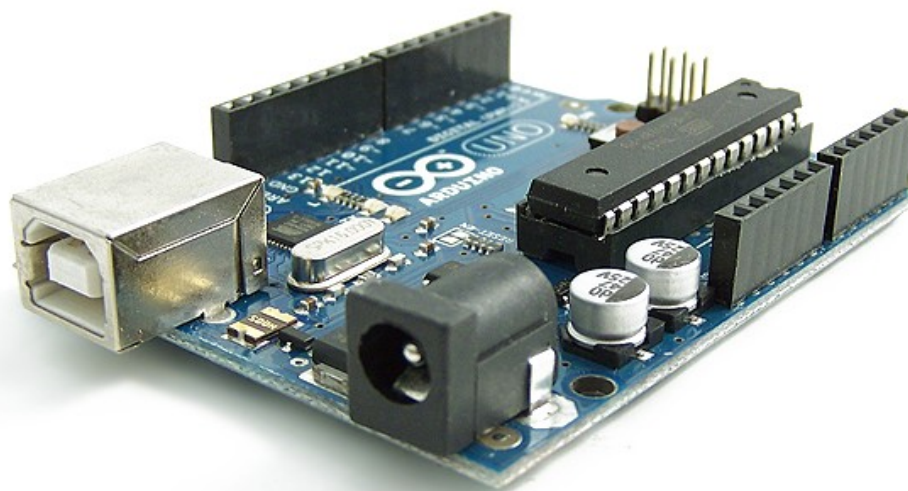


ARDUINO en CABAL

Un taller introductorio, estilo Cabal.



