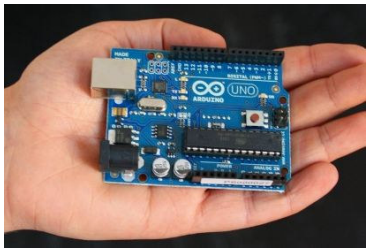
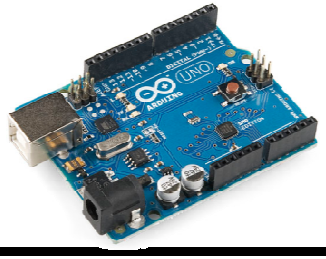
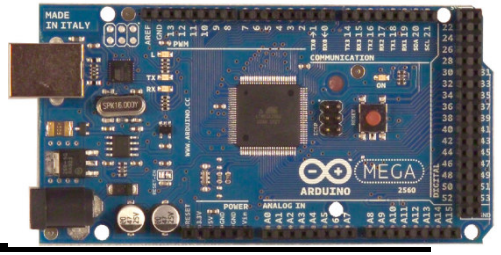




Programación de Arduino

Pedro Corcuera

Dpto. Matemática Aplicada y
Ciencias de la Computación

Universidad de Cantabria

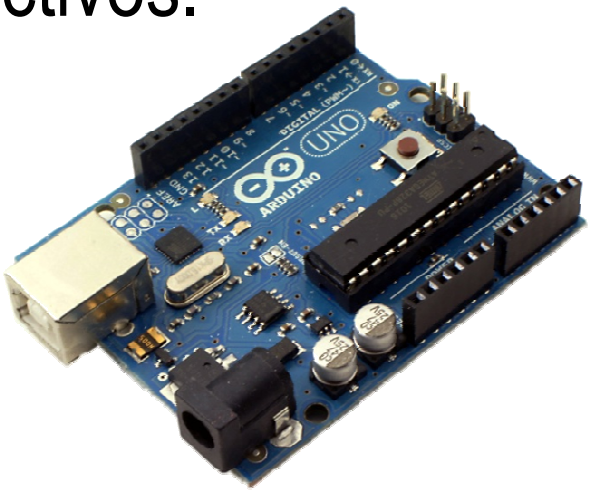
corcuerp@unican.es

Índice General

- Introducción
- Ecuaciones diferenciales
- Método Euler

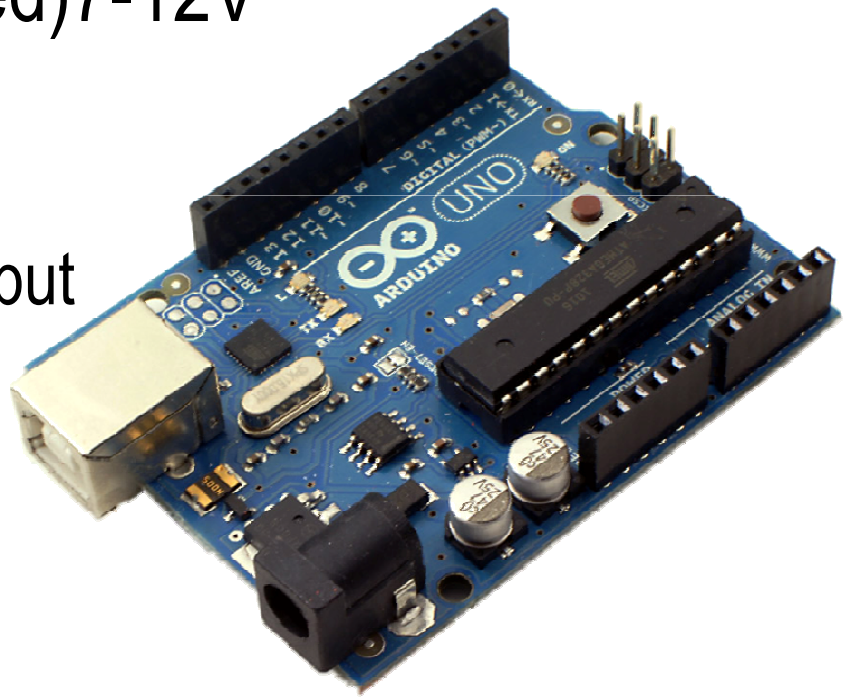
Introducción

- Arduino (http://www.arduino.cc/) es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.



