IES CELSO DÍAZ

GUIA PRÁCTICA ELECTROCARDIOGRAMA

PROGRAMACIÓN CONEXIÓN DEL MÓDULO AD8232 ECG DE PULSO CARDÍACO UTILIZANDO CÓMO PLACA DE CONTROL EL ARDUINO UNO.

1. Introducción

El corazón funciona gracias a impulsos eléctricos que hacen posible su contracción y el bombeo de la sangre por todo el cuerpo. Estos impulsos pueden registrarse mediante un **electrocardiograma (ECG)**, una técnica que permite representar la actividad eléctrica del corazón y comprobar que el ritmo cardíaco es el adecuado.

En esta práctica utilizaremos el **módulo AD8232** junto con una placa **Arduino Uno** para captar la señal eléctrica del corazón mediante unos electrodos colocados sobre la piel. El sensor enviará la información al Arduino, que la mostrará en el ordenador en forma de una gráfica en tiempo real. De esta manera podremos observar cómo cada latido genera una señal eléctrica y comprender mejor el funcionamiento del aparato circulatorio desde un punto de vista práctico.

Esta actividad permite relacionar los contenidos estudiados en la asignatura de **Biología y Geología de 3.º de ESO** sobre el aparato circulatorio con el uso de tecnologías actuales empleadas en medicina e investigación, fomentando además el aprendizaje de técnicas básicas de experimentación y el tratamiento de datos.

2. Objetivos

- Comprender que el corazón genera impulsos eléctricos que coordinan cada latido.
- Conocer qué es un electrocardiograma (ECG) y para qué se utiliza.
- Aprender a colocar correctamente los electrodos para obtener una señal de calidad.
- Montar el circuito conectando el módulo AD8232 a una placa Arduino Uno.
- Programar el Arduino para registrar la actividad eléctrica del corazón.
- Visualizar en el ordenador la señal del electrocardiograma e identificar los latidos cardíacos.
- Relacionar la frecuencia cardíaca observada con el funcionamiento del aparato circulatorio.
- Desarrollar hábitos de trabajo en el laboratorio, realizando el montaje y la toma de datos de forma ordenada y segura.

3. Material

Para realizar esta práctica el material necesario es:

a) AD8232 ECG Módulo Monitor de Pulso Cardíaco

b) Arduino Uno
c) Cables Dupont cortos HH-MM

4. Colocación de los electrodos

Lo primero que hay que hacer es conectar las almohadillas. Para ello hay que despegar el protector de plástico que protege al gel conductor y colocarlos como se ve en la imagen de acuerdo al color del electrodo y función de acuerdo al triángulo de Einthoven.

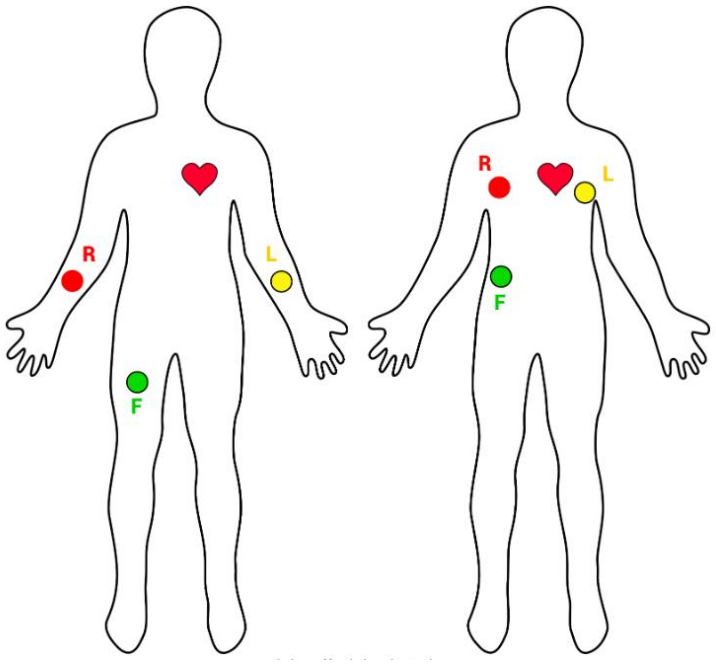


Imagen1: Colocación de los electrodos (fuente: programación conexión AD8232 ECG Arduino Uno?)

Los sensores se pueden colocar en los antebrazos y las piernas como se muestra en la imagen de arriba. Pero de igual manera se pueden colocar en el pecho cerca de los brazos y sobre el abdomen inferior derecho. Cuanto más cerca del corazón estén las almohadillas, mejor será la medición.

5. Diagrama de Conexión

El Módulo AD8232 ECG tiene 6 pines de los cuales solo utilizaremos 5 para conectarlo al Arduino UNO. Lo primero que debes conectar es la alimentación, para ello conecta el pin de 3.3V del Módulo a la salida de 3.3V del Arduino, después conecta el pin GND del Módulo al pin GND del Arduino, el pin OUTPUT es la salida analógica del sensor por lo que tendrás que conectar a una entrada analógica de Arduino, para este tutorial utilizaremos A0.

Por ultimo conectaremos los pines LO- y LO+ a dos pines digitales del Arduino Uno, en este tutorial utilizaremos los pines 10 y 11 del Arduino UNO. En la imagen 2 está el diagrama de las conexiones necesarias:

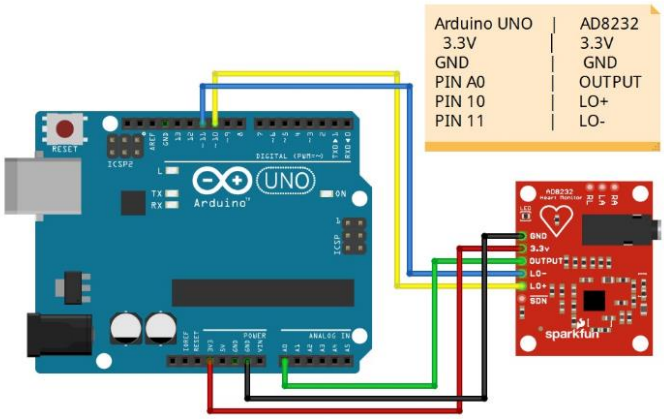


Imagen2: Conexión del Módulo AD8232 con Arduino Uno (fuente: programación conexión AD8232 ECG Arduino Uno?)

6. Código

Para realizar la programación del módulo AD8232 se ha utilizado el IDE de Arduino o Processing. ya qué estos programas nos permiten visualizar de manera gráfica las lecturas del Sensor.

El código utilizado es el siguiente:

```
void setup(){
 // Inicializar la comunicación en serie:
  Serial.begin(9600);
  pinMode(10, INPUT); // Configuración para la detección LO +
  pinMode(11, INPUT); // Configuración para la detección LO -
}
void loop() {
if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
Serial.println('!');
}
else{
// Imprimir la lectura del puerto A0
Serial.println(analogRead(A0));
}
//Espere un poco para evitar que los datos en serie se saturen
delay(1);
}
```

Después de desarrollar la Programación y conexión del AD8232 ECG, subir y copilar el código, se debe abrir el monitor serie para ver las lecturas del sensor. De igual manera se puede abrir el Serial Plotter para visualizar de manera gráfica el pulso cardíaco.

5. Conclusiones

En esta práctica se desarrolla un sistema básico de electrocardiografía utilizando el módulo AD8232 y Arduino UNO, demostrando su viabilidad para adquirir y visualizar señales ECG. Esta práctica permite comprobar cómo los conocimientos de Biología pueden combinarse con la electrónica y la programación para estudiar el funcionamiento del cuerpo humano mediante herramientas similares a las utilizadas en el ámbito sanitario.

IES CELSO DÍAZ

GUIA PRÁCTICA XTOOL

REALIZACIÓN DE PROTOTIPOS PARA PROYECTOS DE EMPRENDIMIENTO

Materiales

- Tablerillo
- Vinilo (opcional)
- Impresora corte laser Xtool

Introducción

En este proyecto, los alumnos trabajarán en equipos para crear una propuesta de emprendimiento completa, simulando el proceso que sigue una empresa desde la identificación de una necesidad hasta el lanzamiento de un producto o servicio al mercado.

El reto consiste en detectar un problema, necesidad u oportunidad existente en la sociedad, diseñar una solución innovadora y desarrollar un modelo de negocio viable que permita convertir esa idea en una empresa real. Como resultado final, cada equipo deberá presentar un plan de Empresa y un prototipo que permita visualizar y validar su propuesta.

Este proyecto busca integrar conocimientos de creatividad, innovación, tecnología, marketing, finanzas y gestión empresarial, fomentando además el trabajo colaborativo, la capacidad de análisis y la comunicación efectiva.

