

└ Agenda

1. Recordar que el curso es de iniciación
2. Va a ser un curso intenso, descanso cortito
3. Comentar duración total del curso



1. BricoLabs: Asociación, Domus, Difusión del hardware y software libre, Tecnófilos, Divulgación, wiki,
2. OSHWDem evento barcamp

- @ctemes
- @pepdiz
- @Mto_1008
- @salvari

1. Para cualquier duda podéis contactar con nosotros en nuestros twitters

1. No hay mucho más que añadir

- ¿Quién ha programado antes?
- ¿Quién conoce el Arduino?

1. No hay mucho más que añadir

- Presentación

- ¿Quienes somos?

- Asistentes

- ¿Quién ha programado antes?
- ¿Quién conoce el Arduino?
- ¿Quién conoce Processing?

1. No hay mucho más que añadir

- Presentación

- ¿Quienes somos?

- Asistentes

- ↳ ¿Quién ha programado antes?
- ↳ ¿Quién conoce el Arduino?
- ↳ ¿Quién conoce Processing?
- ↳ ¿Traéis los deberes hechos? :)

1. No hay mucho más que añadir



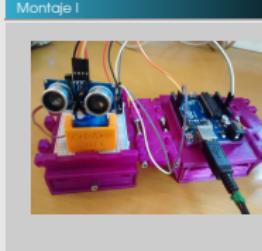
1. Separamos a los que tienen hecha la instalación y los que no, los primeros prueban el Blink y a los segundos los ayudamos.
2. Hora límite 17.00
3. Blink: Fichero::Ejemplos::Básicos::Blink



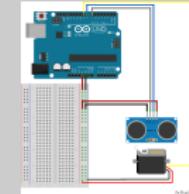
Página Principal

Foto Familia

1. SODAR, igual que RADAR SONic Detection and Range
2. Básicamente dos partes movimiento y sensor
3. Una tercera parte será la estación de usuario
4. *****Corregir GRAFICOS*****
5. Empezamos con el Arduino
6. Año 2005
7. Procesadores de la familia Atmel AVR (AtMega)
8. IDE (basado en processing)
9. CPP con librerías variadas
10. Familia Arduino
11. Ecosistema asociado: shields, sensores, etc



1. ***** CORREGIR FOTO *****
2. Tenemos que fijar el servo (cinta doble cara a la BB)
3. Tenemos que fijar el sensor a la corona del servo
4. Y por último montar la corona



1. ¡Mucho ojo con los cables del servo!
2. Rojo: 5 volt
3. Marron: tierra
4. Naranja: Señal

```
Escritorio - Microcontroller
Nombre: M001_000028.h
Formato: Ajedrezado
Familia: ATmega328P
Categoría: Programa
Este archivo incluye el principio del programa. El resultado es un archivo binario.
void setup() {
  // Crea una variable para el principio del programa. El resultado es un archivo binario.
  void loop() {
    // Crea una variable para el principio del programa. El resultado es un archivo binario.
  }
}
```

1. Tres partes
2. Primera parte imports y variables – Tiempo de compilación
3. Tiempo de ejecución – dos partes setup y loop
4. setup al arrancar, después de un reset hay un arranque
5. loop para siempre jamás (mentira)

```
int medición(int instantes){  
    int result; // Declaración de la variable result  
    result = printLine("Cuantos instantes");  
}
```

1. Explicamos una función Arduino

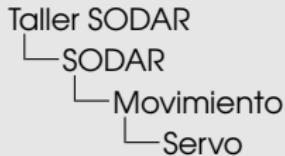
```
Serial.begin(9600);  
Serial.print("Hello world"); // Escribir en la linea  
Serial.println("world"); // Con retorno de linea  
Serial.print(10, DEC); // Escribir 100, 100, 100, 100  
Serial.println(10, DEC, 2); // Imprime 10,10
```

1. Explicamos como se usa el Serial

Taller SODAR
└ SODAR
 └ Movimiento
 └ Servo



1. Que es un servo? Un motor y un pequeño circuito
2. Protocolo de comunicación, anterior a la era digital
3. PWM: 20ms de periodo para los servos
4. Tipos de servos, los nuestros son de 180 grados, 1,6 kg



1. Importar Librería
2. Declarar objeto servo
3. Asociar PIN
4. Ordenar una posición

2014-11-07

Taller SODAR
└ SODAR
 └ Movimiento
 └ Barridos

Barridos



1. ¿Como os imagináis que debe moverse el radar?