<http://www.arduino.cc/>

1. Que es arduino.

1a. Características generales del Atmega328

2. El bootloader de Arduino, descripción breve.

3. Es arduino la única plataforma libre?.

4. Instalación y descarga de códigos de ejemplo.

5. Leds & switches, Debouncing.

5a. Consideraciones importantes

6. Entradas analogicas

7. Comunicación serial.

8. Funciones

9. Corrigiendo Errores

8. Creacion de Librerias

9. El "big main loop" , una estrategia de programación

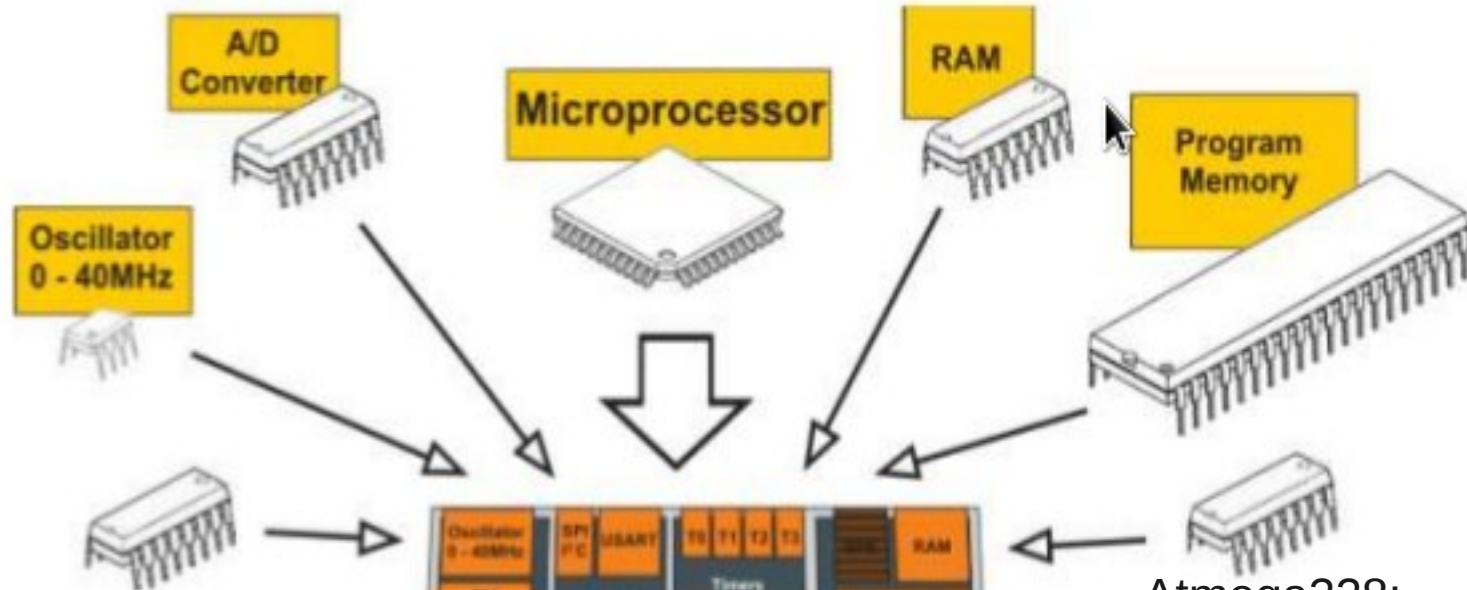
10. Real time clock. I2c port

javierzavalaponce@gmail.com

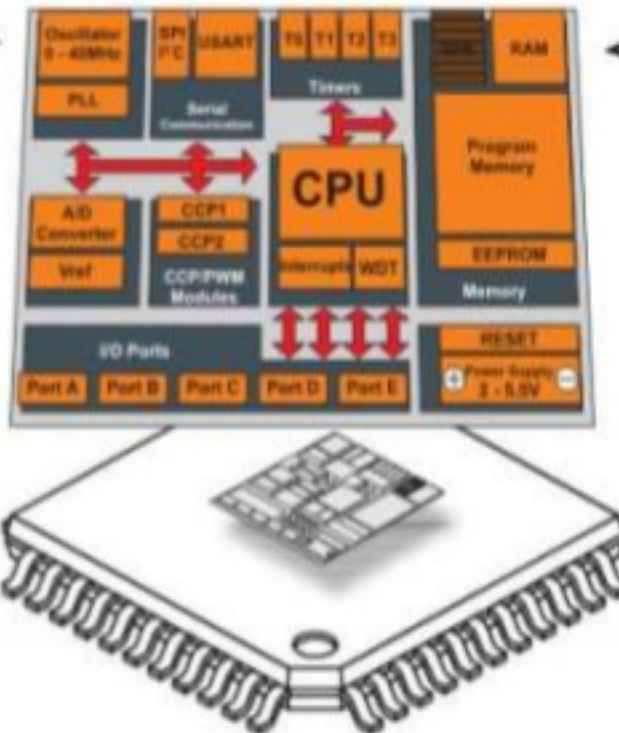
Arduino es...

- Placa con un microcontrolador: ATmega328
- Facil comenzar en el mundo de los microcontroladores. Eagle files
- open-source electronics prototyping platform
- Gran comunidad apoyando el proyecto.





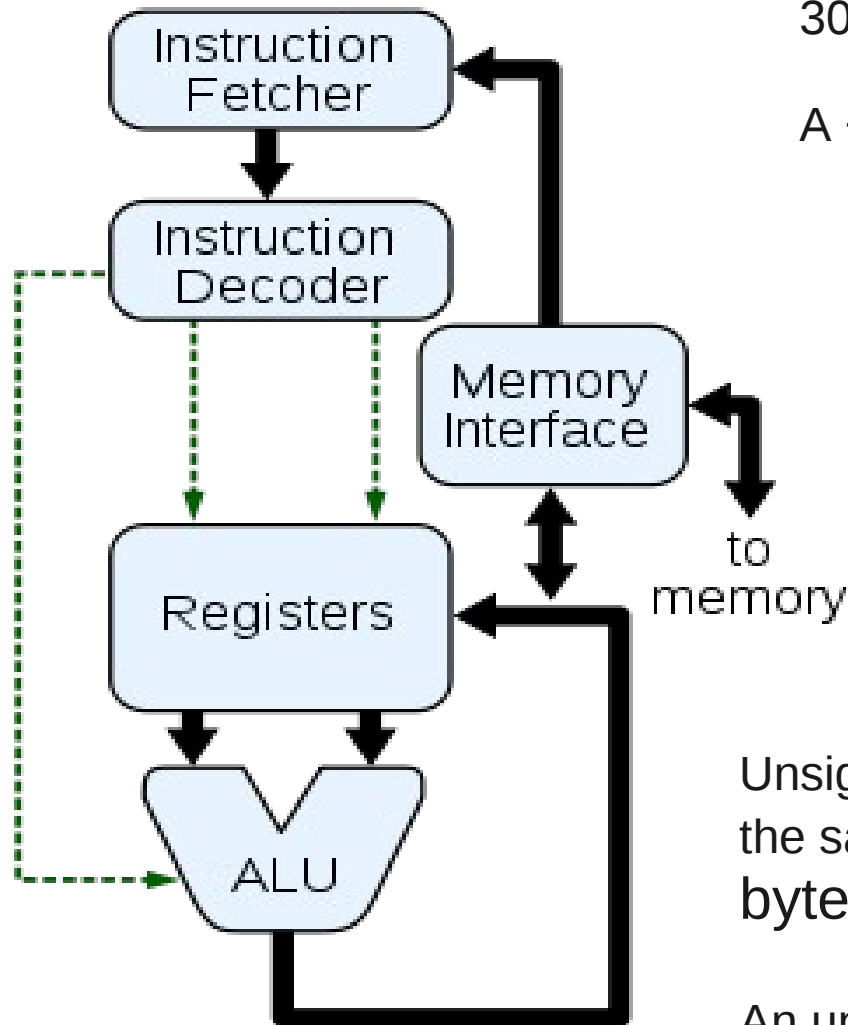
Small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals



