ARDUINO

- Documentación
- Prácticas

Qué sueño

R Memoria

- Proyectos

A.- Documentación:

La documentación sobre Arduino es muy extensa, para empezar:

Aprende:nj gt

- Introducción a Arduino. Presentación en Prezi
- Lenguaje de Programación. Web de Arduino
- Resumen del lenguaje la programación. PDF Resumen

1. Presentaciones:
    2. P0: Introducción.
    3. P1: Entradas y Salidas Digitales.t
    4. P2: Entradas y Salidas Analógicas.
    - P3: Motores y Potencia.
- Documental sobre Arduino
- Videoturoriales en youtube (inglés) y en español
- Más Info sobre Arduino, webs, proveedores...
- Arduino vs Scratch

Practica:

- Ejercicios de Arduino para principiantes. Arduteka
- Instructables: página con proyectos de todo tipo
- Ejercicios resueltos de Arduino para Secundaria en la Web Arduino

B.- Prácticas de Arduino:

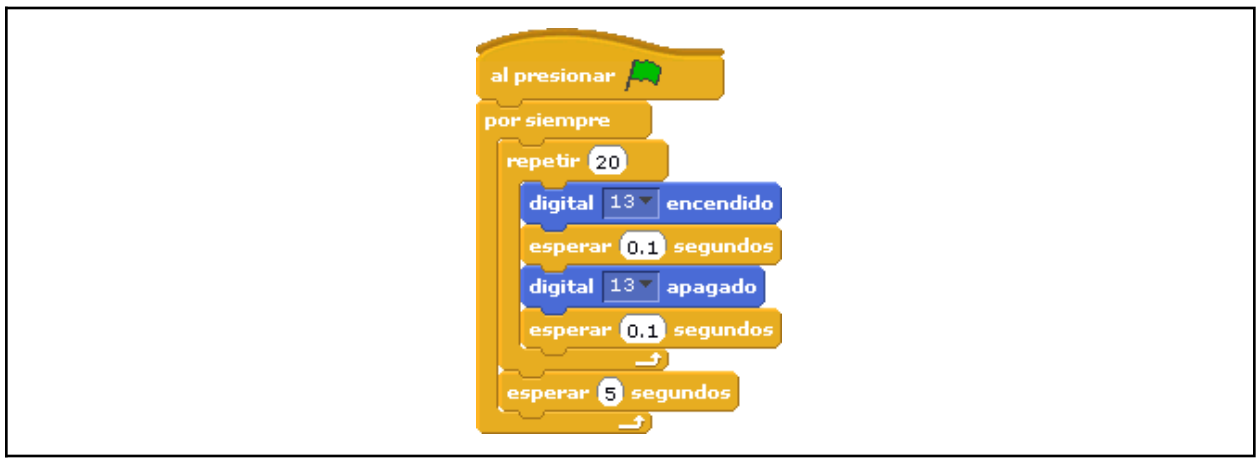
Realiza las siguientes prácticas de programación y guarda en un archivo de texto los programas realizados y los esquemas de programación en S4A.
El objetivo de estas prácticas es que nos familiaricemos con el lenguaje de programación de Arduino, para ello al principio se ha intentado variar de hardware lo menos posible de forma que nos podamos concentrar en el Software. Se aconseja realizar primero la programación en S4A, tal como viene en los dos primeros ejemplos y luego traducirlo al lenguaje de programación de Arduino.

1.- Prácticas LED y Puerto Serie:

1.1- Cargar el programa *Blink* y probar a cambiar la velocidad con la que titila a 0,5 s. Añádele variables para el pin del led y para el tiempo. Es el Hola Mundo de Arduino, ver la entrada de Arduteka para el hardware.

