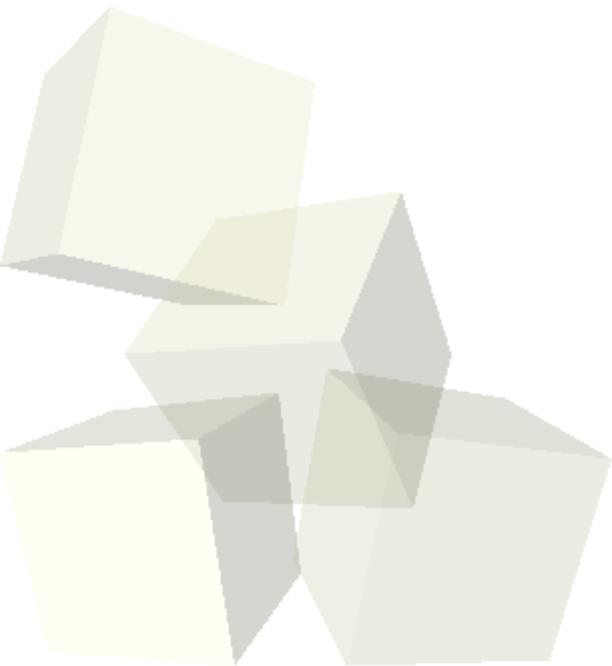




Prácticas de Laboratorio
MODULO: PLATAFORMAS DE DESARROLLO

Manuel J. Bellido Díaz

Septiembre de 2017



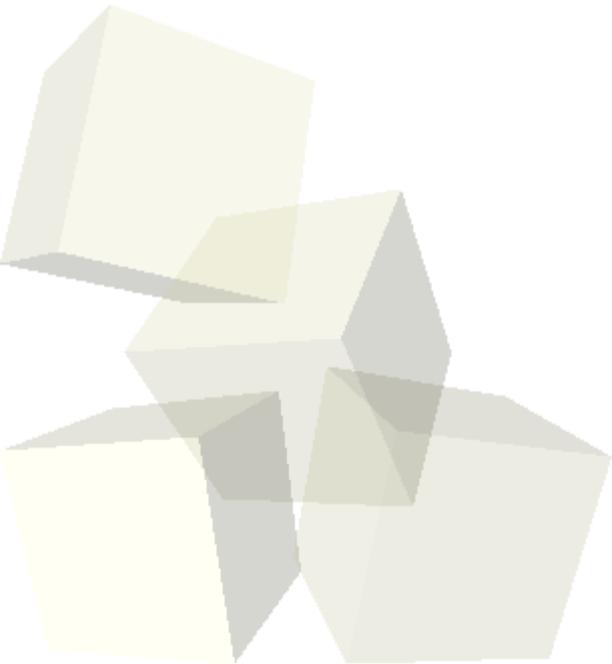


Cuaderno de Laboratorio

- Durante el desarrollo de todas las prácticas de laboratorio siempre es aconsejable tomar nota de todo lo que se esta haciendo, marcando los problemas encontrados, las soluciones a cada problemas y los resultados que se van concluyendo.
- Esta información deberá ser parte de la memoria que se entregue
- Los últimos 10 minutos de cada sesión deben emplearse para guardar el material correctamente.



- **Plataforma Arduino**
- Plataforma Raspberry PI
- Proyecto Mysensor.org





■ Objetivos mínimos del trabajo:

- Conocer la plataforma arduino, sus características, sus variantes, sus modos de programación.
- Conocer una serie de componentes básicos de hardware típicos de aplicaciones de sistemas empuotrados
- Preparar el PC para que funcione el entorno de desarrollo de arduino
- Realizar ejemplos básicos de funcionamiento sobre arduino
- Desarrollar otros ejemplos de uso de arduino manejando diversos componentes hardware





■ Introducción a Arduino

- ¿Que es arduino?
- Arduino es una plataforma de desarrollo de hardware abierta (open Hardware) basada en software y hardware flexibles y fáciles de usar. Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros tipo de actuadores.
- Basado inicialmente en microcontroladores de 8 bits AVR (Atmega 8, 128,328,1280) aunque con alguna versión basada en ARM de 32 bits (Arduino Due)
- Ventajas principales de arduino:
 - Un entorno de desarrollo muy facil de manejar (basado en processing, <https://processing.org/>),
 - Un amplio conjunto de librerías de manejo de periféricos
 - Una comunidad de desarrolladores muy grande.