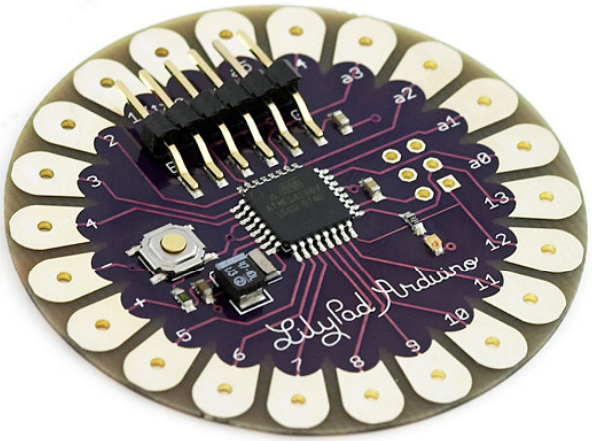
Arduino UNO

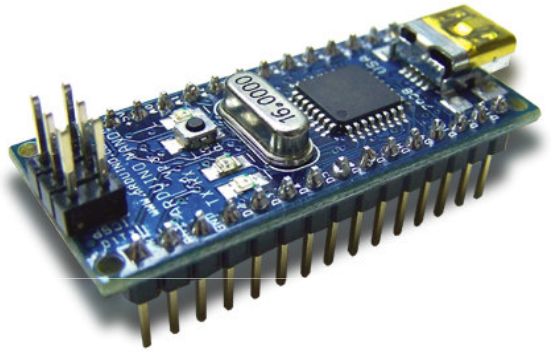
- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended)7-12V
- Input Voltage (limits)6-20V
- Digital I/O Pins14
  - of which 6 provide PWM output
- Analog Input Pins 6
- Flash Memory 32KB
- SRAM 2KB
- EEPROM 1KB
- Clock Speed 16 MHz

