



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño



*BBC Micro:Bit. Introducción a la
mecatrónica en estudios preuniversitarios.*

MEMORIA PRESENTADA POR:

Juan Alberto Marí Alegre

Máster Universitario en Ingeniería Mecatrónica

DIRECTOR:

Juan José Serrano Martín

Septiembre 2017

AGRADECIMIENTOS

A mi tutor Juan José Serrano por el apoyo prestado en este proyecto.

A mi familia y en especial a mi mujer y mi hija por su infinita paciencia.

ÍNDICE GENERAL

I. MEMORIA.....	1
Índice de la Memoria	3
1. Introducción	5
2. Estado del arte	7
3. Comparación entre microcontroladores libres	9
3.1. Raspberry PI	9
3.2. Lego Mindstorms.....	10
3.3. Arduino.....	11
3.4. BBC Micro:Bit	12
3.5. Conclusiones.....	13
4. Microcontrolador Micro:Bit	15
4.1. Descripción del microcontrolador.....	15
4.2. Programación del microcontrolador	17
4.2.1. Editor de bloques JavaScript	17
4.2.2. MicroPython.....	19
4.2.3. Aplicación para los sistemas Android e iOS	20
5. Diseño de la estructura	23
5.1. Componentes	23
5.2. Estructura	24
5.3. Proceso de fabricación	24
6. Código programado.....	27
6.1. Editor de bloques JavaScript	27
6.2. Micropython.....	31
7. Conclusiones.....	33
Bibliografía	35
Índice de figuras	37
II. PRESUPUESTO	39
Índice del Presupuesto.....	41
1. Introducción	43

2. Costes de desarrollo del proyecto.....	45
2.1. Costes de personal	45
2.2. Costes de software y material.....	45
3. Costes de implantación	47
4. Coste total del proyecto	49
III. PLIEGO DE CONDICIONES.....	51
Índice del Pliego de Condiciones.....	53
1. Condiciones Generales.....	55
1.1. Objeto del proyecto y lugar de emplazamiento	55
1.2. Personal.....	55
1.3. Responsabilidades.....	55
1.4. Otras condiciones no especificadas en este pliego.....	55
2. Cláusulas administrativas	57
2.1. Documentación	57
3. Prescripciones técnicas y particulares	59
3.1. Especificaciones de elementos empleados en el proyecto.....	59
3.2. Especificaciones para la implantación del sistema	59
3.3. Especificaciones de la ejecución	59
IV. ANEXOS	61
Índice de Anexos	63
ANEXO I: Programación con bloques JavaScript	65
ANEXO II: Programación con MicroPython	69

I. MEMORIA

Índice de la Memoria

1. Introducción	5
2. Estado del arte	7
3. Comparación entre microcontroladores libres	9
3.1. Raspberry PI	9
3.2. Lego Mindstorms.....	10
3.3. Arduino.....	11
3.4. BBC Micro:Bit	12
3.5. Conclusiones.....	13
4. Microcontrolador Micro:Bit	15
4.1. Descripción del microcontrolador.....	15
4.2. Programación del microcontrolador	17
4.2.1. Editor de bloques JavaScript	17
4.2.2. MicroPython.....	19
4.2.3. Aplicación para los sistemas Android e iOS	20
5. Diseño de la estructura	23
5.1. Componentes	23
5.2. Estructura	24
5.3. Proceso de fabricación	24
6. Código programado.....	27
6.1. Editor de bloques JavaScript	27
6.2. Micropython.....	31
7. Conclusiones.....	33
Bibliografía	35
Índice de figuras	37

1. Introducción

Idear soluciones para algunas de las necesidades cotidianas en el día a día fue uno de los motivos por los cuales decidí estudiar Ingeniería. Creo firmemente que tenemos la obligación moral de tratar de mejorar la sociedad actual a través del conocimiento, al igual que otras personas consiguieron antes crear dispositivos que han conseguido facilitar nuestra vida cotidiana. En ese sentido, considero que una de las vías que tengo para, en la medida de mis posibilidades, ayudar a la sociedad en esta línea es transmitiendo mi pasión y parte de mis conocimientos a las futuras generaciones, para tratar de conseguir que el mundo sea un sitio mejor.

Así, he decidido orientar este proyecto a hacer visible entre los más jóvenes el mundo de la Mecatrónica e introducirlos en el ámbito de la Ingeniería de una manera amena, divertida y adaptada a su edad y conocimientos.

El presente proyecto se estructura de la siguiente manera:

- Primero haremos un repaso por el mundo de los microcontroladores en el Reino Unido, país de procedencia del Micro:bit.

- Seguidamente revisaremos los diferentes microcontroladores libres que existen en el mercado, comparando su hardware así como las ventajas y desventajas de cada dispositivo para explicar el porqué de nuestra elección en el microcontrolador: BBC Micro:bit.

- Después realizaremos una descripción del BBC Micro:bit, además de abordar todas las posibilidades en cuanto a lenguaje de programación se refiere. También comentaremos el control mediante la aplicación de los sistemas operativos iOS y Android.

- A continuación, definiremos la estructura mecatrónica seleccionada para este proyecto, así como los componentes necesarios para su construcción.

- Para finalizar, veremos el código de programación para el control, tanto en un lenguaje más básico, como son los bloques de JavaScript, así como otro más avanzado, el MicroPython.

2. Estado del arte

A principios de 1980, la BBC (British Broadcasting Corporation) inició lo que se conoce como el *BBC Computer Literacy Project* (Proyecto de conocimiento en Informática de la BBC). La idea del proyecto era dar a conocer un microcomputador capaz de realizar varias tareas a través en una serie de televisión llamada *The computer programme* (El programa del Computador). La lista de temas incluía programación, gráficos, música, teletexto, controlar hardware externo, inteligencia artificial, etc. Para ello elaboró una especificación bastante ambiciosa y solicitó ofertas a los fabricantes británicos de computadoras.

El equipo de la empresa *Acorn Computers* había estado trabajando en una mejora de su ordenador microcomputador Atom, que todavía era un prototipo. El equipo de Acorn, que dependía en su mayoría de estudiantes de la universidad de Cambridge, trabajó duro para conseguir un modelo funcional que presentar a la BBC. El *Acorn Proton*, modelo mostrado, no solamente era la única máquina que cumplía con la especificación de la BBC, sino que la excedió en casi cada campo.

La máquina fue lanzada a finales de 1981. Su éxito en el Reino Unido fue debido en gran parte a su aceptación como computadora “educativa”. La gran mayoría de escuelas británica usaron los BBC micro, como fueron bautizados, para enseñar habilidades en conocimientos de informática y tecnologías de la información.

La empresa Acorn con el “Beep”, como pronto comenzó a ser conocido por sus usuarios, llegó a vender más de 1 millón de BBC Micros.

En 2005, gracias a su capacidad de expansión disponible y a las funciones I/O (input/output), todavía hay numerosas BBCs en uso, y una comunidad retroinformática de usuarios dedicados a encontrar nuevas cosas que hacer con el viejo hardware.

En 2006 se ideó el proyecto “*Raspberry PI*” y en 2009 fue creada la Fundación Raspberry PI en Caldecote, (South Cambridgeshire, Reino Unido), como una asociación caritativa regulada por la Comisión de Caridad de Inglaterra y Gales.

Dicha fundación estaba formada por un conglomerado de 25 empresas y entidades públicas en diferentes escalones de inversión, entre los que figuran ARM, Google, Oracle, la agencia espacial inglesa, Samsung o la Universidad de Cambridge

La fundación nació con un objetivo en mente: desarrollar el uso y entendimiento de los ordenadores en los niños. La idea era conseguir ordenadores portables y muy baratos que permitieran a los niños usarlos sin miedo, abriendo su mentalidad y educándolos en la ética del “ábrelo y mira cómo funciona”. David Braven, el ideólogo del proyecto, afirmaba que su objetivo

era que los niños pudieran llegar a entender el funcionamiento básico de un ordenador de forma divertida y fueran ellos mismos los que desarrollaran y ampliaran dispositivos.

El cofundador, Eben Upton, se puso en contacto con un grupo de profesores, académicos y entusiastas de la informática para crear un ordenador con la intención de animar a los niños a aprender, como lo hizo en 1981 en ordenador Acron BBC Micro.

El primer prototipo basado en ARM fue montado en un paquete del mismo tamaño que una memoria USB. En agosto de 2011 se fabricaron 50 placas del modelo inicial (modelo A). En diciembre de ese mismo año, 25 placas del modelo B fueron ensambladas y probadas.

Debido al anticipado anuncio de puesta a la venta, a finales de febrero de 2012 la fundación sufrió un colapso en sus servidores web debido a la gente interesada en la compra de la placa. Las primeras ventas comenzaron el 29 de febrero de 2012 (Modelo B). En los 6 meses siguientes llegarían a vender 500.000 unidades.

En ese mismo año, 2012, como parte del programa *BBC Computer Literacy Programme* (Proyecto de conocimiento en Informática de la BBC) se planeó un nuevo dispositivo para la enseñanza llamado BBC Micro:Bit. Se programó la salida de este dispositivo para julio de 2015, así que la BBC se comprometió con otros 29 socios -como por ejemplo Windows, ARM, Samsung o la universidad de Lancaster-, para la fabricación, diseño y distribución del dispositivo.

Micro:bit fue diseñado para animar a los niños a programar y construir nuevas cosas, de modo que no fueran simples consumidores de dispositivos. Se había diseñado para trabajar con otros sistemas como Raspberry PI o el legado de la BBC, el dispositivo BBC Micro para la enseñanza. La BBC planeó dar un dispositivo a cada niño de 11 años para la fecha de lanzamiento, entregando los dispositivos con un mes de antelación a los profesores para formarse y poder impartir la enseñanza con dicho dispositivo. Sin embargo, los problemas retrasaron el lanzamiento hasta el 22 de marzo de 2016.

Esta gran apuesta de la BBC se materializó con el despliegue de un millón de placas BBC Micro:bit durante el otoño de 2016. Se eligió a los niños de 11 años debido a que era un segmento de edad que, con mayor probabilidad, iba a seguir trabajando con la placa en su casa durante su tiempo libre. El director ejecutivo, Zach Shelby, apuntó que deseaba un despliegue internacional del dispositivo. “Nuestro objetivo es llegar a 100 millones de personas con Micro:bit y cambiar sus vidas con la tecnología”, dijo Shelby.

La organización hizo posible que el dispositivo se comercializara en 2016 en toda Europa y para el presente 2017 planea apuntar hacia América del Norte y China, coincidiendo con una actualización del hardware. “Le agregaremos más potencia informática y buscaremos nuevos tipos de sensores”, dijo Shelby.

3. Comparación entre microcontroladores libres

3.1. Raspberry PI

Raspberry PI está definida como SCB (Single Board Computer). Es un microcontrolador de bajo coste. Se podría decir que es un ordenador reducido. El concepto es de un ordenador al cual le han quitado todos los accesorios, teniendo un funcionamiento básico.

Es necesario instalar un sistema operativo. Normalmente se suele decantar por una versión de Linux. Para programar se puede realizar en python o en C. En ambos casos, se suele utilizar el mini editor “nano”.