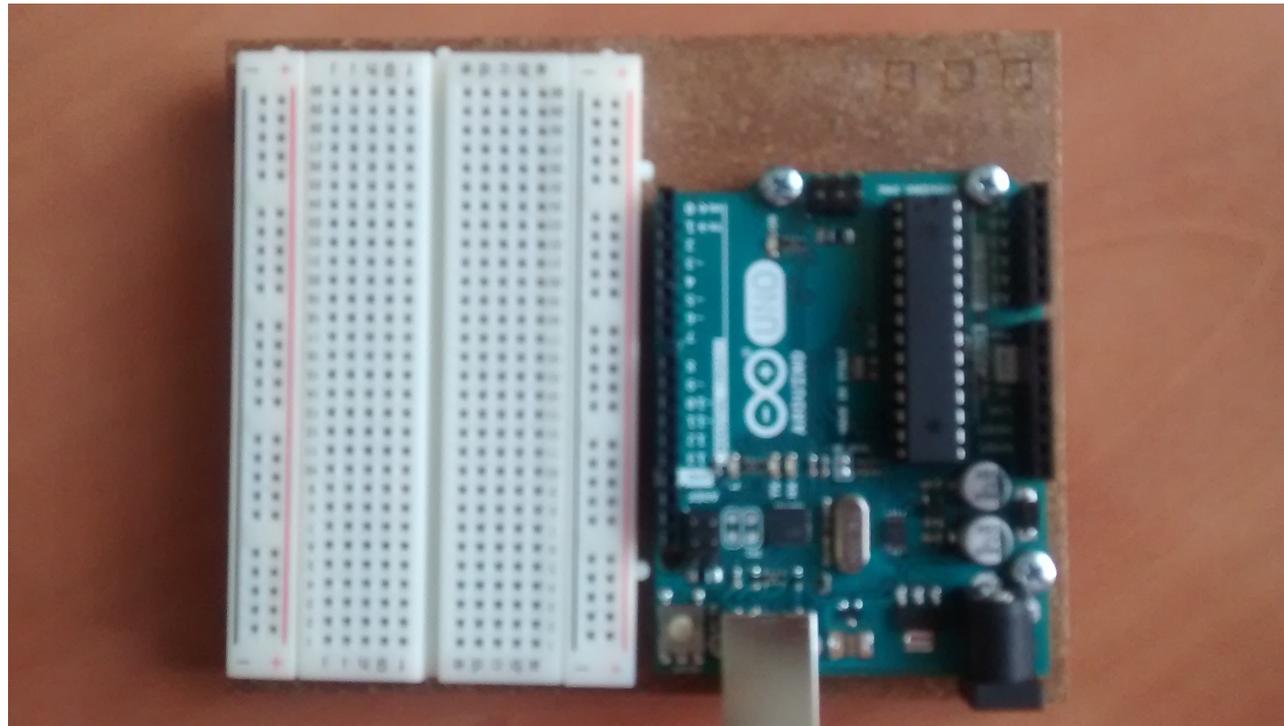


- En 2015 hubo una división en el equipo de arduino:
 - ♦ Web inicial: www.arduino.cc ----> Cuatro de los cinco fundadores
 - ♦ Nueva web: www.arduino.org ----> Un fundador
 - ♦ Al tener registrada la marca Arduino en Italia el fundador que se separó, el antiguo equipo ha creado una marca para uso fuera de EEUU de nombre GENUINO
 - ♦ En 2017 han vuelto a unirse: www.arduino.cc
- arduino.cc: What is arduino?
 - ♦ Discover the official arduino boards
 - Repasar plataformas, UNO, ZERO, mkr1000
 - Se dispone de los diseños de PCBs
 - Definir SHIELD
 - Learning: Getting Started --> enviroment : writing sketches



■ Plataforma Arduino empleada

- Arduino UNO + Protoboard (regleta)

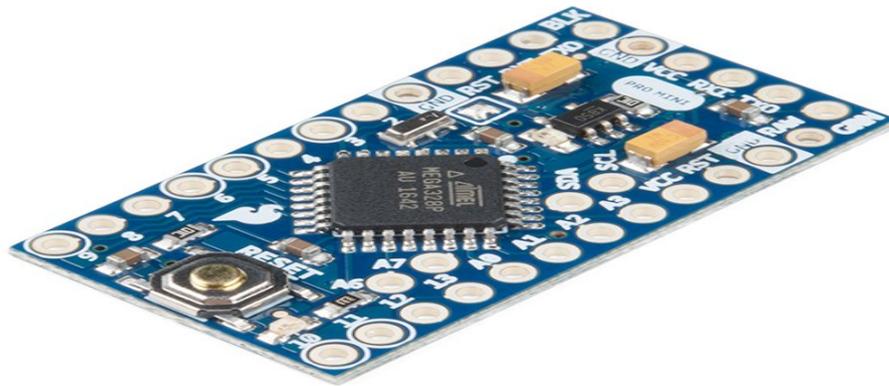


- ATMEGA 328P, 32K Flash memory,
- 14 GPIO (6 pueden ser PWM), 6 analog input
- 16Mhz clock
- <https://www.arduino.cc/en/Main/ArduinoBoardUno>



■ Plataforma Arduino empleada

- Arduino Pro Mini: <https://www.arduino.cc/en/Guide/ArduinoProMini>
- Diseñado por Sparkfun para implementaciones de sistemas autonomos: problemas: alimentación, programación..



- ATMEGA 328P, 32K Flash memory,
- 14 GPIO (6 pueden ser PWM), 6 analog input
- Dos versiones: 5v, 16Mhz clock; **3.3V, 8MHz (la nuestra)**
- Apto para incluirse en una PCB
- Precio: aliexpres: 1-2€; Amazon 2-10€



■ Otros componentes hardware:

- Shield o placa de expansión:

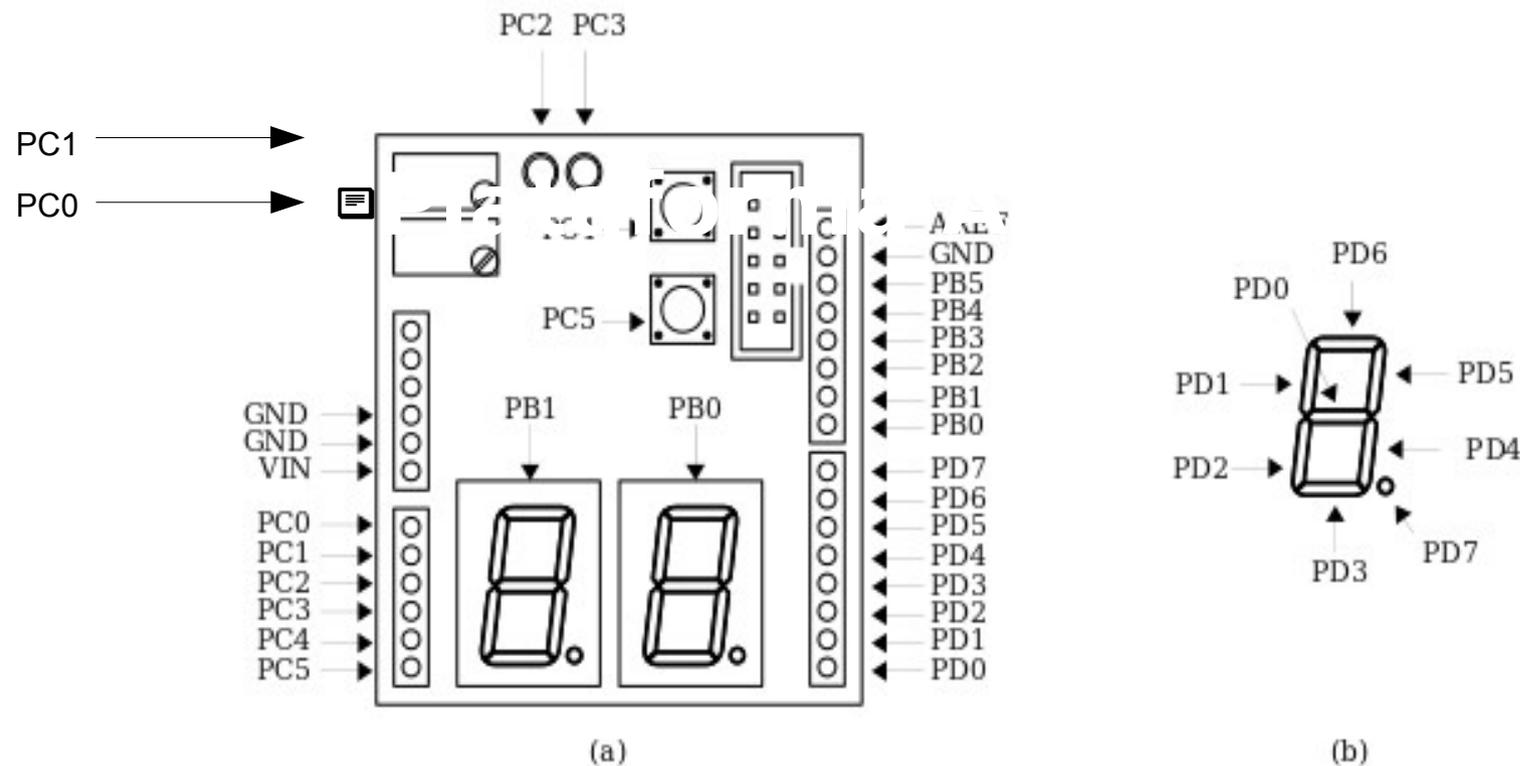


Figura 8. a) Placa de expansión E/S para Arduino. b) Detalle de conexión de los segmentos a los puertos.



■ Otros componentes hardware:

- ◆ Componentes habituales:

- Resistencias, condensadores, leds, diodos, pulsadores, interruptores (switches), display 7-segmentos, cables de interconexión, tiras de pines

- ◆ Otros componentes habituales en sistemas hardware:

- **Potenciómetro:** resistencia variable; componente con tres terminales. En modo normal los extremos se conecta a la alimentación (VDD, GND) de manera que, con la resistencia variable, el tercer terminal puede cambiarse de forma mecánica entre Vdd y GND



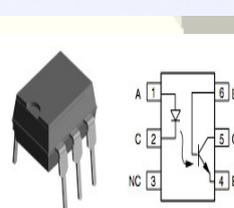
- **Fotoresistencia (célula fotovoltaica):** Componente biterminal que varía su resistencia con la luz.



- **Sensor de inclinación simple:** cuatro terminales; una bola de metal interconecta dos de los terminales en función de la inclinación del componente



- **Optoacoplador:** permite activar/desactivar un switch de un circuito desde otro circuito separado (fuentes de energía distintas)





■ Otros componentes hardware:

- ◆ Otros componentes habituales en sistemas hardware:



→ **Sensor de temperatura (LM36GZ)**: Tres terminales; Dos, de alimentación y el tercero de señal. Varía la tensión en función de la temperatura.

→ **Transistores**: Componentes de tres terminales. Una aplicación muy típica en sistemas hardware es emplearlos como interruptores, activando/desactivando a través del terminal de gate. Existen transistores de potencia para actuar de interruptores en sistemas de mas potencia controlados por señales digitales

→ **Puente H**: un circuito que permite intercambiar la polaridad en las señales de salida con señales de control de entrada.

→ **Relé**: Funciona como interruptor controlado por señal eléctrica (generalmente el control es de una señal electrónica de pocos voltios, mientras la salida puede ser señal de electricidad





■ Otros componentes hardware:

- ♦ **Motores** (desde el punto de vista del microcontrolador):



- **Motor de corriente continua continua:** Una tensión de continua activa el giro del motor. Gira en dos sentidos según la polaridad de la corriente. Un microcontrolador puede cambiar el giro de rotación, para ello se emplea hardware añadido (típicamente un **punte H**)



- **Servo motor:** básicamente un motor DC con circuitería de control ya incluida. La función del servomotor es girar hasta colocarse en un ángulo determinado y mantenerse en esa posición mientras se desee. Suelen tener un ángulo de giro de 0 a 180grados. El ángulo de giro en el que se posicionan depende del tamaño del pulso que se envíe por la señal de control. Típicamente se usa una señal PWM para colocar el servo en un determinado ángulo de giro.



- **Motor paso a paso:** Son motores que giran un determinado ángulo por cada impulso eléctrico. El ángulo puede ser grande (90grados) o pequeño (1,8grados). Típicamente se controlan con puentes H.



■ Otros componentes hardware:

◆ NECESARIOS PARA ARDUINO PRO MINI:

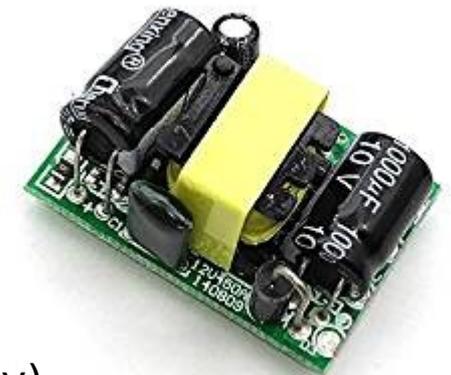
→ Fuente de alimentación DC 5v (AC220 a DC 5v)

- HLK-PM01 (salida 5v 0,5A); Aliexpress: 2-5€



→ Otras Fuentes 220v ac a 5v dc:

- <https://es.aliexpress.com/item/Precision-AC-85-265V-50-60Hz-DC-5V-700mA-3-5W-Isolated-Switching-Power-Supply-AC/32829739090.html?spm=a219c.search0302.4.80.mg17j9>
- Precio: 2-3€



→ Cable Programación Arduino Pro Mini (cable USB-Serie 3.3v)

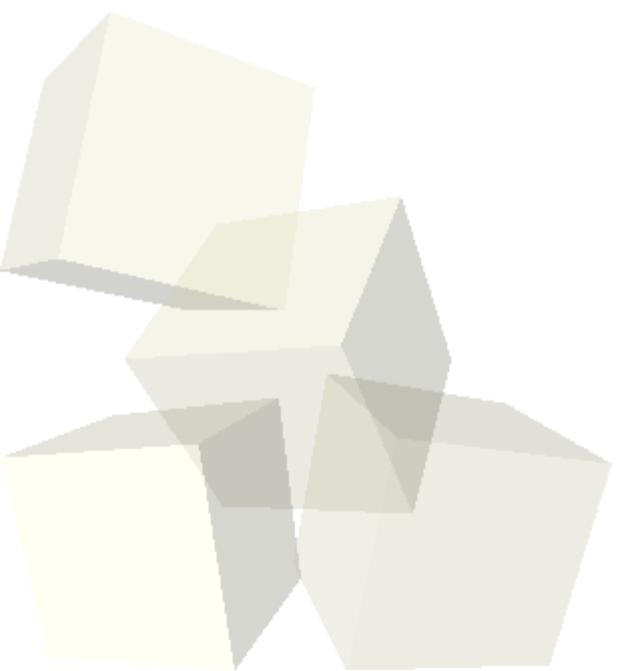
- Precio: 10€-20€





Plataforma Arduino

- Instalación del entorno de desarrollo de Arduino
 - ◆ <https://www.arduino.cc/en/Main/Software>