Atmega328:
 32K memoria de programa
 2K de ram
 ADC
 PWM
 Uart
 I2c
 Timer 8&16 bits, etc...

Microcontroller

8 BITS. Max 0xff , = 255,
300?
500?



300 + 500

A + B;

a) unsigned int i;

b) unsigned char i;

For(i =0;i<200;++i)

Unsigned ints (unsigned integers) are the same as ints in that they store a 2 byte value.

An unsigned char type occupies 1 byte of memory.

+ opciones aparte de arduino

<http://designsomething.org/>

<http://mbed.org/>



Eth
Usb
can

Microcontrollers are getting cheaper, more powerful and more flexible, but there remains a barrier to a host of new applications; someone has to build the first prototype! With mbed, we've focused on getting you there as quickly as possible

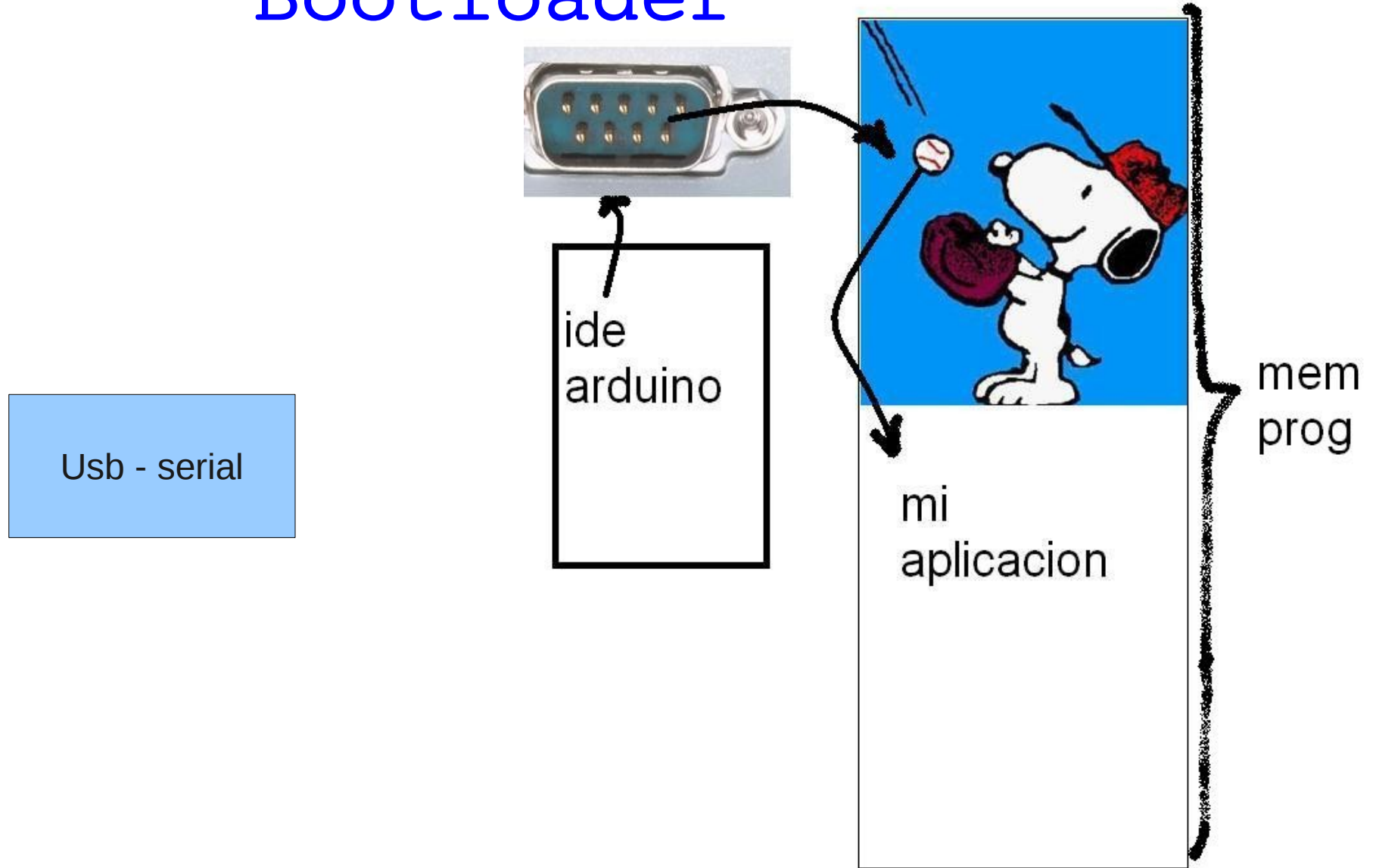


http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en021940

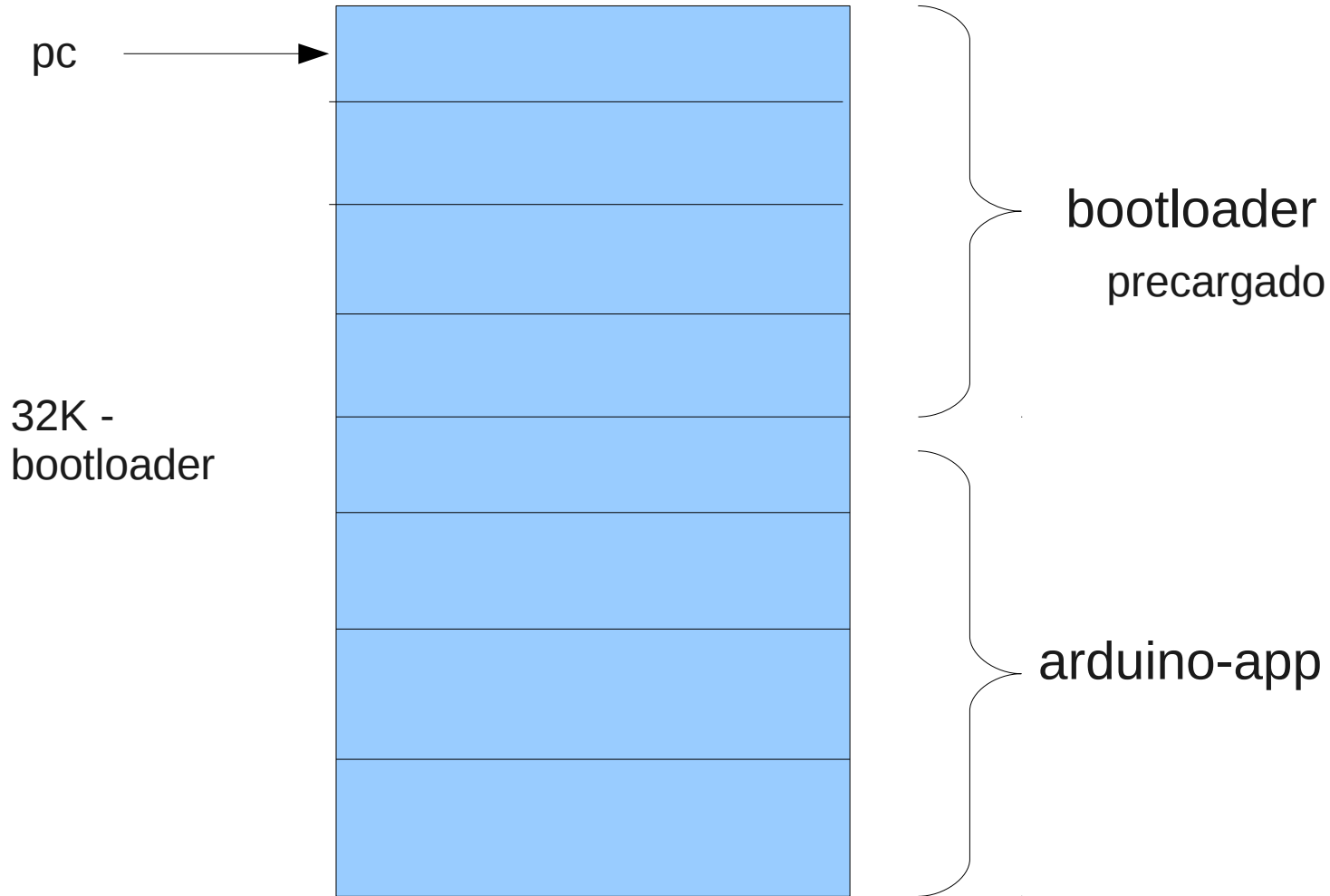
Oct.1.2011

Arduino Cabal

Bootloader



Cuando el micro va a reset.