Existen varias generaciones de Raspberry PI (en concreto 3), así como varios modelos. Es normal, puesto que ya llevan varios años en funcionamiento y desarrollo.

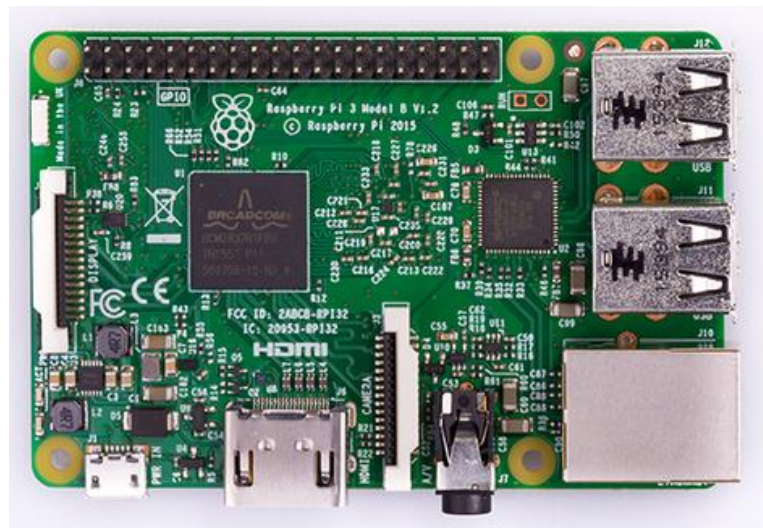


Figura 1. Raspberry PI 3

El diseño de la Raspberry PI (3ª generación) model B que podemos ver en la Figura 1 incluye:

- 1.2 GHz 64-bit quad-core ARMv8 CPU.
- 1Gb de RAM.
- VideoCore IV 3D graphics core.
- Ethernet port.
- Combined 3.5mm audio Jack and composite video.
- Full HDMI port.
- 4 USB2 ports.
- 40 GPIO pins.
- MicroSD card slot

A finales de 2015, Raspberry creó su modelo más ambicioso hasta el momento: la Raspberry Zero. Un modelo muy competitivo tanto por sus características y su reducido tamaño, como por su atractivo precio de 5,50€ + gastos de envío. No obstante, el exceso de demanda hace que sea complicado conseguir un microcontrolador de estos. Podemos ver el modelo en la Figura 2.



Figura 2. Raspberry Pi zero

3.2. Lego Mindstorms

Este tipo de dispositivo se basa en la famosa marca de juguetes LEGO. A sus montajes de piezas le han colocado un microcontrolador y actuadores para que los juguetes interactúen y se muevan. Podemos ver dos ejemplos en la Figura 3.

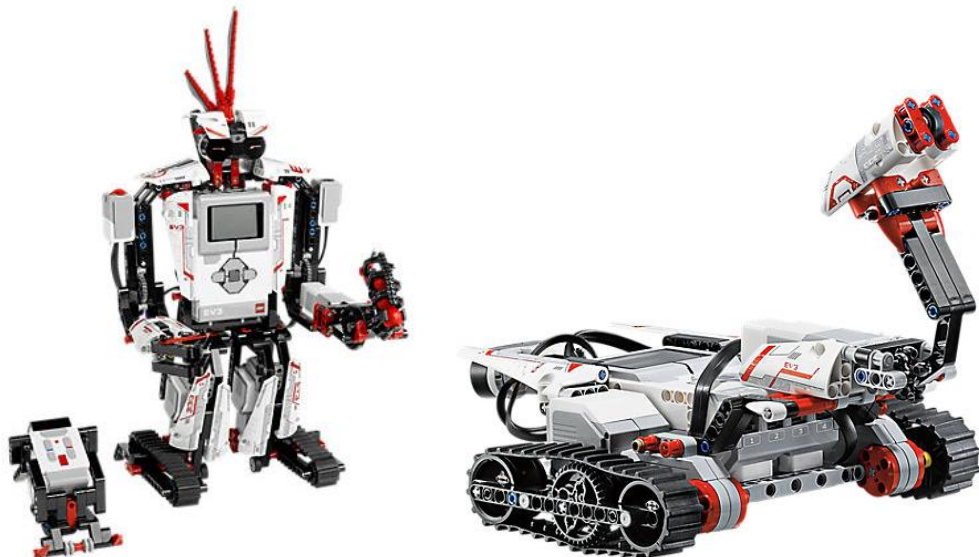


Figura 3. Lego Mindstorms

Los últimos modelos utilizan un procesador ARM9, tienen un puerto USB para proporcionar funciones wifi para su conexión a internet, además de disponer de lector de tarjetas Micro SD, botones retroiluminados y 4 puertos de motor.

Han desarrollado una aplicación para iOS y Android, de manera que se puede diseñar, programar y controlar. La programación se realiza mediante una interfaz en iconos de fácil uso, como se puede observar en la Figura 4.

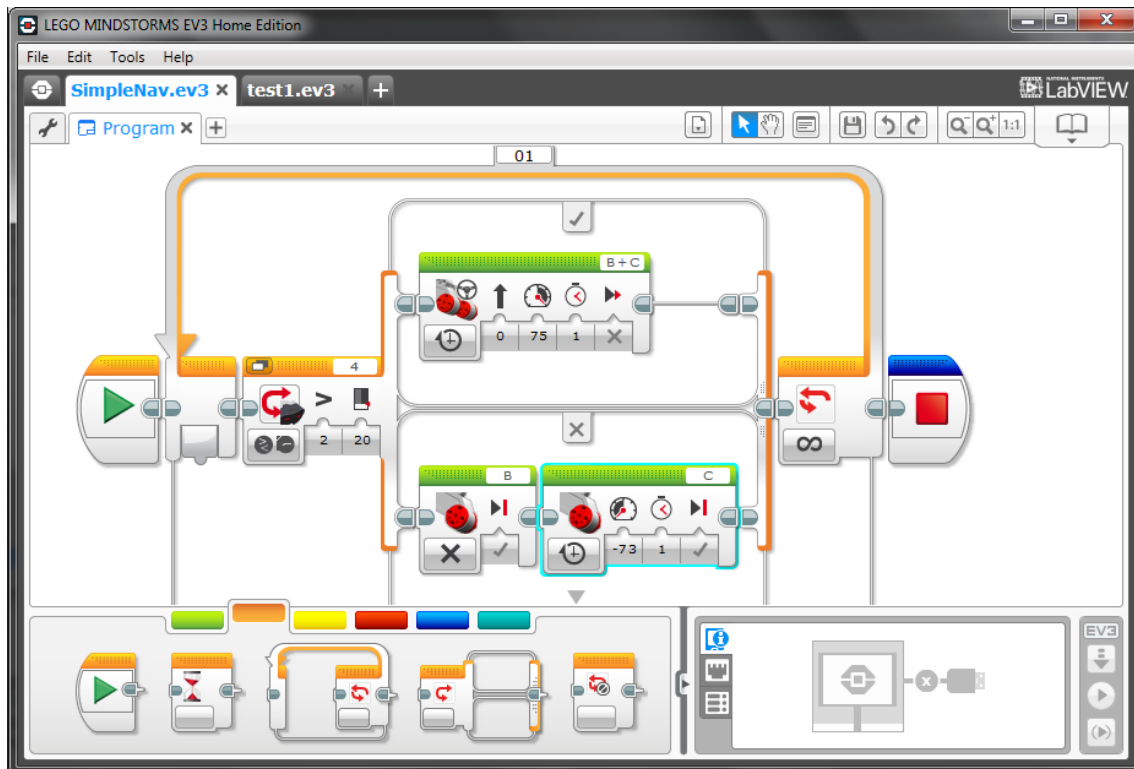


Figura 4. Entorno de programación de Lego Mindstorms

Como se puede observar, la programación es muy intuitiva, pero queda muy lejos de ser similar a un lenguaje de programación corriente.

Aunque el precio es elevado, es algo diferente, ya que es como darle vida a los juguetes de toda la vida. Los niños pueden construir fácilmente alguna forma y animarla mediante la programación del módulo.

3.3. Arduino

Arduino se inició en el año 2005 como un proyecto para estudiantes en el Instituto IVREA, en Ivrea (Italia). Este dispositivo nunca se creó como una idea de negocio, sino que se planteó como una necesidad para los estudiantes, ya que en la época en que se ideó, los alumnos pagaban aproximadamente unos 100 dólares estadounidenses por el microcontrolador BASIC Stamp.

Lo bueno de este microcontrolador es que, como está creado específicamente para los estudiantes, es de hardware y código abierto y resulta sencillo encontrar en la red información sobre cómo funciona y cómo programarlo.

Existen varios microcontroladores bajo esta marca, desde micros de nivel básico, pasando por específicos para IoT, otros más de tipo wearable e incluso para fabricar impresoras 3D. Podemos destacar entre los más potentes el modelo Arduino Due, que tiene las siguientes características:

- 84 MHz 32-bit AT91SAM3X8E CPU.
- 54 I/O Pins (12 PWM).
- 4 puertos UART.
- Comunicación CAN.
- 2 conversores A/D

Podemos ver una imagen del microcontrolador en la Figura 5.

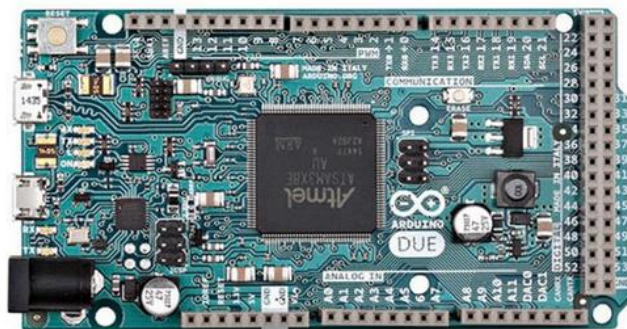


Figura 5. Microcontrolador Arduino DUE

Este microcontrolador de más básico que el Raspberry, puesto que no es necesario instalar ningún sistema operativo. Simplemente nos descargamos el IDLE, escribimos el código y lo introducimos en la tarjeta.

3.4. BBC Micro:Bit

Micro:bit es una pequeña placa de desarrollo de open hardware que permite ejecutar código en ella y tener acceso a todo el hardware para poder interactuar con ella.