Los objetivos principales del proyecto son:

- Identificar oportunidades de negocio en su entorno.
- Diseñar soluciones innovadoras a problemas reales.
- Analizar el mercado y las necesidades de los clientes.
- Elaborar un plan de empresa estructurado y coherente.
- Evaluar la viabilidad económica y comercial de una idea.
- Desarrollar un prototipo que represente su producto o servicio.
- Presentar profesionalmente una propuesta empresarial (Storytelling/Elevator Pitch).

Al finalizar el proyecto, cada equipo habrá desarrollado una propuesta empresarial completa capaz de transformar una idea en una posible oportunidad de negocio, demostrando su utilidad mediante un prototipo y justificando su viabilidad a través de un Plan de Empresa estructurado y fundamentado. El objetivo es que los estudiantes experimenten de forma práctica cómo nacen, se diseñan y se desarrollan los proyectos emprendedores en el mundo real.

MISIÓN: PREPARAR PROTOTIPOS QUE REPRESENTEN SUS PRODUCTOS O SERVICIOS.

El objetivo final de la actividad es que los alumnos de 2º de bachillerato que cursan Fundamentos de Administración y Gestión presenten sus proyectos al concurso Startinnova (programa educativo dirigido principalmente a estudiantes de bachillerato que trabajan en equipo, tutorizados por un profesor, para desarrollar un proyecto empresarial cuyo objetivo es fomentar el espíritu y las habilidades emprendedoras entre los jóvenes). En la fase final de dicho concurso, deben defender su propuesta presencialmente ante un comité evaluador. Para ello es importarte utilizar una técnica narrativa que permita comunicar el modelo de negocio de una manera atractiva para provocar una conexión emocional con el público. Los alumnos utilizan distintas técnicas de comunicación como son el Storytelling y el Elevator Pitch que se ensayan y graban en colaboración con el proyecto Almazuela.

Es en dicha fase donde los prototipos cobran especial relevancia y dotan de realismo el proyecto. El prototipo es una representación de la solución propuesta. Su objetivo no es necesariamente ser un producto terminado, sino demostrar cómo funcionaría la idea en la práctica. El prototipo debe permitir comprender claramente el funcionamiento y las ventajas de la propuesta.

Puede consistir en:

- Una maqueta física.
- Un modelo impreso en 3D.
- Una aplicación o página web básica.
- Un diseño digital interactivo.
- Un vídeo demostrativo.

En nuestro caso los alumnos han utilizado la cortadora laser Xtool para preparar sus prototipos en formato físico. Los pasos para realizar dichos prototipos (imagen de la portada de la guía) han sido los siguientes:

1. Preparación del diseño

- **Elegir o crear el diseño:** Puedes dibujar tu prototipo directamente en el programa oficial **xTool Creative Space (XCS)** o importar vectores desde programas como Adobe Illustrator, Inkscape o AutoCAD en formatos compatibles como .svg, o .dxf.

  También es posible utilizar programas de diseño, con formatos .png por ejemplo, para la capa de grabado siempre que la capa de corte se realice en los programas descritos anteriormente que permitan obtener los formatos mencionados.

- **Configurar los colores (capas):** Para diferenciar qué hará el láser, asigna colores distintos a cada acción. Por ejemplo: usa el color **rojo** para las líneas de corte (Vector) y el **negro** para las líneas que solo quieres grabar o marcar levemente.

2. Preparación del material y la máquina xTool

- **Seleccionar el material:** Existen distintos grosores y tonos entre los que poder elegir. Como se observa en la imagen de portada los alumnos seleccionaron distintas tonalidades en función del efecto final que querían lograr en sus prototipos.

  Se recomienda:

  - Utiliza madera plana y sin deformaciones.
  - Verifica que el material sea apto para corte láser.
  - Evita materiales con colas o recubrimientos desconocidos.

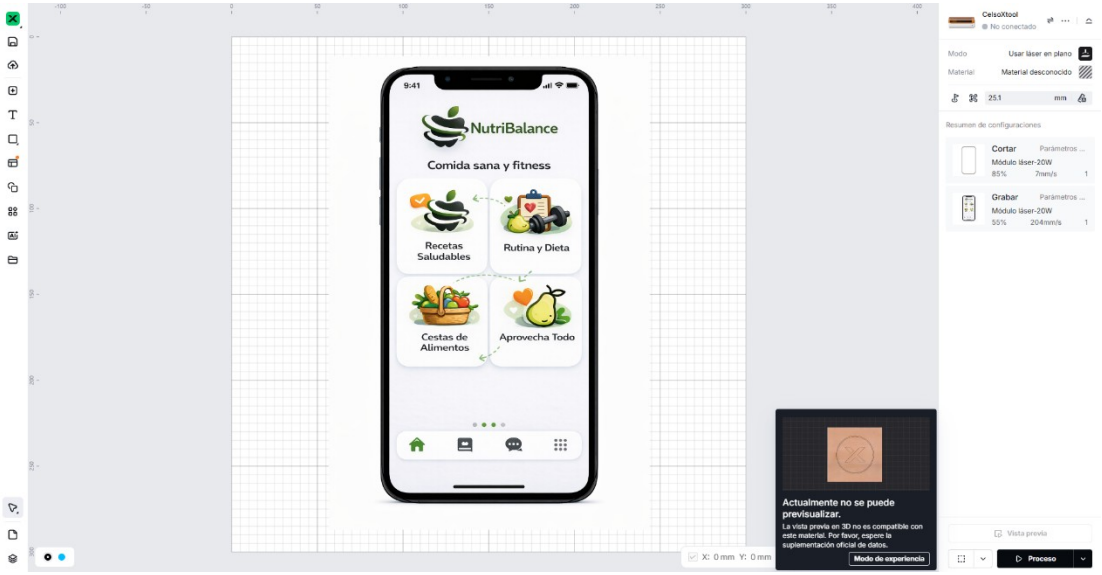
- **Colocar el material:** Poner el material (tablerillo) sobre la cama de nido de abeja. Asegúrate de que esté lo más plano posible. Si está curvado, el enfoque del láser fallará. Usa imanes o pinzas para fijarlo.

- **Extracción de humos:** Encender el sistema de purificador de aire. Asegúrate de que el tubo de extracción esté sacando el humo hacia el exterior.

- **Enfoque correcto:** Mide la distancia focal de manera automática o manual según te indique tu modelo xTool (por ejemplo, usando la regla provista o el sensor de la máquina).

3. Configuración en el software (XCS)

- **Seleccionar tu máquina y material:** Abrir **xTool Creative Space,** conectar tu máquina vía WIFI e importar el/los archivo/s (.svg, .dxf, .png).
- **Ajustar parámetros de corte y grabado (potencia y velocidad):** Ajustar los parámetros de potencia y velocidad a tu diseño.



CONSEJO IMPORTANTE

Realizar una prueba de material: Cada máquina y lote de madera es ligeramente distinto. Es recomendable hacer una pequeña matriz de prueba (cuadros con distintas combinaciones de potencia/velocidad) en un trozo sobrante para encontrar los valores perfectos para tu prototipo.

4. Orden de las operaciones

Las capas deben procesarse en el siguiente orden:

1. Primero: Grabado
2. Segundo: Marcado
3. Tercero: Corte

Si cortas primero, la pieza puede moverse y hacer que el grabado posterior no quede centrado.

5. Encuadre y corte

- **Función Frame (Encuadre):** Antes de iniciar, usa la herramienta de encuadre en el programa. Esto hará que el cabezal del láser se mueva dibujando un rectángulo alrededor de la zona donde va a trabajar, permitiéndote verificar visualmente que el diseño cabe exactamente en tu tablerillo.
- **Ejecución:** Cierra la cubierta de seguridad, asegúrate de que la extracción de humos esté encendida y presiona "Start" (Iniciar).
- **Revisión:** No levantes la tapa ni muevas el material cuando termine el trabajo. Si notas que una zona no se cortó por completo o el grabado no se ejecutó con la potencia deseada, puedes ejecutar un segundo pase sin alterar la posición de la pieza.

6. Retirar y limpiar las piezas

Esperar unos 10 segundos antes de abrir la tapa. Con cuidado, retira las piezas cortadas y limpiar el hollín sobrante.

