

# Universidad de Alcalá

## Escuela Politécnica Superior

Grado en Ingeniería Electrónica y Automática Industrial



**Trabajo Fin de Grado**

Diseño e implementación de un sistema de prueba de tarjetas

Micro:bit

ESCUELA POLITECNICA

**Autor:** Juan Diego Durán López

**Tutor/es:** Julio Pastor Mendoza

2024

UNIVERSIDAD DE ALCALÁ  
Escuela Politécnica Superior

**Grado en Ingeniería Electrónica y Automática Industrial**

Trabajo Fin de Grado  
Diseño e implementación de un sistema de prueba de tarjetas  
Micro:bit

**Autor:** Juan Diego Durán López

**Tutor/es:** Julio Pastor Mendoza

**TRIBUNAL:**

**Presidente:** Miguel González Herráez

**Vocal 1º:** Ernesto Martín Gorostiza

**Vocal 2º:** Julio Pastor Mendoza

**FECHA:** 13/02/2024

## **Agradecimientos**

Dedico este espacio de mi proyecto a todas las personas que me han apoyado en este camino.

Primeramente, a mi familia, por su incondicional apoyo en cualquiera de las circunstancias. Sus valores de esfuerzo y perseverancia transmitidos hacia mí desde que tengo uso de conciencia me hacen poder entregar este trabajo, sin ellos nunca habría conseguido ser la persona que soy hoy en día.

En segundo lugar, quería agradecer a los compañeros y amigos que he encontrado en la Universidad de Alcalá. Han formado un papel fundamental para hacer más fácil este proceso y juntos nos hemos apoyado unos a otros para seguir siempre avanzando. Me llevo recuerdos que son para toda mi vida.

Y, por último, quería dedicar este espacio a mi pareja, quien ha puesto todos los medios y ha estado dispuesta a ayudarme en todo lo que estuviera en sus manos.

Bendecido me siento de estar rodeado de todas estas personas. Han creído en mí y con ellos me siento respaldado y capaz de conseguir todo aquello que me proponga.

## Contenido

Agradecimientos .....	3
Contenido .....	4
Ilustraciones .....	5
Tablas .....	6
Glosario de acrónimos y abreviaturas .....	7
Resumen .....	8
Abstract .....	9
CAPITULO 1: INTRODUCCIÓN.....	10
Introducción .....	12
CAPITULO 2: TARJETA MICRO:BIT .....	15
2.1.Introducción a la tarjeta Micro:bit.....	16
2.2.Características Hardware.....	17
CAPITULO 3: CARACTERÍSTICAS SOFTWARE Y PROGRAMACIÓN .....	27
3.1.Características software.....	28
3.2.Programación .....	29
CAPITULO 4: PRUEBAS DE FUNCIONALIDAD.....	33
4.1. Matriz de LEDs (0) .....	36
4.2. Botones programables (1 y 2) .....	37
4.3. Logo sensible al contacto (3) .....	38
4.4. Micrófono (4) .....	39
4.5. Altavoz (5) .....	41
4.6. Acelerómetro (6) .....	42
4.7. Brújula (7) .....	43
4.8. Sensor de temperatura (8) .....	44
4.9. Pines de entrada o salida. (9).....	45
4.9.1 Placa de circuito impreso (PCB) .....	45
4.9.2 Pruebas sobre los puertos de la micro:bit.....	45
CAPITULO 5. DISEÑO DE LA PLACA DE CIRCUITO IMPRESO .....	50
5.1.Esquema general de conexiones versión v1 .....	51
5.2.Esquema general de conexiones versión v2.....	55
5.3.Listado de componentes de ambas versiones .....	58
CAPITULO 6: CONCLUSIONES Y TRABAJOS FUTUROS .....	60
CAPITULO 7: PRESUPUESTO .....	62
6.1. Tarjeta de pruebas .....	63
6.2. Material hardware .....	63
6.3. Material software.....	63
6.4. Mano de obra.....	64
6.5. Presupuesto total .....	64
CAPITULO 8: BIBLIOGRAFIA.....	66
CAPITULO 9: ANEXOS.....	77
9.1. Proyecto Make It Digital .....	78
9.1.1 Comparativa con otras tarjetas del mercado.....	79
9.2 Código cargado en la tarjeta puesta a prueba o hija .....	82
9.3 Código cargado en tarjeta referencia o madre.....	103
9.4 Modulo Bluetooth .....	116
9.5 Comunicación radio .....	121
9.6 Manual de usuario sistema de prueba de tarjetas Micro:bit .....	122

## Ilustraciones

Ilustración 1. Diagrama de bloques del procesador Cortex-M0.....	18
Ilustración 2. Mapa de pines de la tarjeta Micro:bit V1 .....	20
Ilustración 3. Tarjeta micro:bit V1 .....	23
Ilustración 4. Tarjeta micro:bit V2.....	25
Ilustración 5. Diagrama de bloques de la Micro:bit .....	25
Ilustración 6. Sistema de software de la micro:bit .....	28
Ilustración 7. Entorno de programación MakeCode .....	29
Ilustración 8. Organigrama de las pruebas de funcionalidad .....	35
Ilustración 9. Prueba de funcionalidad de la matriz de LEDs. ....	36
Ilustración 10. Mensaje en el informe para la prueba de funcionalidad de la matriz de LEDs... ..	36
Ilustración 11. Prueba de funcionalidad de los botones programables. ....	37
Ilustración 12. Transmisión prueba botones.....	38
Ilustración 13. Prueba de funcionalidad del logo sensible. ....	38
Ilustración 14. Mensaje transmitido prueba del logo .....	39
Ilustración 15. Prueba de funcionalidad del micrófono .....	40
Ilustración 16. Mensaje en el informe para la prueba del micrófono. ....	40
Ilustración 17. Prueba de funcionalidad del altavoz.....	41
Ilustración 18. Mensaje transmitido prueba altavoz.....	42
Ilustración 19. Transmisión prueba acelerómetro .....	42
Ilustración 20. Transmisión prueba brújula.....	43
Ilustración 21. Mensaje en el informe para la prueba del sensor de temperatura.....	44
Ilustración 22. Multiplexor CD74HC4067.....	46
Ilustración 23. Formato del mensaje vía serie para la prueba de pines. ....	47
Ilustración 24. Esquemático del PCB v1 .....	51
Ilustración 25. Diseño del PCB v1 .....	53
Ilustración 26. Simulación 3D de la placa en versión prototipo v1.....	54
Ilustración 27. Tarjeta construida en v1. ....	55
Ilustración 28. Esquemático PCB v3.....	56
Ilustración 29. Diseño PCB v2.....	57
Ilustración 30. Simulación 3D de la placa en v2.....	57
Ilustración 31. Menú general APP móvil .....	116
Ilustración 32. Código de conexión Bluetooth.....	118
Ilustración 33. Secuencia de enlace dispositivo. ....	118
Ilustración 34. MakeCode en APP .....	119
Ilustración 35. Transmisión de un programa de la APP a la tarjeta .....	120
Ilustración 36. Esquema módulo Bluetooth. ....	121
Ilustración 37. Configuración del puerto serie .....	122

## Tablas

Tabla 1. Mapa de pines de la Micro:bit.....	22
Tabla 2. Coste Tarjeta de pruebas. ....	63
Tabla 3. Coste material hardware.....	63
Tabla 4. Coste Software. ....	63
Tabla 5. Coste mano de obra. ....	64
Tabla 6. Presupuesto total. ....	64
Tabla 7. Especificaciones de la tarjeta Raspberry Pi Pico. ....	79
Tabla 8. Especificaciones de la tarjeta Arduino UNO rev3. ....	80

## Glosario de acrónimos y abreviaturas

PCB	Printed Circuit Board
LED	Light-Emitting Diode
GND	Ground
IoT	Internet of things
ARM	Advanced RISC Machine
RAM	Random Access Memory
MEMS	Microelectromechanical Systems
I/O	Input/Output
I2C	Inter-Integrated Circuits
SPI	Schedule Performance Index
UART	User Acceptance Testing
BLE	Bluetooth Low Energy
USB	Universal Serial Bus
EN	Enable
OWH	Output Write High
OWL	Output Write Low
ORH	Output Read High
ORL	Output Read Low

# Resumen

El objetivo de este trabajo es la creación de un sistema de control de calidad de tarjetas Micro:bit, un tipo de placa de desarrollo. Para ello, se ha desarrollado un determinado hardware basado en una placa de circuito impreso en el que se insertan dos tarjetas micro:bit: una maestra que marcará las pruebas a realizar sobre la otra tarjeta, que será la probada. Todo ello va sustentado por un software diferente descargado en cada micro:bit y que permite realizar las determinadas pruebas de cada uno de sus componentes y periféricos, obteniendo una respuesta de validación.

Los resultados de las pruebas se recibirán a través de la propia tarjeta y serán enviados vía serie , garantizando el correcto funcionamiento.

## Palabras clave

Tarjeta Micro:bit, BBC, tarjeta programable.



# **Abstract**

The main objective of this work is to build a quality control system of micro:bit cards, which are a type of development board. To do so, a hardware has been developed based on a printed circuit assembly on which two micro:bit cards are inserted: a master card that will determine the tests to be carried out on the other card, which will be the tested one. All this is supported by a separate software downloaded on each micro:bit, which allows to perform certain tests on every component and peripherals, obtaining a validation response.

The test results will be received either through the card itself and sent via serial, ensuring correct operation.