Este pequeño dispositivo incorpora una matriz de 5x5 leds que pueden mostrar mensajes de una manera relativamente sencilla. También tiene dos botones programables que pueden utilizarse para interactuar con el microcontrolador. Incluye acelerómetro, brújula, una antena bluetooth de bajo consumo de energía, sensor de temperatura e incluso permite realizar transmisiones de comunicación vía radio con una antena de 2,4GHz, de manera que se puede comunicar inalámbricamente con otras tarjetas Micro:bit

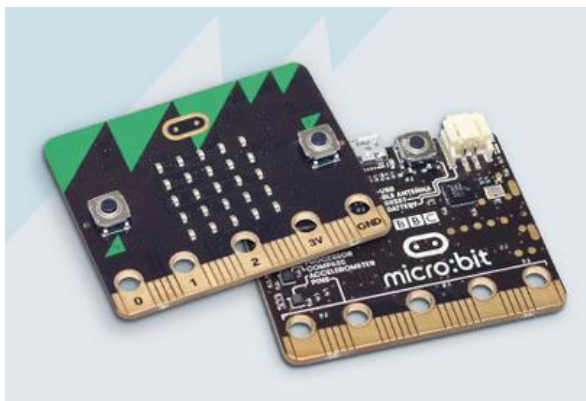


Figura 6. Placa Micro:bit

Podemos ver una imagen del microcontrolador en la Figura 6. Sus características son las que se detallan a continuación:

- 16 MHz 32 bits ARM Cortex M0 CPU
- 2 botones programables.
- 19 GPIO I/O de las cuales 6 cuentan con convertor A/D (10 bits), se pueden programar 3 señales PWM simultáneamente
- Además está preparado para trabajar con protocolos de comunicación I2C y SPI.

El BBC Micro:bit se puede programar tanto con un lenguaje de bloques JavaScript, como con Java o con Micropython, que es un lenguaje similar al python pero diseñado para funcionar en pequeños microcontroladores como el BBC Micro:bit. Se puede programar con una aplicación a través de los sistemas Android e iOS.

3.5. Conclusiones

Para concluir vamos a realizar una comparación entre todos los microcontroladores mostrados anteriormente.

-Los Lego son menos genéricos, ya que utilizan sus propios sensores y su propia programación. Es decir, sólo muestran los valores que se pueden modificar o los movimientos que quieres realizar. En este caso no se puede aplicar cualquier actuador genérico, ya que no se puede generar la señal que se desees. No obstante lo anteriormente comentado, son unos microcontroladores educativos muy extendidos entre los colegios, ya que se organizan competiciones entre diferentes centros. De esta forma, motivan de una manera extra a los alumnos para poder realizar mejores diseños y programaciones.

-En cuanto a las Raspberry pi, son los más potentes pero por contra también los más complejos de programar. Primero hay que instalar un sistema operativo y realizar la programación a través del mismo con alguna aplicación. Esto no resulta nada intuitivo y a veces nos podemos encontrar con dificultades añadidas que a veces son complicadas de solucionar.

Con este tipo de microcontroladores es necesario tener cierto nivel de programación para poder realizar diseños y controles más atractivos. Existe mucha información en la red sobre cómo realizar las instalaciones necesarias, pero por este mismo motivo, se puede tender a copiar programación sin llegar a entender lo que se está realizando realmente.

-Las placas Arduino son sencillas de programar, ya que alguien que tenga conocimientos básicos de programación podría realizarlo. Simplemente con el IDLE de la propia marca se puede seleccionar el tipo de placa que se está utilizando y en qué puerto está conectada y después de escribir el código y cargarlo, ya puede funcionar.

En este caso, también vamos a poder encontrar mucha información en la red, al igual que para la Raspberry pi y también podemos tender a copiar programaciones y no llegar a entender lo que estamos realizando. Aunque también podemos encontrar proyectos muy diversos y divertidos.

-En el caso del BBC Micro:bit, nos encontramos con una placa muy novedosa y polivalente, ya que sus múltiples lenguajes de programación hace que lo pueda programar tanto un niño de 11 años como uno de 17, ya que se puede empezar con un nivel básico con la programación de los bloques y luego se puede programar con Micropython y empezar a iniciarse dentro del mundo de la programación de hardware, conociendo más las entradas y salidas, programando señales PWM, etc.

En el caso de estos tres últimos microcontroladores, tenemos la ventaja de que podemos introducir prácticamente cualquier tipo de sensor, pero también tenemos que tener conocimientos básicos de electrónica para saber qué estamos realizando. Es decir, tenemos que saber cómo funcionan los sensores para saber qué datos vamos a introducir al microcontrolador y, de esta manera, poder trabajar con ellos para conseguir nuestro objetivo.

De esta manera, se puede aprender mucho más, ya que también se necesita diferenciar entre qué tipo de señal se debe generar para poder manejar ciertos actuadores, como por ejemplo un servo, un motor DC, etc.

Debido a los múltiples lenguajes de programación del BBC Micro:bit, lo que permite disponer de una gran adaptación al nivel del programador, me parece una plataforma ideal para introducirnos en el ámbito de la Mecatrónica. Posteriormente veremos las ventajas que ofrece esta plataforma para poder pasar del sencillo lenguaje de programación basado en bloques a uno estructurado, propio de lenguajes de programación más avanzados.

Además, conforme se vaya aumentando el nivel de programación, podemos continuar utilizando este microcontrolador o incluso utilizar otros de los descritos en este proyecto empezando por Arduino y posteriormente realizar el salto a Raspberry pi.

4. Microcontrolador Micro:Bit

4.1. Descripción del microcontrolador

En la Figura 7 podemos ver unas imágenes del aspecto física que presenta la placa BBC Micro:bit, tanto la parte delantera como la trasera.

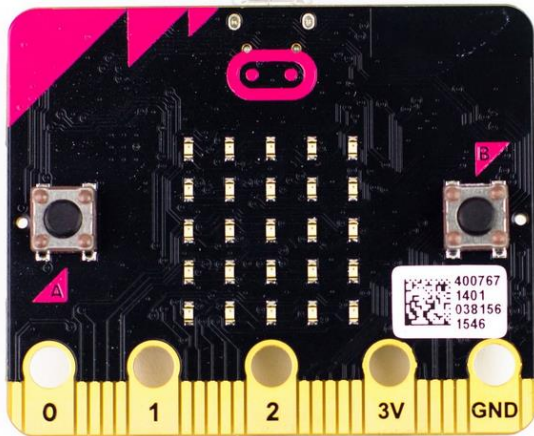


Figura 7a. Parte delantera del BBC Micro:bit



Figura 7b. Parte trasera del BBC Micro:bit

En la parte delantera, tal y como se puede ver en la Figura 7a, cuenta con una matriz de leds de 5x5 y dos botones nombrados como A y B. De esta forma, podemos interactuar con la placa.

En la parte trasera, tal y como se observa en la Figura 7b, la placa cuenta con un botón de reset, de manera que si tenemos cualquier tipo de problema podremos reiniciar de una sencilla manera nuestro dispositivo.

También cuenta con un puerto mini usb, desde el cual podremos conectar a nuestro PC para poder programar, como veremos más adelante, de una forma muy sencilla. También tenemos un conector para poder introducir alimentación mediante un portapilas 2xAA, aunque también dispone de un par de pines donde podemos soldar unos cables o un portapilas para una batería tipo CR2032. Además, como se puede observar en la Figura 7b, los elementos principales se encuentran marcados. De esta manera, el estudiante también puede empezar a familiarizarse con el hardware.

BBC Micro:bit dispone de 25 conexiones externas en el borde de la placa, a las cuales nos referimos como pins. Tiene 5 pins más grandes y otros 20 que son más pequeños, tal y como se puede ver en la Figura 7. Realizar la conexión con los pins más grandes es relativamente sencillo e incluso se pueden gastar pinzas de cocodrilo. Pero a la hora de conectar los pins pequeños, podemos tener algún problema. Por eso también existen conectores como el mostrado en la Figura 8. Se puede comprar ya soldados o listos para soldar. Precisamente, de esta última manera podemos conectar únicamente lo que necesitamos.



Figura 8. Micro:bit Edge connector

En la Figura 9 podemos ver una descripción gráfica de la configuración de todos los pines que tiene la placa BBC Micro:bit en su orden correspondiente.

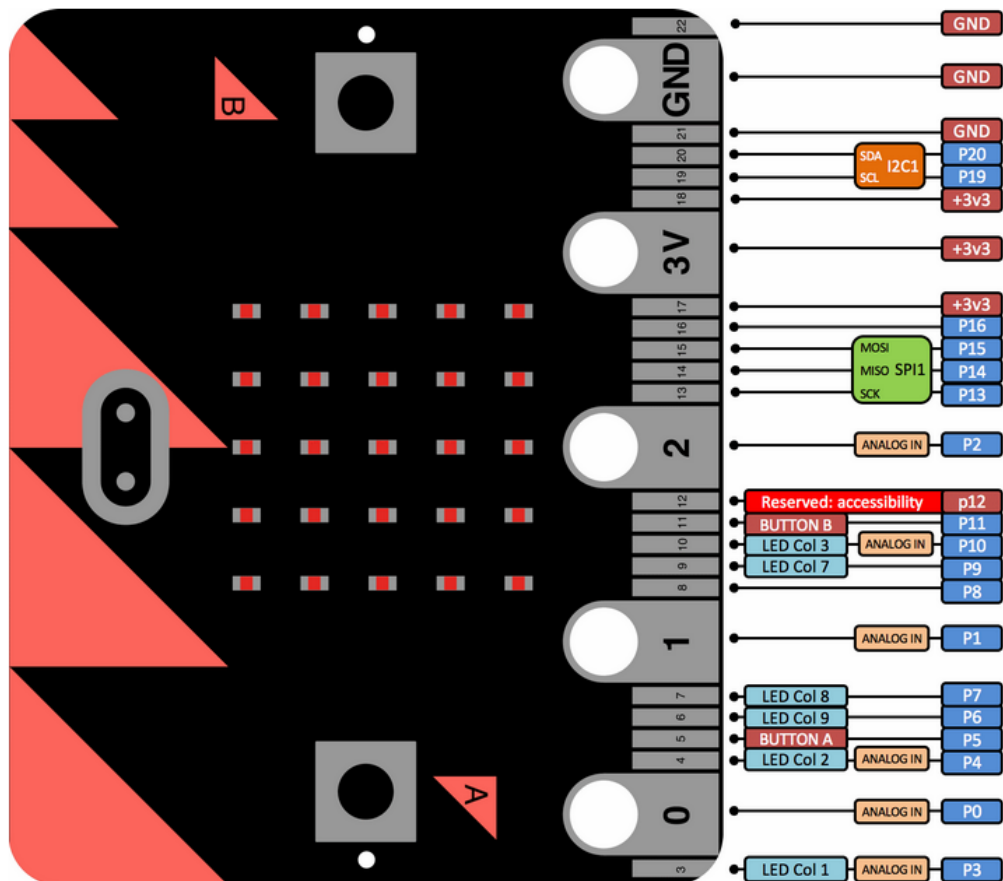


Figura 9. Descripción de los pines de BBC Micro:bit

4.2. Programación del microcontrolador

BBC Micro:bit se puede programar de diferentes maneras. Tenemos la opción de utilizar un editor de bloques JavaScript- Es un lenguaje fácil de entender por los más jóvenes (aproximadamente sobre los 11 años). Podemos elegir entre JavaScript de Code Kingdoms o el block editor de Microsoft. Se puede utilizar también un editor para principiantes llamado Touch Develop de Microsoft, donde tenemos precargados las funciones y variables. Esta manera de escribir el código consiste en seleccionar unas casillas con estructura de un lenguaje de programación avanzado.