IES CELSO DÍAZ

IES CELSO DÍAZ

GUÍA DIDÁCTICA DE FILOSOFÍA

LÓGICA FILOSÓFICA Y CONDUCCIÓN AUTÓNOMA: PENSAR, ARGUMENTAR, PROGRAMAR

Materiales

- Textos filosóficos sobre lógica (Aristóteles, Frege, Wittgenstein)
- Textos sobre ética aplicada a vehículos autónomos (Philippa Foot, dilema del tranvía)
- Ordenador (software educativo para lógica o tabla de verdad digital)
- Tarjetas con enunciados lógicos y falacias
- Opcional: simulador de conducción autónoma o vídeos de sistemas reales (Tesla, Waymo)

IES CELSO DÍAZ

Introducción

La conducción autónoma no es solo ingeniería y sensores. Es, ante todo, lógica aplicada. Cada decisión de un coche eléctrico que circula sin conductor –*cada if al detectar un peatón, cada then al elegir frenar o esquivar*– es una expresión de reglas lógicas que la filosofía ha estudiado durante siglos. Desde los silogismos de Aristóteles hasta la lógica de predicados de Frege, el pensamiento lógico es el fundamento tanto del razonamiento filosófico como del código de los vehículos autónomos.

Imagina un coche autónomo circulando por una ciudad. Su comportamiento se basa en reglas del tipo: SI el radar detecta un obstáculo ENTONCES (→) reducir velocidad. SI hay un peatón cruzando ENTONCES (→) detenerse. Esta estructura condicional es idéntica a la de un silogismo o a la de una regla de inferencia lógica. *Si la lógica falla –si hay una falacia en el razonamiento humano que diseñó el código*– el coche cometerá errores potencialmente mortales. Por eso, saber lógica filosófica no es algo ajeno a la tecnología de los coches autónomos: es la base para programar mejor, depurar errores y diseñar sistemas más seguros y éticos.

Además, los coches autónomos plantean problemas éticos ineludibles: ¿cómo debe decidir un vehículo autónomo ante un accidente inevitable? ¿Debe priorizar la vida del pasajero o la de los peatones? ¿Quién es responsable cuando un coche autónomo comete un error? Estas preguntas pertenecen a la lógica deóntica (la lógica de las normas) y a la ética aplicada.

En esta guía, el alumnado no solo aprenderá qué es un silogismo o una falacia. Aplicará la lógica filosófica al análisis de sistemas de conducción autónoma, comprendiendo que programar la decisión de un coche es, en el fondo, formalizar razonamientos morales.

MISIÓN: DEPURAR LA LÓGICA DE UN COCHE AUTÓNOMO

El IES CELSO DÍAZ, en colaboración con un proyecto de innovación en movilidad sostenible, ha diseñado un prototipo de coche eléctrico autónomo a pequeña escala. El sistema debe circular por un circuito, detectar obstáculos, peatones y señales, y tomar decisiones seguras siguiendo reglas lógicas programadas por el alumnado.

Sin embargo, durante las pruebas iniciales se ha detectado un problema: el coche autónomo toma decisiones incorrectas porque las reglas lógicas del código contienen errores de estructura argumentativa. Es decir, el programa comete "falacias lógicas": confunde condiciones, invierte implicaciones (frena cuando debería acelerar), usa negaciones incorrectamente o genera bucles lógicos sin salida.

La situación es preocupante porque el prototipo será presentado en una feria de innovación tecnológica ante empresas y otros centros educativos. Ante esta situación, el equipo docente ha creado un grupo especial de ingenieros lógicos formado por el alumnado de 1º de Bachillerato.

Vuestra misión será analizar el código del coche autónomo, identificar los errores lógicos (falacias, negaciones mal aplicadas, condicionales invertidos, problemas en el dilema del tranvía) y corregirlos aplicando las reglas de la lógica filosófica. Si conseguís depurar la lógica del sistema, el coche circulará con seguridad y demostraréis que la filosofía es indispensable para la tecnología autónoma.

Se realiza una explicación al alumnado sobre la estructura de un condicional lógico (SI $p \rightarrow q$), la diferencia entre condición necesaria y suficiente, las leyes de De Morgan (negación $\neg$ de conjunciones $\wedge$ y disyunciones $\vee$) y las falacias formales más comunes (afirmación del consecuente, negación del antecedente). A continuación, se presenta el problema del tranvía adaptado a coches autónomos y se entrega un código defectuoso (en pseudocódigo) para que el alumnado identifique los fallos lógicos y "depure" el sistema de conducción autónoma.

Parte 1. Modificar un código con errores lógicos. Depurando la lógica del coche autónomo

Primer error lógico (condicional invertido). Se introduce un error en el código donde el coche aplica la regla inversa: en lugar de "SI hay un peatón ENTONCES frenar" ($p \rightarrow q$), el programa hace "SI frenar ENTONCES hay un peatón" ($q \rightarrow p$). Esto es una falacia lógica de afirmación del consecuente. El alumnado debe identificar el error y reescribir la regla correctamente aplicando la estructura del condicional material.

Segundo error lógico (negación incorrecta). Se introduce un error en la función que detecta cuándo NO debe detenerse el coche. El código original usa mal las leyes de De Morgan: la negación de "hay obstáculo O hay peatón", es decir $\neg(p \vee q)$, se programa como "NO hay obstáculo O NO hay peatón" ($\neg p \vee \neg q$) en lugar de "NO hay obstáculo Y NO hay peatón" ($\neg p \wedge \neg q$). El alumnado debe corregir la negación aplicando las leyes lógicas de De Morgan.

Tercer error lógico (bucle sin condición de salida). Se genera un bucle infinito en el sistema de navegación porque la condición de parada está mal formulada: MIENTRAS (verdadero) HACER... en lugar de MIENTRAS (no haya llegado al destino) HACER... El alumnado debe reformular la condición de salida utilizando una variable de destino y una negación lógica correcta (¬destino).

Actividad de profundización. Reto. El coche autónomo debe decidir entre frenar o esquivar según el tipo de obstáculo. Diseñar una tabla de verdad con las cuatro combinaciones posibles (peatón / objeto inanimado / animal / señal de stop) y programar las reglas lógicas correspondientes usando SOLO los operadores Y (∧), O (∨), NO (¬). Demostrar que las reglas son lógicamente consistentes (sin contradicciones).

Parte 2. Mejorar el sistema con lógica modal y deóntica (ética del coche autónomo)

El alumnado dispone de un código previo que funciona correctamente desde el punto de vista lógico-formal, **pero plantea un problema ético clásico**: *el dilema del tranvía adaptado a coches autónomos*. El coche circula y de repente: frenar implica atropellar a un peatón; desviarse implica atropellar a otro peatón; no hacer nada implica un accidente grave para el pasajero.

El objetivo es ampliar el código con reglas de lógica deóntica (obligación, permiso, prohibición) para que el coche pueda tomar decisiones éticas explícitas. Por ejemplo:

- PROHIBIDO (atropellar a un peatón SI hay alternativa)
- OBLIGATORIO (priorizar la vida del pasajero SOBRE la del peatón)
- PERMITIDO (desviarse SI la víctima es un animal NO humano)

El alumnado debe modificar el código original para incorporar tres reglas deónticas y explicar filosóficamente por qué estas reglas no son solo técnicas, sino también éticas (basándose en Philippa Foot, Judith Jarvis Thomson o Hans Jonas).

Actividad de profundización. Reto.

Conseguir que el coche autónomo distinga entre dos tipos de decisiones aplicando un silogismo hipotético:

Premisa 1: SI se detecta un peatón ENTONCES se debe frenar ($p \rightarrow q$)

Premisa 2: SI se debe frenar ENTONCES la velocidad debe ser 0 ($q \rightarrow r$)

Conclusión lógica: SI se detecta un peatón ENTONCES la velocidad debe ser 0 (p → r)

El alumnado debe programar esta cadena de inferencia y demostrar su validez mediante una tabla de verdad o una derivación lógica formal. Después, debe responder: ¿qué ocurre si hay dos peatones en direcciones opuestas?

Parte 3 (ampliación). Filosofía de la lógica autónoma: ¿quién es responsable del error del coche?

Debate filosófico a partir de la siguiente pregunta:

"Si un coche autónomo comete un error lógico (por ejemplo, confunde un peatón con una sombra por una mala negación en el código) y atropella a alguien, ¿quién es el responsable moral? ¿El programador? ¿El fabricante? ¿El algoritmo? ¿El propietario del vehículo? ¿Nadie?"

Textos de apoyo: fragmento de Aristóteles (Ética a Nicómaco sobre la responsabilidad) y un texto breve de Hans Jonas (El principio de responsabilidad aplicado a la tecnología).

Producto: texto argumentativo (1 página) o grabación de un debate simulado entre un ingeniero, un filósofo, un abogado y un fabricante de coches.