■ Desarrollo de ejemplos de programación sobre Arduino: sketches

- Información básica: <http://arduino.cc/en/Guide/Environment>
- Lenguaje de programación: similar al C, conjunto de funciones propias: <http://arduino.cc/en/Reference/HomePage>
- Estructura general de un programa:

```
void setup() {
```

```
    // put your setup code here, to run once:
```

```
}
```

```
void loop() {
```

```
    // put your main code here, to run repeatedly:
```

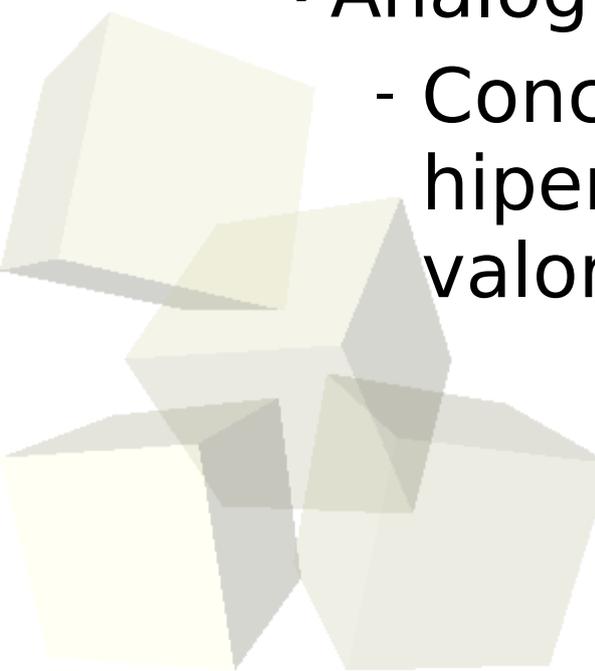
```
}
```

- Estructura propia de sistema empujado ejecutando aplicación en modo stand-alone



- Desarrollo de ejemplos de programación sobre Arduino: sketches
 - ♦ Compilación de ejemplos de la base de datos de arduino:
 - Blink:

Conceptos: estructura general del programa; escritura en pines digitales; retrasos
 - Fade:
 - Conceptos: escritura analógica;
 - AnalogInOutserial:
 - Conceptos: escritura en puerto serie, hiperterminal; lectura analógica; adecuación valor de lectura a valor de escritura.





■ ARDUINO:

- ♦ Ejecutar programas ejemplos de funcionamiento de arduino:
 - Blink; fade
- ♦ Variaciones sobre blink:
 - Sobre blink: Controlar el encendido apagado de un led con un switch/pulsador conectado a VCC y GND
 - Controlar el encendido -apagado del LED desde el PC vía puerto serie
 - Información: buscar en internet
 - . Propuesta: programa en Python que envíe comandos de encendido y apagado a través del puerto serie



■ ARDUINO:

- ♦ Alternativas sobre control encendido-apagado led:
 - **Controlar encendido-apagado de una bombilla mediante un relé.**
 - Mucho cuidado con los calambres.
 - Conectar primero la bombilla al relé antes y el rele a arduino antes de enchufar la bombilla a 220v
 - Probar el código antes de enchufar para comprobar que el rele conmuta
 - Encendido apagado de un led mediante pulsador
 - Empleando las interrupciones en arduino:
 - <https://www.arduino.cc/en/Reference/AttachInterrupt>
 - Utilice un pulsador en configuración de pull-up o pull-down para activar una interrupción en alguno de los pines de interrupciones
 - **¿Que problema detecta con el pulsador?**



■ ARDUINO:

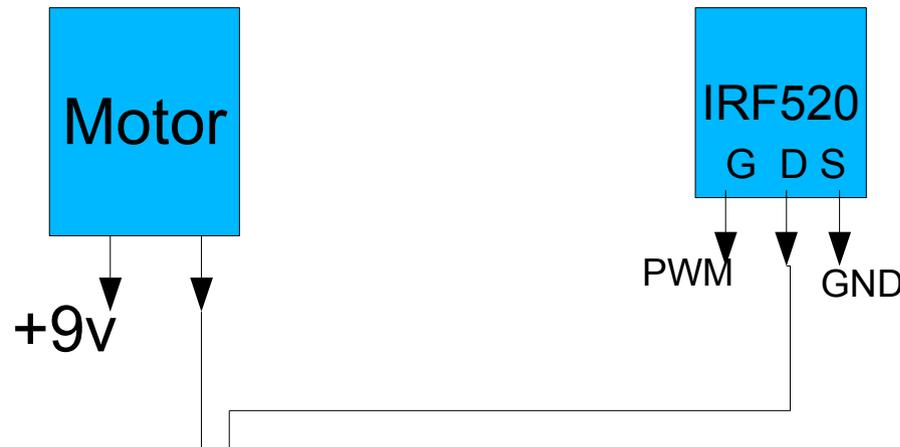
- Control de la posición de un servo motor con un potenciómetro.
 - Información: librería **servo** de arduino
- Variación del programa: Controlar el ángulo de posicionamiento del servo desde el PC. Enviando el valor de ángulo (desde 0 a 180 grados) a través del puerto serie.
 - Información: buscar en internet
 - Propuesta: Programa en Python en PC que envíe el valor del ángulo a través del puerto serie.
 - . Utilizar `parseInt()` de la librería serial



■ ARDUINO:

- ♦ Control de la velocidad de giro de un motor de continua con un potenciómetro.