## **Key words**

Micro:bit card, BBC, programmable card.

# CAPITULO 1: INTRODUCCIÓN



# Introducción

La finalidad de este trabajo ha sido crear un sistema de validación de tarjetas Micro:bit, una placa de desarrollo utilizada en el departamento de Electrónica de la Universidad de Alcalá.

A través de este se obtiene una información precisa del estado en el que se encuentra la tarjeta y los resultados que se han obtenido sobre cada uno de los puntos comprobados sobre ella.

Este trabajo no solo aborda el estudio de la funcionalidad técnica de las tarjetas Micro:bit, sino que también tiene un impacto significativo en el aspecto pedagógico. Al asegurar que las tarjetas funcionan correctamente, se garantiza una experiencia de aprendizaje fluida y efectiva para los estudiantes del grado.

Es un paso importante para mantener y mejorar la calidad de la educación tecnológica. Al invertir en la confiabilidad de las tarjetas ya adquiridas, se garantiza un proceso de aprendizaje óptimo y se reducen gastos en la adquisición de nuevas tarjetas sin haber sido sometidas previamente a un control de calidad.

Para ello, se hace uso de un desarrollo software y hardware, que complementados precisan toda la información al usuario que esté realizando el control de calidad de una tarjeta Micro:bit, permitiéndole de una manera sencilla e intuitiva conocer cuáles son las deficiencias que encuentra la tarjeta si es que las tuviese.

El sistema se basa en un circuito impreso al cual son conectadas dos tarjetas Micro:bit, una de ellas se concibe como “maestra”, teniéndose la garantía de que se encuentra en condiciones óptimas de funcionamiento y la otra es llamada “esclava” o “hija”, siendo ésta sobre la que se validarán todas las características de la tarjeta. Ambas quedan conectadas vía radio y circuito impreso, siendo la tarjeta maestra la que marca el orden secuencial de pruebas que serán realizadas sobre la tarjeta esclava. Todos los resultados obtenidos serán reflejados en un informe vía serie emitido desde la tarjeta a prueba.

El proyecto se estructura de la siguiente manera: Primeramente, se realiza un desarrollo del software necesario para conectar ambas tarjetas y simular un orden establecido de pruebas a realizar. Después, se desarrollan aquellas pruebas que únicamente necesitan de una activación para ser probadas sobre la misma tarjeta hija, siendo estas puramente periféricos como la pantalla de LEDs, los botones, el sensor de temperatura, etc.

Una vez definida esta primera parte, el siguiente paso es la comprobación del hardware de la tarjeta a probar, donde se desarrolla simultáneamente el software que sustenta las pruebas, la realización del prototipo del PCB y sus correspondientes pruebas, la selección y adquisición de materiales y por último la propia fabricación del circuito. La PCB diseñada en este proyecto contiene varias versiones, siendo la tarjeta construida una versión prototípica. En el proyecto se definen diferentes mejoras planteadas para futuras versiones.

Sobre la tarjeta construida se han soldado todos los componentes adquiridos y se realizaron todas las pruebas de funcionamiento que abalaran que el proyecto cumplía con todos los requerimientos iniciales.

El trabajo se inicia comentando todas las características hardware y software de que las que se dota la tarjeta, para que el lector conozca cuáles son sus capacidades y elementos que deben ser validados por este sistema. Después se describen las pruebas de funcionalidad que serán realizadas, y por último se describirá el método de diseño del circuito impreso y el presupuesto necesitado para este mismo.



# CAPITULO 2:

## TARJETA MICRO:BIT

## 2.1. Introducción a la tarjeta Micro:bit

La tarjeta Micro:bit, anunciada a partir de la campaña “Make It Digital” de la BBC y lanzada el 12 de marzo de 2015 [9], es una placa de desarrollo programable creada con el objetivo de apoyar la enseñanza de la programación y tecnología en las escuelas. La intención de crear esta placa fue proporcionar un millón de éstas entre las escuelas de Reino Unido.

La fundación Micro:bit aparte de distribuir millones de estas placas alrededor del mundo, ofrece múltiples recursos para ayudar a los estudiantes en la programación de las placas. Para ello, facilitan múltiples tutoriales, eventos, proyectos y actividades que los estudiantes pueden visualizar o realizar [8].

Micro:bit proporciona la capacidad de iniciar a los más jóvenes en el ámbito de la programación de una forma dinámica y sencilla, abriendo en ellos un mundo de posibilidades gracias a los ilimitados proyectos realizables. Brinda la oportunidad de utilizar la tecnología de una forma creativa y estimular a los jóvenes en su capacidad de imaginación.

Esta placa tiene un tamaño reducido, con dimensiones de 43 mm x 52 mm, por lo que es fácilmente portable y adaptable a múltiples aplicaciones. Incluye una gran cantidad de periféricos y brinda la capacidad de ser una plataforma IoT, convirtiéndola en un dispositivo muy interesante para cualquier tipo de usuario [2].

Cuenta con un procesador, memoria y diversos sensores y componentes, como un acelerómetro, botones, leds, etc., siendo todos estos controlables y programables para crear proyectos interactivos. La programación es muy versátil, dando la capacidad de trabajar con lenguajes de programación como JavaScript, Python, C++, Blockly, y más. Además, incluye un entorno gráfico “Editor MakeCode” que posibilita la programación mediante concatenación de bloques de código, brindando la posibilidad de programar a cualquier usuario y pudiendo traducir este código resultante a otro lenguaje de programación [10]. Se puede acceder a este editor a través de la web oficial o a partir de la aplicación disponible para dispositivos móvil y tablet, dando plena accesibilidad al usuario.

Por lo tanto, es una herramienta poderosa y asequible con la cual enseñar a los más jóvenes y no tan jóvenes los conceptos básicos de programación y tecnología de manera lúdica y creativa.



## 2.2. Características Hardware

La placa Micro:bit es un sistema embebido compuesto por una combinación de hardware y software basado en una arquitectura ARM. Tiene un tamaño reducido, inferior al de una tarjeta de crédito (43mm x 52 mm) y posee diferentes elementos hardware que convierten a esta placa en una de las más interesantes de este tipo en el mercado actual.

En su primera versión (V1), lanzada en 2015, la placa poseía las siguientes especificaciones hardware [7]:

- **Procesador Nordic nRF51822 ARM Cortex-M0:** Es uno de los componentes clave de esta placa de desarrollo, ya que desempeña un papel fundamental en la ejecución de programas y el control de los diversos componentes.

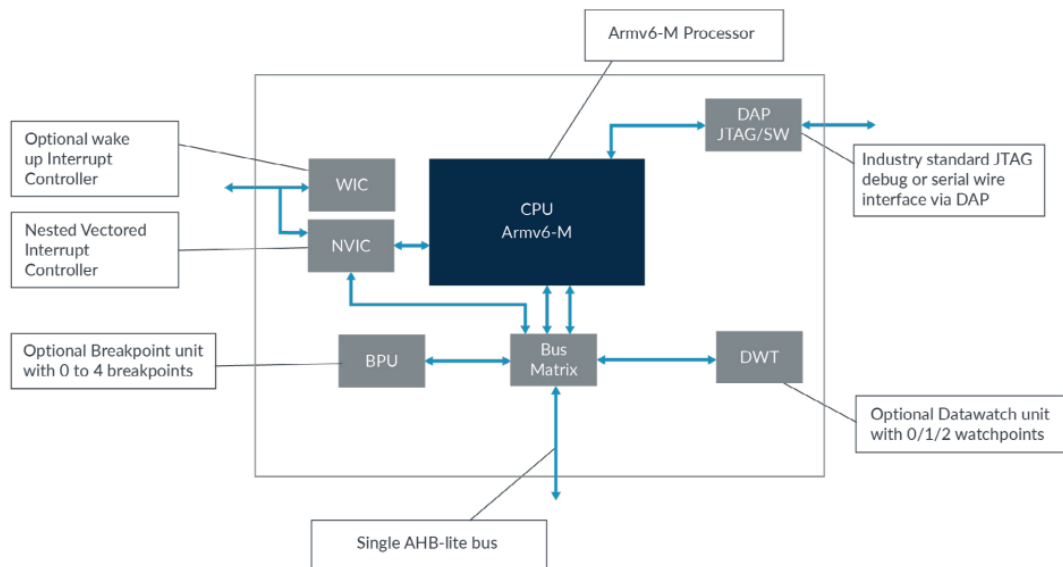
El procesador de la familia ARM [13] (Advanced RISC Machine) es una arquitectura RISC (Reduced Instruction Set Computer), conocido por su eficiencia energética y su capacidad de rendimiento en aplicaciones embebidas y sistemas integrados. El Cortex-M0 [15] se basa en un conjunto reducido de instrucciones programables que se ejecutan en ciclos fijos de reloj.

La velocidad del reloj funciona a 16 megahercios (Mhz), inferior en comparación a procesadores de computadores generales, pero más que suficiente para ejecutar programas y aplicaciones diseñadas específicamente para la micro:bit.

El procesador tiene una memoria flash de 256 kilobytes (KB) ofrece la capacidad de almacenar programas y datos. Por otra parte, cuenta con 16 KB de memoria RAM que permite a los usuarios almacenar datos de forma temporal.

Una de las características más importantes, presente en todo este tipo de procesadores, es su eficiencia energética. Opera a bajo consumo, lo que es vital para este tipo de dispositivo, alimentados habitualmente por una batería para aplicaciones en campo, lo que permite que la placa funcione durante largos periodos de tiempo con una batería AAA.