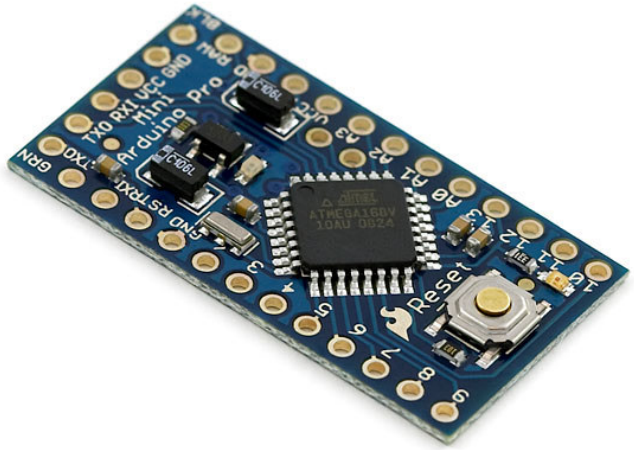


22 €

Otras placas Arduino

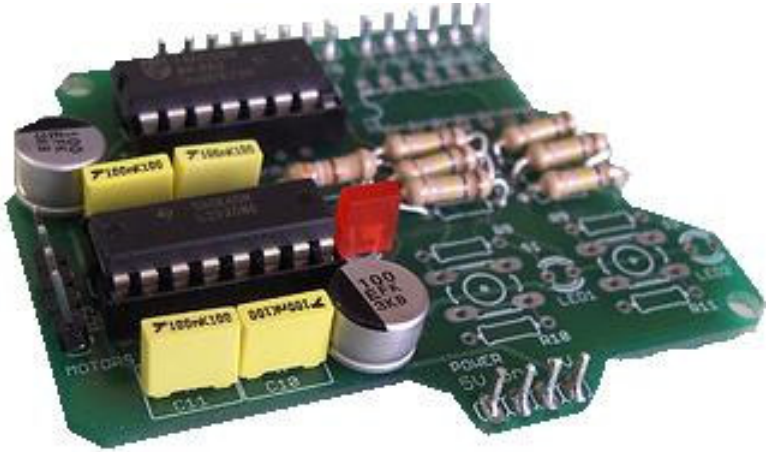




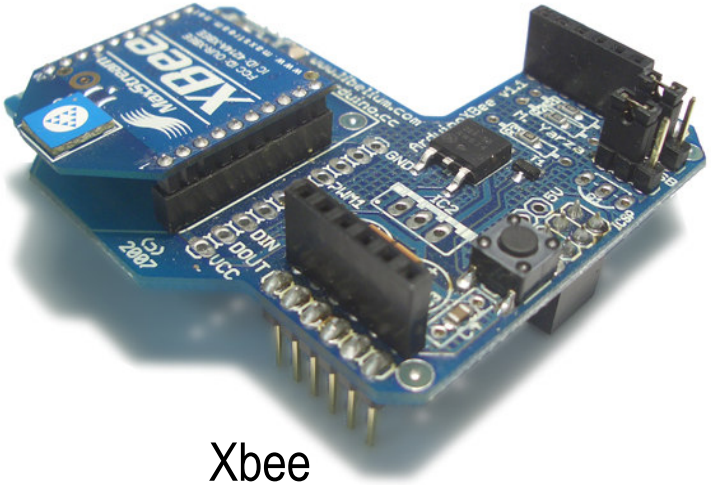


Algunos Shields

Ethernet



Motor

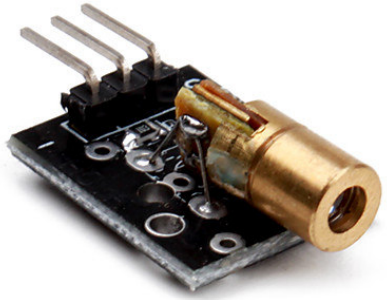


Xbee

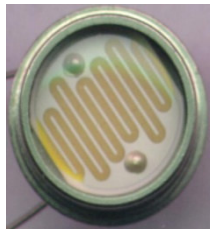
LCD

Algunos Sensores

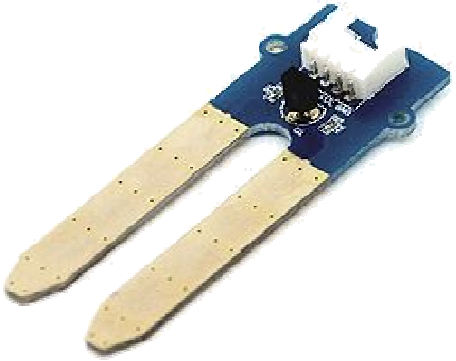