Evaluación

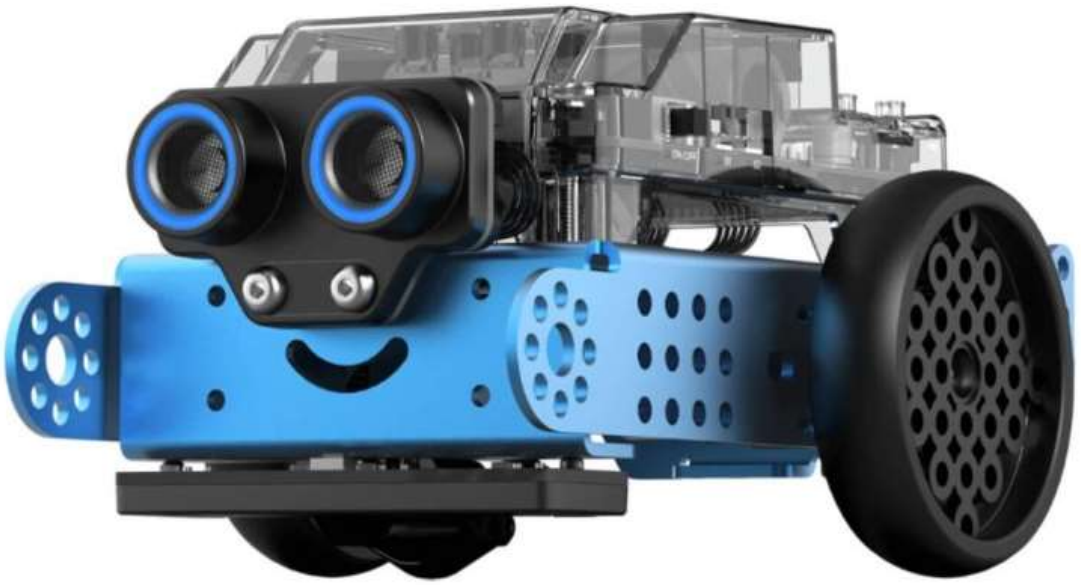
Indicador	Instrumento
Identifica y corrige errores lógicos en reglas condicionales	Ficha de depuración lógica (Parte 1)
Aplica correctamente las leyes de De Morgan (negación de enunciados compuestos)	Ejercicio de negación de enunciados compuestos
Diseña tablas de verdad consistentes	Reto de profundización (Parte 1)
Incorpora reglas deónticas al código ético	Código modificado con explicación filosófica
Argumenta sobre la responsabilidad en sistemas autónomos	Texto argumentativo o debate simulado

Recursos complementarios

- Aristóteles, Órganon (selección de Primeros Analíticos sobre el silogismo)
- Philippa Foot, "El problema del tranvía" (artículo original de 1967)
- Tabla de verdad interactiva (herramienta digital)
- Vídeo: "El dilema del coche autónomo" (YouTube: MIT Moral Machine)
- Vídeo: "Lógica proposicional aplicada a la programación de coches"
- Guía rápida: Leyes de De Morgan con ejemplos de conducción
- Podcast: "Ética y coches autónomos" – Filosofía en la red
- Simulador online: Moral Machine (MIT) https://www.moralmachine.net/hl/es

IES CELSO DÍAZ

GUIA PRÁCTICA mBot2



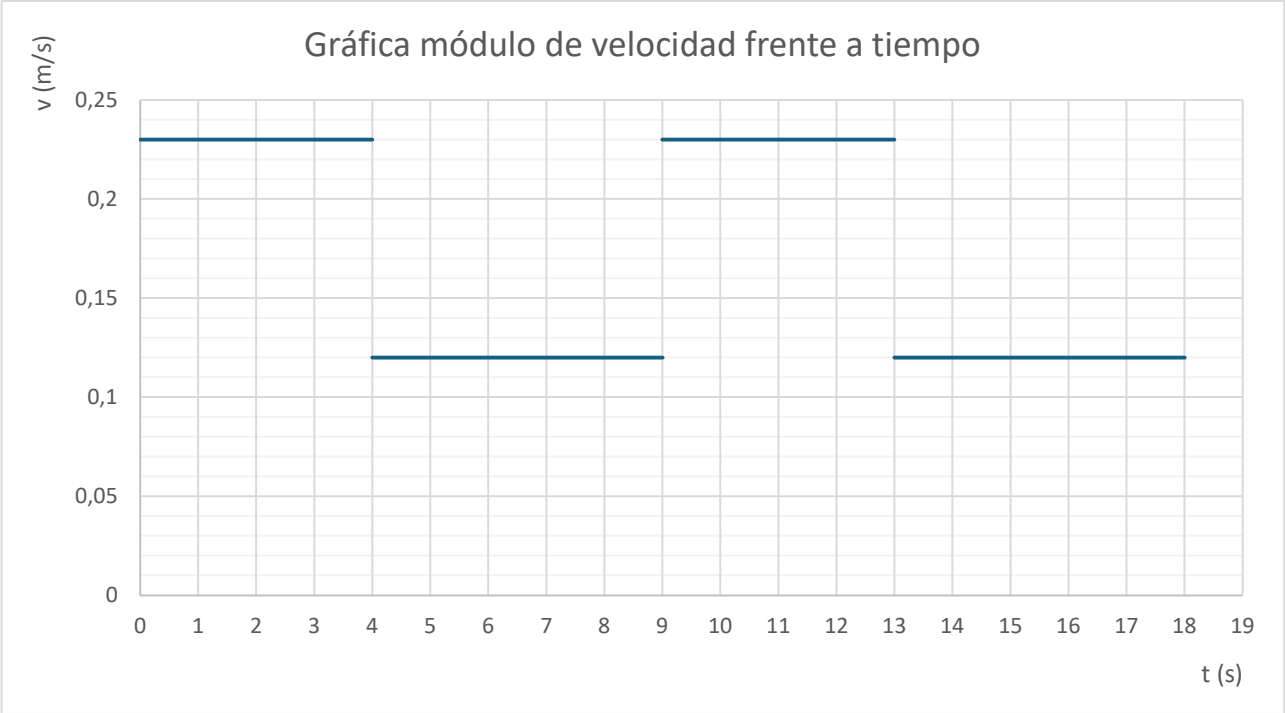
MATERIAL:
* Chromebook
* Robot MBot2
* Cinta métrica
* Calibre

OBJETIVOS

- Interpretar gráficas velocidad-tiempo y extraer datos.
- Comprender la relación entre velocidad lineal y velocidad angular.
- Programar un robot mBot con los parámetros de velocidad calculados previamente.

PARTE I Cálculo de la velocidad angular a partir de la gráfica velocidad / tiempo

En esta parte debes analizar el problema planteado: debes programar un robot para que describa un recorrido rectangular sobre cuatro mesas que habrás puesto juntas, sin caer al suelo. El movimiento en cada tramo vendrá descrito por la siguiente gráfica:



El robot no admite valores de velocidad lineal, sino velocidad de giro de sus ruedas, ω (rev/min). Tienes que usar lo aprendido sobre movimiento circular: $v=\omega \cdot R$, sabiendo que en dicha fórmula, ω se introduce en rad/s, y la equivalencia: 1 rev = $2 \cdot \pi$ rad. Puedes medir con una regla lo que consideres oportuno.

CUESTIONES:

(1 pto.) ¿Qué tipo de movimiento lleva el robot en cada tramo? Justifica la respuesta.

(1 pto.) ¿Qué tipo de movimiento tiene cada rueda en cada tramo, tomando como referencia su centro? Justifica la respuesta.

(4 ptos.) En base a la gráfica proporcionada y a las medidas obtenidas, calcula la velocidad angular de las ruedas (en rev/min o r.p.m.), en los dos primeros tramos.

(3 ptos.) Halla la distancia total recorrida por el robot, y el desplazamiento en todo el recorrido.

(1 pto.) Calcula la velocidad media en todo el recorrido. ¿Significa que ha estado en reposo todo el tiempo?

PARTE II. Programación del robot en tu Chromebook:

Accede a la web: https://mblock.cc/ y pulsa el botón inferior izquierdo: “Code with block”.

En la parte inferior de la columna central pulsa el botón: 

, y añade la extensión: ”mBot2 shield”.

Programa como en Scratch, incluyendo una primera instrucción que sea: “Al pulsar el botón A”…

Las instrucciones de movimiento aparecen en los bloques azules de la extensión.

Cuando tengas el código completo, avisa a tu profesor/a para que lo revise y te ayude a transferirlo al robot.

PARTE III. Ejecución del programa en el robot:

Conociendo las distancias que recorre el robot en cada tramo y vuestro programa, colocad el robot en una posición que asegure que no caerá de las mesas.