En la imagen 1 se puede observar el diagrama de bloques que sigue el procesador [46]:



*Ilustración 1. Diagrama de bloques del procesador Cortex-M0*

- **Matriz de LEDs:** La tarjeta viene equipada con una matriz de 5x5 diodos emisores de luz en el anverso de la placa. Cada LED puede ser controlado de manera independiente, brindando al proyecto un gran dinamismo según las necesidades.

La iluminación de los LEDs es programable, lo que permite crear diferentes patrones de movimientos, generar letras, números o figuras o incluso mostrar imágenes.

Esta especificación brinda a la placa de una excelente herramienta a través de la cual crear una interfaz con el usuario sin depender de componentes adicionales, permitiendo de una manera sencilla utilizar esta matriz para representar simulaciones, pantallas de información o proyectos visuales.

- **Botones programables:** La micro:bit incluye dos botones físicos, etiquetados con la nomenclatura A y B. Están situados en la parte frontal de la placa, permitiendo que sean pulsados y utilizados fácilmente.

Son completamente programables, lo que permite asignarles funciones específicas al repetir patrones definidos. Los botones funcionan en un modo eléctrico invertido típico, donde una resistencia pull-up garantiza un '1' lógico cuando se suelta el botón y un '0' lógico cuando se presiona el botón [38]. De esta manera en la programación se puede detectar cuando es presionado uno o ambos botones, y en función de la función definida se realizarán las acciones consecuentes.

- **Acelerómetro:** Es un componente clave entre los componentes de la placa. Es un acelerómetro tipo MEMS de la empresa NXP, en concreto el modelo MMA8653FC [9]. Es un sensor de tres ejes que permite detectar la aceleración a lo largo de tres dimensiones espaciales: X, Y y Z.

Para realizar esta detección, mide la fuerza gravitatoria y las aceleraciones lineales, lo que permite que la placa detecte desplazamientos y cambios en su orientación respecto a la gravedad. Su rango de medición es configurable, desde  $\pm 2g$  (gravedades) a  $\pm 8g$  en cada uno de los ejes.

En algunos casos puede ser necesario una calibración del acelerómetro para asegurar su correcto funcionamiento en sus mediciones. Esta calibración supone establecer valores nulos a cada eje y se podría realizar en el código de programa.

- **Magnetómetro:** Es uno de los diferentes sensores incluidos en la placa. Este sensor permite detectar campos magnéticos, y en el caso específico de la micro:bit, es utilizado para medir la dirección e intensidad de los campos magnéticos externos. Esto permite detectar la orientación de la micro:bit en relación con el campo magnético de la Tierra.

El magnetotérmico utilizado es un sensor de Efecto Hall, de la familia NXP Semiconductors y en concreto el modelo MAG3110 [9]. Es capaz de medir campos magnéticos en microteslas ( $\mu T$ ) en el marco de las tres dimensiones: X, Y y Z.

Al igual que en el caso del acelerómetro, este sensor también puede requerir una calibración que requiera establecer valores de referencia para cada eje.

- **Sensor de temperatura:** Se incluye un sensor de temperatura que permite la medida de la temperatura ambiente. Se trata de un termistor a partir de una resistencia eléctrica que modifica su impedancia en función de la temperatura [38].

El rango de medida se encuentra entre  $-40$  a  $85$  °C, lo que es más que suficiente para las mediciones que se van a realizar con la placa.



Los pines de la tarjeta están divididos en varios grupos que cumplen diferentes funciones, y los podemos desglosar en 6 diferentes:

### **1. Pines de Alimentación:**

- Pin 3V (3.3V): Este pin proporciona una tensión de alimentación de 3.3 voltios para alimentar componentes externos. La tarjeta permite una corriente máxima de 15mA a través de estos pines.
- GND (Tierra): Es el pin de tierra (GND), utilizado para completar el circuito y proporcionar una referencia de tierra.

### **2. Pines de E/S Digital:**

- P0 a P16: Estos pines son de entrada/salida digital utilizados para conectar sensores, LEDs, y otros componentes.

### **3. Pines Analógicos:**

- P0-P4, P10: Estos pines, aparte de ser utilizados para E/S digital, algunos de ellos también son capaces de realizar funciones analógicas, como la lectura de valores analógicos de sensores. El margen de tensiones a la entrada es de 3,3V, siendo el máximo permisible antes de su deterioro 3,9V [39].

### **4. Pines de Comunicación:**

- I2C (P19, P20): Estos pines se utilizan para la comunicación I2C, que es un protocolo de comunicación serie utilizado para conectar sensores y dispositivos externos.
- SPI (P13, P14, P15): Estos pines se utilizan para la comunicación SPI, otro protocolo de comunicación serie utilizado para conectar dispositivos externos.
- UART (P0-P16,P19,P20): Estos pines se utilizan para la comunicación UART, que permite la comunicación serie con otros dispositivos, como módulos Bluetooth.

### **5. Pantalla LED:**

- P3, P4, P6, P7, P10: Tienen la funcionalidad de controlar respectivamente las 5 columnas de la matriz de LEDs integrada en la tarjeta.

## 6. Botones programables:

- El pin 5 es útil para controlar el botón A mientras el pin 11 lo es para el B. El P2 permitirá acceder al botón sensible al contacto.

El mapa de pines de la tarjeta para la V2 es el siguiente [39]:

<b>GPIO</b>	<b>Asignación</b>	<b>Interfaz (KL27/nRF52)</b>	<b>Puerto</b>
<b>P0.00</b>	ALTAVOZ	KL27_DAC / IF_SPEAKER	
<b>P1.05</b>	LEDCOL(4)	N	P6
<b>P0.02</b>	GPIO, ANALOG, TOUCH, PWM, UART	N	P0
<b>P0.03</b>	GPIO, ANALOG, TOUCH, PWM, UART	N	P1
<b>P0.04</b>	GPIO, ANALOG, TOUCH, PWM, UART	N	P2
<b>P0.05</b>	(GPIO), LEDCOL(4), (PWM), (UART)	N	
<b>P0.06</b>	UART_INT_RX	PTA18 / P0.03	
<b>P1.08</b>	UART_INT_TX	PTA19 / P0.02	
<b>P0.08</b>	I2C_INT_SCL	PTC1 / P0.29	
<b>P0.10</b>	GPIO, PWM, UART (NFC2)	N	P8
<b>P0.09</b>	(GPIO), (PWM), (UART), (NFC1)	N	P9
<b>P0.11</b>	(GPIO), LEDCOL(2), (PWM), (UART)	N	P7
<b>P1.02</b>	GPIO, PWM, UART	N	P16
<b>P0.19</b>	LEDFILA(5)	N	
<b>P0.14</b>	(GPIO), BUTTON(A), (PWM), (UART)	N	P5
<b>P0.23</b>	(GPIO), BUTTON(B), (PWM), (UART)	N	P11
<b>P1.04</b>	TOUCH_BTN	N	
<b>P0.16</b>	I2C_INT_SDA	PTC2 / P0.28	
<b>P0.17</b>	GPIO, SPI(SCLK), PWM, UART	N	P13
<b>P0.01</b>	GPIO, SPI(MISO), PWM, UART	N	P14
<b>P0.13</b>	GPIO, SPI(MOSI), PWM, UART	N	P15
<b>P0.20</b>	ACT_MIC	N	
<b>P0.21</b>	LEDFILA(1)	N	
<b>P0.22</b>	LEDFILA(2)	N	
<b>P0.15</b>	LEDFILA(3)	N	
<b>P0.24</b>	LEDFILA(4)	N	
<b>P0.25</b>	COMBI_SENSOR_INT	PTA1 / P0.09	
<b>P0.26</b>	(GPIO), I2C(SCL), (PWM), (UART)	N	P19
<b>P1.00</b>	(GPIO), I2C(SDA), (PWM), (UART)	N	P20
<b>P0.12</b>	(GPIO), ACCESIBILIDAD, (PWM), (UART)	N	P12
<b>P0.28</b>	LEDCOL(1)	N	P4
<b>P0.31</b>	LEDCOL(3)	N	P3
<b>P0.30</b>	LEDCOL(5)	N	P10

*Tabla 1. Mapa de pines de la Micro:bit*

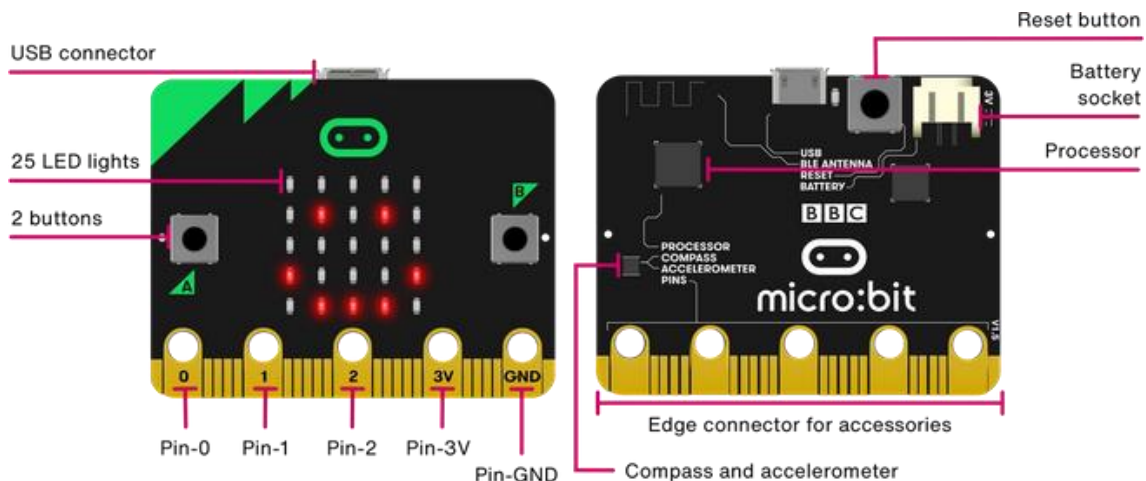
- **Sensor fotovoltaico:** Incluye un sensor de luz incorporado que permite detectar la intensidad lumínica en el ambiente. Está basado en un fototransistor [38] y es usado para determinar la luz incidente sobre la placa.