→ Información: Giro en un sentido. Alimentación del motor con 9v. Driver transistor de potencia IRF520

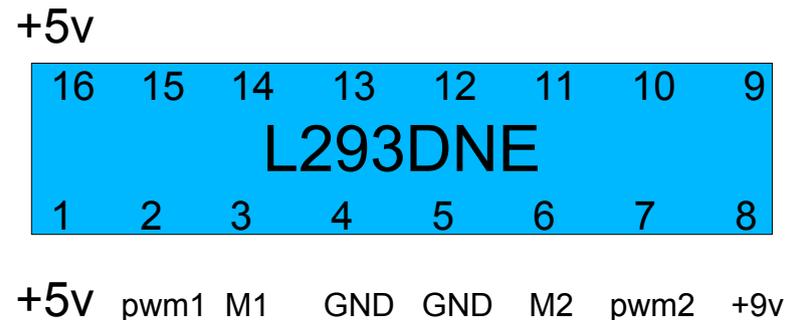
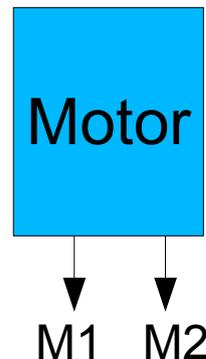


- ♦ Variación: Controlar la velocidad desde el PC vía puerto serie (enviando un valor entre 0-parado- y 255-máxima velocidad)



■ ARDUINO:

- ♦ Control de la velocidad de giro de un motor de continua en doble sentido con un potenciómetro. (mitad señal pot, giro pos, otra mitad señal pot., giro neg)
 - Información: Alimentación del motor con 9v. Driver circuito driver en puente H (aporta potencia y cambia polaridad)





■ ARDUINO:

- Impresión en una pantalla LCD.
- Pantalla LCM1602C , basada en controlador Hitachi HD44780 o compatible.
- Librería arduino: LiquidCrystal
- Desarrollar ejemplos.
- Variación 1: Imprimir en dos filas con desplazamiento a la izquierda:
 - Fila 1: nombre de la asignatura
 - Fila 2: nombre alumno
- Variación 2: imprimir información que envía el pc vía serie



■ ARDUINO:

- ◆ Empleando el sensor de temperatura tmp36GZ diseñar un termómetro con la temperatura en el display LCD
 - **Buscar información en internet**

- ◆ Contador 00 a 59 en display 7-seg cada segundo



■ ARDUINO:

◆ **Aplicación:** Contador 00 a 59 en display 7-seg cada segundo

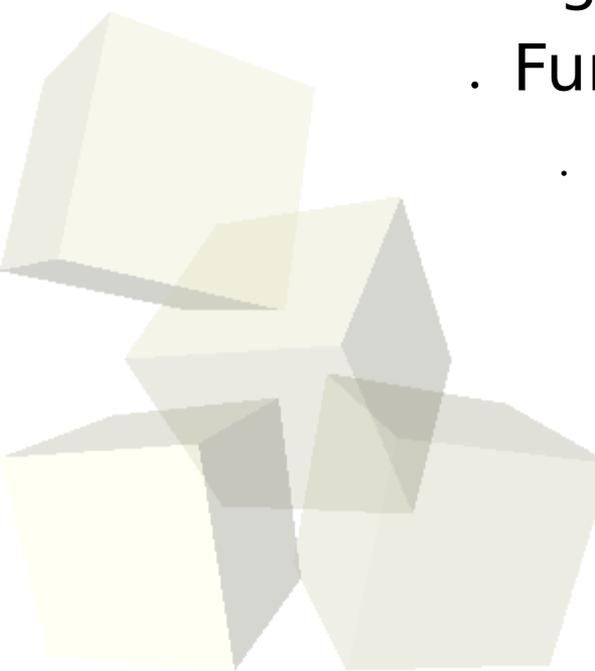
→ Una posible forma de hacerlo

- Estructura del programa:

- Dos enteros, dat1 y dat0 asociados a los displays 7-seg 1 y 0

- . Función principal (loop())

- . Bucle infinito que se ejecuta 1 vez por segundo. Calcula el valor correcto de dat1 y dat0 partiendo de dat1=0, dat0=0. Llama a refresh(dat1, dat0)





- ♦ **Función refresh** (int dat1, int dat0)

- Función que va “refrescando” (escribiendo en los displays) los displays 7-seg alternativamente cada Xms y que en total tiene que durar un tiempo total=1seg=1000 ms.

Para activar display1 o display0 se emplean los bits PB0 y PB1 de la placa de expansión (uno activo y el otro desactivo y así alternativamente cada Xms).

El refresco lo puede hacer con llamada a la función write_data(int dato)

- ♦ **Función write_data**(int dato)

- Función que dado un dato entero (dígito de 0 a 9) calcula el valor correcto de representación en los displays 7-seg y lo escribe en la función mediante write7seg(unsigned char arg)



- Función `write7seg`(unsigned char)

```
int segPins[] = {0, 1, 2, 3, 4, 5, 6, 7};
```

.....

```
void write7seg (unsigned char arg) {
```

```
    unsigned char segmen = 0x01;
```

```
    unsigned char display1;
```

```
    display1 = arg;
```

```
    for (int i = 0; i < 8; i++) {
```

```
        if ((display1 & segmen) == 0x00)
```

```
            digitalWrite(seg[i], LOW);
```

```
        else
```

```
            digitalWrite(seg[i], HIGH);
```

```
        segmen <<= 1;
```

```
    }
```

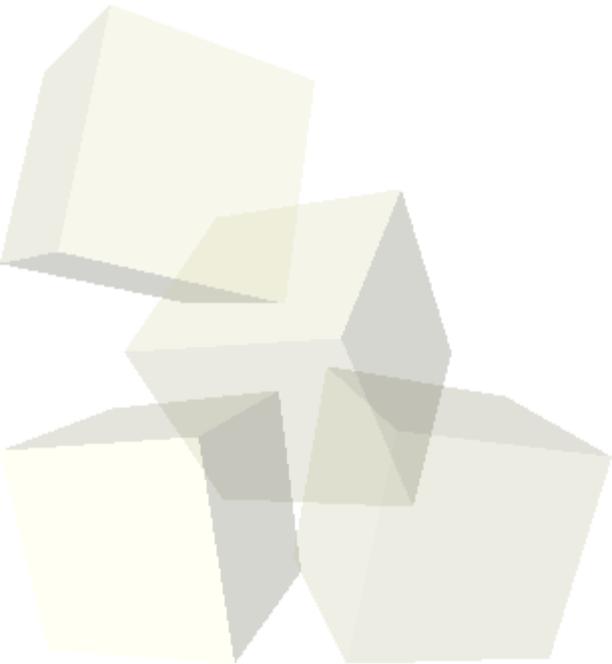


- ♦ **Función refresh** (int dat1, int dat0)
 - Función que va “refrescando” (escribiendo en los displays) los displays 7-seg alternativamente cada Xms y que en total tiene que durar un tiempo total=1seg. El refresco lo ,puede hacer con llamada a la función write_data(dato)
- ♦ **Función write_data**(int)
 - Función que dado un dato entero (dígito de 0 a 9) calcula el valor correcto de representación en los displays 7-seg y lo escribe en la función mediante write7seg(unsigned char)
- ♦ **Función write7seg**(unsigned char)