Fotodiodos



Láser



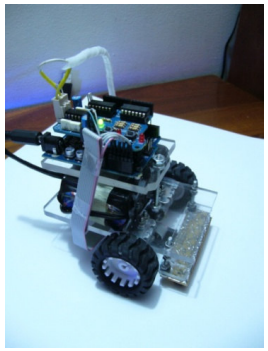
Fotoresistencias



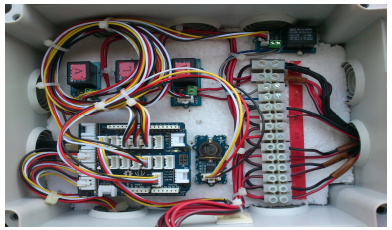
Humedad

Ultrasonido

Algunos desarrollos



Robot



Control de Invernadores

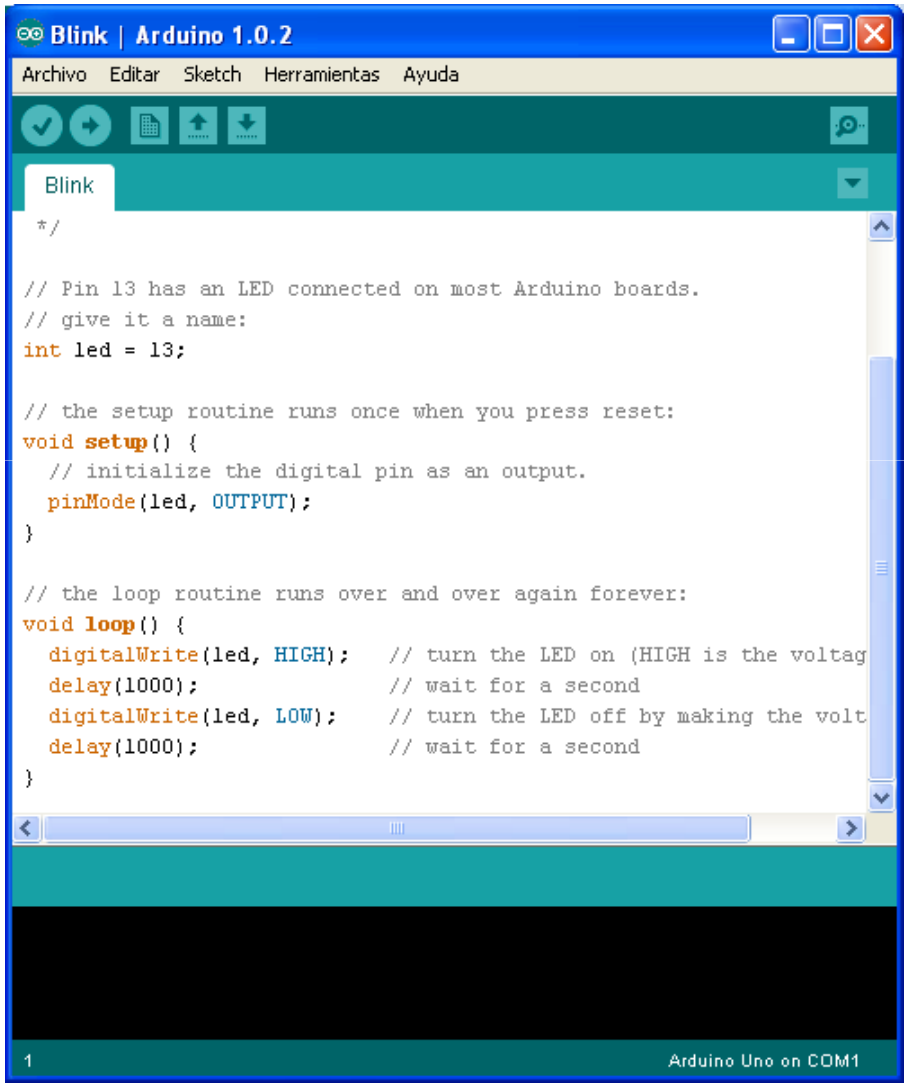
Massimo Banzi at TED (http://www.youtube.com/watch?v=UoBUXOOdLXY)