Además, cabe la posibilidad de programarlo con Micropython. Es una ligera variación del lenguaje de programación python que existe para microcontroladores pequeños, de manera que resulta un poco más sencillo programar ciertas cosas. También existe la posibilidad de interactuar y programar mediante JavaScript con una aplicación que existe tanto para la plataforma Android como iOS. Se sincroniza el microcontrolador con el dispositivo mediante bluetooth y después de escribir el código se puede cargar directamente en la tarjeta, como veremos más adelante en el apartado correspondiente.

4.2.1. Editor de bloques JavaScript

Uno de sus principales atractivos es que dispone de un lenguaje muy intuitivo para los niños, por lo que resulta fácil de utilizar y entender para ellos. El idioma español todavía no está disponible para los tutoriales, pero en la actualidad muchos niños ya entienden el inglés y, si no es el caso, se les puede guiar, ya que las palabras utilizadas son muy descriptivas y fáciles de entender.



Figura 10. Ejemplo de programación de bloques con JavaScript

La configuración de la pantalla, como se puede ver en la Figura 10, es muy acertada. En la parte izquierda tenemos un BBC Micro:bit, en el cual podremos visualizar el código que hemos configurado en la parte derecha. De esta manera es muy sencillo probar el código que hemos realizado para ver si realmente hace lo que pretendíamos. Con este método los niños también pueden probar diferentes funciones, aunque no sepan muy bien qué significan. Tenemos la opción de ocultarlo si queremos tener más pantalla para crear el código y, cuando lo necesitemos, volver a visualizarlo para probar el código.

La aplicación web tiene la opción de poner la simulación de la placa en modo pantalla completa, de manera que veamos sólo el Micro:bit para poder interactuar mejor o, simplemente, para mostrar la programación sin necesidad de ver el código. Muy útil por si los niños desean enseñar a compañeros un programa sin tener que mostrar el código.

En la parte central de la pantalla podemos ver los grupos que tenemos de funciones para poder elegir. Una vez que pinchamos en el grupo deseado, se despliega un menú con todas las funciones para poder seleccionar la que deseemos.

Como se puede observar, tiene como una forma de puzle, de manera que si se necesita dar una variable u otra función, la continuación será diferente y el programa no deja enlazarlo. De esta manera se utiliza el método para evitar fallos y los niños puedan aprender la diferencia entre los diferentes tipos de inputs que se pueden dar.

En la parte superior aparece el logotipo de la placa y, si pinchamos con el ratón en él, nos direcciona a la página oficial de BBC Micro:bit. Seguidamente tenemos un botón que nos abre un explorador de archivos en el cual nos podemos encontrar los diferentes proyectos que hemos utilizado y programado. También podemos guardar nuestros códigos, por lo que es una opción muy útil.

Una vez que tenemos nuestro código finalizado, si lo publicamos, podemos obtener un enlace para poder compartirlo con quien deseemos y por el medio digital que queramos.

En la parte central superior vemos que aparecen dos botones. En uno pone “Blocks” y el otro “{} JavaScript”. De esta manera, podemos pasar del diagrama de bloques que hemos realizado a programación JavaScript, con lo que es posible realizar una transición de programación de bloques a un lenguaje más de programación.

Incluso si tenemos activado el modo JavaScript, bajo de la simulación de la placa se nos muestra un explorador donde podemos desplegar y visualizar el código interno para controlar el hardware de la placa. Podemos ver, por ejemplo, la programación de la configuración del módulo de radio para las conexiones con otras placas. El código también está comentado para una mejor comprensión, por lo que si somos muy curiosos podemos ver cómo se programa el micro completo. Por motivos obvios, se puede ver pero no modificar.

También cuenta con un botón de ayuda donde podemos encontrar desde las páginas de proyectos oficiales a la descripción de lo que hacen los bloques, la descripción del hardware, dónde comprar, etc.

Al lado de este botón aparece uno de herramientas donde podemos gestionar las propiedades, nombre, etc... de nuestros proyectos, o si deseamos añadir algún otro bloque de funciones o cambiar el idioma, entre otros.

En la parte superior derecha nos encontramos con el logo de Microsoft. Si pinchamos nos direcciona a la página de Microsoft Make code, donde podemos encontrar todas las diferentes plataforma que apoyan para programación en proyectos educativos como Minecraft (del propio Microsoft), tarjetas de Adafruit, Sparkfun y Chibitronics. Los microcontroladores que se utilizan en estas diferentes plataformas son todas salidas después del pionera BBC Micro:bit.

4.2.2. MicroPython

Python es un lenguaje de programación que es fácil de aprender y de programar y es muy popular debido a sus potentes funciones y su alta comprensión del código. Es decir, que es fácil de entender. MicroPython es una versión derivada específica para microcontroladores pequeños y es un software libre que puede ser utilizado por cualquier persona.

En este caso se puede programar directamente desde el editor existente en la web oficial del dispositivo igual que JavaScript. Existe también una versión beta para programar offline, aunque siempre podemos precargar el editor desde el sitio web cuando tengamos conexión y después sin conexión funcionará también.

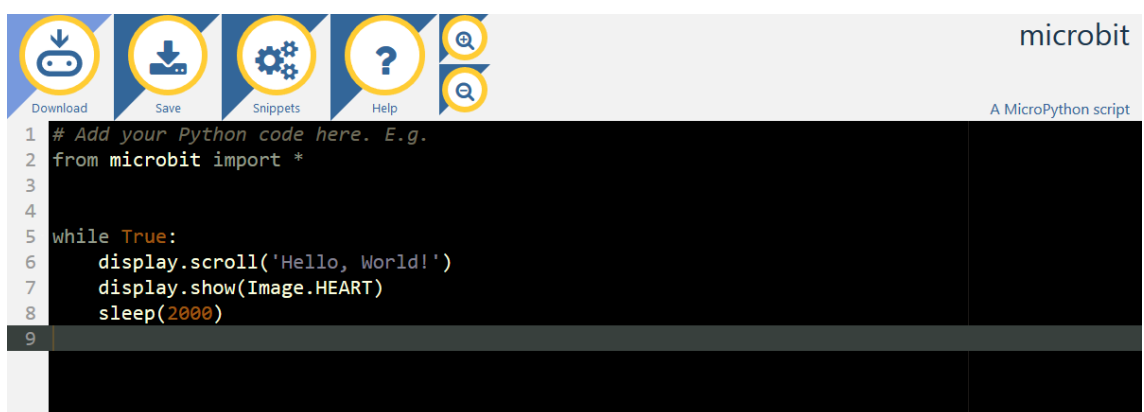


Figura 11. Vista del editor web para programar con microPython

En la Figura 11 podemos ver cómo es el editor web para programar BBC Micro:bit con MicroPython. Se trata de un diseño muy sencillo e intuitivo. La manera de programar es muy accesible: una vez tenemos listo el código, no tendremos más que darle a “Download” para descargarnos un archivo .hex. Una vez tengamos este archivo, conectamos el dispositivo y, si no

se abre una ventana del explorador de archivos, vamos a “Mi PC” donde aparecerá nuestra tarjeta como si fuera un dispositivo de almacenamiento. Hacemos doble click sobre el icono y se abrirá una ventana. Únicamente arrastramos el archivo .hex y soltamos. El microcontrolador cargará los datos y veremos que parpadea un led que se encuentra al lado del botón de reset. Cuando acaba deja de parpadear y arranca la programación.

También existe un programa que sube los archivos directamente al Micro:bit sin necesidad de arrastrar y soltar. Cuando detecta una descarga de un archivo .hex nuevo con el navegador, si tenemos conectado el microcontrolador lo carga automáticamente. Este programa es compatible con cualquier navegador.

Podemos ver en la Figura 11 que la mayor parte de la pantalla la ocupa el editor en sí, que será donde implementemos nuestro código de programación. En la parte superior del editor web tenemos la barra de herramientas, como en cualquier programa. En este caso tenemos 6 iconos, de los cuales vamos a describir sus funciones.

El primero con el que nos encontramos es el que pone “Download”. Sirve para descargar el archivo .hex una vez que tengamos la programación finalizada. Después nos encontramos con uno que pone “Save”. Si nos registramos, al igual que antes, también tendremos un espacio disponible para dejar nuestros programas y volver a cargarlos cuando queramos. Al igual que se pueden guardar también se pueden importar archivos programados con otro editor de python.

Seguidamente tenemos el botón de “Snippets” que son sugerencias de funciones de programación como los bucles if, for, etc. Otro icono de “Ayuda”, que nos direcciona a una página web donde podemos encontrar para que sirve cada botón, así como una introducción a la utilización del BBC Micro:bit con MicroPython.

Para finalizar, a la derecha del todo podemos ver dos iconos que sirven para aumentar y disminuir el tamaño del texto del editor.

4.2.3. Aplicación para los sistemas Android e iOS

La aplicación móvil está implementada por Samsung, que es una de las muchas marcas involucradas en este proyecto educativo, como hemos visto anteriormente. Permite enviar código al micro:bit utilizando la tecnología del bluetooth.

Como se puede observar en la Figura 12, la pantalla de la aplicación cuenta con cuatro botones, los cuales pasamos a describir a continuación:

El primero de arriba a la izquierda es el de “CONNECT”. Como su nombre indica, éste es el que nos lleva a la aplicación para poder emparejar el dispositivo con nuestro BBC Micro:bi. Es muy sencillo y seguro. Cuando se detectan ambos dispositivos, aparece un patrón en la matriz

de leds del microcontrolador, que después deberemos de introducir en la pantalla de la aplicación. Para poder realizar el emparejamiento, tenemos que tener en cuenta que el código que tenga en memoria el microcontrolador tiene que haber sido programado con JavaScript.

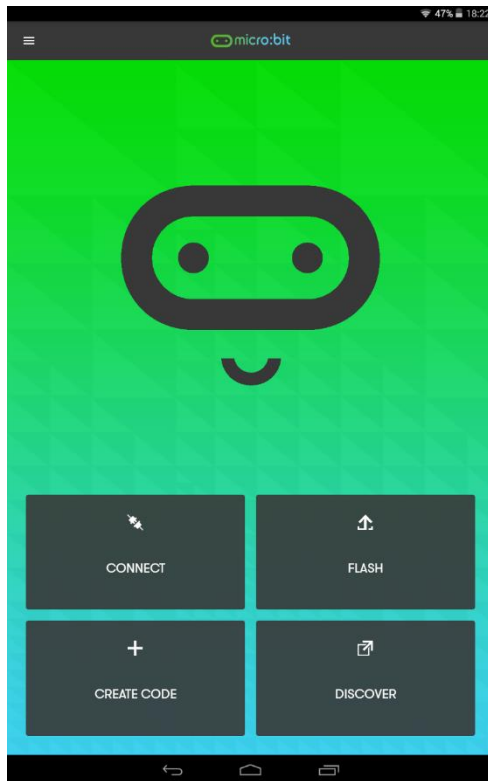


Figura 12. Aplicación para el S.O. Android

La siguiente con la que nos encontramos es “FLASH”. Aquí es donde accederemos para enviar el archivo .hex previamente codificado y descargado desde la web.

Otro de los botones con los que cuenta esta aplicación es el de “CREATE CODE”, que nos reenvía a la página web oficial al apartado de programación. De esta manera, podremos elegir el editor web para poder programarlo.

