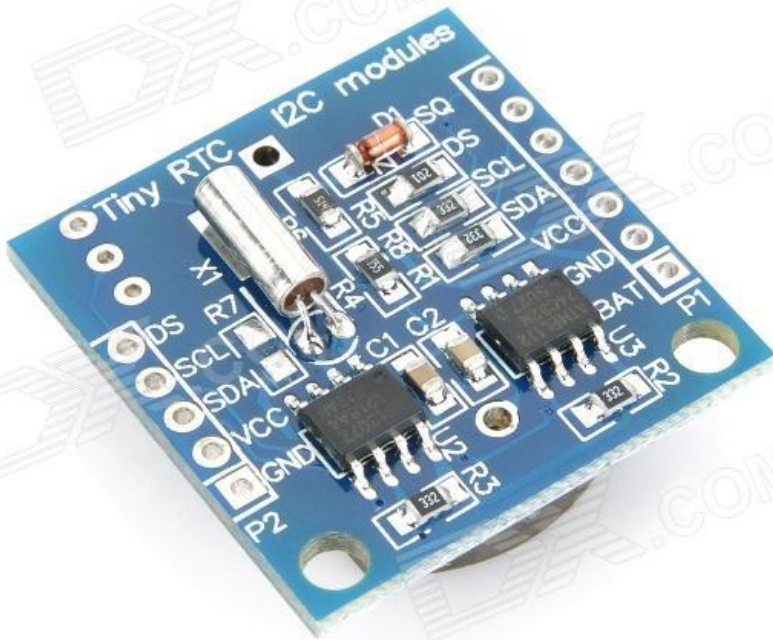


<http://club.dx.com/forums/forums.dx/threadid.1227253>

<http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>



# RTC: fechas



```
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 RTC;

void setup () {
    Serial.begin(57600);
    Wire.begin();
    RTC.begin();

    if (! RTC.isrunning()) {
        Serial.println("RTC is NOT running!");
        // following line sets the RTC to the date & time this
        // sketch was compiled
        RTC.adjust(DateTime(__DATE__, __TIME__));
    }
}

void loop () {
    DateTime now = RTC.now();
    DateTime future (now.unixtime() + 7 * 86400L + 30);
}
```

<http://learn.adafruit.com/ds1307-real-time-clock-breakout-board-kit/understanding-the-code>





# Tareas: Librería Timer

```
#include "Timer.h"

Timer t;

int pin = 13;

void setup()
{ Serial.begin(9600);
  pinMode(pin, OUTPUT);
  t.oscillate(pin, 100, LOW);
  t.every(1000, takeReading);}

void loop()
{ t.update();}

void takeReading()
{ Serial.println(analogRead(0));}
```

Ejecución de tareas  
repetitivas cada cierto  
tiempo

<http://playground.arduino.cc/Code/Timer>



# Programación dirigida a eventos

```
#include "Event.h"
```

```
EventManager evtManager;  
int ledPin = 12;  
boolean ledOn = false;  
struct LEDListener : public EventTask  
{ using EventTask::execute;  
  void execute(Event evt)  
  { if (ledOn == true) { digitalWrite(ledPin, LOW); }  
    else { digitalWrite(ledPin, HIGH); }  
  } } LEDListener;
```

```
void setup()  
{ pinMode(ledPin, OUTPUT);  
  evtManager.subscribe(Subscriber("led.on", &LEDListener));
```

```
Event toggleLed = Event("led.on");  
evtManager.triggerInterval(TimedTask(2000, toggleLed)); }
```

```
void loop()  
{ evtManager.tick();}
```

## Librería Event

[http://www.cainsvault.com/blog/arduino-event-manager?  
goback=%2Egde\\_1268377\\_member\\_256043556](http://www.cainsvault.com/blog/arduino-event-manager?goback=%2Egde_1268377_member_256043556)



# RTOS:

## Sistema Operativo de Tiempo Real

RTuinOS <https://github.com/PeterVranken/RTuinOS>

Duinos <https://code.google.com/p/duinos/>

FreeRTOS <http://maniacbug.wordpress.com/2012/01/31/freertos/>



# Conclusiones

Gracias por vuestra atención