Programación

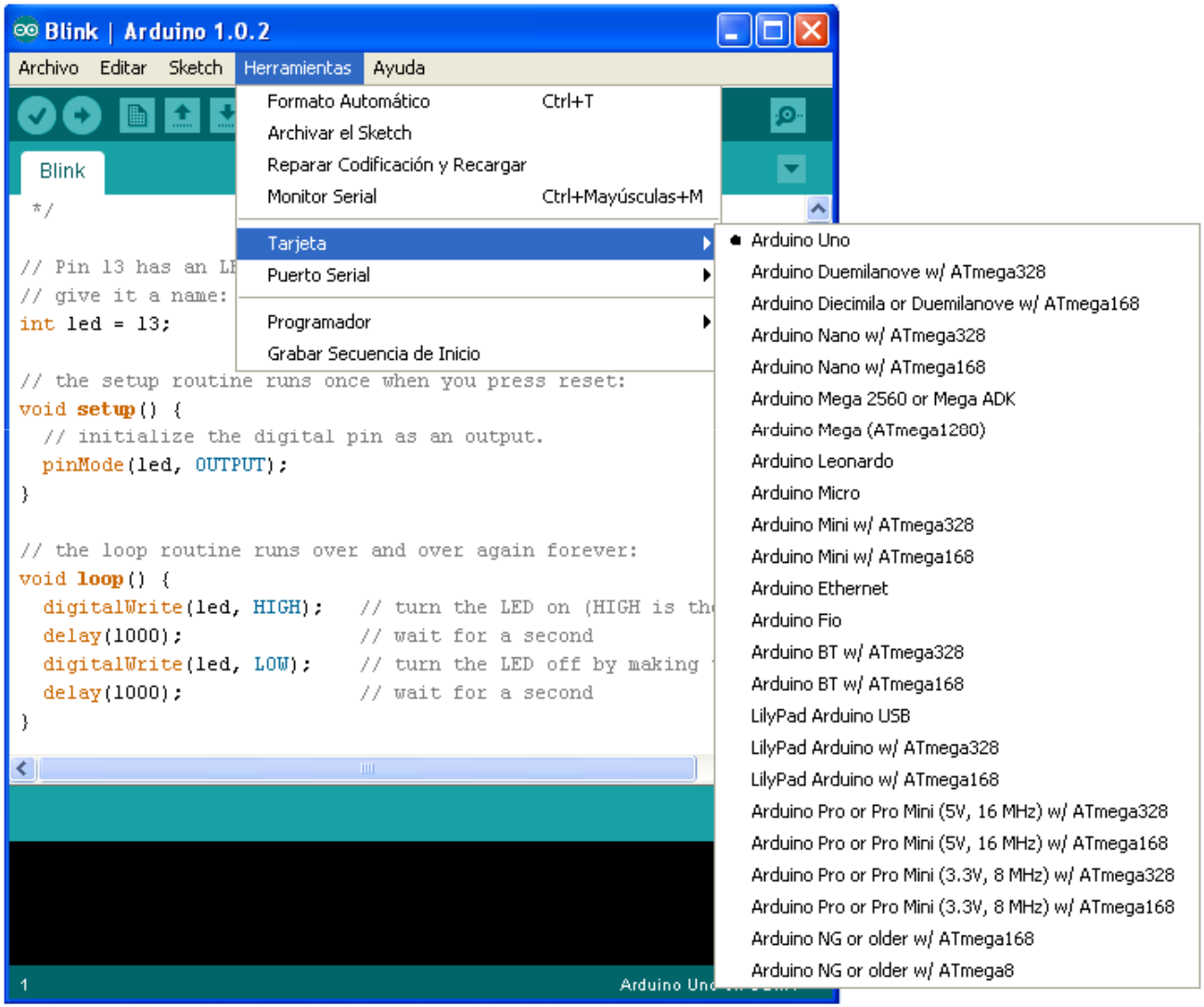
- Lenguaje de programación Arduino (basado en Wiring, a su vez basado en C/C++)
  - El entorno Arduino (basado en Processing, a su vez desarrollado en Java) es abierto.
  - Se descarga de http://arduino.cc/es/Main/Software
- Otros entornos:
  - Autónomos
    - Minibloq, Ardubloq, Amici, ModKit, VirtualBreadBoard, Matlab
  - Esclavos
    - Etoys (Squeak), S4A (Scratch), Labview, Firefly, MyOpenLab