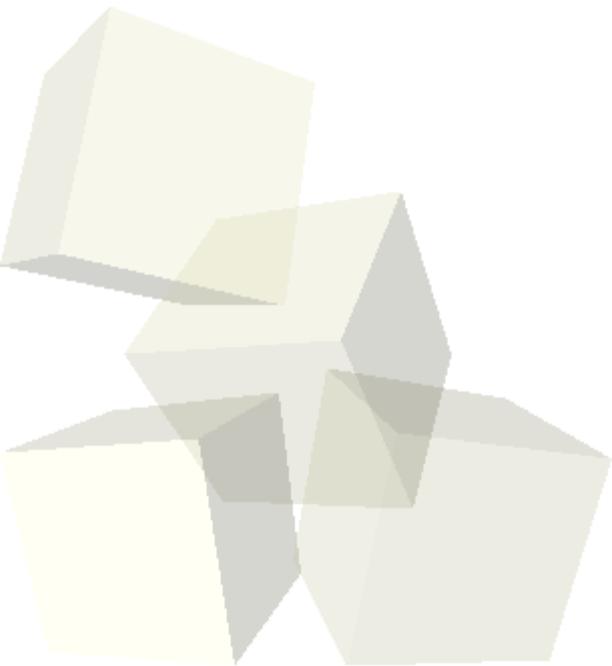
- Plataforma Arduino
- **Plataforma Raspberry PI**
- Proyecto Mysensor.org





■ Objetivos mínimos del trabajo:

- Preparar la plataforma Raspberry Pi para que puedan cargarse diferentes versiones de Sistema Operativo
- Arrancar y comprobar el funcionamiento de la placa Raspberry Pi
- Desarrollar ejemplos de utilización de los pines de expansión GPIOs





Plataforma Raspberry Pi

■ Información sobre Raspberry Pi:

- <http://www.raspberrypi.org/>

■ Instalación de sistemas operativos preparados:

- <http://www.raspberrypi.org/documentation/installation/installing-images/linux.md>
 - Vamos a instalar la imagen de **UBUNTU MATE**
- **Información de ubuntu Mate para Raspberry Pi:**
 - <https://ubuntu-mate.org/raspberry-pi/>
- Descargar la imagen de Coria:
 - <https://coria.dte.us.es/~bellido/> <https://10.1.15.78/~bellido>
 - Imagen a descargar:
 - [Ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz](#)

■ Una vez preparada la tarjeta:

- Conectar la SD a RaspberryPi (RPI)
- Desconectar Teclado, ratón y cable Ethernet del PC y conectar a RPI
- Conectar cable HDMI-DVI a RPI y Monitor.
- Conectar cable USB-Micro usb a RPI (microusb) y PC para alimentar la RPI



Plataforma Raspberry Pi

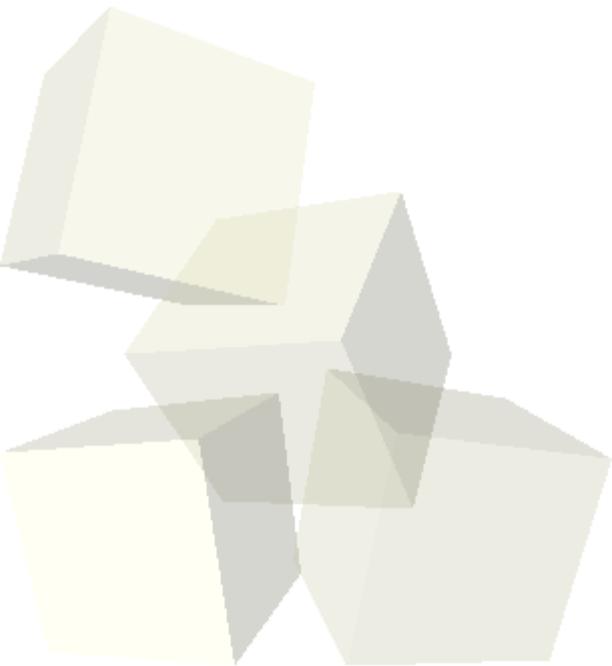
■ Configurar ubuntu-mate:

- ♦ Seguir los pasos en el arranque configurando idioma, teclado, localización, fecha y hora
- ♦ **IMPORTANTE:** guardar información sobre usuario y passwd elegidos.
- ♦ Configuraremos la red manualmente en un terminal o mediante network manager. Los datos son::
 - IP address **10.1.15.xx** (xx-numero del pc: RD-xx)
netmask **255.255.252.0**
gateway **10.1.15.78**
dns-nameservers **8.8.8.8**
 - Probar conexión: \$ ping www.google.es
 - Abrir firefox y conectarse a google.
 - Si falla, revisar la **fecha y hora** del sistema. Corregirla.
- ♦ A considerar: Ubuntu-Mate nombra al interfaz de red ethernet con “enx” + “digitos de la mac”. Esto significa que si introducimos la tarjeta SD en otra Raspberry Pi diferente nos fallará la conexión de red (habría que volver a configurarla)



Plataforma Raspberry Pi

- En el primer arranque de Ubuntu Mate es cuando se realiza una configuración del sistema y se completa la instalación del sistema de ficheros
- También se puede realizar el ajuste del tamaño del sistema de ficheros al tamaño de la imagen tal y como se cuenta en la página web de información de Ubuntu Mate para Raspberry Pi:
 - ◆ En la página de bienvenida (welcome) de ubuntu-mate





Plataforma Raspberry Pi

■ CONEXIÓN MEDIANTE SSH

- Para activar el servicio SSH en el arranque del sistema debe ejecutarse el siguiente comando:
 - `$ sudo systemctl enable ssh`
- Resetando el sistema debería funcionar SSH

■ Configurar Fecha y Hora para que en el arranque se inicialice correctamente

- En Ubuntu-Mate viene instalado por defecto el servicio de hora NTP (Network Time Protocol) que es el que, en el arranque, debería configurar correctamente la hora.
- Sin embargo, en esta versión de Ubuntu-Mate parece que algo falla con ntp
- Para configurar correctamente fecha y hora desinstalaremos ntp e instalaremos ntpdate:
 - `$ sudo apt-get remove ntp`
 - `$ sudo apt-get install ntpdate`
- Por último modificaremos el fichero `/etc/ntp.conf` añadiendo un servidor:
`server hora.rediris.es`



Plataforma Raspberry Pi

- Tutorial sobre GPIOs en RPI:
 - ♦ http://elinux.org/RPi_Low-level_peripherals
- Se pueden desarrollar algunos ejemplos de manejo de GPIOs en diferentes lenguajes de programación:
 - ♦ http://elinux.org/RPi_GPIO_Code_Samples
- **TAREA1:** Vamos a ver el manejo de GPIOs desde línea de comandos:

<https://geekytheory.com/tutorial-raspberry-pi-gpio-parte-1-control-de-un-led/>

- ♦ **AVISO: USAR UN LED CON RESISTENCIA**

- **TAREA2:** Vamos a ver ejemplos sencillos con PYTHON:

<https://geekytheory.com/tutorial-raspberry-pi-gpio-parte-2-control-de-leds-con-python/>

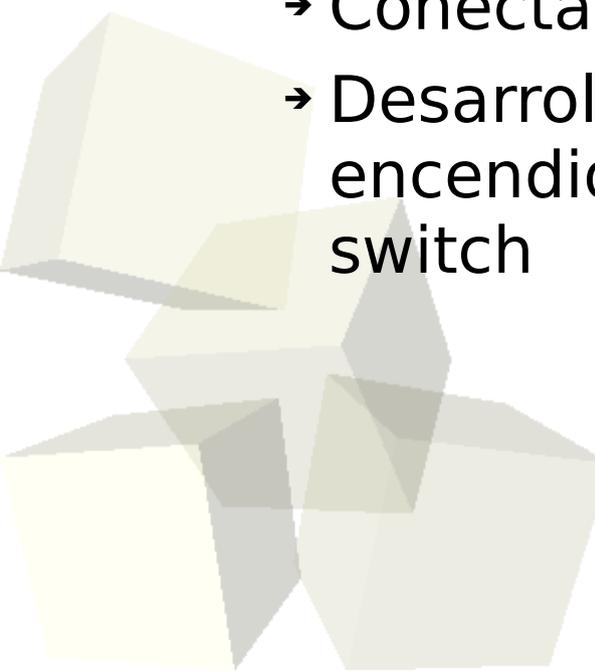
- ♦ **ATENCIÓN:**

→ Con respecto a este tutorial hay que tener en cuenta que en la versión de ubuntu-mate instalada ya viene preinstalada la librería Rpi.GPIO



Plataforma Raspberry Pi

- **TAREA2:** Vamos a ver ejemplos sencillos con PYTHON:
<https://geekytheory.com/tutorial-raspberry-pi-gpio-parte-2-control-de-leds-con-python/>
 - ◆ **ATENCIÓN:**
 - Con respecto a este tutorial hay que tener en cuenta que en la versión de ubuntu-mate instalada ya viene preinstalada la librería Rpi.GPIO
 - ◆ Modificación sobre el tutorial (**Tarea a controlar**):
 - Conectar dos switches a dos GPIOs de la Raspberry Pi
 - Desarrollar un programa en Python que controle el encendido/apagado de cada uno de los leds con un switch



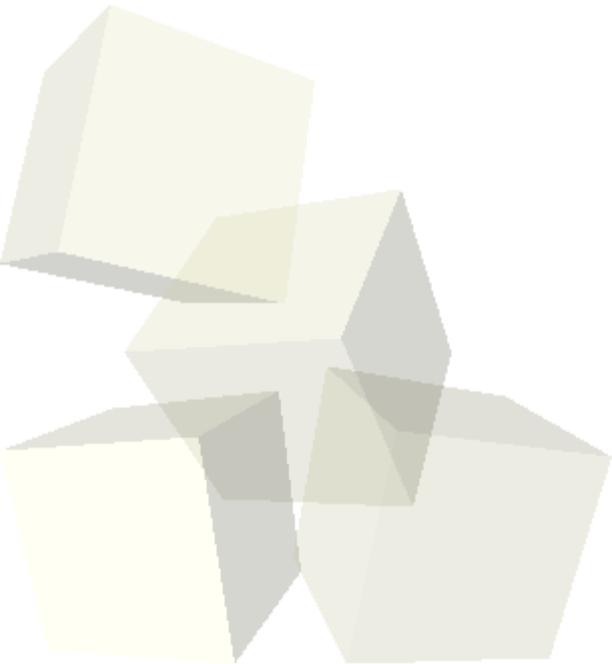


Plataforma Raspberry Pi

- Existen dos formas básicas de acceder a una consola de la Raspberry Pi sin tener teclado, ratón ni monitor:
 - ♦ 1.- Conexión a Raspberry Pi mediante cable serie-usb
 - ♦ 2.- Conexión por red , por ejemplo con SSH
- **TAREA3:** Vamos a realizar el segundo método
 - ♦ Apagar raspberry Pi, desconectar teclado, monitor, ratón
 - ♦ Conectar solo cable ethernet de red
 - ♦ Encender Raspberry Pi
 - ♦ Conexión por SSH a raspberryPI desde el PC:
 - ♦ `$ ssh -X <usuario>@<IPadressRasp>`
 - ♦ Ejemplo:
 - `$ ssh -X pi@10.1.15.XX`
- **El resto de tareas vamos a desarrollarlas en este modo de conexión vía SSH**



- Plataforma Arduino
- Plataforma Raspberry Pi
- ***Proyecto Mysensor.org***





■ <https://www.mysensors.org/>

- ♦ ***“MySensors is an open source hardware and software community focusing on do-it-yourself home automation and Internet of Things”***
- ♦ **“Do it yourself!”**
- ♦ **“We've combined the Arduino platform with a small radio transceiver into a fun, flexible world of low cost wireless sensors.”**
- ♦ **“1. Connect the parts. 2. Install the MySensors library, which includes ready-to-run examples. 3. Start measuring and controlling the world!”**
- ♦ **Open Source Hardware asociado a mysensor:**
 - **<https://www.openhardware.io/>**



- **Objetivo: controlar las “cosas”.....¿Pero como?**
 - ♦ **¿Como recoger información y donde se guarda o procesa?**
 - ♦ **¿que acciones debo hacer en función de la información?**
 - ♦ **Elementos involucrados:**
 - **Recoger información: Sensores con información digital y capacidad de transmitir la información**
 - **Controlador del sistema. Para procesar adecuadamente la información de los sensores y realizar la toma de decisiones**
 - **Actuadores que ejecuten las decisiones del controlador. Necesitan capacidad de recibir información**
- **Componentes básicos del sistema:**
 - ♦ **Para los sensores y/o actuadores:**
 - **Sensor/actuador + arduino + Modulo comunicación (ej NRF24I01)**
 - ♦ **Para el controlador:**
 - **Gateway (módulo comunicación) + host con software del controlador**