http://es.wikipedia.org/wiki/Unidad_central_de_procesamiento

Vistazo al bootloader

```
/* Request programmer ID */
/* Not using PROGMEM string due to boot block in m128 being
beyond 64kB boundry */
/* Would need to selectively manipulate RAMPZ, and it's only 9
characters anyway so who cares. */
else if(ch=='1') {
    if (getch() == ' ') {
        putchar(0x14);
        putchar('A');
        putchar('V');
        putchar('R');
        putchar(' ');
        putchar('I');
        putchar('S');
        putchar('P');
        putchar(0x10);
    } else {
        if (++error_count == MAX_ERROR_COUNT)
            app_start();
    }
}
}
```

Oct.1.2011

Arduino Cabal

Instalación

Help!!

YUM INSTALL ARDUINO

En las primeras pruebas con el arduino y fedora 15 tuvimos algunos problemas

1) Comunicación con el puerto serial, dándole permisos:

```
chmod 777 /dev/ttyACM0
```

2) La versión de avr-gcc que se corrige con un downgrade

```
yum downgrade avr-gcc avr-gcc-c++
```

queda avr-gcc avr-gcc'c++ en versión 4.6.1

1) CONFIGURAR...

Tools --board

Tools – serial port

A TOUR OF THE ARDUINO
INTERNALS: HOW DOES HELLO
WORLD ACTUALLY WORK?

[http://urbanhonking.com/ideasfordozens/
2009/05/18/an_tour_of_the_arduino_inte
rna/](http://urbanhonking.com/ideasfordozens/2009/05/18/an_tour_of_the_arduino_interna/)

File - examples – Basics –blink

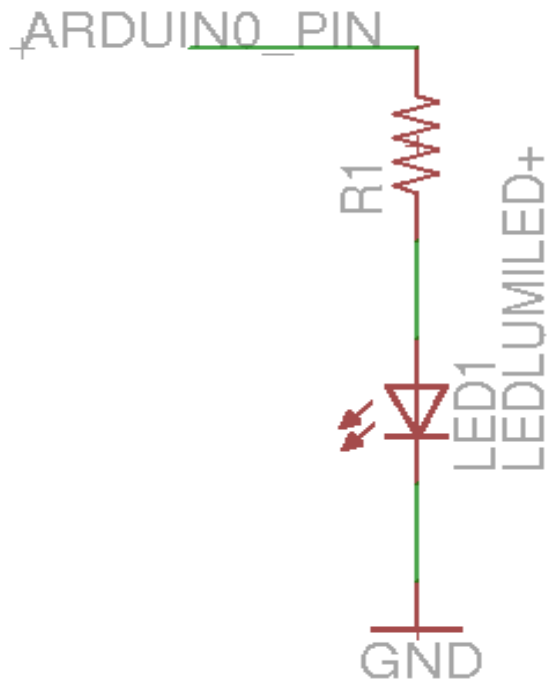
(hay otra opcion en linea...)