Entorno de programación

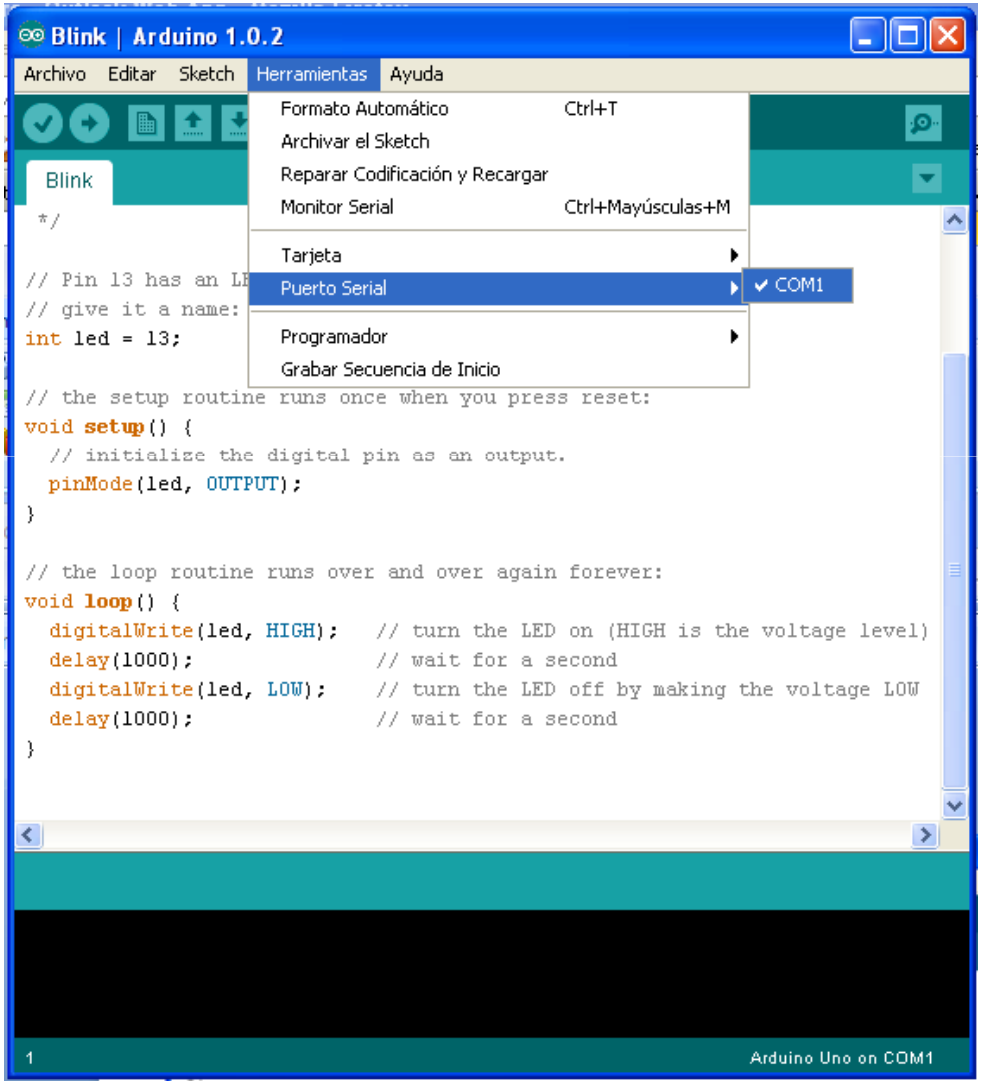
Programas = Sketches



Tipo de tarjeta (board)



Puerto serie / COM



Partes de un Sketch

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Programación: estructura

```
void setup() {
      ….se ejecuta una vez cuando se ejecuta
      ….un programa Arduino
}
void loop() {
      …. se ejecuta después de setup. Se ejecuta
      …. de manera repetida hasta que se quita la
      …. tensión
}
```

Programación: Comentarios y sintaxis

- Igual que en C
  - Multilínea /* Comentario */
  - Línea // Linea
- { } para encerrar código
- ; para terminar instrucciones

Programación: Operadores de comparación

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)