Este sensor proporciona una lectura analógica, permitiendo una configuración de rango de valores en los que se puede determinar los niveles de luz bajos o altos.

- **Radio integrada:** La placa permite la comunicación inalámbrica vía radio a partir de este componente, lo que crea la posibilidad de comunicar entre sí varias placas micro:bit u otros dispositivos como computadores y teléfonos.

La radio de 2.4 GHz [38], más que suficiente para este tipo de dispositivo, está diseñada para ser usada fácilmente. La tarjeta utiliza un protocolo de radio personalizado basado en la norma Bluetooth Low Energy (BLE), ideal para un terminal que requiere de un reducido consumo de energía. Este protocolo ofrece una comunicación con una tasa de transferencia de 1 Mbps en la capa física.

En la ilustración 3 se observa como todos estos componentes se distribuyen de la siguiente forma sobre la tarjeta [11]:



*Ilustración 3. Tarjeta micro:bit V1*

La placa en la segunda versión (Micro:bit V2), lanzada el 13 de octubre de 2020 presentó diferentes novedades respecto la versión inicial. Entre ellas se encuentran:

- **Micrófono digital MEMS con indicador LED:** La micro:bit monta un micrófono MEMS (micro-electromechanical systems) que permite la medida de una señal de entrada para el sonido captado. Se encuentra incorporado en la parte frontal e incluye un LED que se activa mientras el micrófono se encuentre encendido.

El micrófono diseñado por la empresa Knowles, un líder global proveedor de micrófonos acústicos avanzados, se trata concretamente del modelo SPU0410LR5H-QB-7 [38]. Este tiene una sensibilidad de  $-38 \text{ dB} \pm 3\text{dB}$  e incluye una relación señal-ruido (SNR) de 63 dB

- **Altavoz integrado:** Además de permitir la posibilidad de emitir sonido conectando un altavoz a través de los pines en una señal PWM, en este nuevo modelo se incluye un altavoz magnético montado en la PCB (circuito impreso) por el cual se puede emitir la salida de sonido.

El altavoz con modelo MLT-8530 [38] de la compañía Jiangsu Huaneng permite una presión sonora de 80 dB a 5V, con una frecuencia de resonancia propia de 2700 Hz.

- **Logo con botón táctil:** El conocido logo incluido en la tarjeta a la vez hace las funciones de botón táctil. Este botón es capacitivo y permite añadir funcionalidades a los ya presentes botones de la micro:bit: botón A, botón B, botón A+B y botón táctil.

Este nuevo modelo incorpora nuevas funciones, como el modo de espera, que detendrá el programa en curso hasta que el botón de reinicio no sea pulsado. Esta función sirve para momentos en los que el proyecto sea pausado sin la necesidad de desconectar la micro:bit.

Además, contribuye a reducir el consumo de energía en los momentos en los que no sea necesario tener activo el programa cargado, optimizando así el funcionamiento general de la placa.

Por último, cabe destacar que este último modelo consta de un procesador actualizado, el Nordic nRF52833, basado en un ARM Cortex M4 de 64MHz + FPU [9], cuatro veces más rápido que el anterior. Además, la memoria de almacenamiento Flash se duplica (512 KB) y la memoria RAM pasa a ser de 128 KB.



En cuanto a las nuevas funciones y elementos incorporados, existe una compatibilidad de hardware y software respecto a anteriores modelos, de forma que todos los programas pueden ser reconstruidos para funcionar con este último modelo de la micro:bit

En definitiva, viendo el éxito que tuvo la placa en un primer lanzamiento, desde Micro:bit decidieron ampliar aún más las posibilidades que brindaba esta placa, contribuyendo a su propósito general de enseñar a programar fácilmente de una manera divertida y al alcance de cualquier usuario.

En la imagen 4 se muestra la distribución de dispositivos y sensores de la más novedosa tarjeta micro:bit V2 en su parte frontal y trasera [11]:

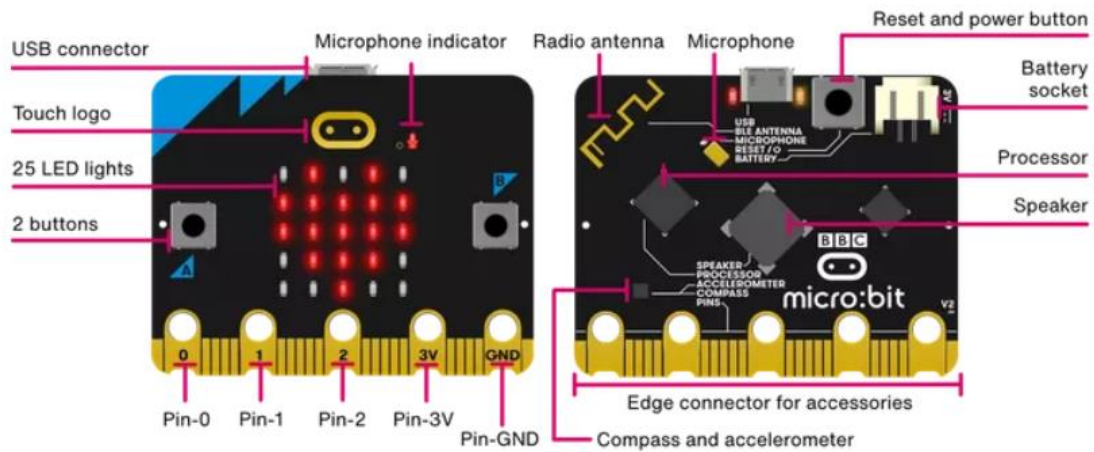


Ilustración 4. Tarjeta micro:bit V2

El diagrama de bloques de ambas tarjetas es el incluido en la imagen 5 [38]:

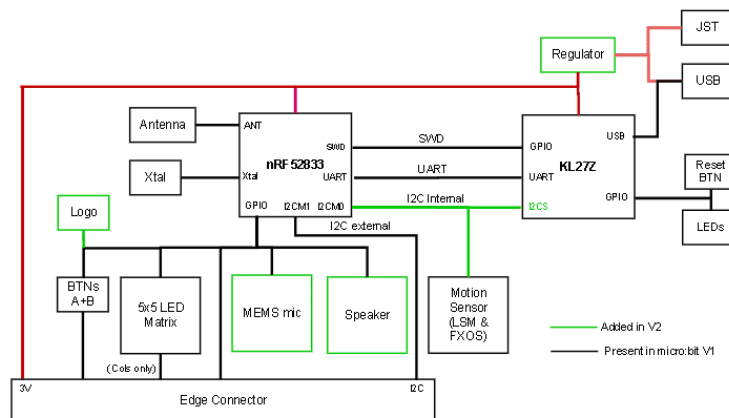


Ilustración 5. Diagrama de bloques de la Micro:bit



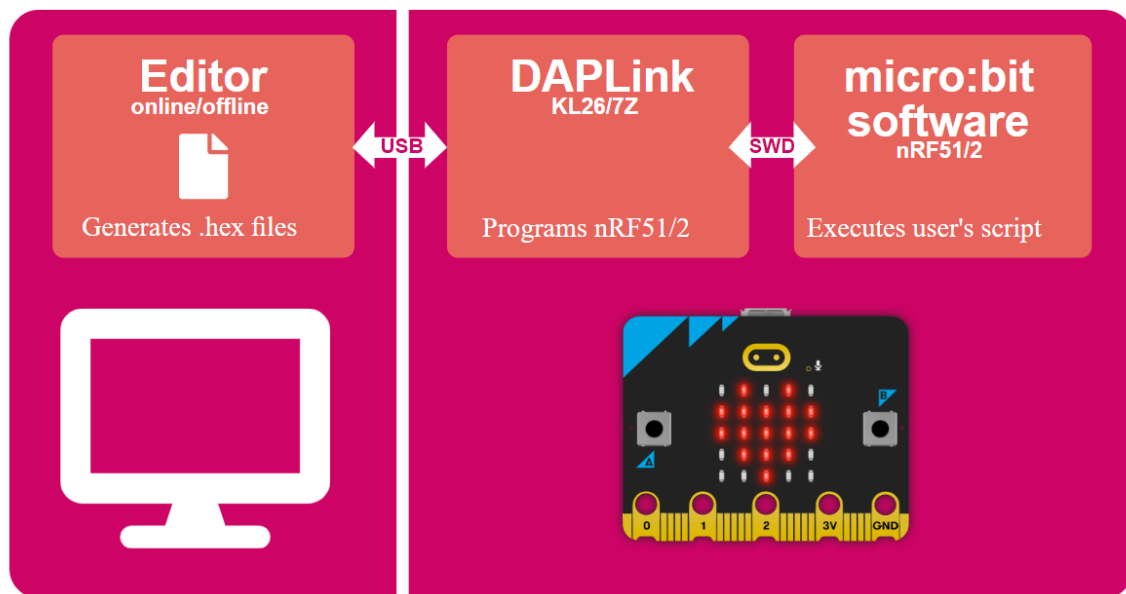
# **CAPITULO 3: CARACTERÍSTICAS SOFTWARE Y PROGRAMACIÓN**

### 3.1. Características software

El software que ofrece la tarjeta micro:bit está basado en dos grupos principales:

1. El software ejecutado en una computadora / dispositivo móvil-tablet
2. El software ejecutado sobre la propia tarjeta

El transcurso habitual para iniciar un proyecto está basado en diseñar inicialmente el programa de una manera simulada a partir del editor online/offline y más tarde, cuando se haya validado el correcto funcionamiento del boceto, transferirlo a la tarjeta micro:bit para ejecutar el programa en la realidad. Para ello, la tarjeta consta de una vía de comunicación puerto USB. Se ilustra en la imagen 6 la conexión entre dispositivos [10]:



*Ilustración 6. Sistema de software de la micro:bit*

La tarjeta monta dos chips, uno encargado de ejecutar el software DAPLink para facilitar la actualización del programa cargado (KL26/KL27) y otro que ejecuta el código diseñado por el usuario (nRF51/nRF52).

## 3.2. Programación

Los lenguajes de programación que se pueden aplicar a la tarjeta se pueden clasificar en dos categorías diferentes [10]:

1. **Lenguajes compilados:** El programa se compila en el lenguaje ensamblador ARM o cualquier otro tipo de código de bytes antes de ejecutarlo en la tarjeta.
2. **Lenguajes interpretados:** Se ejecuta el programa línea por línea y a la vez se ejecuta cada comando.

En la práctica, y sobre todo entre los usuarios a los que ha sido destinada la micro:bit, siempre será programada en un idioma interpretado. A continuación, se estudiará cual es el método que se llevará a cabo para los diferentes lenguajes disponibles.

El entorno que ofrece micro:bit para la programación es MakeCode (previamente conocido como PXT). MakeCode es un entorno de programación online que hace uso de bloques funcionales y encadenables para generar programas. Este entorno es muy accesible para cualquier usuario que quiera iniciarse en la programación.

Cabe destacar que se puede programar en otros lenguajes de programación como MicroPython, C/C++, pero primeramente se aborda este lenguaje más sencillo. En la imagen 7 se muestra el entorno [3]:

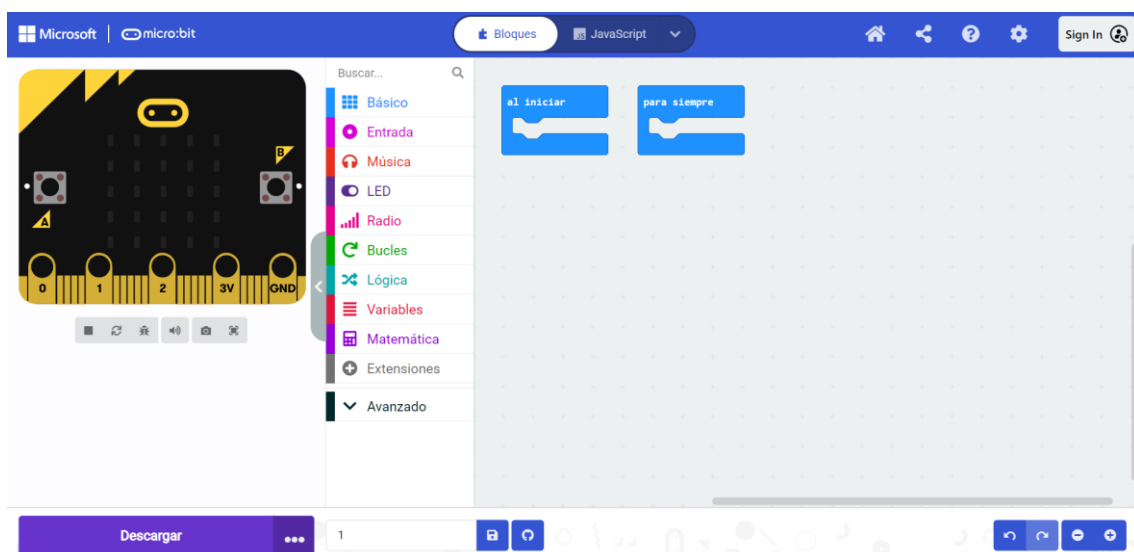


Ilustración 7. Entorno de programación MakeCode

Para programar a partir de MakeCode, a través del arrastre y conexión de bloques de código se generan programas. Los bloques representan funciones o instrucciones y permiten programar interacciones simples.

En este mismo entorno se pueden realizar las simulaciones necesarias antes de aplicar el programa a la tarjeta real, revisando el correcto funcionamiento a partir de la imagen dinámica que se muestra en la parte izquierda del entorno de programación.

Una vez el programa haya sido creado, se puede descargar en la tarjeta Micro:bit conectando la placa con el ordenador a partir de una unidad USB. Simplemente se debe marcar “Conect Device” en los tres puntos que se encuentran junto a la pestaña de “Descargar” y seguir las indicaciones. Mientras el programa se ejecute se mantendrá representado en la tarjeta micro:bit.

Si el anterior paso ha sido realizado, el programa quedará guardado en la memoria de la tarjeta, por lo que puede ser desconectado del PC y ejecutarlo de forma independiente conectándolo a una fuente de alimentación.

Otro método de descarga del programa en la tarjeta es a partir de la descarga del archivo .hex y cargarlo directamente en la unidad micro:bit.

La otra versión popular de lenguaje de programación que ofrece micro:bit es MicroPython, una alternativa a muy similar al lenguaje Python. Es un lenguaje adaptable a múltiples entornos:

- Editor Python en la web de micro:bit
- Editor MU sin conexión
- Cualquier editor en el que se creen archivos Python

MicroPython es una reimplementación completa de Python 3, lo que ofrece características avanzadas que no se encuentran en ninguno de los otros lenguajes. Y al igual que MakeCode, MicroPython se ejecuta en el propio micro:bit o en su entorno de simulación, sin la necesidad de un compilador.

La máquina virtual de Python que se encuentra en MicroPython se compila a partir de código C++. Su función es leer códigos de bytes de Python e interpretarlos uno por uno.

En este proceso, llama a funciones C++ de nivel inferior, cada una diseñada para cumplir una tarea específica.

Este enfoque utiliza un intérprete de códigos de bytes y permite a MicroPython crear una máquina virtual con su propio conjunto de instrucciones virtuales, las cuales son virtuales, ya que no están directamente integradas en el hardware del sistema, sino que se implementan en el software. Esta característica es lo que facilita que MicroPython sea portable y adaptable a diferentes sistemas informáticos con diversos tipos de procesadores.





# **CAPITULO 4:**

## **PRUEBAS DE FUNCIONALIDAD**

Las tarjetas del departamento de Electrónica de la Universidad de Alcalá son utilizadas por estudiantes que no tienen conocimientos técnicos en el uso de componentes electrónicos y en ocasiones realizan conexiones que deterioran los componentes de las placas. Cuando se realiza una actividad relacionada con la electrónica, se requiere que el usuario, ya sea un niño o un estudiante universitario, tenga a su mano un material que cumpla los estándares óptimos de rendimiento y funcionalidad. En este contexto, el trabajo se enfoca en la creación de un sistema de control de calidad específicamente diseñado para las tarjetas Micro:bit, las cuales tienen aplicación dentro de la Universidad de Alcalá, en concreto en varias asignaturas del grado de Magisterio.

El uso de las tarjetas a lo largo de los años ha provocado que algunas fallen en algunos componentes y funcionalidades y el objetivo de este proyecto es crear un sistema práctico que compruebe el correcto funcionamiento de los diferentes componentes de las placas.

El sistema de control de calidad lleva a cabo pruebas exhaustivas en cada uno de los componentes y periféricos de la tarjeta Micro:bit. Esto incluye verificar el funcionamiento del microcontrolador, los sensores, la matriz de LEDs y otros elementos clave. Las pruebas se llevan a cabo siguiendo procedimientos marcados que aseguren resultados confiables.

En primer lugar, se recopilan resultados a través de la propia tarjeta Micro:bit, que es programada para realizar autoevaluaciones y proporciona resultados detallados sobre el estado y funcionamiento del componente a prueba. Para marcar el orden de las pruebas a realizar, se lanzan transmisiones a través de un dispositivo conectado que actúe como un control externo, en este caso otra micro:bit. Este punto garantiza que el proceso de control de calidad sea preciso y efectivo.

En los siguientes puntos se van a exponer las distintas pruebas a las que va a ser sometida la tarjeta micro:bit con el objetivo de confirmar el correcto funcionamiento de cada una de las partes. Se hace una explicación de la prueba con una prueba gráfica. El código empleado se encontrará en los anexos del documento.

La dinámica que tiene el proceso asegura la máxima automatización de la prueba requiriendo de la menor intervención humana posible. La tarjeta probadora, o usualmente llamada en este proyecto como “madre”, es la encargada de enviar vía radio a la tarjeta puesta a prueba (o hija) que funcionalidad se quiere poner a prueba. Una vez se haya concluido la prueba particular de funcionalidad, vía comunicación serie se informa de los resultados obtenidos. De esta forma, cuando la prueba de calidad sobre la tarjeta hija haya sido finalizada, se obtiene un informe detallado del estado de cada punto examinado.