■ Controlador

- Existe un lista de mas de 20 controladores compatibles con mysensor.
- Existe una comparación de alguno de estos controladores:
 - <https://www.mysensors.org/controller>
- En estas prácticas vamos a trabajar con [mycontroller.org](http://www.mycontroller.org)
 - <http://www.mycontroller.org/#/home>
- Mycontroller.org cubre prácticamente toda la red de mysensor
- Se puede instalar en RASPBERRY PI





■ Instalando mycontroller.org:

- ◆ <https://github.com/mycontroller-org/mycontroller>
- ◆ 1.- Descargar mycontroller
 - <http://www.mycontroller.org/#/downloads>
- ◆ 2.- Extraer los ficheros:
 - Carpeta **mycontroller/***
- ◆ 3.- Cambiar aspectos de configuracion:
 - Fichero **mycontroller/conf/mycontroller.properties**
 - En nuestro caso vamos a usar protocolo http en vez de https
- ◆ 4.- Ejecutar **start.sh** o **stop.sh**:
 - En carpeta **mycontroller/bin/**
- ◆ 5.- Comprobar funcionamiento conectándose a:
 - <http://localhost:8443>



- **Instalando el gateway en la Raspberry Pi**
 - Seguiremos las instrucciones de:
 - <https://www.mysensors.org/build/raspberry>
 - Utilizaremos la configuración de ethernet en el puerto 5003
 - Una vez compilado y ejecutado podremos configurarlo en mycontroller
 - **ATENCIÓN:** Para que no tengamos conflicto entre los nodos/gateway de cada alumno, vamos a montar cada red de mysensor en un canal diferente.
 - El módulo de radio frecuencia nrf24 tiene 126 canales, desde el 0 hasta el 125.
 - Para seleccionar el canal en configure usar opción: **--my-rf24-channel=XX**
 - **XX** - numero del pc en el que estamos sentado



- **Configurando un nodo con un sensor**
 - ◆ **Configurando el entorno de arduino:**
 - **Debemos instalar la librería mysensor. En el siguiente enlace se cuenta como se hace:**
 - **<https://www.mysensors.org/about/arduino>**
 - ◆ **Conectado la radio con arduino:**
 - **Usaremos el módulo de radio frecuencia NRF24L01. La conexión a arduino se indica en:**
 - **https://www.mysensors.org/build/connect_radio**
 - **Arduino Uno se conecta de igual forma que arduino Pro Mini o Nano**





- **Configurando un nodo con un sensor**
 - ♦ **ATENCIÓN:** Para que no tengamos conflicto entre los nodos/gateway de cada alumno, vamos a montar cada red de mysensor en un canal diferente.
 - ♦ El módulo de radio frecuencia nrf24 tiene 126 canales, desde el 0 hasta el 125.
 - ♦ Para seleccionar el canal cada sketch de arduino debemos incluir un define:

```
#define MY_RF24_CHANNEL XX
```

- ♦ **XX** - numero del pc en el que estamos sentado



■ Configurando un nodo con un sensor II

- El controlador identificará cada uno de los nodos con un ID.
- Por defecto, en MyConfig.h esta definido el ID de forma automática (es decir, sera el controlador quien asigne ID)
- **ATENCIÓN:** En nuestro caso, vamos a asignar ID para poder detectar cual es nuestro nodo y distinguirlo del resto de nodos de los restantes alumnos:
 - Incluir en sketch:

```
#define MY_NODE_ID <NUM>
```

- **IMPORTANTE:**

→ **CADA VEZ QUE CONFIGUREMOS UN NODO DISTINTO DEBEREMOS USAR UN ID DIFERENTE**



■ Configurando un nodo con un sensor II

- ◆ Ejemplo 1: Interruptor (o pulsador)
 - Existe un ejemplo preparado en mysensor:
 - **Door/Window/Buttom**
 - Emplear los interruptores que vienen con arduino UNO.
 - Comprobar que mycontroller detecta el nodo y el sensor y que detecta los cambios en el switch
- ◆ **Modificación 1:**
 - Modificar el hardware y el código para usar dos interruptores
 - Comprobar los cambios en los dos interruptores en mycontroller
- ◆ **Modificación 2:**
 - Incluir un rele y activarlo cada vez que cambie un interruptor (**carga hacia delante -forward payload-**)



■ Configurando un nodo con un sensor III

- ♦ Configuraremos ahora un nodo con
 - **arduino pro mini 3.3v 8mhz.**
- ♦ La idea es dejar en arduino UNO los dos interruptores y montar el rele en arduino pro mini.
- ♦ Configurar arduino pro mini (3.3v 8mhz).
 - Emplearemos la placa de alimentación con transformador de 220 AC a 5v DC
 - La alimentación se hace a través de la entrada RAW
 - Para programar arduino pro mini, emplearemos el cable TTL RS232 RPI
 - La alimentación del relé debe ser 5v
- ♦ Opcional: Una vez funcionando se puede conectar la bombilla



■ OPCIONAL: Configurando un nodo con un sensor IV

- ◆ Configuraremos ahora un nodo con arduino pro mini y un sensor DHT11 (temperatura y humedad).
- ◆ La idea es montar un sensor **“portatil”**
- ◆ Configurar arduino pro mini.
 - Emplearemos la batería LIPO (3.7 a 4.2 v) conectándola a RAW
 - Para programar arduino pro mini, emplearemos el cable TTL RS232 RPI
- ◆ **Modificación Importante:** Ser capaz de medir el nivel de la batería.
- ◆ Información sobre baterías LIPO:
 - <http://www.robotshop.com/media/files/pdf/hyperion-g5-50c-2s-1300mah-lipo-battery-user-guide.pdf>
- ◆ Sobre nivel de batería en Mysensor.org:
 - <https://www.mysensors.org/build/battery>



■ OPCIONAL: Configurando un nodo con un sensor IV

- ◆ Nivel de batería en Mysensor:
 - El código que aparece en la página de Mysensor esta pensado para una batería formada por dos pilas AA (pilas de 1,5v ==> 3v)
 - Debemos adaptarlo para la batería LIPO:
 - Celda LIPO 100% de carga ==> 4,2V
 - Celda LIPO 0% de carga ==> 3.0V
 - Usaremos como resistencias:
 - . R1= 1Mohm R2= 270Kohm





■ Diseñar Panel en mycontroller

- ♦ Hay que diseñar un panel que incluya botones para los sensores:
 - Interruptor 1
 - Interruptor 2
 - Rele
- ♦ Si se ha implementado el sensor DHT11 incluir en el panel:
 - Gráfica de la temperatura durante el último día
 - Gráfica de la Humedad durante el último día