File - examples – Comunications – ASCII table

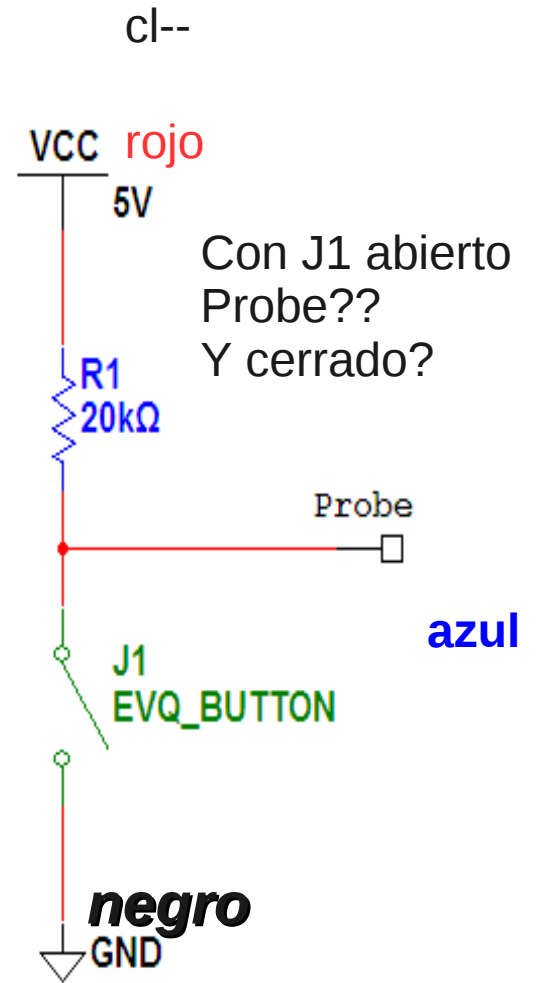
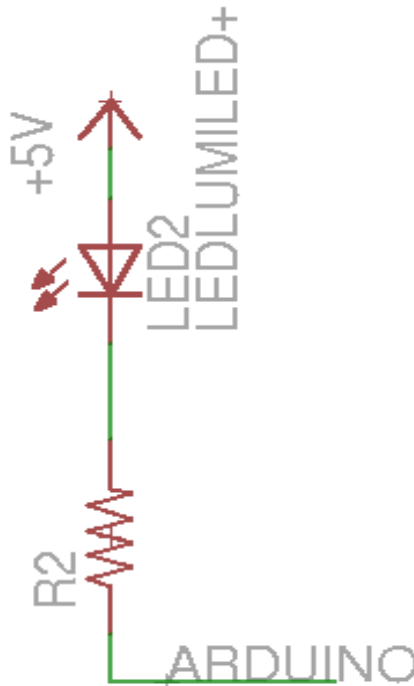
pausa

Leds & Switches

$3V/R = 20mA$
Standar
220,330ohms



- 1- R de que valor?
- 2- puedo conectar asi 8 leds



<http://www.engscope.com/pic-example-codes/basic-io-button-debounce/>

Datasheet

Although each I/O port can sink more than the test conditions (20 mA at VCC = 5V, 10 mA at VCC = 3V) under steady state

conditions (non-transient), the following must be observed:

ATmega48PA/88PA/168PA/328P:

- 1] The sum of all IOL, for ports C0 - C5, ADC7, ADC6 should not exceed 100 mA.
- 2] The sum of all IOL, for ports B0 - B5, D5 - D7, XTAL1, XTAL2 should not exceed 100 mA.

1)Examples-Digital-Button

DIFICIL APRECIAR EL REBOTE.

2)Modificacion de el programa.

En setup:

```
Serial.begin(9600);  
Serial.println("Linux Cabal!!!");
```

Dentro de if:

```
Serial.println("high!!!");
```

El mundo es analogico....

File – examples-analog-analogInput
modificado

```
-----  
int sensorPin = A0; // pin for the potentiometer  
int ledPin = 13; // select the pin for the LED  
int sensorValue = 0; // variable to store the value  
void setup() {  
    // declare the ledPin as an OUTPUT:  
    pinMode(ledPin, OUTPUT);  
    Serial.begin (9600);  
    Serial.println("Linux Cabal!!!"); /*say hello*/  
}  
  
void loop() {  
    // read the value from the sensor:  
    sensorValue = analogRead(sensorPin);  
    delay(800);  
    Serial.println(sensorValue);  
}
```

Funciones

<http://arduino.cc/estemp/Reference/FunctionDeclaration>

Anatomy of a C function

Datatype of data returned,
any C datatype.

"void" if nothing is returned.

Parameters passed to
function, any C datatype.

Function name

```
int myMultiplyFunction(int x, int y){  
  int result;  
  result = x * y;  
  return result;  
}
```

Return statement,
datatype matches
declaration.

Curly braces required.

Las funciones ayudan al
programador a ser
organizado

sólo deben ser depuradas de
errores una vez.

hacen el sketch mas
pequeño y mas compacto por
que las secciones de código
se reutilizan varias veces.

Se declaran antes de que se
puedan usar.

Funciones...

```
int sensorPin = A0; // select the input pin for the potentiometer
int ledPin = 13; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the sensor
```

```
//prototipo
```

```
int convertir(int dato);
```

```
void setup() {
```

```
    // declare the ledPin as an OUTPUT:
```

```
    pinMode(ledPin, OUTPUT);
```

```
    Serial.begin (9600);
```

```
    Serial.println("Linux Cabal!!"); /*say hello*/
```

```
}
```

```
void loop() {
```

```
    // read the value from the sensor:
```

```
    delay(800);
```

```
    sensorValue = analogRead(sensorPin);
```

```
    sensorValue = convertir(sensorValue);
```

```
    Serial.print(sensorValue);Serial.println("mV")
```

```
}
```

```
//implementacion
```

```
int convertir(int dato)
```

```
{
```

```
    return (dato * 5);
```

```
}
```

*Ejemplo para pasar de
0 a 1023 a un valor en
mVolts*

Doxygen!

