Diapositiva 2

Cargar el ejemplo **movSERVOROTACIONCONTINUA.bly** , conectando el robot.

1. Elección de placa y puerto
2. Añadir bloques
3. Cómo aparece el código a la derecha (código de ARDUINO)
4. Menú derecho en un bloque: duplicar bloques,colapsar, etc.
5. Cómo borrar bloques (papelera)

Explicar preferencias: carpeta de ARDUINO IDE y lenguaje.

Diapositiva 3

- ¿Por qué hemos llegado hasta aquí?
- Explicar la PONENCIA introducción a PRINTBOT
  - Entrar en bitbloq y bitbloq1 (***ya no disponible***) **¡¡Google CHROME!!** (posibilidad de usarse en clase). Problema de 64 bits.

Diapositiva 4

- Acceder a la web de visualino: http:visualino.net (explicarla un poco). Acceder al foro, a la página de download. Explicar cómo no te contestan y por qué hace tiempo que no se actualiza. ¿¿??

Diapositiva 5

- ¿Cómo trabajé con los alumnos? Uso de bitbloq1 y VISUALINO
- Abrir los apuntes de PROCOMÚN sobre VISUALINO. Explicarlos un poco. Están en internet y en la carpeta.
- Abrir alguno de los programas de los chavales para la torre de lEDS y POV.

Diapositiva 6

1. Conectar la placa con un sensor de ultrasonidos.
2. Enseñar el programa PENDULO.PDF (explicar que no se haría así, que es una primera aproximación al problema). Explicar problema del timeout y un poco el programa.
3. Demostración del péndulo. Monitor. Gráfico.
4. Programa BLUETOOTH (cambiar placa y puerto). Montar el robot (placa ZUM BT tiene Bluetooth). Abrir BLUETOOTH.PDF .
5. Explicar el programa.
6. Conectarse con el móvil, ROBOPAD++, y controlar el robot con el móvil.
7. Conectarse con el **IDE ARDUINO**, conectar la placa con el sensor de temperatura **DS 18B20**, y hablar de las limitaciones de **ARDUINO**. Por ejemplo, no es capaz de conectarse adecuadamente con bibliotecas y objetos. Ver ejemplo **intento_temperatura.bly**

Diapositiva 7

1. Explicar dónde están en la red los proyectos **ROBOBLOCKS (proyecto bitbloq1 de bq)** y **BLOCKLY**. Son de código abierto. Explicar que ***roboblock es una adaptación de blockly que genera un código arduino.***
2. Objetivo final: conseguir el fichero ***roboblocks.js***, que es el que usará visualino (carpeta /usr/share/visualino/html)
3. Enseñar el bloque “Escribir en el pin digital ____ el valor [0,1] __” (propio)

Diapositivas 8, 9, 10 , 11 y 12

1. Procedimiento para la generación del fichero roboblocks.js en Ubuntu. Leer diapositiva. Dificultades (diapositiva 8)
2. (Diap. 9) Cómo se crea un bloque. Ejemplo: ***inout_digital_write_var***
   - Crearla en la carpeta src/blocks.
   - “Copiarse” o ir fijándose en cómo están hechos los otros bloques. Porque ya más información de cómo se programa estaría en la documentación de blockly.
   - Enlace hacia BLOQUE_INOUT_DIGITAL_WRITE_VAR (ver su contenido)
   - PRIMERO: fichero README.md
   - SEGUNDO: Código de programación en javascript **[nombre].js**
     - Ver que está dividido en dos funciones, una que va llamando a objetos (plantillas) y va generando una variable alfanumérica code que es el código que se va añadiendo a la salida de VISUALINO a ARDUINO.
     - Segunda función: el diseño del bloque-→ nombre, colores, previo o siguiente…
   - TERCERO: plantillas de código. Bien explicados en las fichas. Plantillas que se llaman desde el fichero anterior e indican cómo modificar el código.
3. Diapositiva 12. Cómo se instala la última versión desde GITHUB, donde está mi código y el de otros (pero el proyecto parece estar parado y no contestan; la última versión deb no lo incluye – no he visto la de windows -).