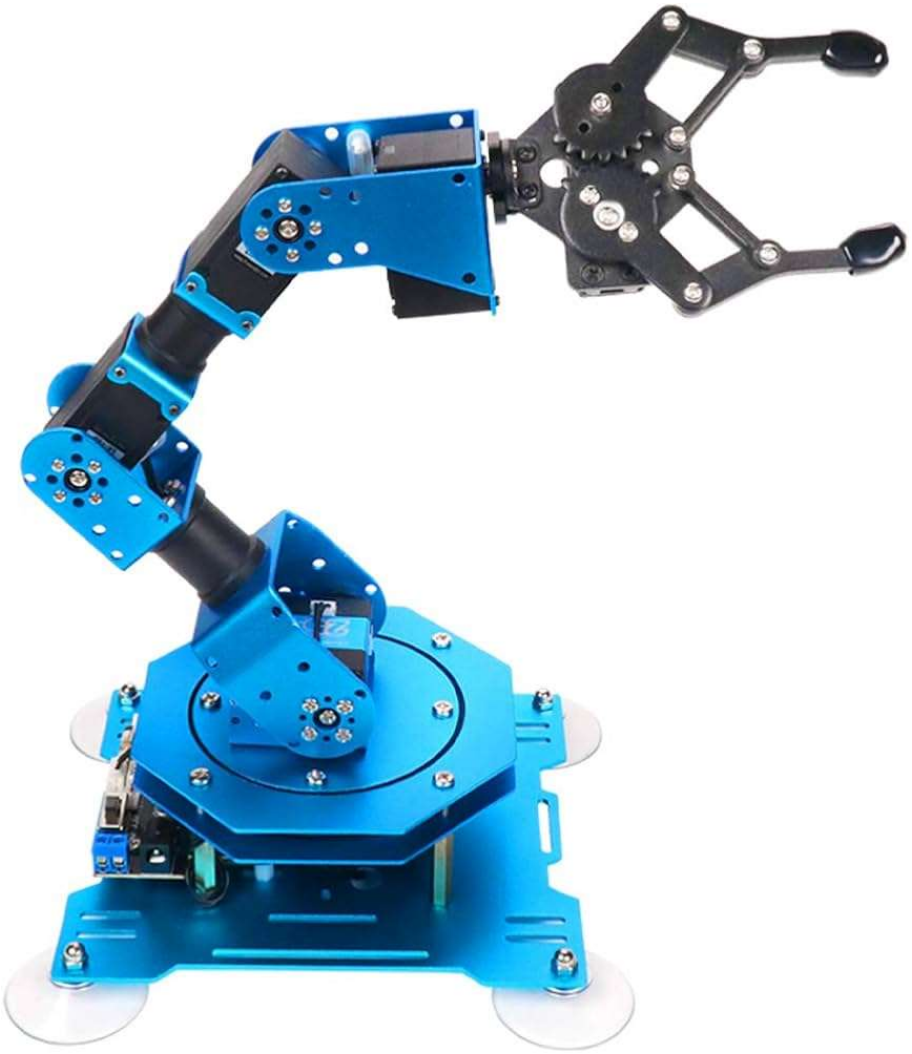
No ejecutes el programa hasta que tu profesor/a haya supervisado la posición. Presta especial atención durante el recorrido para detener el robot si se acerca demasiado al borde de la mesa.

Cuando tu profesor/a dé el visto bueno, podrás ejecutar el programa. ¡Mucha suerte!

IES CELSO DÍAZ

GUIA PRÁCTICA BRAZO ROBÓTICO

REPARAR UNA PLANTA DE RECICLAJE AUTOMATIZADA



Materiales

- Brazo xArm AI
- Bloques EVA
- Ordenador (Software WonderCode)

Introducción

Cada día generamos una enorme cantidad de residuos: botellas de plástico, latas, envases de cartón, vidrio y otros materiales terminan en los contenedores de reciclaje. Sin embargo, para que estos materiales puedan reutilizarse es necesario clasificarlos correctamente.

Imagina una planta de reciclaje moderna. Miles de objetos llegan cada hora a través de cintas transportadoras. Sería muy difícil para las personas identificar y separar manualmente todos esos residuos con rapidez y precisión. Por ello, muchas instalaciones utilizan sensores, cámaras y robots capaces de realizar esta tarea automáticamente utilizando la inteligencia artificial.

En una de estas plantas, una cinta transportadora lleva residuos mezclados hasta una estación de clasificación. Allí, un sensor identifica el material de cada objeto y envía la información a un brazo robótico que recoge el objeto y lo deposita en el contenedor adecuado. Gracias a este proceso, miles de materiales pueden ser reutilizados para fabricar nuevos productos, reduciendo el consumo de materias primas y la contaminación.

MISIÓN: REPARAR UNA PLANTA DE RECICLAJE AUTOMATIZADA

El IES Celso Díaz, en colaboración el proyecto CEHS, ha puesto en marcha un innovador proyecto de sostenibilidad y robótica para mejorar la gestión de los residuos generados en el centro educativo.

Cada día, cientos de envases, hojas de papel, botellas y latas son depositados en los contenedores del instituto. Con el objetivo de fomentar el reciclaje y aplicar los conocimientos adquiridos en clase, se ha diseñado una **planta de reciclaje automatizada a pequeña escala** utilizando un brazo xArm y diferentes sus sensores.

El sistema debía identificar automáticamente los residuos y depositarlos en el contenedor correspondiente. Sin embargo, durante las pruebas realizadas por el profesorado se ha detectado un problema: **muchos objetos están siendo clasificados incorrectamente.**

La situación es preocupante porque la planta será presentada próximamente en la **Feria Tecnológica de La Rioja**, donde otros centros educativos visitarán el proyecto para conocer cómo la robótica puede ayudar a mejorar la sostenibilidad.

Ante esta situación, el equipo docente ha decidido crear un **grupo especial de ingenieros formado por el alumnado de 3ºESO.**

Vuestra misión será investigar el funcionamiento del sistema, analizar el código de programación y localizar los errores que impiden que los residuos lleguen al contenedor correcto.

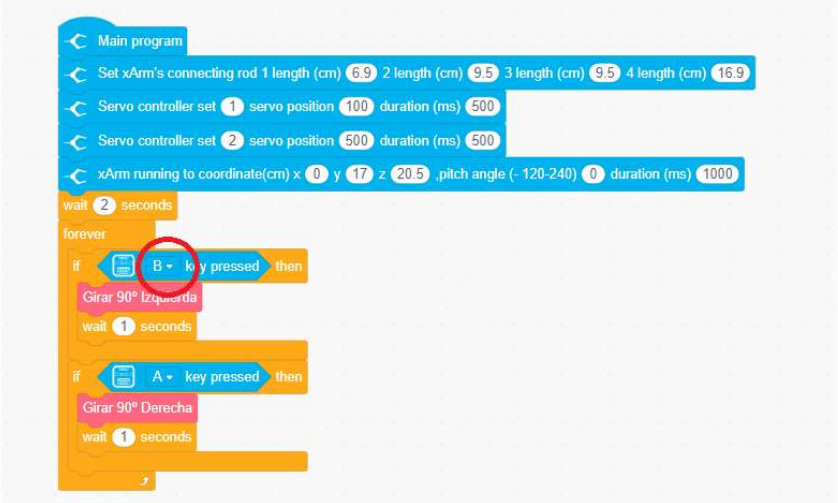
Si conseguís resolver el problema, la planta podrá presentarse con éxito en la feria y demostrar como la tecnología puede contribuir al cuidado del medio ambiente.

Se realiza una explicación al alumnado sobre el funcionamiento del código, en que consiste cada uno de los pasos y una breve introducción al sistema de referencia en tres dimensiones. A continuación, se explican las conexiones del robot, incluyendo como conectarlo al ordenador y cargar un programa con el código diseñado. Finalmente se entrega por TEAMS un código defectuoso para que el alumno sea capaz de identificar que falla y “arreglar” la planta de reciclado.

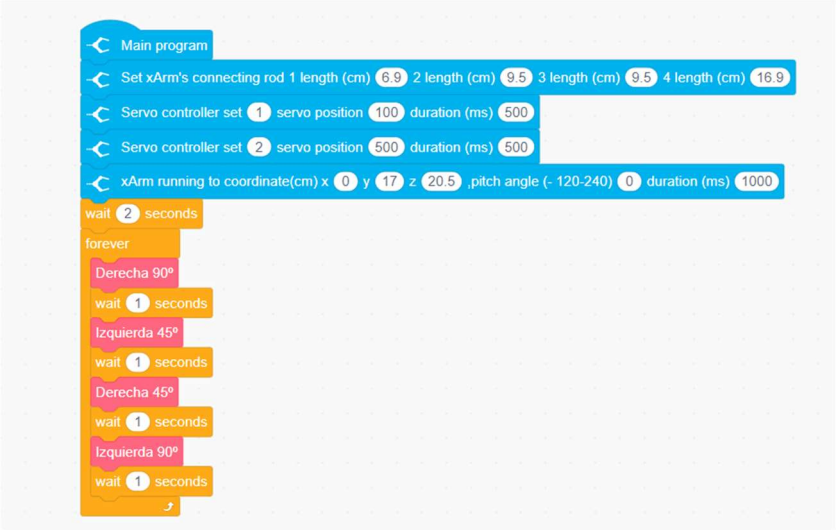
Parte 1. Modificar un código con errores. Calibrando la máquina