La tarjeta probadora tiene cargado un programa que establece inicialmente una frecuencia de radio para la comunicación con la tarjeta probada y fija un contador a cero. El valor de este contador marca la prueba a realizar sobre la tarjeta hija, aumentando cada vez que el botón “A” es presionado y disminuyendo al pulsar “B”, permitiendo retroceder entre las pruebas.

En la ilustración 8 se encuentra el cronograma de la prueba para conocer la secuencia que es llevada y entre los anexos se encuentra un manual de usuario para conocer rápidamente los pasos a llevar a cabo.



*Ilustración 8. Organigrama de las pruebas de funcionalidad*

## 4.1. Matriz de LEDs (0)

La matriz de LEDs es uno de los componentes más distintivos y útiles de la Micro:bit. Permite mostrar información visual, crear juegos y proyectos interactivos, y es una parte esencial de muchas aplicaciones. Asegurarse de que esta matriz funcione correctamente es crucial para el uso exitoso de la tarjeta en una variedad de proyectos.

La primera prueba de funcionalidad sobre la tarjeta consiste en un programa muy sencillo que examina cada uno de los LEDs de la matriz. Para ello, se hace uso de una secuencia de encendido y apagado de la matriz, y se enviará el diagnóstico vía serial. Al tratarse de un punto no comprobable sin alterar el esquema electrónico de la tarjeta micro:bit, lo que modificaría su diseño de fábrica, se enviará un mensaje vía serie USB que haga comprobar al usuario el correcto encendido de la matriz.

Esta es la prueba más sencilla de todas, pero es clave para conocer el estado de los LEDs y comprobar que son completamente funcionales. El proceso será el mostrado en la ilustración 9 [3]:

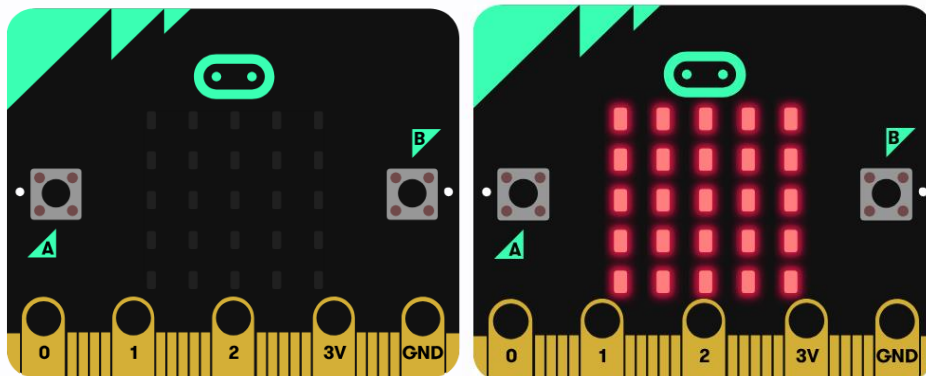


Ilustración 9. Prueba de funcionalidad de la matriz de LEDs.

Mensaje recibido en el informe vía serie USB será como se muestra en la ilustración 10:

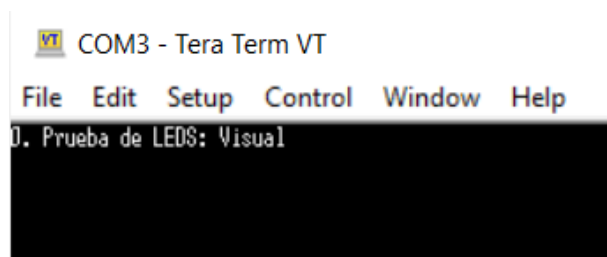


Ilustración 10. Mensaje en el informe para la prueba de funcionalidad de la matriz de LEDs

## 4.2. Botones programables (1 y 2)

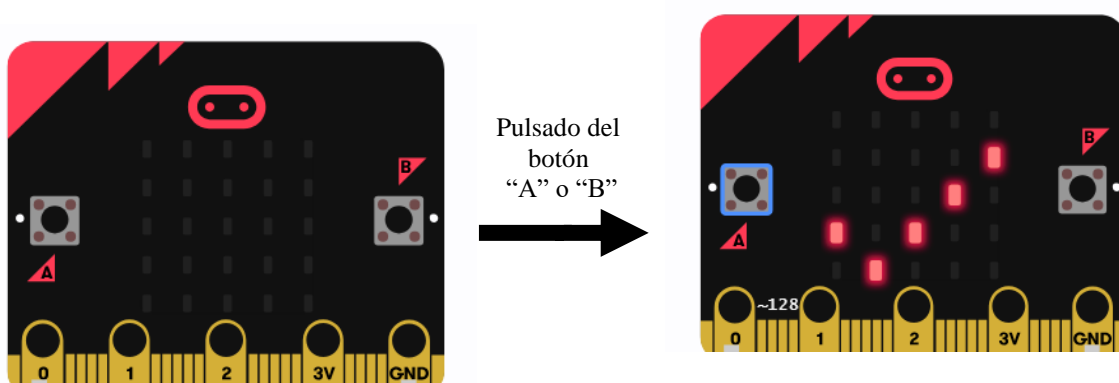
Los botones son la interfaz principal para interactuar con la Micro:bit. Si no funcionan correctamente, la experiencia del usuario se verá afectada negativamente. Las pruebas garantizan que los usuarios puedan utilizar los botones de manera efectiva, lo que es especialmente importante en proyectos y aplicaciones interactivas.

Los botones incorrectamente configurados o defectuosos pueden llevar a errores en la entrada de datos o en la ejecución de programas. Las pruebas aseguran que los botones se comporten según lo previsto y evitan problemas de funcionamiento no deseado.

La prueba consiste en un programa diseñado para que cuando presione el botón "A" en la prueba 1 y el botón "B" en la 2 en la tarjeta Micro:bit probada, el programa muestre un ícono de verificación, y luego borra la pantalla para enviar un mensaje vía serie en el que indica que los botones se encuentran en correcto estado. En el caso en el que no se detecte su pulsado, el botón no funcionará.

En el caso que se detecte que el botón se mantiene pulsado tras dejar de presionarlo, lo que supondría un defecto de este, se transmitirá un mensaje de error y se mostrará un "X" en la matriz.

La representación gráfica de la prueba sería la representada en la ilustración 11 [3]:



*Ilustración 11. Prueba de funcionalidad de los botones programables.*

A continuación, sobre la ilustración 12 se muestra un ejemplo de botón A en correcto estado y un botón B deteriorado:

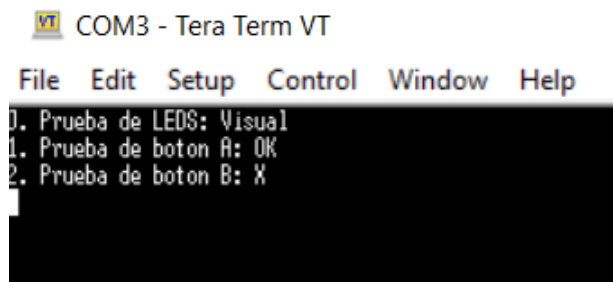


Ilustración 12. Transmisión prueba botones

### 4.3. Logo sensible al contacto (3)

El logo sensible al contacto es un componente específico de la Micro:bit y forma parte de su conjunto de características únicas. Realizar pruebas en este componente es esencial para garantizar que funcione como se espera y que los usuarios puedan interactuar con él de manera efectiva.

La prueba que se ha planteado para comprobar su correcta funcionalidad se basa en un programa muy sencillo que está diseñado para responder a eventos en los que el logo sensible al tacto de la Micro:bit es presionado. El programa muestra un ícono de verificación en la pantalla de LEDs durante 1 segundo y luego borra la pantalla, por último, emite un mensaje vía serie de confirmación de la prueba. Si el logo inteligente fuese defectuoso, no se reflejaría en la matriz de LEDs el código de validación ni se transmitiría el mensaje.

En el caso que la tarjeta puesta a prueba se trate de una versión V1, la cual no incluye este botón, se transmitirá vía serie el mensaje “Prueba de logo no disponible en V1”.

En la tarjeta quedaría representado como en la imagen 13 [3]:

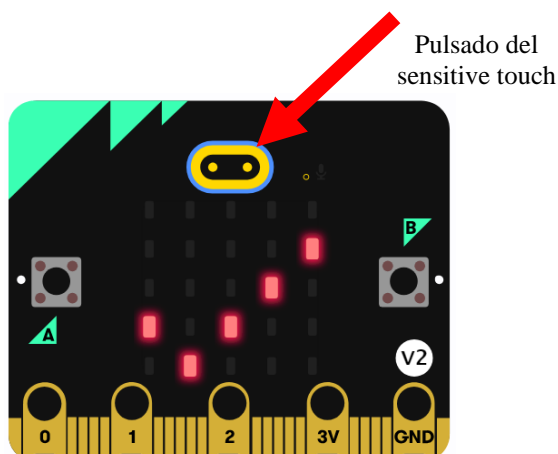
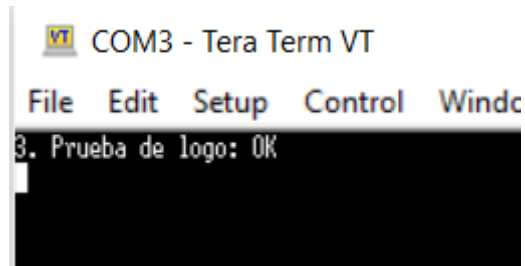


Ilustración 13. Prueba de funcionalidad del logo sensible.