Documentacion

<http://www.stack.nl/~dimitri/doxygen/>

Ejemplo:

[http://www.stack.nl/~dimitri/doxygen/
examples/qtstyle/html/class_test.html](http://www.stack.nl/~dimitri/doxygen/examples/qtstyle/html/class_test.html)

Oct.1.2011

Arduino Cabal

Correcting Arduino Compiler Errors

<http://diyroboticslab.wordpress.com/2009/06/05/correcting-arduino-compiler-errors/>

```
/*--- Blink an LED ---//
```

```
//Associate LEDs with an Arduino Digital pin.
```

```
//The Arduino already has a built-in LED that we can use on Digital Pin 13.
```

```
int ledPin = 23; \\We're using Digital Pin 23 on the Arduino.
```

```
void setup();
```

```
{  
  pinMode(ledPin OUTPUT); //Set up Arduino pin for output only.  
}
```

```
loop()
```

```
(  
  /The HIGH and LOW values set voltage to 5 volts when HIGH and 0 volts LOW.  
  digitalWrite(ledPin, high); //Setting a digital pin HIGH turns on the LED.  
  delay(1000); //Get the microcontroller to wait for one second.  
  digitalWrite(ledPin, LOW); //Setting the pin to LOW turns the LED off.  
  Delay(1000); //Wait another second with the LED turned off.  
  }  
}
```

Oct.1.2011

Arduino Cabal

1 * *linea*

Librerías

<http://arduino.cc/es/Hacking/LibraryTutorial>

Para una librería necesita al menos dos archivos: un archivo de cabecera (con extensión. H) y el código fuente (extensión. cpp).

Una clase es simplemente una colección de funciones y variables agrupadas en un mismo lugar. Estas funciones y variables pueden ser públicas, o privadas,

Cada clase tiene una función especial conocida como constructor, que se utiliza para crear una instancia de la clase (o sea, un objeto). El constructor tiene el mismo nombre que la clase, y no devuelve nada. Si no definimos un constructor el compilador creará uno por defecto, sin parámetros, que no hará absolutamente nada

`\\arduino-0022\libraries`

1. crea carpeta dentro de la carpeta librerías "mylib"
- 2.dentro crea archivo mylib.cpp y mylib.h

mylib.cpp

```
#ifndef MYLIB_H
#define MYLIB_H
unsigned complementa (unsigned char dato);

class MYLIBClass
{
public:
    unsigned char add1(unsigned char);
};

extern MYLIBClass MYLIB;

#endif//MYLIB_H
```

```

/*
  MYLIB es una libreria que hvace magia...

*/
/*****
 * Includes
 *****/
#include "MYLIB.h"

/*****
 * Constructors
 *****/

/*****
 * User API
 *****/
unsigned complementa (unsigned char dato)
{
  return ~dato;
}

unsigned char MYLIBClass::add1(unsigned char data)
{
  return ++data;
}

MYLIBClass MYLIB;

```

```

#include <mylib.h>

int sensorPin = A0; // select the input pin for the potentiometer
int ledPin = 13;    // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the sensor

void setup() {
    // declare the ledPin as an OUTPUT:
    pinMode(ledPin, OUTPUT);
    Serial.begin (9600);
    Serial.println("Linux Cabal!!!"); /*say hello*/
}
void loop() {
    unsigned char val = 99;
    sensorValue = analogRead(sensorPin);
    delay(800);
    Serial.println(sensorValue);

    val = MYLIB.add1(val);
    Serial.println("MYLIB.add1(val) "); //100
    Serial.print(val,DEC);
    Serial.println("\ncomplementa(val) ");
    val = complementa(val); //155
    Serial.print(val,DEC);
    while(1);
}

```

Usando mylib

```

keywords.txt
#####
# Syntax Coloring Map For Ultrasound
#####
MYLIB      KEYWORD1add1 KEYWORD2

```

Big main loop

Programa que presionen un sw y un led cambie (toggle)
Presionar - prender
Vuelves a presionar – apagar ...

```
#include <LED.h>
const int buttonPin = 2;
int buttonState = 0;
LED led = LED(13);
```

```
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin (9600);
}
```

```
void loop(){
  if (digitalRead(buttonPin) == LOW)
  {
    while(digitalRead(buttonPin) == LOW);//no hago nada!! solo esperar
    led.toggle();
    Serial.println("hey se presiono-solto el boton!");
  }
}
```

<http://www.arduino.cc/playground/Code/LED>

Maquina de estado

http://www.i-micro.com/pdf/articulos/Maquina_edos.pdf

Las máquinas de estado son una parte integral de la programación de software. Las máquinas de estado hacen al código más eficiente, más fácil de depurar y ayudan a organizar el flujo del programa.