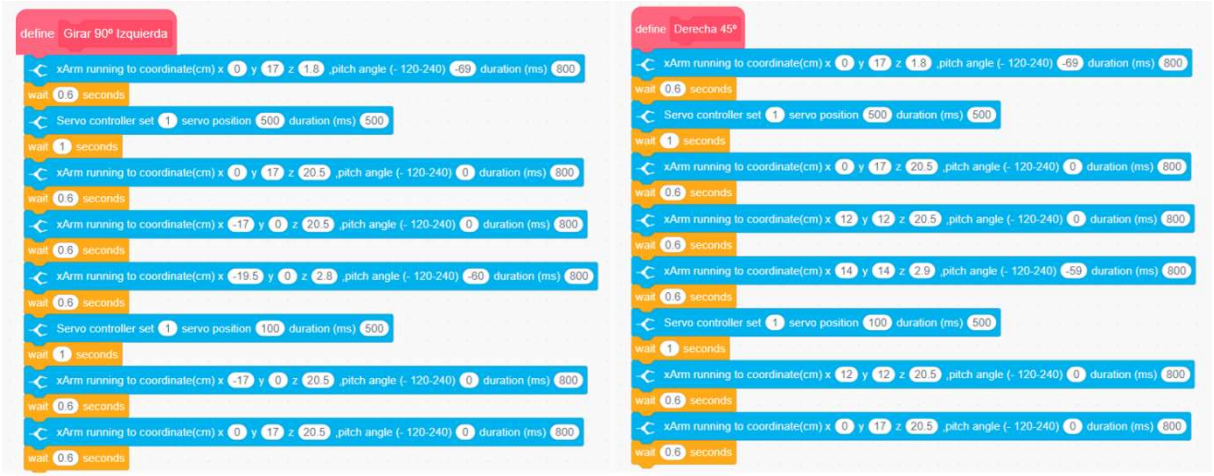
Primer error. Se introduce un error en el código donde el brazo no ajusta lo suficiente la altura para coger el objeto. Para solucionarlo el alumnado debe modificar el valor de la variable z a un valor entre 0 y 1. El fallo está presente en las cuatro funciones que están programadas para controlar el brazo, deben modificar las cuatro.

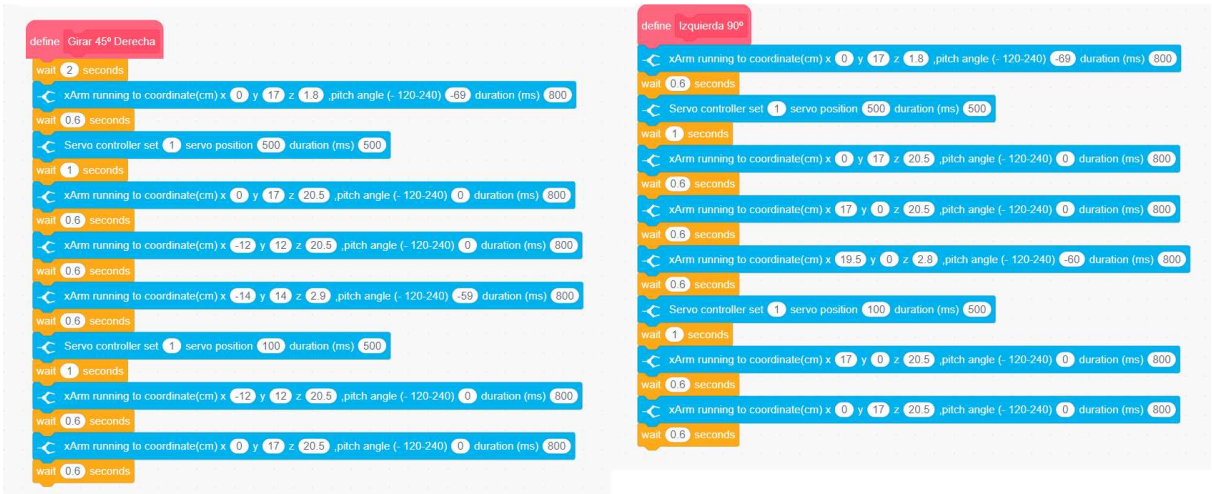
Segundo error. Una vez comprobadas las posiciones extremas a 90º se deben calibrar las intermedias, para ello el alumnado debe cambiar las funciones del programa principal por “Girar 45º Izquierda” y “Girar 45º Derecha”. Ahora el error está en los pulsadores, cuando pulsas izquierda se va a la derecha y viceversa. Se puede solucionar de varias formas, cambiando el nombre de la función y reasignando el código o cambiando el pulsador en función de la orden para invertir el programa.



Tercer error. Puesta en marcha. Se generar un bucle, para que el brazo trabaje de forma automática cogiendo los diferentes objetos y repitiendo los movimientos a los diferentes contenedores recorriendo un ángulo llano en cuatro pasos.





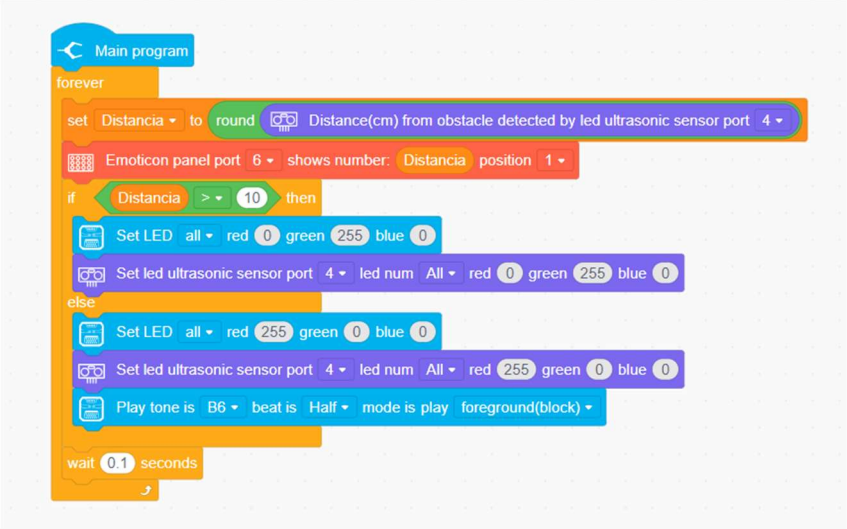


Actividad de profundización. Reto. El brazo está diseñado para repetir el bucle sin parar, no obstante, en un caso real es necesario añadir una parada de emergencia para evitar accidentes. Modifica el código para añadir esta opción.

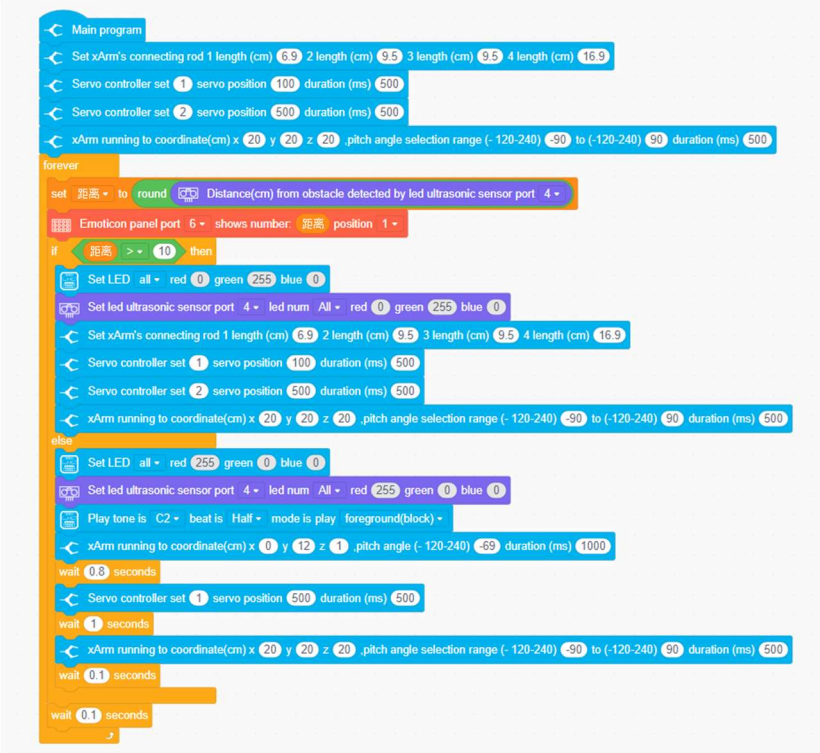
Parte 2. Mejorar el sistema con sensor de proximidad.