En la imagen 14 se muestra el mensaje transmitido vía serie:



*Ilustración 14. Mensaje transmitido prueba del logo*

## 4.4. Micrófono (4)

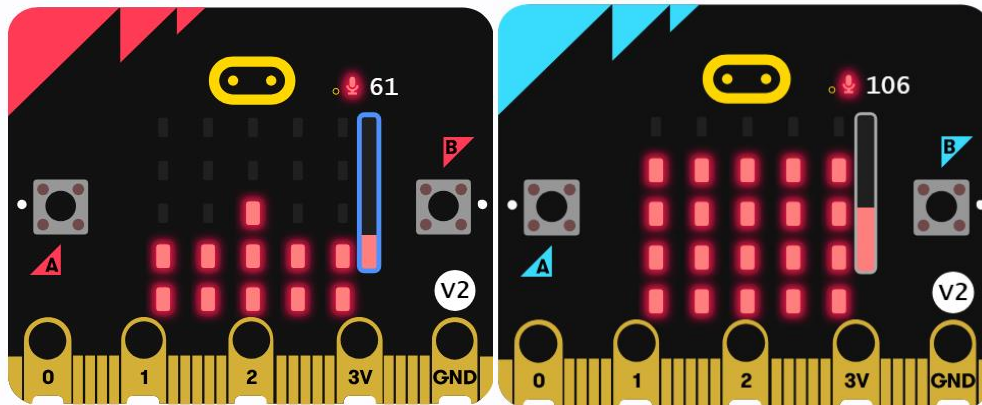
El micrófono permite a la Micro:bit detectar y registrar sonidos del entorno. Puede ser utilizado en una variedad de aplicaciones, como proyectos de detección de sonido, grabación de voz y proyectos de música. Las diferentes pruebas que se proponen aseguran que el micrófono funcione correctamente en estas aplicaciones.

Para llevar a cabo esta prueba, se debe hacer uso de la tarjeta Micro:bit V2, ya que sus anteriores versiones no tenían incluido este componente. De todas formas, haciendo uso de un periférico externo se podría adaptar ligeramente el código para realizar la prueba sobre este mismo.

La primera prueba que se plantea se trata de un programa que permite medir y mostrar niveles de sonido en una Micro:bit, al tiempo que registra y muestra el nivel máximo de sonido detectado. La lógica del programa es la siguiente: Si vía radio se está recibiendo la señal que activa la prueba (contador es igual a 4), envía cada vez que se pulsa el botón A de la tarjeta probada vía serie el valor registrado junto un mensaje que indica el valor de sonido percibido en dB, mientras muestra continuamente el nivel de sonido actual en forma de un gráfico de barras en la pantalla LEDs.

El programa de la tarjeta madre se encargará de enviar constantemente vía radio a la tarjeta hija el valor del contador (4) mientras este se mantenga. Hasta que no se vuelva a pulsar el botón A de la tarjeta madre, el valor del contador no se verá incrementado.

Se muestran casos gráficos de funcionamiento del programa en la imagen 13[3]:



*Ilustración 15. Prueba de funcionalidad del micrófono*

A partir de los valores percibidos por el micrófono, la pantalla de LEDs mostrará un gráfico de barras con valores más o menos pronunciados.

La escala de valores que puede percibir el micrófono es de 100 Hz hasta 80 kHz con una sensibilidad de  $-38\text{dB}\pm 3\text{dB}$  a  $94\text{dB}$ , que son representados en este caso en una escala de 0 a 255.

En la ilustración 16 se incluye el mensaje recibido vía serie para esta prueba:

```
VT COM3 - Tera Term VT
File Edit Setup Control Window Help
0. Prueba de LEDs: Visual
1. Prueba de boton A: OK
2. Prueba de boton B: OK
3. Prueba de logo: OK
4. Valores nivel sonido: 7dB
4. Valores nivel sonido: 45dB
7
```

*Ilustración 16. Mensaje en el informe para la prueba del micrófono.*



## 4.5. Altavoz (5)

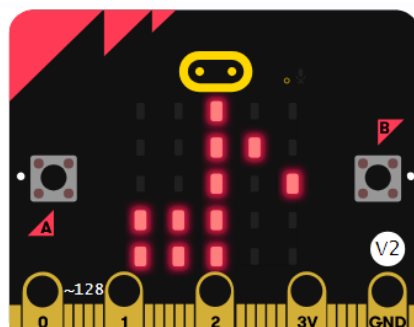
Aunque el altavoz ya ha entrado dentro del programa de prueba de funcionalidad de los botones programables, es necesario tener una prueba independiente para el altavoz. Esta nos servirá para que el método de comprobación del altavoz no dependa de otros periféricos completamente diferentes.

Por ello se va a incluir un programa sumamente sencillo que genere una melodía cuando la constante del contador llegue a 4, permitiendo así comprobar el estado del altavoz.

En definitiva, este programa hace que la Micro:bit muestre un icono de una nota musical [3] y reproduzca una melodía musical de manera única. La melodía se compone de una serie de notas musicales definidas en cadena. En el caso de que se reproduzca la melodía el altavoz funcionará correctamente, de lo contrario se dará como “KO”. Vía serie se transmitirá un mensaje para que el usuario este atento a la reproducción de la melodía.

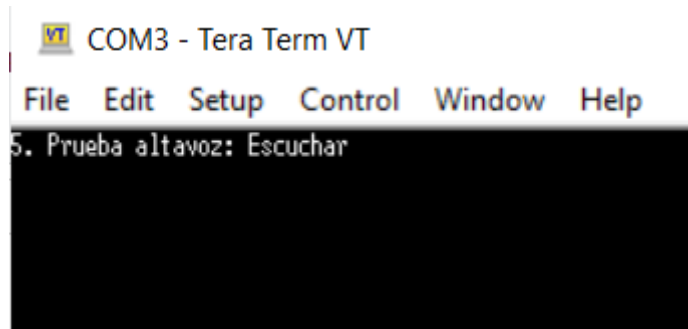
En el caso que la tarjeta puesta a prueba se trate de una versión V1, la cual no incluye este botón, se transmitirá vía serie el mensaje “Prueba de altavoz no disponible en V1”.

Visualmente en la tarjeta hija se mostraría tal y como se representa en esta ilustración 17 [3]:



*Ilustración 17. Prueba de funcionalidad del altavoz.*

En la ilustración 18 se incluye el mensaje transmitido vía serie:



*Ilustración 18. Mensaje transmitido prueba altavoz*

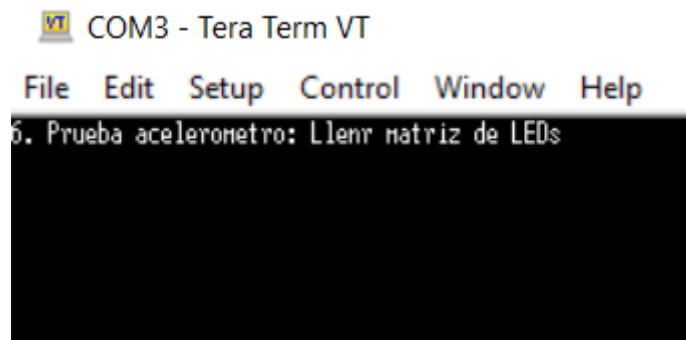
## 4.6. Acelerómetro (6)

En esta parte del proyecto se va a comprobar el correcto funcionamiento del sensor integrado en la Micro:bit, acelerómetro. Para ello, se va a diseñar un sensor de aceleración sobre las coordenadas X,Y y Z.

La calibración es un paso importante para asegurar que la brújula señale correctamente. Para realizar esta, se seguirán las indicaciones al iniciar el programa. Esto implica mover la Micro:bit en patrones específicos hasta completar el encendido de todos los LEDs de la matriz.

Al igual que en la prueba del micrófono, la tarjeta madre se encargará de enviar constantemente vía radio a la tarjeta hija el valor del contador (6) mientras este se mantenga. Hasta que no se vuelva a pulsar el botón A de la tarjeta madre, el valor del contador no se verá incrementado.

El objetivo de la prueba será completar el encendido de la matriz de LEDs mediante uso de giros y desplazamientos de la tarjeta micro:bit, consiguiendo así validar el correcto funcionamiento del acelerómetro. Para iniciar la prueba se debe pulsar el botón A de la tarjeta hija, mostrándose posteriormente el mensaje incluido en la ilustración 19:



*Ilustración 19. Transmisión prueba acelerómetro*

## 4.7. Brújula (7)

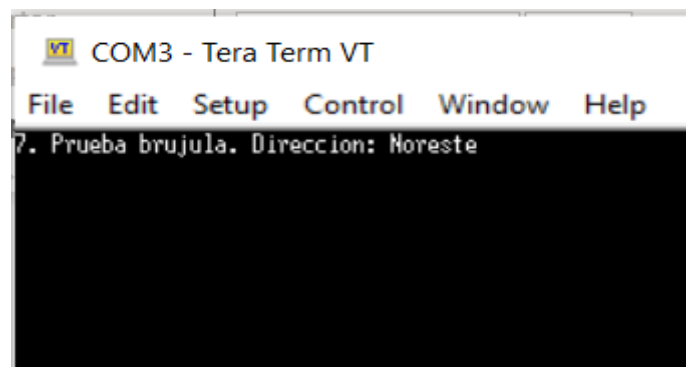
En esta prueba, para comprobar la correcta funcionalidad de la brújula, la tarjeta madre enviará vía radio el valor del contador que activa esta prueba (7). El programa de la tarjeta hija funciona a partir del valor recibido vía radio y el pulsado del botón A para enviar el valor.