Para finalizar, el último botón es el que pone “DISCOVER”. Aquí nos redirecciona a la web oficial, pero en este caso al apartado de proyectos, donde primeramente deberemos seleccionar si somos profesores o simplemente queremos realizar proyectos por nuestra cuenta. Esto es debido que posteriormente puedes ver proyectos más relacionados con la educación o no.

Una de las noticias que se puede visualizar en esta página de proyectos es que, según una encuesta, el 70% de las niñas dice que quiere continuar sus estudios relacionados con ciencias computacionales.

5. Diseño de la estructura

Tal y como se puede ver en la Figura 13, en este caso me he decantado por realizar un diseño que es muy común entre los juguetes de cualquier niño.



Figura 13. Vehículo diseñado para la programación

La tecnología de las impresiones 3D se encuentra cada vez más presente en nuestra sociedad ya que cada vez existe más variedad y su precio es moderado.

La idea de este proyecto es que el vehículo pueda ser fabricado por los alumnos en su totalidad para contar con una motivación extra. Con esta premisa se ha diseñado mediante el software SolidWorks un vehículo que será fácilmente imprimible así como visualmente atractivo.

5.1. Componentes

Para introducir en la mecatrónica a los niños he decidido utilizar servomotores para la movilidad del vehículo, de esta manera no es dar tensión a un motor y ya está, sino que hay que introducirlos en el concepto de los que es una señal PWM y jugar con los tiempos de esta señal para poder moverlo.

Con este tipo de motores podremos hacer que se mueva hacia delante, hacia atrás, así como que pueda girar a la derecha y a la izquierda.

Para introducir el BBC Micro:bit utilizaremos un zócalo como el que vimos en la Figura 8, de manera que sea fácil de introducir y soltar. Como se puede programar con un Smartphone o Tablet a través de bluetooth, no será necesario estar desconectando y conectando la tarjeta. Además el cable para la comunicación se quedará visible por si queremos programarlo mediante pc.

Para la alimentación utilizaremos un portapilas de 4 pilas tipo LR03 size AAA. De esta manera, podremos trabajar con dos voltajes: 3V para alimentar la tarjeta y 6V para poder

alimentar a los servos. El acceso al portapilas será fácil para su sustitución en caso de tener que cambiarlas.

5.2. Estructura

El modelado se ha realizado de una manera sencilla e intuitiva, de forma que se pueda incluso personalizar por parte de una persona cualificada y que tenga un conocimiento básico sobre el software SolidWorks.



Figura 14. Piezas del vehículo

En la Figura 14 podemos ver las piezas del vehículo. En la imagen de la izquierda está la base preparada para montar los diferentes componentes, los servomotores, la ranura para insertar el zócalo del BBC Micro:Bit, y también los agujeros para posteriormente poder colocar la carrocería.

En la imagen de la derecha podemos ver todos los componentes comentados anteriormente ensamblados cada uno en su posición, incluyendo el portapilas que será el encargado de alimentar el vehículo.

5.3. Proceso de fabricación

Para la producción del vehículo se ha decidido utilizar el proceso de fabricación aditiva, más comúnmente conocida como impresión 3D.

Actualmente en el mercado existe una gran variedad de impresoras 3D, con posibilidad de imprimir en varios materiales, principalmente plásticos ABS y PLA. A su vez, éstos se presentan con diferentes formatos, como hilo, polvo, etc. Incluso existen impresoras con varias extrusoras en la punta de impresión para utilizar varios tipos de materiales diferentes en una misma impresión.

La principal diferencia entre los dos plásticos más utilizados para las impresiones 3D son los acabados. Cuando finaliza la impresión ambos tienen la misma textura, pero al ABS se le puede aplicar un tratamiento con acetona y la superficie se vuelve lisa. Con el PLA no tenemos esta opción y, para conseguir la misma superficie, se debería de colocar una capa de masilla y lijar.

Independientemente de la impresora que se utilice, así como del material, todo modelo necesita un pre-procesado para ver que la impresión será correcta y evitar malgastar material y tiempo. El formato que se necesita para una impresión 3D es STL. El SolidWorks da la opción de guardar directamente en dicho formato, como se ha realizado en este caso.

Dentro de las opciones de guardado en el SolidWorks de este formato, se pueden configurar ciertos parámetros como, por ejemplo, la tolerancia en el ángulo a la hora de realizar una circunferencia, las unidades de todas las opciones seleccionadas, o la codificación del formato (ASCII o binario), etc.

Se han tenido en cuenta, para elegir estos parámetros, la precisión de la impresora 3D y se ha elegido una calidad alta, ya que las opciones disponibles para imprimir soportan este formato.

Una vez se tiene el archivo STL, se tiene que procesar el archivo con un programa fileteador. Es decir, un programa que realiza los cortes de la figura y realiza una simulación de las capas que vamos a tener que imprimir para conseguir el conjunto final. Existen varios programas para realizar esto como Slic3r, KISSlicer, Skeinforge. En este caso ha utilizado el Cura, un programa gratuito y además con resultados muy buenos. Una vez se ha procesado el archivo por todos estos programas, ya está listo para imprimir, así que se debe copiar el archivo en una memoria USB y transferir a la impresora para finalizar el proceso.

En este caso hemos impreso con ABS la base y la carrocería, que son las piezas clave del vehículo. Al final se ha optado por no darle acabado con acetona para que se vea bien la impresión 3D. La tapa de las pilas y la estructura que las sujeta se ha impreso en PLA. De este modo, también hemos seleccionado dos tonos de colores para jugar con los contrastes.

Para mejorar la impresión hemos realizado la carrocería en dos partes y la unión que es visible en el centro de la misma se ha reforzado con unos soportes y se ha pegado con una resina epoxi de alta resistencia. Podemos ver una imagen del software a continuación.

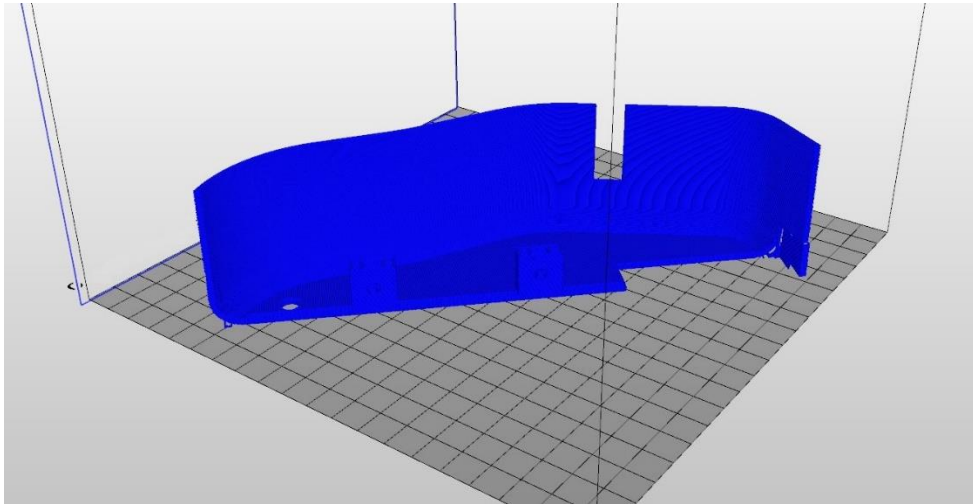


Figura 15. Imagen interior de la carrocería para su fabricación aditiva

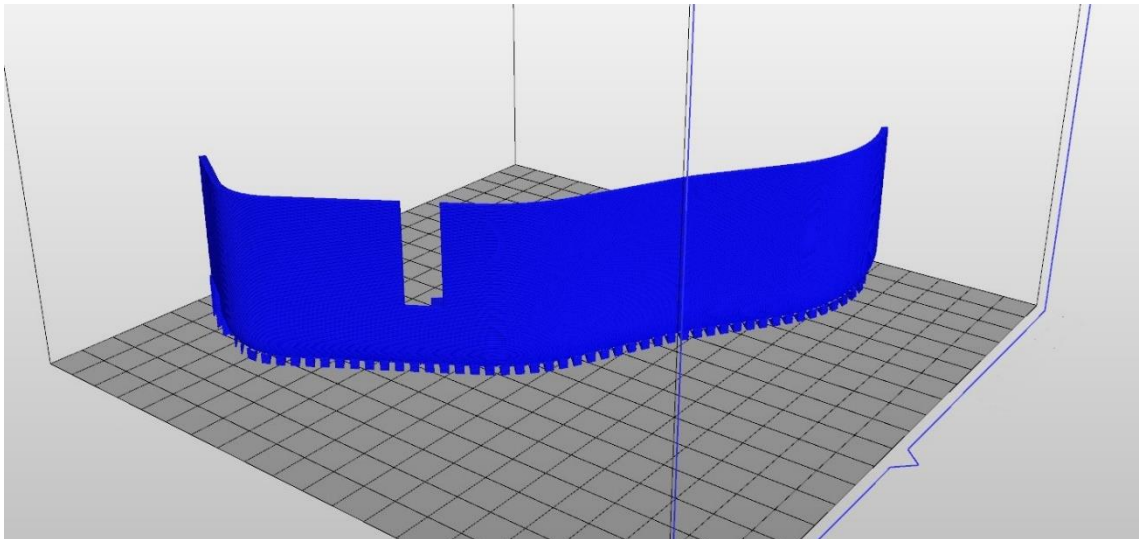


Figura 16. Imagen exterior de la carrocería para su fabricación aditiva

Como se puede observar en la Figura 16, se seleccionó este método con dos partes para su producción ya que sólo se necesitaba un poco de soporte para la estructura. De esta manera la fabricación es mucho más limpia.

6. Código programado

A continuación vamos a ver la programación realizada para el vehículo diseñado en ambos idiomas, primeramente con el editor de bloques y posteriormente en MicroPython.

En ambos casos hemos codificado las señales PWM, tal y como se indica en el datasheet de los servomotores. Después, como es habitual en este tipo de actuadores y viendo el resultado obtenido, se ha ajustado a mano para el tener el movimiento deseado. La fabricación de la electrónica nunca es perfecta, y este tipo de procesos hay que realizarlos siempre.

Para realizar la programación, se han definido funciones para los diferentes tipos de movimientos deseados. En el bucle sin fin se ha colocado la llamada de dichas funciones, de manera que el manejo del vehículo sea más sencillo. Para que el vehículo empiece a moverse, es necesario apretar un botón de la tarjeta. Se programa de esta manera como medida de seguridad y para la repetición de los movimientos.

6.1. Editor de bloques JavaScript

Este es el tipo de programación que se utiliza de forma más específica para la gente que no tenga conocimientos en programación. A continuación, se detallan los bloques referentes a la programación de las funciones.