El alumnado dispone de un código previo que devuelve la distancia de un objeto al brazo, después enciende las luces y finalmente emite sonido cuando se acerca algo al objetivo (para que el personal de la fábrica se aleje).

El objetivo es modificar este código, basándose en el anterior, para que el brazo agarre el objeto y lo lance a otra posición. Quitando el obstáculo del campo de detección.



Ejemplo de respuesta:

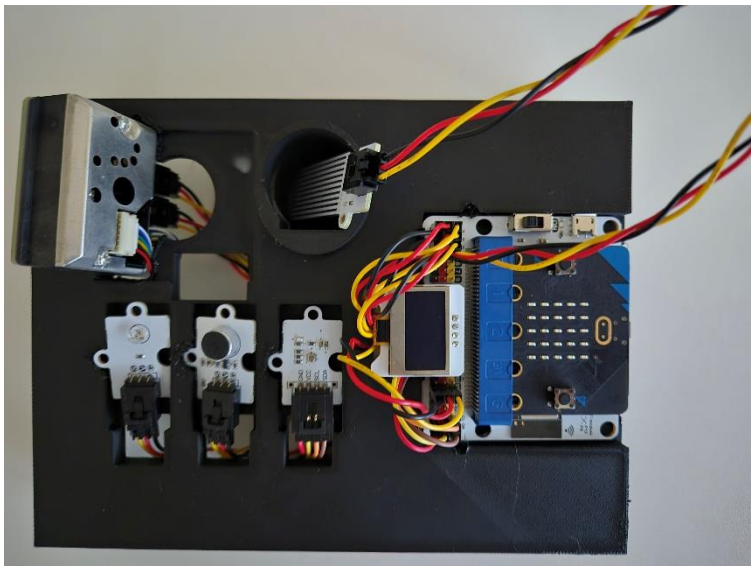


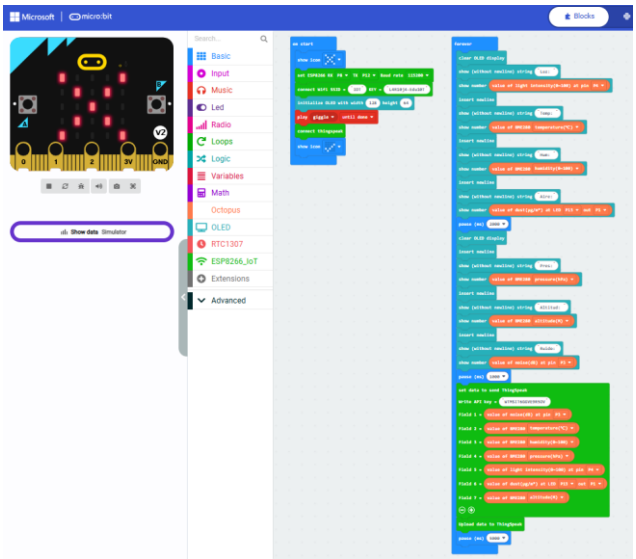
Actividad de profundización. Reto. Conseguir replicar el programa de forma que el primer objeto que detecta lo mueve a la izquierda y lo deposita en un contenedor y el segundo objeto que detecta lo mueve a la derecha y lo deposita en otro contenedor diferente.

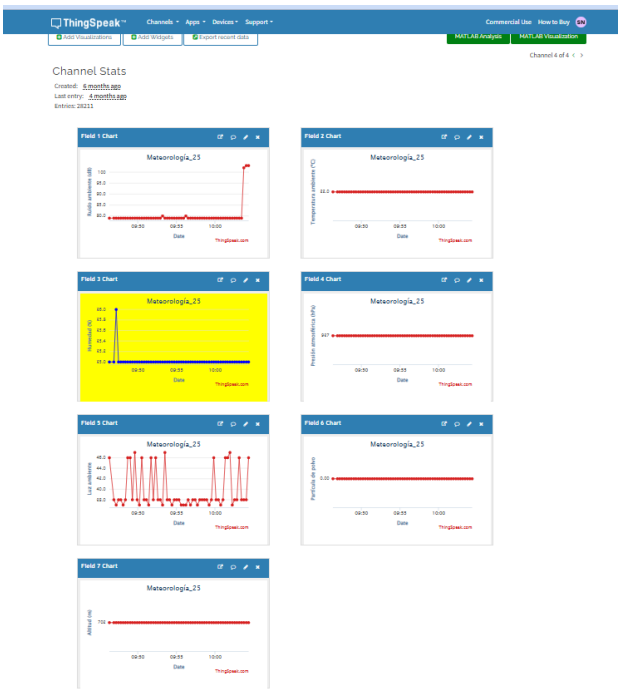
IES CELSO DÍAZ

GUIA PRÁCTICA ESTACIÓN METEOROLÓGICA

UTILIZACIÓN DE micro:bit e IoT KIT PARA EL MONTAJE, PROGRAMACIÓN, ALMACENAMIENTO Y VISUALIZACIÓN DE DATOS EN LA NUBE DE UNA ESTACIÓN METEOROLÓGICA.







1. MATERIALES Y DISPOSITIVOS UTILIZADOS.

- IoT Kit: ELECFREAKS®
- Micro:bit®
- Impresora 3D Bamboo Lab
- Filamento de impresion 3D PLA
- Ordenador con conexión a internet
- Software online:
    - Tinkercad | Panel principal,
    - Microsoft MakeCode for micro:bit
    - IoT Analytics - ThingSpeak Internet of Things

2. INTRODUCCIÓN.

En esta actividad, los alumnos desarrollarán una estación meteorológica inteligente utilizando tecnologías de diseño e impresión 3D, programación de sistemas embebidos e Internet de las Cosas (IoT). El proyecto integra distintas competencias relacionadas con el diseño digital, la fabricación aditiva, la programación y el tratamiento de datos en la nube.

El reto consiste en diseñar mediante Tinkercad un soporte personalizado capaz de alojar los diferentes componentes electrónicos del kit ELECFREAKS Smart IoT. Una vez finalizado el diseño, se procederá a su fabricación mediante impresión 3D y a la comprobación del correcto encaje y disposición de todos los elementos que forman parte de la estación meteorológica.

Posteriormente, los alumnos realizarán el montaje físico de los componentes electrónicos y programarán el sistema utilizando Microsoft MakeCode para micro:bit. La estación será capaz de medir diferentes parámetros ambientales como temperatura, humedad, presión atmosférica, intensidad luminosa, nivel de ruido, altitud y calidad del aire. También se podrá integrar un pluviómetro y medir el nivel de humedad del suelo.

Finalmente, se configurará una cuenta en la plataforma ThingSpeak para almacenar y visualizar los datos recogidos por los sensores. De esta forma, los estudiantes podrán analizar

la información obtenida en tiempo real mediante gráficas accesibles desde cualquier dispositivo con conexión a Internet.

3. OBJETIVOS PRINCIPALES DEL PROYECTO:

- Diseñar piezas funcionales mediante herramientas de modelado 3D.
- Fabricar prototipos utilizando tecnologías de impresión 3D.
- Integrar sensores y dispositivos electrónicos en un proyecto real.
- Desarrollar programas mediante programación por bloques con MakeCode.
- Utilizar plataformas IoT para la adquisición y almacenamiento de datos.
- Analizar y visualizar información obtenida a partir de sensores ambientales.
- Aplicar conocimientos de tecnología, programación y comunicación digital para resolver problemas reales.

Al finalizar el proyecto, los alumnos habrán desarrollado una estación meteorológica completamente funcional capaz de adquirir datos ambientales, transmitirlos a Internet y representarlos gráficamente para su posterior análisis.

4. MISIÓN: DISEÑAR, FABRICAR Y PONER EN FUNCIONAMIENTO UNA ESTACIÓN METEOROLÓGICA IoT.

El objetivo de esta actividad es desarrollar una estación meteorológica inteligente capaz de medir diferentes parámetros ambientales y publicar la información obtenida en una plataforma de almacenamiento y visualización de datos en la nube.

Para ello, los alumnos deberán diseñar mediante Tinkercad una estructura o soporte que permita alojar de forma ordenada y segura los diferentes componentes del kit ELECFREAKS Smart IoT. El diseño deberá tener en cuenta tanto aspectos funcionales como estéticos, garantizando el correcto cableado y acceso a los sensores y dispositivos electrónicos que forman parte del sistema.

Una vez finalizada la fase de diseño, se procederá a la fabricación del soporte mediante impresión 3D utilizando filamento PLA. Tras la impresión, los alumnos comprobarán el ajuste

y encaje de todos los componentes, realizando las modificaciones necesarias hasta obtener un montaje adecuado.