Cuando se implementa el concepto de la maquina de estados, se debe de elaborar una lluvia de ideas de todos los estados que se necesitan para una determinada aplicación. Una vez hecho esto se debe identificar el primer estado. Acto seguido debemos responder la siguiente pregunta
¿Que condición se necesita para salir de este estado y que estado es el siguiente?


```

#include <LED.h>
const int buttonPin = 2;
unsigned char estado = 1;
int buttonState = 0;
LED led = LED(13);
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin (9600);
}
void loop(){
  Serial.println("otra tarea!");
  ///esto ya puede ser encapsulado en otra funcion:
  switch (estado)
  {
  case 1:
    if (digitalRead(buttonPin) == LOW){
      led.toggle();
      Serial.println("hey se presiono el boton");//como filtro.
      estado = 2;
    }
  case 2:
    if (digitalRead(buttonPin) == LOW){;} // no hagas nada
    else estado = 1;
  }
  ////////////
}

```

Crear tarea2());

Esto esta
mmmmucho
mejor

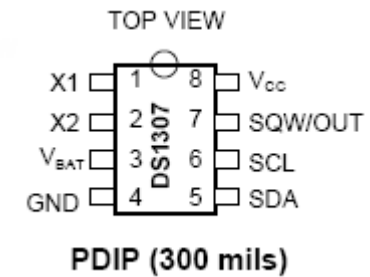
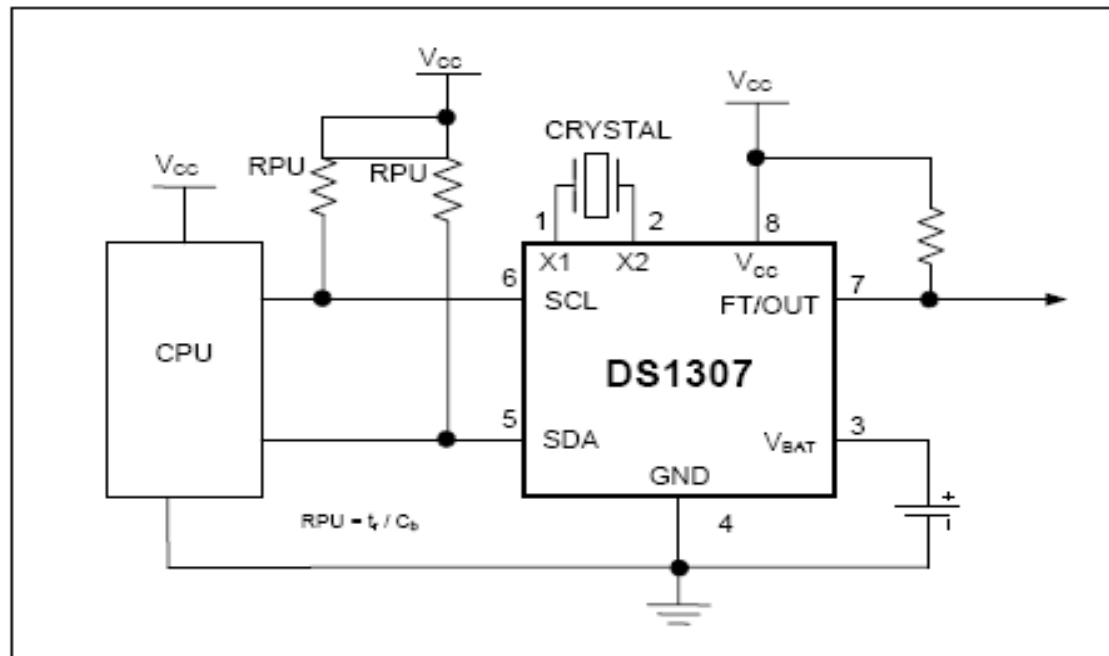
i2c

<http://www.ladyada.net/learn/breakoutplus/ds1307rtc.html>

<https://github.com/adafruit/RTClib>

En arduino
Sda pin A4
Scl pin A5

TYPICAL OPERATING CIRCUIT



5v scl sda nc gnd vista por abajo de mi pcb...



Oct.1.2011

Arduino Cabal