Programación del sistema en S4A

1.2- Hacer que el led titile 20 veces con una frecuencia de 0,1s y luego pare durante 5s, luego vuelve a repetir el ciclo. Usar la estructura *for*



Programación del sistema en S4A

1.3- Hacer que el led se encienda 2s y se apague 0,5s, además debe informar por el *puerto serie* cuando está encendido y cuando está apagado. Hay que usar Comunicación Serial

1.4.- Hacer que el led se encienda y se apague 20 veces con periodo de 0,1s, y luego permanezca apagado. Hay varias formas de hacerlo, una de ellas es utilizar la condición if y cambiar una variable de estado. Para repetir 20 veces usar for

1.5- Realiza un programa que cuente cada segundo que pasa y lo muestre por el puerto serie, con un mensaje tipo "Han pasado x segundos". *Concepto de contador*
Opcional para x=1 que escriba en singular "Ha pasado 1 s". Usar *if -else*

1.6- Controla la intensidad luminosa del LED usando *analogWrite* y un bucle *for*. La subida y la bajada deben durar 5 s cada una. Programa Fading. Tenemos que usar uno de los pines Digitales que tienen PWM.

1.7- Cargar el programa *millis* (Examples->Digital->BlinkwithoutDelay) y "españolizarlo" tanto en las variables como en los comentarios. Modificarlo para que el periodo sea de 0,5s. Este programa hace lo mismo que el programa Blink inicial, pero sin utilizar Delay. Nos será útil en el futuro, por ahora basta con comprender cómo funciona. Práctica opcional

2.- Prácticas LED y Pulsador

Para conectar el pulsador seguir los esquemas

2.1- Controla el encendido de un led con un pulsador, el led se enciende *cuando estamos* accionando el pulsador (*if* y *digitalRead*)

2.2- Si pulsamos, el led parpadea 20 veces con intervalos de 0,1s. Usar for

2.3- Cuando pulsamos mandamos un mensaje que dice "Hola" por el puerto serie, (un es singular). *Solucionar el problema de rebote*, una forma sencilla de hacerlo es añadiendo un pequeño delay

2.4- Realiza un contador con un pulsador. Salida por pantalla serie
Opcional: Añádele al programa anterior que un led parpadee cuando pulsamos

2.5- Añade al sistema anterior un segundo botón que nos resetee el contador.
Opcional: con un solo pulsador haz que el contador se resetee si pulso durante 3s. Ver artículo

2.6- Controlar el encendido del led con el pulsador como si fuera un interruptor. Tenemos que

introducir el concepto de *Memoria de estado, es decir una variable que conozca el estado del led. Debemos solucionar también el problema de rebote. Hacer diagrama de flujo con S4A. Todo esto lo soluciona el Programa Debounce*

3.- Sensores Analógicos

Ver documentación

3.1- Realiza un sistema de **control** el encendido de un **led** por la **luz ambiental** (medida con una LDR)
Primero necesitamos ver que valores toma el sensor para fijar el umbral, imprimir su valor por el puerto serie. Para conectar la LDR seguir el esquema

3.2.- **Realizar un medidor de luz:** con 3 leds realizar un medidor de la cantidad de luz, de forma que a oscuras no se encienda ningún led, con poca luz 1 led, intermedia 2 leds, y a tope de luz se enciendan los 3 leds. Opcional: a tope de luz los leds deben parpadear

3.3.- Controlar la intensidad luminosa de un **LED** en función de la luz ambiental (**LDR**) de forma que cuando no haya luz el LED funcione a tope de intensidad y se vaya apagando gradualmente conforme la luz vaya aumentando. Usar función *map.* Práctica opcional

3.4.-Control de intensidad luminosa de un **LED** con **potenciómetro**. Usar función *map* para convertir los valores Analógicos entrada 0-1023, en valores de 0-255 para PWM

3.5.- Controla el sonido de un **altavoz** variando el tiempo de encendido y apagado mediante un **potenciómetro**. Usa un transistor BC547 (ver datasheet) para la salida del altavoz.

3.6.- Realiza una **alarma** sonora con un **sensor de distancia** por infrarrojos como detector de presencia. Hardware: usar un Zumbador y un sensor de distancia
Sensor de distancia Sharp G2Y Data Sheet

3.7.- Alarma de temperatura, si t<40º no pasa nada, si 40º<t<50º el led se enciende, si t>50º el led se vuelve intermitente. Añadir un botón de apagado de la alarma. Hardware sensor temperatura divisor tensión pull down

3.8.- Piezoeléctrico: utilizando un piezoeléctrico realiza una aplicación. Por ejemplo que se encienda un LED. Hardware piezoeléctrico

3.9.- Piano aéreo: usando el sensor de distancia realiza un piano aéreo. Ver tone. Se explica también en “Entradas y Salidas Analógicas”. Práctica opcional

4.- Motores

Ver Motores y Potencia.