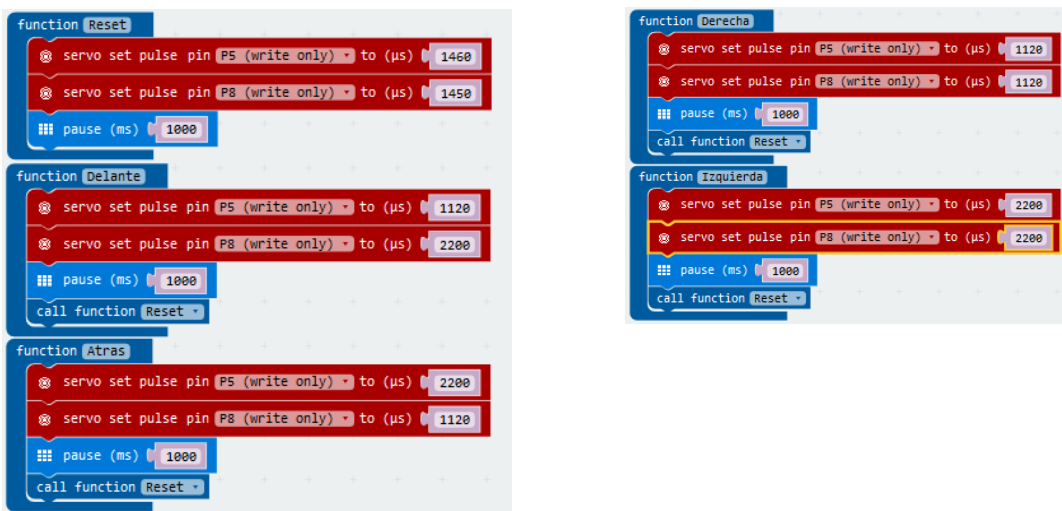


Figura 17. Programación de las funciones mediante bloques JavaScript

Todas las funciones son iguales. Primero se configura una señal PWM en un pin y después en el otro. Para ello se ha utilizado la función “Servo set pulse”. Simplemente le tendremos que introducir en micro segundos el valor de duración de la señal PWM.

Como se puede observar, pese a que el movimiento es el mismo para dos motores, los valores de las señales PWM son diferentes, como por ejemplo en el movimiento de “Delante”. Las funciones vienen definidas por su nombre. Así, “Reset” sirve para parar el vehículo y el resto indican la dirección hacia la que se moverá. En la Figura 18 podemos ver la programación del bucle sin fin.

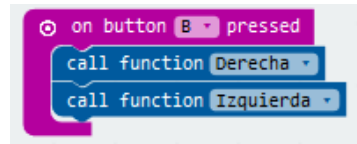


Figura 18. Programación bucle sin fin con bloques JavaScript

Hasta que no se pulse el botón B, no se iniciará el movimiento del vehículo. Vemos un ejemplo donde el vehículo se mueve hacia la derecha, se para y después se mueve hacia la izquierda y se queda detenido. Se ha optado por incorporar la función reset dentro de cada movimiento, así no nos tenemos que preocupar de realizar un movimiento y después parar el vehículo, directamente le damos la dirección que deseamos y cuando alcance el tiempo marcado se detendrá.

Se pueden introducir tantos movimientos como se deseen y después volver a cargar la programación por bluetooth o cable. Si lo deseamos podríamos ver la programación en JavaScript con el traductor de la web de programación.

Esta sería la opción sencilla, pero la repetitividad de los movimientos dejaría mucho que desear. Para solucionar este problema nos podemos aprovechar de los diversos componentes que incorpora la tarjeta. En este caso utilizaremos la brújula, donde podemos leer la posición en grados, desde 1 hasta 359.

A la hora de programar esta opción es más compleja debido a que tenemos que tener en cuenta que cuando gira hacia la derecha los grados van aumentando y cuando gira a la izquierda los grados van disminuyendo. Además de esto deberemos tener en cuenta que cuando está girando hacia la izquierda y llega a 1 después pasa a 359 y viceversa para el movimiento de giro hacia la derecha. También tendremos que tener en cuenta que la brújula hay que calibrarla cada vez que se resetee el microcontrolador, sino puede dar falsas lecturas.

En este caso hemos utilizado una auxiliar para saber cuándo se pasa por cero y cambiamos de 1 a 359 y viceversa. En la Figura 19 podemos observar el flujograma de trabajo del giro a la derecha y en la Figura 20 el del giro a la izquierda. Son iguales, simplemente van a cambiar algún valor y las comparaciones, ya que en un caso gira a la derecha y en el otro a la izquierda. Podemos ver el código completo en el Anexo I, y en el siguiente enlace podemos verlo on-line <https://makecode.microbit.org/hUd7jeDLPg9V>

Flujograma de la función girar a la derecha con brújula

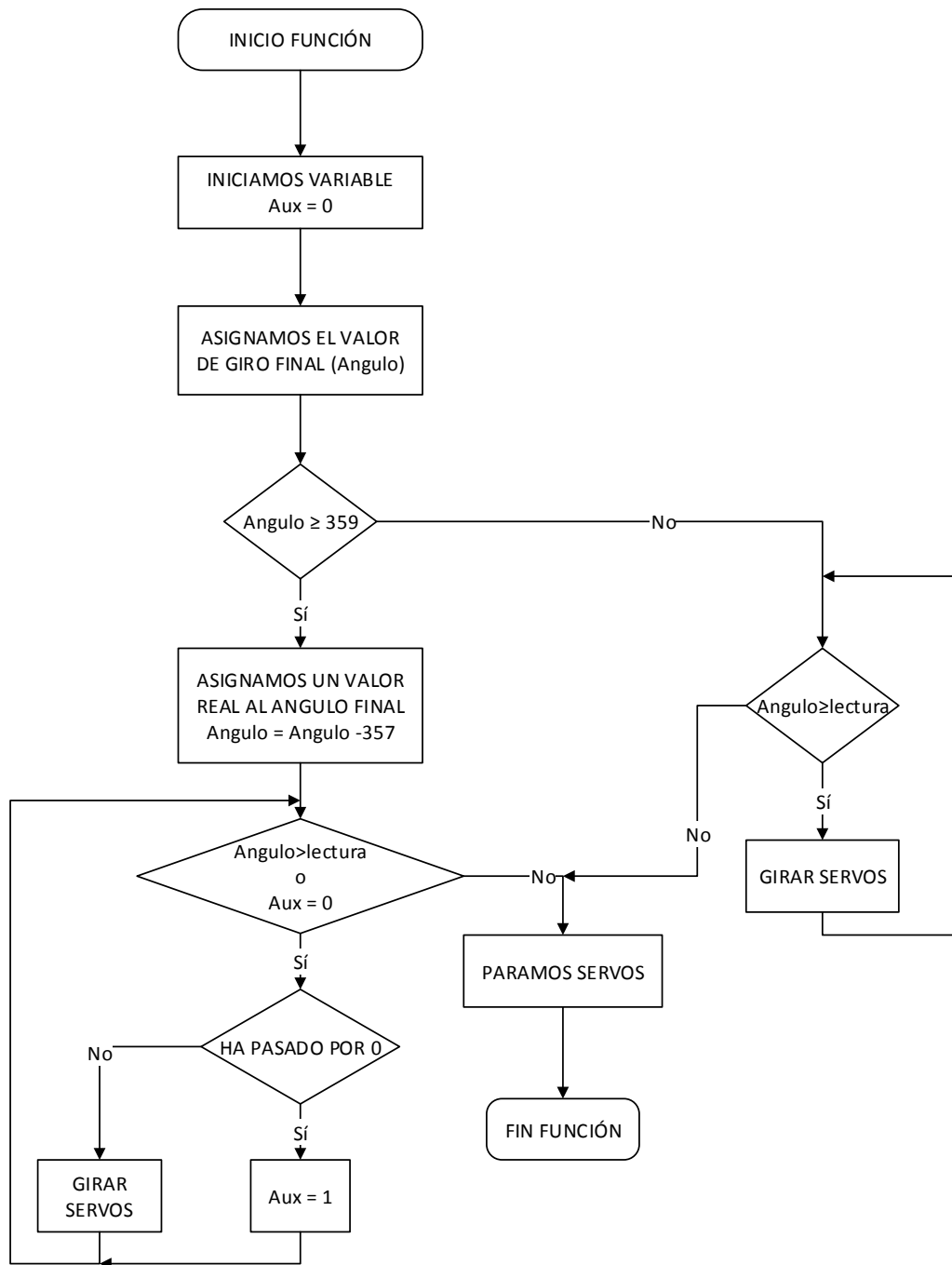


Figura 19. Flujograma de la función giro a la derecha con la brújula

Flujograma de la función girar a la izquierda con brújula

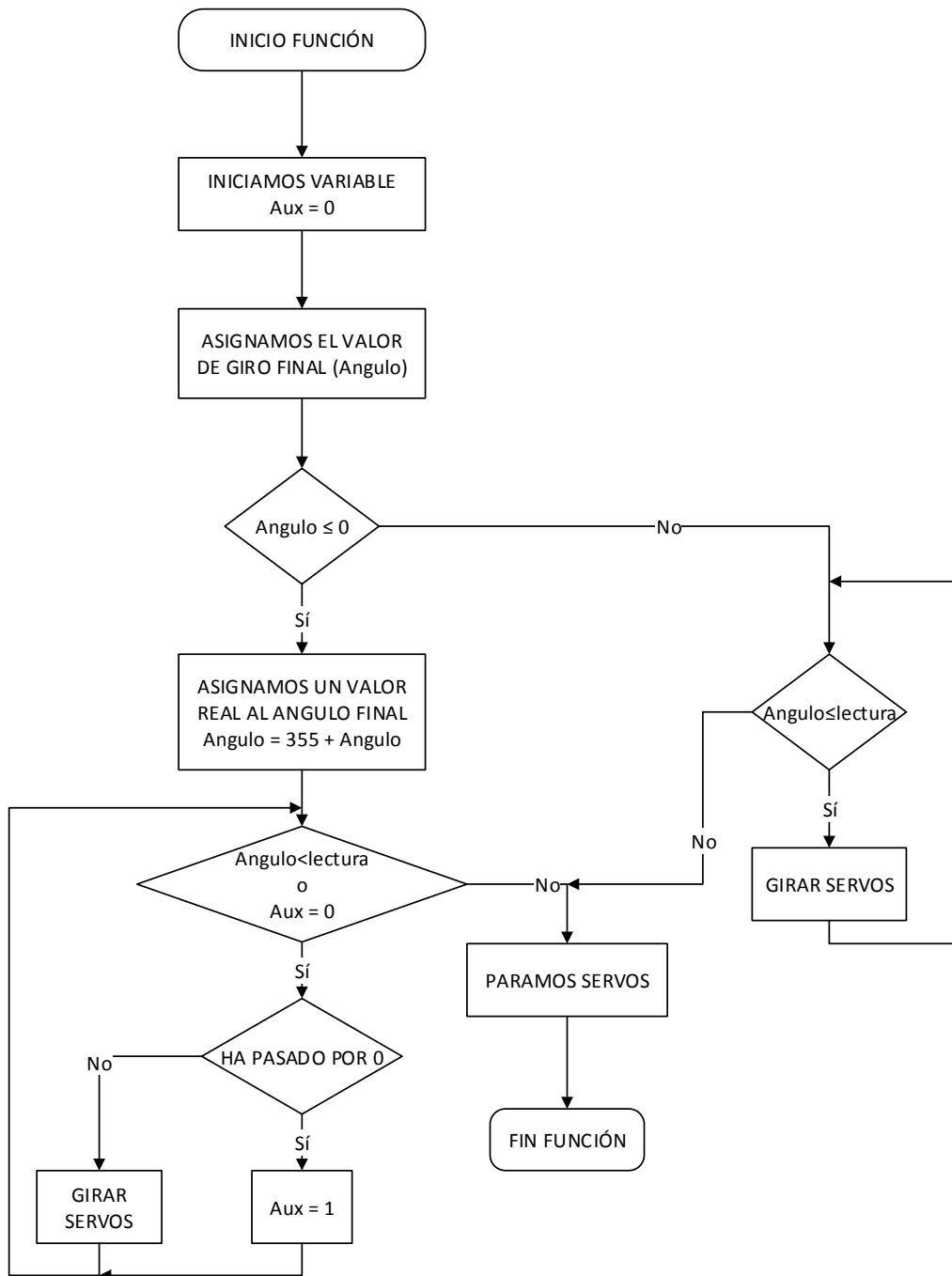


Figura 20. Flujograma de la función giro a la izquierda con la brújula

6.2. Micropython

Al igual que en el apartado anterior vamos a ver las dos programaciones, sin recurrir a la brújula y utilizándola.