Posteriormente, se llevará a cabo el ensamblaje de la estación meteorológica y la programación de la placa micro mediante Microsoft MakeCode. El sistema deberá ser capaz de adquirir datos procedentes de los distintos sensores integrados en el kit, tales como temperatura, humedad relativa, presión atmosférica, intensidad luminosa, nivel de ruido o calidad del aire.

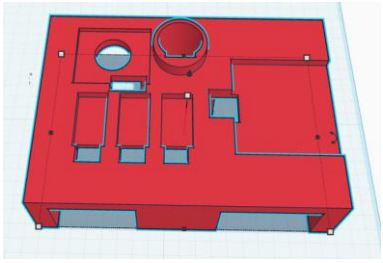
Finalmente, se configurará una cuenta en la plataforma ThingSpeak para permitir el almacenamiento de los datos recogidos y su representación gráfica en tiempo real. Los alumnos verificarán el correcto funcionamiento de la estación mediante la observación de los valores registrados y el análisis de las gráficas generadas.

El resultado final será una estación meteorológica completamente operativa que combine diseño 3D, fabricación digital, electrónica, programación e Internet de las Cosas en un único proyecto tecnológico integrado.

Los pasos para realizar la estación meteorológica han sido los siguientes:

1º. Diseño del soporte mediante Tinkercad.

a. Creación de una clase virtual en la plataforma Tinkercad por parte del profesor.
b. Creación de una actividad específica para el proyecto de estación meteorológica y asignación de la misma al alumnado.
c. Acceso de los alumnos a la actividad mediante el código proporcionado por el profesor.
d. Diseño individual del soporte de la estación meteorológica teniendo en cuenta las dimensiones y características de los componentes electrónicos del kit ELECFREAKS Smart IoT.
e. Supervisión y revisión de los diseños directamente desde la plataforma Tinkercad, permitiendo al profesor acceder en tiempo real al trabajo realizado por cada alumno.
f. Realización de las correcciones necesarias hasta obtener un diseño funcional que garantice el correcto alojamiento de todos los elementos de la estación.



2º. Fabricación mediante impresión 3D.

a. Descarga del modelo en formato STL.
b. Preparación del archivo para impresión mediante software laminador.
c. Impresión del soporte utilizando filamento PLA.
d. Comprobación de dimensiones y ajuste de los componentes electrónicos sobre la pieza impresa.
e. Realización de modificaciones y reimpresión si fuese necesario.

3º. Montaje y cableado de la estación meteorológica.

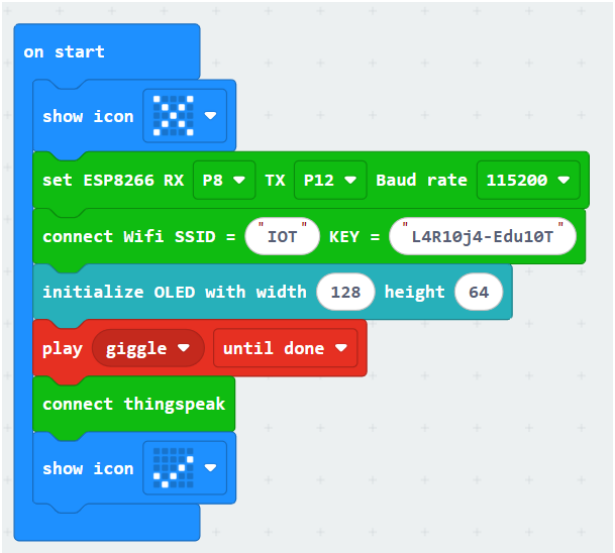
a. Instalación de los sensores y dispositivos electrónicos sobre el soporte impreso.
b. Conexionado de los diferentes elementos del kit ELECFREAKS Smart IoT.
c. Verificación del correcto montaje mecánico y eléctrico de la estación.

4º. Configuración de la plataforma ThingSpeak.

a. Acceso a la plataforma ThingSpeak y creación de una cuenta de usuario.
b. Creación de un nuevo canal destinado al almacenamiento de los datos de la estación meteorológica.
c. Configuración de los diferentes campos asociados a los parámetros que serán medidos por los sensores.
d. Obtención y configuración de las claves API necesarias para el envío de información desde la estación meteorológica.
e. Comprobación del correcto funcionamiento del canal mediante el análisis de los datos recibidos.

5º. Programación de la estación meteorológica mediante MakeCode.

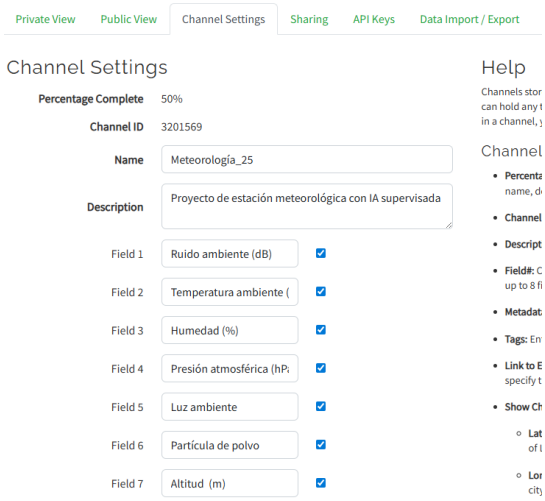
a. Acceso al entorno de programación Microsoft MakeCode para micro:Bit.
b. Incorporación de las extensiones necesarias para el funcionamiento del kit ELECFREAKS Smart IoT.
c. Programación de la lectura de los diferentes sensores de la estación meteorológica.
d. Programación de la visualización local de los datos mediante la pantalla OLED incorporada al sistema.
e. Programación del envío periódico de los datos recogidos a la plataforma ThingSpeak utilizando la conexión WiFi del dispositivo.
f. Transferencia del programa a la placa micro y puesta en funcionamiento de la estación.

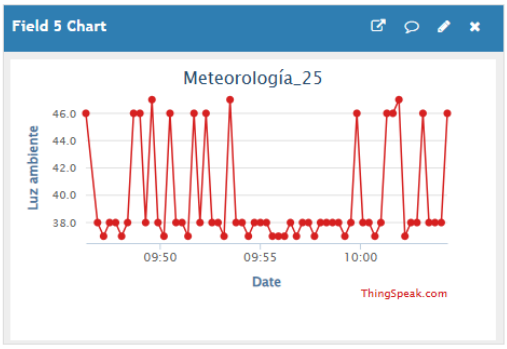

on start
show icon
set ESP8266 RX P8 TX P12 Baud rate 115200
connect Wifi SSID = "IOT" KEY = "L4R10j4-Edu10T"
initialize OLED with width 128 height 64
play giggle until done
connect thingspeak
show icon


forever
clear OLED display
show (without newline) string "Luz:"
show number value of light intensity(0~100) at pin P4
insert newline
show (without newline) string "Temp:"
show number value of BME280 temperature(°C)
insert newline
show (without newline) string "Hum:"
show number value of BME280 humidity(0~100)
insert newline
show (without newline) string "Aire:"
show number value of dust(µg/m³) at LED P13 out P1
pause (ms) 2000
clear OLED display
insert newline
show (without newline) string "Pres:"
show number value of BME280 pressure(hPa)
insert newline
show (without newline) string "Altitud:"
show number value of BME280 altitude(M)
insert newline
show (without newline) string "Ruido:"
show number value of noise(dB) at pin P3
pause (ms) 1000
set data to send ThingSpeak
Write API key = "WTMSIT6GGVE9RSOV"
Field 1 = value of noise(dB) at pin P3
Field 2 = value of BME280 temperature(°C)
Field 3 = value of BME280 humidity(0~100)
Field 4 = value of BME280 pressure(hPa)
Field 5 = value of light intensity(0~100) at pin P4
Field 6 = value of dust(µg/m³) at LED P13 out P1
Field 7 = value of BME280 altitude(M)
Upload data to ThingSpeak
pause (ms) 5000

6º. Comprobación y análisis de resultados.

a. Verificación del correcto funcionamiento de todos los sensores integrados en la estación meteorológica.
b. Comprobación de la recepción de datos en la plataforma ThingSpeak.
c. Visualización de los datos mediante gráficas en tiempo real.
d. Comparación de los valores obtenidos con las condiciones ambientales del entorno.
e. Detección y corrección de posibles errores de montaje, configuración o programación.
f. Validación final del sistema y comprobación de su funcionamiento autónomo.





5. RESULTADO FINAL:

Al finalizar la actividad, los alumnos dispondrán de una estación meteorológica completamente operativa, diseñada y fabricada por ellos mismos, capaz de adquirir datos ambientales mediante sensores, mostrarlos localmente y transmitirlos a una plataforma en la nube para su almacenamiento y visualización en tiempo real. El proyecto integra conocimientos de diseño 3D, fabricación digital, electrónica, programación e Internet de las Cosas, permitiendo desarrollar una solución tecnológica completa a partir de un problema real.