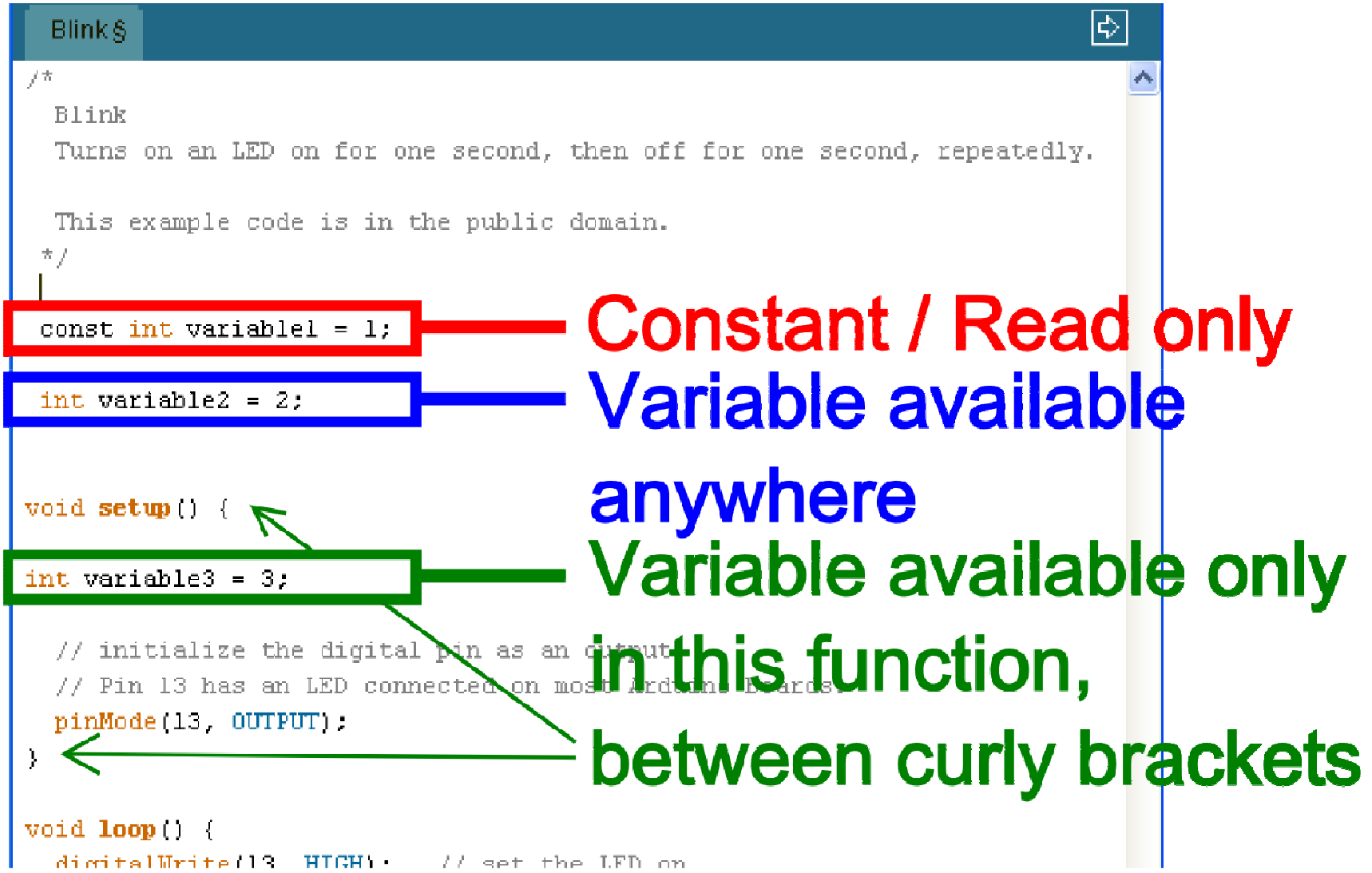
Programación: Operadores matemáticos

- = (assignment)
- % (módulo)
- +
- -
- *
- /

Programación: Variables

- Tipos básicos:
  - int
  - long
  - boolean
  - float
  - char

Programación: Alcance de variables



Programación: estructura de control if

```
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

If Statement

Programación: estructura de control loop

Blink

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeat

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Loop

Programación: estructura de control for

```
void setup()
{
  //Set each pin connected to an LED to output mode (pulling high
  for(int i = 0; i < 8; i++){                    //this is a for loop and will r
      pinMode(ledPins[i],OUTPUT); //we use this to set each LED p
  }                                              //the code this replaces is

  /* (commented code will not run)
   * these are the lines replaced by the for loop above they do e
   * same thing the one above just uses less typing
  pinMode(ledPins[0],OUTPUT);
  pinMode(ledPins[1],OUTPUT);
  pinMode(ledPins[2],OUTPUT);
  pinMode(ledPins[3],OUTPUT);
```

For loop

Programación: estructura de control while

```
void loop() {
  // while the button is pressed, take calibration readings:
  while (digitalRead(buttonPin) == HIGH) {
    calibrate();
  }
  // signal the end of the calibration period
  digitalWrite(indicatorLedPin, LOW);

  // read the sensor:
  sensorValue = analogRead(sensorPin);

  // apply the calibration to the sensor reading
  sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);

  // in case the sensor value is outside the range seen during calibration
  sensorValue = constrain(sensorValue, 0, 255);

