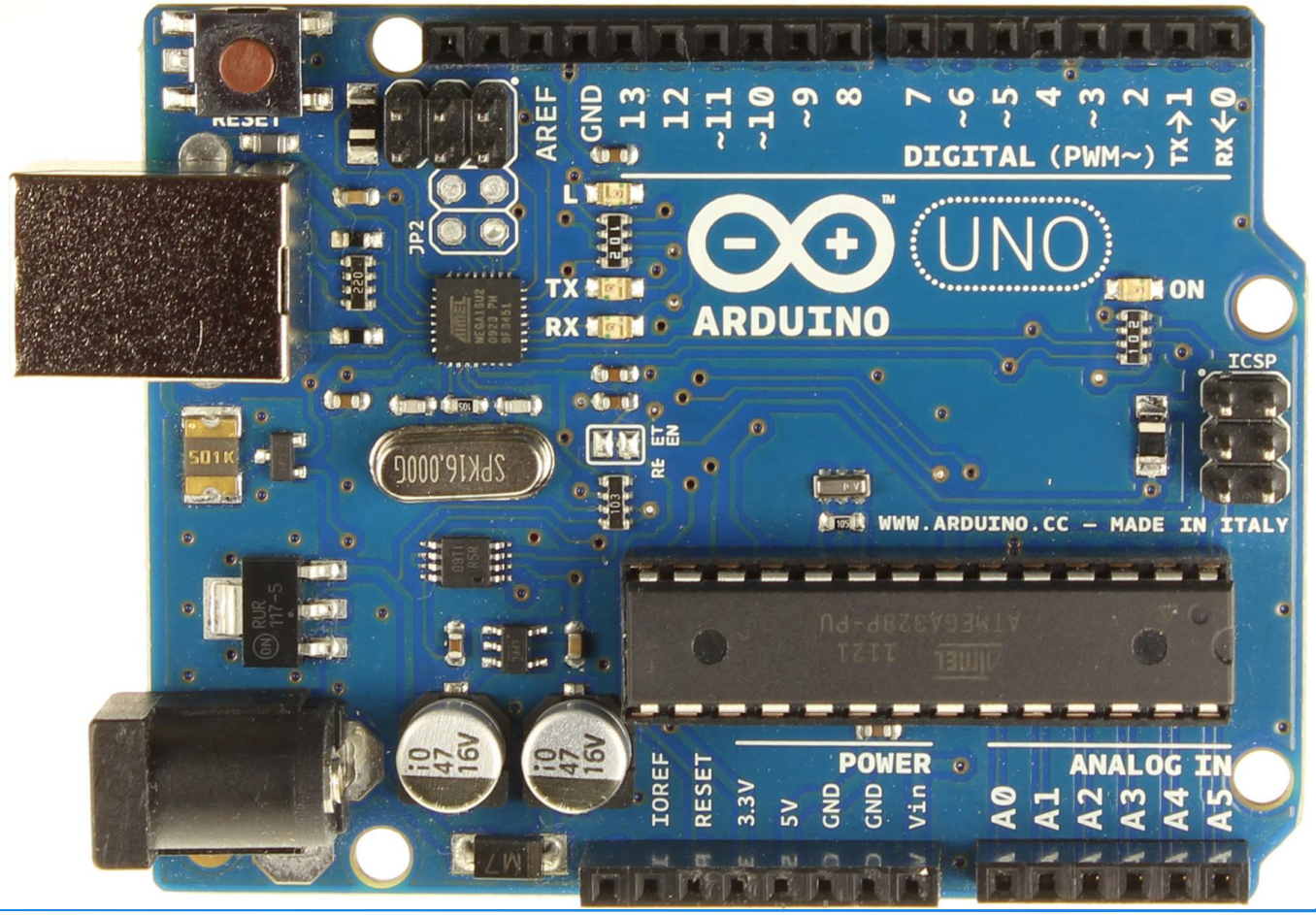
Curso avanzado sobre Arduino

Arduino Avanzado

ElCacharreo.com

Introducción a Arduino: Presente



Introducción a Arduino: Presente

José Antonio Vacas Martínez

blog

javacasm@elcacharreo.com

twitter
linkedin

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

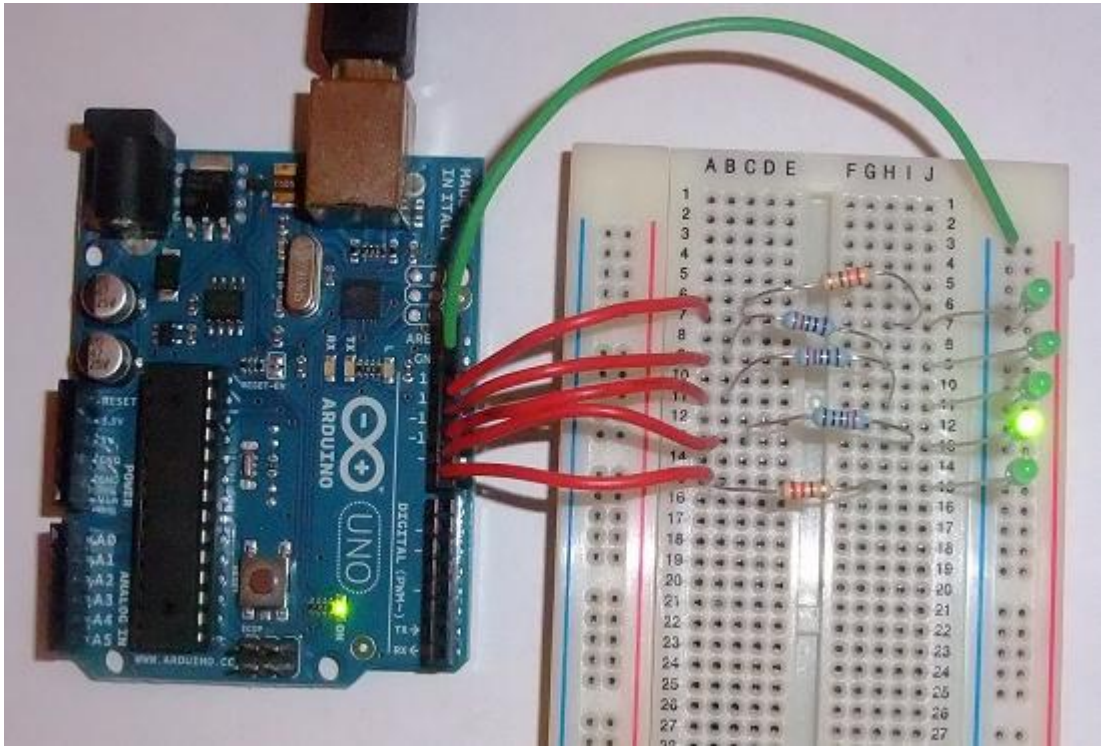
- delay(milisegundos): espera los milisegundos indicado
- delayMicroseconds(microsegundos): espera los microsegundos indicado. El número máximo que soporta es 16383

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: cylon (kit) 3.2.2



Versión avanzada

Tiempos: semáforo 3.2.3

Usar 3 leds hacer que se arranquen en la secuencia correcta: R-A-V

(Mejora, hacer que el Amarillo parpadee)

Tiempos: semáforo con pulsador 3.2.4

Modificar el semáforo anterior añadiendo un pulsador que acelera el paso al rojo

Añadir la opción de que si se ha pulsado hace poco no se vuelva a activar la secuencia

Tiempos: parpadeo sin delay 3.3.5

```
const int ledPin =  13;      // the number of the LED pin
int ledState = LOW;          // ledState used to set the LED
long previousMillis = 0;     // will store last time LED was updated

long interval = 1000;        // interval at which to blink (milliseconds)

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

void loop()
{
  unsigned long currentMillis = millis();
  if(currentMillis - previousMillis > interval) {
    previousMillis = currentMillis;
    if (ledState == LOW)
      ledState = HIGH;
    else
      ledState = LOW;
    digitalWrite(ledPin, ledState);
  }
}
```

Sonido: teoría

Usaremos un piezo speaker para tocar notas musicales.
Producimos una onda cuadrada de la frecuencia correspondiente que el piezo convertirá en una nota

El cálculo de los tonos se hace con la siguiente fórmula:

timeHigh = periodo / 2 = 1 / (2 * toneFrecuencia)