4.1- Controlar el giro de un servomotor de posición con un potenciómetro. Opcional: usar un potenciómetro deslizante

4.2.- Controlar el giro de un servo de rotación continua con dos pulsadores

4.3.- Controlar la velocidad de un motor de corriente continua usando un transistor

4.4.- Controlar la velocidad y el sentido de giro de un motor de continua con un LM293D

C.- MEMORIA DE LAS PRÁCTICAS:

Realizar al menos 1 memoria de cada apartado

1. **Finalidad**: Finalidad del sistema. Esquema de entradas y salidas. Diagrama de estados o cronograma (si el funcionamiento es cíclico).
2. **Hardware**: El circuito electrónico: lista de materiales, protoboard y esquema electrónico. Se puede usar el programa Fritzing, o usar imágenes de internet.
3. **Software**: Diagrama de flujo o diagrama de bloques mediante S4A. El programa con comentarios. Cada línea debe de llevar su comentario realizado por nosotros.
4. **Comandos**: Comandos nuevos usados, para qué sirven. Explicación detallada de cada comando o función nueva usado. Variables, constantes... en fin comentar todo lo que sea nuevo (lo que se haya explicado en la memoria anterior no hay que repetirlo)
5. **Funcionamiento**: Vídeo de como funciona, también podemos incorporar fotografías, capturas de pantalla del terminal serie...

Al final de las memorias añadir **Valoración personal** y **Aportación Individual**
Recordad que la presentación es importante

Tiempo: 5 semanas
Realización: Realizar una Memoria por semana
Fecha de entrega:
Formato de entrega: Memoria PDF + 1 archivo PDF con todas las prácticas realizadas (sin cuidar el formato)

D.- PROYECTOS

Con Arduino se puede hacer casi cualquier cosa, buscar un proyecto que os interese y al que esteis dispuesto a dedicarle tiempo. Busca información y comprueba que podemos hacerlo.

Proyectos propuestos 4º ESO:

- Instrumentos musicales: Theremin digital. Arduino y Pure Data
- Cubo de leds
- Seguilíneas
- Vehículo inteligente
- Coche controlado mediante app y bluetooth
- Máquina reflejos Fernando Alonso
- Semáforo activado por sonido. VU meter
- Coche teledirigido. 2 arduinos conectados radiofrecuencia, el mando se puede realizar con un acelerómetro o como querais…
- Photoduino. Está todo el kit comprado
- Máquina Arcade
- Juego de equilibrios. Con acelerómetro
- Mandos para videojuegos: diseñar un mando para un juego específico
- Luces de Navidad con detector de presencia y sensor de temperatura corazón (modificando el cubo de leds)
- Girasol
- …. lo que se te ocurra

Fases para realizar el proyecto:

Finalidad → Búsqueda de información → Descomponer el proyecto → Prueba de las diferentes sistemas → Desarrollo del proyecto

Memoria del Proyecto:

1. **Finalidad**.

- Objetivo y descripción completa del funcionamiento
- Objetivos cercanos y posibles ampliaciones
- Descomposición del problema: usar diagramas
- Esquema de Entradas y Salidas

2. **Búsqueda de información**

Información necesaria para desarrollar el proyecto estructurada por sistemas

3. **Funcionamiento subsistemas**

Con la protoboard realizar un prototipo de los subsistemas comprobando su funcionamiento de forma aislada

De cada sistema:

- **Hardware**. Esquemas de conexión con Fritzing.
- **Programación**: Diagramas de flujo. Archivos de programación comentados.

4. **Desarrollo del proyecto**

Se trata de unir las diferentes sistemas que hemos probado aisladamente. En el **hardware** solo tenemos que conectar cada parte de forma independiente. Para montar el **software** tenemos que realizar primero un diagrama de flujo con las funciones que hemos programado antes.

a. **Hardware**. Entradas y salidas. Listado de materiales en una tabla (nº/ denominación/ cantidad) Esquemas de conexión con Fritzing.
b. **Programación**: Diagrama de flujo general. Archivo de programación comentado.

Recordad el método para resolver problemas en programación.