Taller SODAR

└ SODAR

└ Movimiento

└ Una solución

```
positive 900000_000 // angulo máximo del servomotor (en grados)
positive 900000_180 // angulo mínimo del servomotor (en grados)
positive 900000_90 // paso de servomotor

positive 100000 // pasos por segundo
positive 1000000 // pasos por 180 grados
positive 10000000 // pasos por 360 grados

angle(1) // 180 grados
angle(0) // 0 grados
angle(0.5) // 90 grados

servo.setAngle(angle) // establece el movimiento al ángulo indicado
if angle > 900000_000 | dimension > 10000000
    servo.setAngle(900000_000) // establece el movimiento al ángulo mínimo
    angle = dimension + 900000_000 // establece el movimiento al ángulo máxi
    servo.setAngle(angle) // establece el movimiento al ángulo indicado
}
```

1. Definimos un paso
2. Controlamos el ángulo
3. Usamos el propio loop del Arduino
4. Vemos nuestra solución en el IDE proyectado y comentamos

Taller SODAR

└ SODAR

└ Sensor

└ Sensor ultrasonidos



Sensor de distancia HC-SR04

1. Como funciona el sensor, tren de pulsos y mide el tiempo en que tarda en recibir el eco

Taller SODAR

└ SODAR

└ Sensor

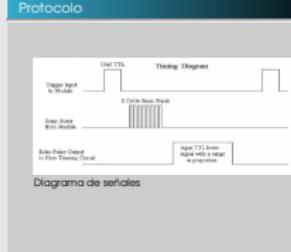
└ Sensor ultrasonidos

Electric Parameter

| | DC 3 V |
|----------------------|---|
| Working Voltage | 1.5Vdc |
| Working Current | 400mA |
| Working Frequency | 40KHz |
| Max Range | 40m |
| Min Range | 1m |
| Measuring Angle | 10 degree |
| Trigger Input Signal | High TTL pulse |
| Echo Output Signal | Low TTL, level signal and the range is proportional |
| Dimensions | 47*29*15mm |

1. Como funciona el sensor, tren de pulsos y mide el tiempo en que tarda en recibir el eco

Taller SODAR
└ SODAR
 └ Sensor
 └ Protocolo



1. Explicamos el protocolo de señales del servo

```
// Función medida
// Se ejecuta una sola vez al principio del programa. Si cambia el ambiente
// se debe re ejecutar.

void setup() {
    Serial.begin(9600); // por si queremos ver los resultados
    Sonido.setVolume(10); // 0=0dB, 25=10dB
    Sonido.setBip(1000); // 1000ms
    Serial.begin(9600); // abre el puerto serial
}
```

1. Una función de bajo nivel

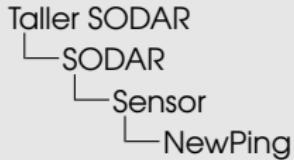


```

    loop {
        if ( sistema_reloj >= reloj_maximo ) break // Salir de la función
        for ( i = 0; i < 10; i++ ) {
            digito1 = digitar(LED0000000000000000, 1000); // Preparación al comienzo
            digito2 = digitar(LED0000000000000000, 1000); // generación de los 10 acoplamientos
            digito3 = digitar(LED0000000000000000, 1000); // Recorrido en pulsos de los 10 acoplamientos
            digito4 = digitar(LED0000000000000000, 1000);
            digito5 = digitar(LED0000000000000000, 1000);

            time = pausar(LED0000000000000000, 1000);
            if ( time < 20 ) // Lector de reloj, tiempo modificando un solo bit
                digitalWrite(LED0000000000000000, 1); // Activación de la alarma
                delay(500); // y la desactivación
        }
    }
}

```



1. Explicamos como se usa la librería NewPing



1. Qué es processing?
2. Processing es java
3. Educativo
4. Interpretado no compilado
5. Artístico
6. Ben Fry and Casey Reas (MIT)