Para programar una señal PWM en este lenguaje, primero debemos configurar el periodo de la señal, que en este caso serán de 20ms.

Una vez configurado el periodo deberemos configurar la señal, donde deberemos de realizar una conversión, donde el valor 0 serán 0 ms y el valor 1022 serán 20ms. Para el manejo de los motores necesitamos que nuestra PWM se encuentre en el rango de 700 – 2300 μ s. Aproximadamente la posición de paro se encuentra en 1500 μ s. Por lo tanto los valores que manejaremos se encontrarán dentro del rango de 36 – 117 aproximadamente.

En el Anexo II podemos ver la programación en este lenguaje, tanto la versión básica programada con tiempos de funcionamiento en los servos, como la más compleja utilizando la brújula.

7. Conclusiones

La selección de este microprocesador novedoso es la ideal para esta función, debido a su gran poder de adaptación y a todos los sensores que incorpora, tal y como hemos estado viendo a lo largo de este proyecto.

En cuanto a la estructura, se trata de un vehículo bien conocido por todos y, por tanto, familiar para los usuarios que decidan realizar este proyecto. Debido a esta particularidad, el producto puede llamar más la atención y motivar más a los futuros estudiantes de Mecatrónica.

Además, se pueden introducir en el mundo del control de servomotores y, debido a que es más complejo que manejar un relé, pueden presentar interés por el ámbito de los prototipos únicos y personales.

Al mismo tiempo, pueden tener contacto con el mundo del diseño 3D, ya que es necesario introducirse en el mundo del software CAD para realizar el prototipo.

También es muy interesante el contacto con el mundo de la fabricación aditiva, ya que en el mundo del prototipado cada vez está más extendido, tanto a nivel de realización de proyectos personales como incluso a la hora de producir prótesis médicas, por ejemplo.

Por todos los diferentes motivos que hemos expuesto anteriormente, como conclusión final podemos decir que se trata de un proyecto muy completo e interesante para realizar en un aula de curso preuniversitario.

Bibliografía

https://es.wikipedia.org/wiki/BBC_Micro

<http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>

http://www.eldiario.es/turing/BBC_micro-bit-utilizaran-escolares-Reino-Unido_0_411209780.html

<https://makecode.com/>

<http://tech.microbit.org/>

<http://microbit.org/>

<https://www.python.org/psf/>

<http://www.educaciontrespuntocero.com/recursos/software-programas-aprender-diseno-3d/30654.html>

<http://www.electan.com/>

Índice de figuras

FIGURA 1. RASPBERRY PI 3	9
FIGURA 2. RASPBERRY PI ZERO	10
FIGURA 3. LEGO MINDSTORMS.....	10
FIGURA 4. ENTORNO DE PROGRAMACIÓN DE LEGO MINDSTORMS	11
FIGURA 5. MICROCONTROLADOR ARDUINO DUE	12
FIGURA 6. PLACA MICRO:BIT	13
FIGURA 7A. PARTE DELANTERA DEL BBC MICRO:BIT	15
FIGURA 8. MICRO:BIT EDGE CONNECTOR.....	16
FIGURA 9. DESCRIPCIÓN DE LOS PINES DE BBC MICRO:BIT	16
FIGURA 10. EJEMPLO DE PROGRAMACIÓN DE BLOQUES CON JAVASCRIPT	17
FIGURA 11. VISTA DEL EDITOR WEB PARA PROGRAMAR CON MICROPYTHON	19
FIGURA 12. APLICACIÓN PARA EL S.O. ANDROID.....	21
FIGURA 13. VEHÍCULO DISEÑADO PARA LA PROGRAMACIÓN	23
FIGURA 14. PIEZAS DEL VEHÍCULO	24
FIGURA 15. IMAGEN INTERIOR DE LA CARROCERÍA PARA SU FABRICACIÓN ADITIVA.....	26
FIGURA 16. IMAGEN EXTERIOR DE LA CARROCERÍA PARA SU FABRICACIÓN ADITIVA	26
FIGURA 17. PROGRAMACIÓN DE LAS FUNCIONES MEDIANTE BLOQUES JAVASCRIPT	27
FIGURA 18. PROGRAMACIÓN BUCLE SIN FIN CON BLOQUES JAVASCRIPT	28
FIGURA 19. FLUJOGRAMA DE LA FUNCIÓN GIRO A LA DERECHA CON LA BRÚJULA	29
FIGURA 20. FLUJOGRAMA DE LA FUNCIÓN GIRO A LA IZQUIERDA	30

II. PRESUPUESTO

Índice del Presupuesto

1. Introducción	43
2. Costes de desarrollo del proyecto.....	45
2.1. Costes de personal	45
2.2. Costes de software y material.....	45
3. Costes de implantación	47
4. Coste total del proyecto	49

1. Introducción

El presente presupuesto hace referencia al desarrollo y fabricación del proyecto. El presupuesto incluirá tanto los materiales necesarios para la fabricación como para la puesta en marcha del vehículo, así como el coste del personal necesario para el desarrollo del proyecto.

También se han tenido en cuenta los recursos computacionales utilizados, calculando la parte proporcional de las licencias de los programas empleados durante el proceso.

2. Costes de desarrollo del proyecto

2.1. Costes de personal

Aquí se han contabilizado los costes de mano de obra utilizados durante el proyecto. Tanto para el desarrollo del vehículo como para la programación del mismo.

Concepto	Cantidad	Coste unitario*	Total
-Tiempo de programación (Javascript blocks)	10 h	10 €/h	100 €
-Tiempo de programación (micropython)	10 h	10€/h	240 €
-Tiempo de diseño del vehículo	10 h	10 €/h	100 €
-Tiempo de impresión del vehículo	32 h	7 €/h	224 €
TOTAL			824 €

Tabla 1. Presupuesto coste personal

2.2. Costes de software y material

En este apartado tenemos en cuenta el software utilizado, así como todo el material empleado tanto para la electrónica como para la impresión.

Concepto	Cantidad	Coste unitario	Total
SOFTWARE:			
-SolidWorks 2016 (Licencia anual 4.500€)	10 h	0,514 €/h	5,14 €
-Cura 3D printing (Licencia gratuita)	2 h	0 €/h	0 €
MATERIAL:			
-Servo motores (Feetech FS90R)	4	6,07 €	24,28 €
-Porta pilas (4 x LR06)	1	0,70 €	0,70 €
-Cables 20 x 0,5	12	0,05 €	0,60 €
-ABS (Material para la impresión 3D)	173 g	0,02 €	3,46 €
-BBC Micro:Bit	1	17 €	17 €
TOTAL			51,18 €

Tabla 2. Coste de software y material

Coste unitario* precio típico de mano de obra de becario

3. Costes de implantación

Este proyecto se ha diseñado con la idea de su implantación en un aula de un colegio. Desde ese punto de vista, tal y como se ha visto en el apartado anterior, los costes más elevados son los dedicados a la formación del profesor que prepare el curso y a la impresión 3D del vehículo.

En ese aspecto y, para poder realizar el proyecto a un coste relativamente bajo, la idea es que en el colegio en cuestión se ofrezca voluntario uno de los profesores titulares o alguno adjunto al que le guste la tecnología, con el fin de que pudiera formarse para ello. Desde su casa debería de empezar a investigar sobre el microcontrolador y comenzar a realizar pruebas, buscando información por internet.

Al mismo tiempo, en el colegio se debería de ir acumulando una partida económica para poder realizar una inversión y adquirir una impresora 3D. En este caso, el coste la producción del vehículo se vería reducido de una manera muy significativa, ya que consistiría simplemente en el precio del material utilizado.

Esta impresora se podría dedicar no solamente a este proyecto, sino a muchos otros que puedan surgir en el futuro, además de permitir realizar prácticas a los alumnos en ámbitos como el modelado en 3D con software adaptado específicamente para los niños, como por ejemplo 3D Slash. Así, podrían realizar una muestra de impresión 3D de algo que hayan modelado previamente.

4. Coste total del proyecto

Realizando un pequeño resumen sobre el precio del proyecto tenemos las siguientes partidas:

- Programación → 440 €
- Material electrónico → 46,04 €

En cuanto al apartado del software, en este caso se ha utilizado SolidWorks, pero se puede valorar la opción de utilizar algún otro tipo de software libre de diseño, como por ejemplo TinkerCAD o 123D de Autodesk, Blender o 3D Slash, donde la licencia sería gratuita y no tendríamos que abonar este gasto. Esta última aplicación es una plataforma web muy intuitiva y ágil, donde los más pequeños podrán comenzar a modelar sus primeros sólidos.

En cuanto al precio del material, que no es muy elevado, existen diversas maneras para poder abaratarlo. Por ejemplo, unir a varios colegios para que participen en el proyecto y pedir una cantidad elevada de microcontroladores. En cuanto a los servomotores, si se comprar con bastante antelación, todavía pueden llegar a ser más económicos.

Por otro lado, hay que tener en cuenta que todo este sector evoluciona de una manera vertiginosa, por lo que constantemente se están actualizando los diseños de los materiales electrónicos y surgen formas más económicas de fabricar este tipo de estructuras. Por este motivo, habría que estar atentos, ya que sería posible encontrar servo motores similares a precios aún más reducidos.

III. PLIEGO DE CONDICIONES

Índice del Pliego de Condiciones

1. Condiciones Generales.....	55
1.1. Objeto del proyecto y lugar de emplazamiento	55
1.2. Personal.....	55
1.3. Responsabilidades.....	55
1.4. Otras condiciones no especificadas en este pliego.....	55
2. Cláusulas administrativas	57
2.1. Documentación	57
3. Prescripciones técnicas y particulares	59
3.1. Especificaciones de elementos empleados en el proyecto.....	59
3.2. Especificaciones para la implantación del sistema	59
3.3. Especificaciones de la ejecución	59

1. Condiciones Generales

1.1. Objeto del proyecto y lugar de emplazamiento

Con el presente proyecto se pretende introducir a la Mecatrónica a alumnos de cursos preuniversitarios. Para ello hemos descrito como utilizar un microprocesador novedoso y de altas características técnicas para poder ser utilizado en diferentes niveles de conocimiento.