  // fade the LED using the calibrated value:
  analogWrite(ledPin, sensorValue);
}
```

Programación: señales digitales

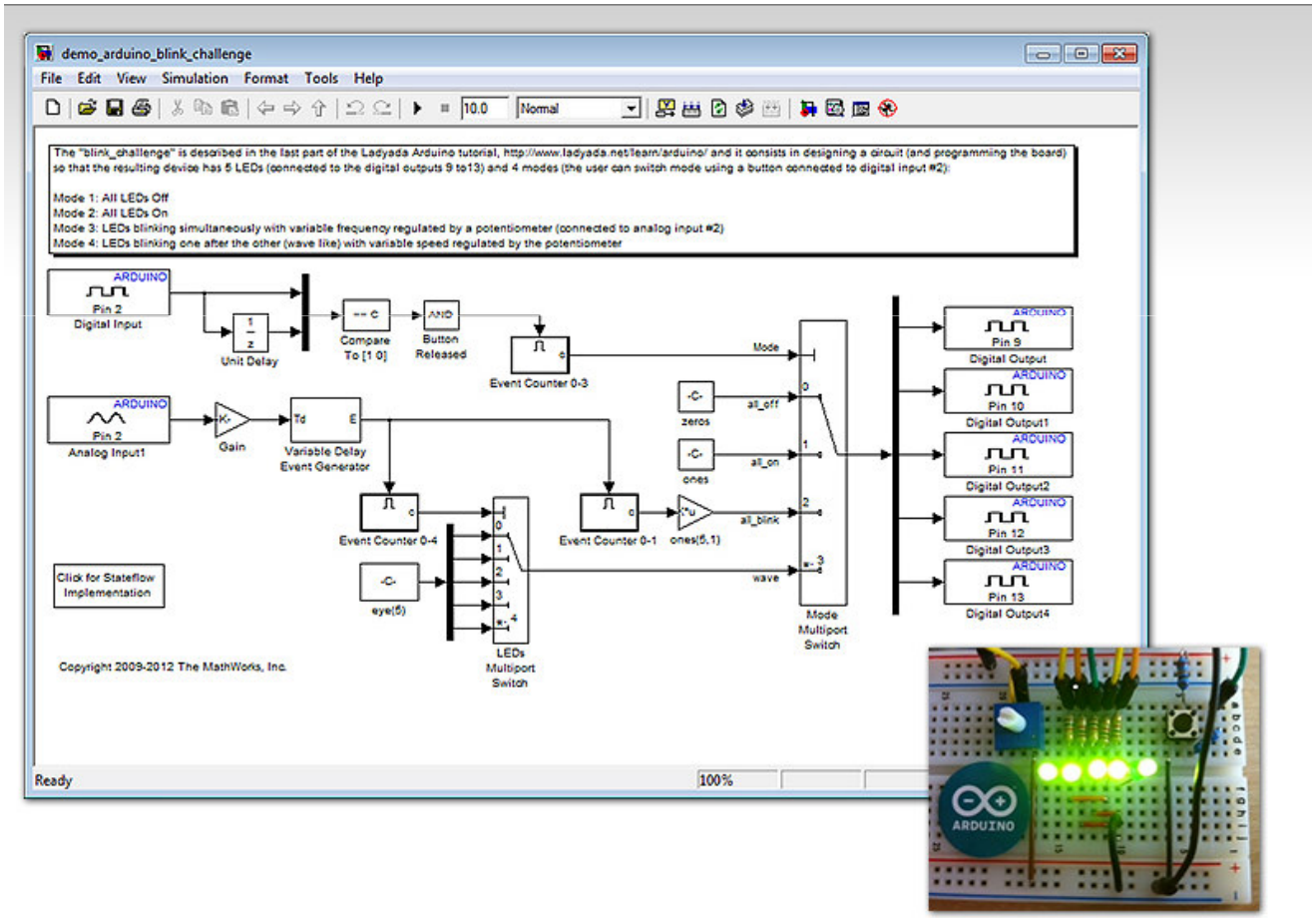
- pinMode (pin, mode) : usado para asignar el modo de un pin. pin es el número del pin que se quiere asignar 0-19. mode puede ser INPUT o OUTPUT.
- digitalWrite (pin, value): una vez un pin es asignado como OUTPUT se puede asignar a HIGH (5 Volts) o LOW(0 volts). Básicamente significa ON y OFF.
- int digitalRead (pin): una vez un pin es asignado como INPUT se puede usar para devolver HIGH (5 Volts) o LOW(0 volts).

Programación: señales analógicas

- analogWrite (pin, value); algunos de lo pines soportan pulsos modulados (PWM) (3, 5, 6 ,9, 10, 11). Esto pone el pin en on o off muy rápido comportándose como una salida analógica. El valor es un número entre 0 (0 v) y 255 (5 v).
- int analogRead (pin); puede leer un voltaje. Devuelve un valor entre 0 (0 v) y 1024 (5 v).

Programación: desde Matlab/Simulink

- http://www.mathworks.es/academia/arduino-software/



Programación: desde LabView

- http://sine.ni.com/nips/cds/view/p/lang/es/nid/209835