Según la siguiente tabla:

nota	frecuencia	periodo	timeHigh
do	261 Hz	3830	1915
re	294 Hz	3400	1700
mi	329 Hz	3038	1519
fa	349 Hz	2864	1432
sol	392 Hz	2550	1275
la	440 Hz	2272	1136
si	493 Hz	2028	1014
do	523 Hz	1912	956

http://www.arduino.cc/en/Tutorial/Melody

Sonido: tone

tone(pin, frecuencia)
tone(pin, frequencia, duracion)

- Genera una señal cuadrada de la frecuencia y duración definidas
- Interfiere con el pwm de los pines 3 y 11

noTone(pin)

- detiene la ejecución del tone actual
- se utiliza para usar varios pines generando sonido

http://www.arduino.cc/en/Tutorial/Melody

Sonido: ModPlayer

- Proyecto avanzado capaz de reproducir 4 voces simultáneamente
- http://www.instructables.com/id/Turn-your-Arduino-into-a-4-voice-wavetable-synth-w/
- Vídeo http://player.vimeo.com/video/41439986?title=0&byline=0

¿Cómo funciona?

Ejemplo Sonido: cancioncilla 3.3.1

```
int speakerPin = 9;

int length = 15; // the number of notes
char notes[] = "ccggaagffeeddc "; // a space represents a rest
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 };
int tempo = 300;

void playTone(int tone, int duration) {
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(tone);
    digitalWrite(speakerPin, LOW);
    delayMicroseconds(tone);
  }
}

void playNote(char note, int duration) {
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014,
956 };
  // play the tone corresponding to the note name
  for (int i = 0; i < 8; i++) {
    if (names[i] == note) {
      playTone(tones[i], duration);
    }
  }
}

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

void loop() {
  for (int i = 0; i < length; i++) {
    if (notes[i] == ' ') {
      delay(beats[i] * tempo); // rest
    } else {
      playNote(notes[i], beats[i] * tempo);
    }

    delay(tempo / 2);
  }
}
```

http://www.arduino.cc/en/Tutorial/Melody

Ejemplo: Sintetizador 3.3.2

A partir de las funciones tone, hacer que suene una nota dependiendo del pulsador usado y del potenciómetro

Ejemplo: Indicador de temperatura 3.3.3

Hacer que dependiendo de la temperatura se produzcan sonidos más agudos

Ejemplo: James Bond

```
#include "pitches.h"

#define NO_SOUND 0 // make the rests in music

int melody[] = {  NOTE_E4,NOTE_F4,NOTE_F4,NOTE_F4,NOTE_F4,NOTE_E4,NOTE_E4,NOTE_E4,  NOTE_E4,
NOTE_G4,NOTE_G4,NOTE_G4,NOTE_G4,NOTE_E4,NOTE_E4,NOTE_E4,   NOTE_E4,NOTE_F4,NOTE_F4,
NOTE_F4,NOTE_F4,NOTE_E4,NOTE_E4,NOTE_E4,   NOTE_E4,NOTE_G4,NOTE_G4,NOTE_G4,NOTE_G4,
NOTE_E4,NOTE_E4,NOTE_E4,   NOTE_DS5,NOTE_D5,NOTE_B4,NOTE_A4,NOTE_B4,   NOTE_E4,NOTE_G4,
NOTE_DS5,NOTE_D5,NOTE_G4,NOTE_B4,   NOTE_B4,NOTE_FS5,NOTE_F5,NOTE_B4,NOTE_D5,NOTE_AS5,
NOTE_A5,NOTE_F5,NOTE_A5,NOTE_DS6,NOTE_D6,NO_SOUND };

// note duration: 1 = whole note, 2 = half note, 4 = quarter note, 8 = eighth note, etc.

int noteDurations[] = {   8,16,16,8,4,8,8,8,  8,16,16,8,4,8,8,8,  8,16,16,8,4,8,8,8,  8,16,16,8,4,8,8,8,  8,2,8,8,1,
8,4,8,4,8,8,  8,8,4,8,4,8,  4,8,4,8,3};

int pace = 1450; // change pace of music("speedy")

void setup() {

  for (int Note = 0; Note <54; Note++) {    int duration = pace/noteDurations[Note];//Adjust duration with the pace of
music

    tone(8, melody[Note],duration); //Play note    // to distinguish the notes, set a minimum time between them.

    delay(duration*1.2);  } }

void loop() {}
//End of Sketch
```

Conclusiones

Gracias por vuestra atención