Así mismo también hemos enumerado el material necesario para conseguir dicho fin en el capítulo de presupuesto, tanto de software como de hardware.

Hemos pasado también por la programación más básica, basada en bloques de JavaScript, hasta la más compleja en Micropython. Entre otros atractivos, ello permite adecuar al nivel de dificultad de la programación a la edad o conocimientos del grupo de estudiantes que vaya a realizar el proyecto.

En este caso, el proyecto se puede realizar con alumnos de a partir de 11 años con la programación más básica, a modo de introducción básica a la Mecatrónica en un aula preparada para tal efecto. El equipamiento mínimo requerido en el aula es el descrito en el apartado 4 del presente capítulo de pliego de condiciones.

1.2. Personal

Para la realización del presente proyecto es necesario únicamente contar con una persona que lleve a cabo las explicaciones, así como las programaciones, además de solucionar los posibles problemas que pudieran surgir durante la realización del proyecto. De este modo, la persona responsable de la iniciativa tendrá que tener conocimientos de electrónica y experiencia en programación de hardware libre similar, como por ejemplo Arduino.

1.3. Responsabilidades

En el caso de realizar alguna modificación, tanto en la parte del software como del hardware, el autor no se hace responsable de los daños derivados de ello. Los resultados obtenidos en el siguiente proyecto son fruto del trabajo con los materiales descritos en el mismo.

1.4. Otras condiciones no especificadas en este pliego

En el presente proyecto se ha descrito la correcta realización de la fabricación y programación de los diferentes elementos. Si faltara alguno de los materiales, o por la

descatalogación de los mismos hubiera que realizar alguna modificación, se requeriría al responsable del trabajo para que realizara una valoración. Única y exclusivamente en caso de que se decida que deben realizarse dichas modificaciones, será bajo las condiciones especificadas del propio responsable.

2. Cláusulas administrativas

2.1. Documentación

En el presente proyecto se describe la utilización de diferentes tipos de hardware libre, lo que implica que posteriormente se pueden utilizar para cualquier otro fin que quiera el propietario del mismo, bajo su propia responsabilidad.

Debido a ello, no resulta necesaria la entrega de ningún tipo de documento al respecto. Además, se puede encontrar amplia información a través de la red de todos y cada uno de los componentes utilizados. Incluso las hojas relativas a las características técnicas de todos ellos.

3. Prescripciones técnicas y particulares

3.1. Especificaciones de elementos empleados en el proyecto

El equipamiento mínimo que requiere este proyecto por alumno es:

- Un ordenador con acceso a internet.
- Un kit compuesto por una tarjeta BBC Micro:Bit, el cable USB para su programación, un portapilas 2 x LR03 y un conector Border Edge para la tarjeta Micro:Bit.
- Cuatro servomotores de 360 grados (Feetech FS90R), en su defecto se pueden adquirir servomotores de 180 grados (Feetech FS90) y adaptarlos.
- Cuatro ruedas para los servomotores de la marca Feetech FS90R 60x8mm
- Portapilas 4 x LR03.

Además, también se debería de tener acceso a una impresora 3D. En el mercado están disponibles desde unos 400€. Si no se quiere adquirir una de ellas, existen tiendas de impresión 3D, aunque los costes todavía son elevados.

3.2. Especificaciones para la implantación del sistema

En este caso, una vez montado y conectado todo el hardware, se debe proceder a la programación del microcontrolador, ya sea por cable o por bluetooth con un Smartphone o una Tablet.

3.3. Especificaciones de la ejecución

La programación insertada en el microcontrolador tiene que ser la especificada en el presente proyecto para garantizar un buen funcionamiento del vehículo.

Lo que sí que se debe realizar es el ajuste fino de las señales PWM de control de los servos ya que, dependiendo de su fabricación, puede variar el funcionamiento óptimo del proyecto final.

IV. ANEXOS

Índice de Anexos

ANEXO I: Programación con bloques JavaScript	65
ANEXO II: Programación con MicroPython	69

ANEXO I: Programación con bloques JavaScript

PROGRAMACIÓN SIN BRÚJULA

```
function Reset
  servo set pulse pin P10 to (µs) 1450
  servo set pulse pin P8 (write only) to (µs) 1500
  pause (ms) 1000

function Delante
  servo set pulse pin P10 to (µs) 1120
  servo set pulse pin P8 (write only) to (µs) 2200
  pause (ms) 1000
  call function Reset

function Atras
  servo set pulse pin P10 to (µs) 2200
  servo set pulse pin P8 (write only) to (µs) 1120
  pause (ms) 1000
  call function Reset

function Derecha
  servo set pulse pin P10 to (µs) 1120
  servo set pulse pin P8 (write only) to (µs) 1500
  pause (ms) 1200
  call function Reset

function Izquierda
  servo set pulse pin P10 to (µs) 1450
  servo set pulse pin P8 (write only) to (µs) 1800
  pause (ms) 800
  call function Reset
```

PROGRAMACIÓN CON BRÚJULA

```
on start
  calibrate compass
```

```
function Reset
  servo set pulse pin P5 (write only) to (μs) 1460
  servo set pulse pin P8 (write only) to (μs) 1450
  pause (ms) 1000

function Delante
  servo set pulse pin P5 (write only) to (μs) 1120
  servo set pulse pin P8 (write only) to (μs) 2200
  pause (ms) 1000
  call function Reset

function Atras
  servo set pulse pin P5 (write only) to (μs) 2200
  servo set pulse pin P8 (write only) to (μs) 1120
  pause (ms) 1000
  call function Reset
```

```
function Derecha
  establecer Aux para 0
  establecer Angulo para compass heading (°) + 80
  si Angulo ≥ 359
  entonces establecer Angulo para Angulo - 357
  mientras Angulo > compass heading (°) < Aux = 0
  ejecutar si compass heading (°) ≤ 25
  entonces establecer Aux para 1
  servo set pulse pin P5 (write only) to (μs) 1120
  servo set pulse pin P8 (write only) to (μs) 1120
  si no mientras Angulo ≥ compass heading (°)
  ejecutar servo set pulse pin P5 (write only) to (μs) 1120
  servo set pulse pin P8 (write only) to (μs) 1120
  call function Reset
```

```

function Izquierda
  establecer Aux para 0
  establecer Angulo para compass heading (°) - 90
  si
    Angulo ≤ 0
  entonces
    establecer Angulo para 355 + Angulo
    mientras
      Angulo < compass heading (°) 0
      ejecutar
        si
          compass heading (°) ≥ 340
          entonces
            establecer Aux para 1
            servo set pulse pin P5 (write only) to (μs) 2200
            servo set pulse pin P8 (write only) to (μs) 2200
    si no
      mientras
        Angulo ≤ compass heading (°)
      ejecutar
        servo set pulse pin P5 (write only) to (μs) 2200
        servo set pulse pin P8 (write only) to (μs) 2200
  call function Reset

```

```

on button B pressed
  call function Derecha
  call function Izquierda
  call function Delante

```


ANEXO II: Programación con MicroPython

PROGRAMACIÓN SIN BRÚJULA

```
1 # Cargamos la librería de control del microcontrolador
2 from microbit import *
3
4 #Configuramos el periodo de la señal PWM
5 #para los pines que vamos a utilizar
6 pin5.set_analog_period(20)
7 pin8.set_analog_period(20)
8
9 # INICIO DE LAS FUNCIONES
10 #PIN5 --> Motor Izquierda
11 #PIN8 --> Motor Derecha
12 def reset():
13     pin5.write_analog(74) #1450 microseg.
14     pin8.write_analog(77) #1500 microseg.
15     sleep(1000)
16
17 def delante():
18     pin5.write_analog(57)
19     pin8.write_analog(112)
20     sleep(1000)
21     reset()
22
23 def atras():
24     pin5.write_analog(112)
25     pin8.write_analog(57)
26     sleep(1000)
27     reset()
28
29 def derecha():
30     pin5.write_analog(61)
31     pin8.write_analog(82)
32     sleep(1000)
33     reset()
34
35 def izquierda():
36     pin5.write_analog(69)
37     pin8.write_analog(84)
38     sleep(1000)
39     reset()
40 #FIN DE LAS FUNCIONES
41
42 #INICIO DEL BUCLE INFINITO
43 while True:
44
45     if button_a.is_pressed():
46         delante()
47         derecha()
48         izquierda()
49
50 # Reseteamos las señales para parar servos
51 pin5.write_analog(0)
52 pin8.write_analog(0)
```

PROGRAMACIÓN CON BRÚJULA

```
1 # Cargamos La librería de control del microcontrolador
2 from microbit import *
3
4 #Configuramos el periodo de La señal PWM
5 #para Los pines que vamos a utilizar
6 pin5.set_analog_period(20)
7 pin8.set_analog_period(20)
8
9 # Si La brújula no está calibrada, La calibramos
10 if (not compass.is_calibrated()):
11     compass.calibrate()
12
13 # INICIO DE LAS FUNCIONES
14 #PIN5 --> Motor Izquierda
15 #PIN8 --> Motor Derecha
16 def reset():
17     pin5.write_analog(74) #1450 microseg.
18     pin8.write_analog(77) #1500 microseg.
19     sleep(1000)
20
21 def delante():
22     pin5.write_analog(57)
23     pin8.write_analog(112)
24     sleep(1000)
25     reset()
26
27 def atras():
28     pin5.write_analog(112)
29     pin8.write_analog(57)
30     sleep(1000)
31     reset()
32
33 def derecha():
34     aux = 0
35     angulo = 0
36     # Calculamos el angulo final
37     angulo = compass.heading() + 80
38     # Si no es un valor valido Lo adaptamos
39     if(angulo >= 359):
40         angulo = angulo - 357
41         while ((angulo > compass.heading()) or aux == 0):
42     # Si ha pasado por cero activamos La variable auxiliar
43         if (compass.heading() <= 25):
44             aux=1
45             pin5.write_analog(61)
46             pin8.write_analog(82)
47     # Si el valor no es necesario adaptarlo
48     else:
49         while (angulo >= compass.heading()):
50             pin5.write_analog(61)
51             pin8.write_analog(82)
52     reset()
53
54 def izquierda():
55     aux = 0
56     angulo = 0
57     angulo = compass.heading() - 80
58     if(angulo <= 0):
59         angulo = 355 + angulo
60         while ((angulo < compass.heading()) or aux == 0):
61             if (compass.heading() > 340):
62                 aux=1
63                 pin5.write_analog(69)
64                 pin8.write_analog(84)
```

```
65     else:
66         while (angulo <= compass.heading()):
67             pin5.write_analog(69)
68             pin8.write_analog(84)
69         reset()
70
71     #FIN DE LAS FUNCIONES
72
73     #INICIO DEL BUCLE INFINITO
74     while True:
75         if button_a.is_pressed():
76             delante()
77             derecha()
78             izquierda()
79
80         pin5.write_analog(0)
81         pin8.write_analog(0)
```