A continuación, el programa muestra una flecha que indica en cada momento el norte magnético en función de la dirección apuntada:

- Si dirección es igual a 0 grados, muestra una flecha hacia el norte (N).
- Si dirección es 90 grados, muestra una flecha hacia el este (E).
- Si dirección es 180 grados, muestra una flecha hacia el sur (S).
- Si dirección es 270, muestra una flecha hacia el oeste (O).
- Si la dirección se encuentra entre 0 y 90 grados, muestra una flecha hacia el noreste (NE).
- Si la dirección se encuentra entre 90 y 180 grados, muestra una flecha hacia el sureste (SE).
- Si la dirección se encuentra entre 180 y 270 grados, muestra una flecha hacia el suroeste (SO).
- Si la dirección se encuentra entre 270 y 0 grados, muestra una flecha hacia el noroeste (NO).

De esta forma, la Micro:bit funciona como una brújula digital. Esta prueba se completará haciendo uso de una brújula perfectamente calibrada, ya sea analógica o digital, que permita al usuario determinar si el sensor integrado se encuentra en óptimas condiciones dentro de la tarjeta.

En la imagen 20 se muestra el mensaje transmitido:



*Ilustración 20. Transmisión prueba brújula*

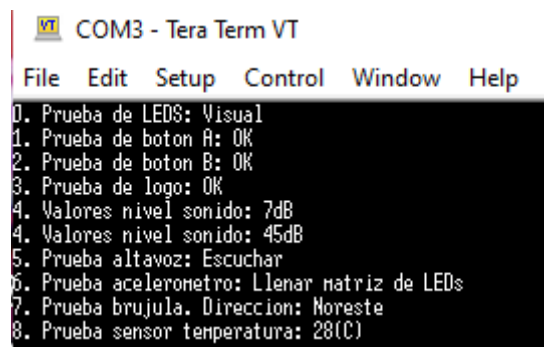
## 4.8. Sensor de temperatura (8)

Una de las funcionalidades de la tarjeta más útiles y aplicables a diferentes proyectos es la capacidad de medir la temperatura ambiental a partir del sensor integrado de temperatura.

Para comprobar la correcta funcionalidad de este sensor, se ejecutará un programa con objetivo de ser comparado con un sensor analógico o digital perfectamente calibrado. Este permite mostrar la temperatura en escala Celsius sobre la matriz de LEDs.

Para activar esta prueba se seguirá el mismo funcionamiento que en las anteriores, cuando el valor del contador se reciba en la tarjeta hija, la prueba comenzará y enviará vía serie el valor obtenido, de forma que se siga completando el informe de pruebas.

En la imagen 21 se muestra el mensaje transmitido:



```
COM3 - Tera Term VT
File Edit Setup Control Window Help
0. Prueba de LEDs: Visual
1. Prueba de boton A: OK
2. Prueba de boton B: OK
3. Prueba de logo: OK
4. Valores nivel sonido: 7dB
4. Valores nivel sonido: 45dB
5. Prueba altavoz: Escuchar
6. Prueba acelerometro: Llenar matriz de LEDs
7. Prueba brujula. Direccion: Noreste
8. Prueba sensor temperatura: 28(C)
```

*Ilustración 21. Mensaje en el informe para la prueba del sensor de temperatura*

## **4.9. Pines de entrada o salida. (9)**

### **4.9.1 Placa de circuito impreso (PCB)**

Para comprobar el correcto funcionamiento de los puertos a testear, en el proyecto utiliza una placa de circuito impreso (PCB - Printed Circuit Board) que sirva para conectar los pines de la tarjeta micro:bit.

Para abordar este asunto de manera completa, se necesita considerar múltiples aspectos del diseño, fabricación y ensamblaje de PCBs. El motivo de probar los puertos sobre una PCB es debido a que estas ofrecen un medio eficiente y seguro para establecer la conexión entre ambas micro:bit. En este proyecto se han tenido en cuenta dos versiones: una inicial prototipo llamada v1 que ha servido de referencia para conocer los puntos a mejorar para una v2 final.

### **4.9.2 Pruebas sobre los puertos de la micro:bit**

#### **4.9.2.1 Transmision de señales digitales**

Inicialmente se transmitirá una señal digital a través de la microbit puesta a prueba y que está sea recibida en la tarjeta “madre”.

Como la micro:bit cuenta con múltiples puertos GPIO, en una primera versión de PCB se hará uso de un multiplexor analógico de 16 canales, el CD74HC4040E de Texas Instruments, un multiplexor, también conocido como MUX, que permite conectar varias señales de entrada en una sola salida. Por otro lado, también puede ser utilizado como demultiplexor o demux, permitiendo distribuir una señal única a múltiples salidas. La señal de salida vendrá determinada por el valor de los pines SELECT del dispositivo.

El CD74HC4067 [26] es versátil, ya que trabaja tanto con señales digitales como analógicas, tanto de entrada como de salida. Esta capacidad de manejar diferentes tipos de señales le otorga la designación de amplificador de entradas y salidas, conocido como I/O expander.

Puede proporcionar un máximo de 20 mA, un valor limitado pero asumible para la aplicación dentro del proyecto de la micro:bit. En cuanto a su tensión de alimentación es de 2 a 6 Voltios.

Es un chip muy adecuado, no solo por las especificaciones que ofrece si no por su reducido coste y operatividad para realizar las pruebas antes de construir la tarjeta. Aproximadamente en el

mercado actual se puede encontrar en diferentes tiendas por un valor que ronda 1 euro, aunque como se comentará posteriormente en el mercado actual se encuentra mayormente descatalogado.

En la imagen 22 se muestra como se suele encontrar habitualmente, con una placa de circuito impreso para facilitar su uso [26]:



*Ilustración 22. Multiplexor CD74HC4067*

El multiplexor va soldado a la placa PCB, de forma que una única señal sea canalizada a la tarjeta “madre”. Este punto será detallado más adelante.

Una vez queda clara la estructura de la prueba, queda definir el desarrollo de esta. Para ello, se hará uso de un programa que ponga a prueba los pines de entrada y salida mediante una placa protoboard.

A continuación, se explican los programas cargados en la tarjeta madre e hija para hacer la prueba de comprobación de pines con señales digitales:

**Tarjeta madre:** En el caso de llegar a la prueba escritura de la tarjeta hija mediante sus pines I/O (valor del contador igual a 9), se escriben vía pines P1, P2, P3 y P4 los valores necesarios para dar salida a la señal CO del multiplexor. Una vez se selecciona esta entrada deseada, se lee en el puerto P0 el valor de esta señal, y en función del valor recibido vía radio se envía un valor consigna.

Este proceso se automatiza para cada uno de los pines contiguos, modificando los valores definidos mediante los pines P1-P4 para dar salida a la señal deseada y diagnosticando el estado de todos los pines de la tarjeta puesta a prueba.

Posteriormente se realiza la misma comprobación para la lectura de una escritura a nivel bajo de la tarjeta hija.

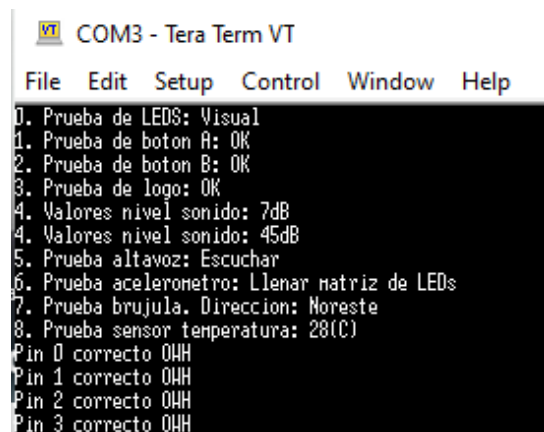
Y, por último, se prueba la lectura de valores digitales de la tarjeta probada, siendo ahora la tarjeta madre quien envíe valores a cada uno de los 16 pines y se haga secuencialmente primero una lectura de un valor alto y una lectura a nivel bajo.

**Tarjeta hija:** Para comprobar que funcionan correctamente los pines I/O de la tarjeta hija examinada, cuando se inicie esta prueba se establece en un bucle un valor alto en todos sus pines para ir comprobando el correcto funcionamiento de cada uno de ellos. Estos irán conectados vía PCB al multiplexor, esperando su selección vía la tarjeta madre.

Por otra parte, vía radio se recibirán las consignas enviadas por la tarjeta madre para indicar el resultado obtenido en la lectura del pin 0 de la tarjeta madre. En el caso de hacer la lectura de un valor en alto, la consigna indicará “Pin X correcto OWH”, y para lo contrario la consigna indicará “Pin X KO OWH” .

Esta misma prueba se repetirá consecutivamente enviando un valor bajo y recibiendo vía radio el resultado obtenido en el pin 0 de la tarjeta madre. En el caso de hacer la lectura de un valor en bajo, la consigna indicará “Pin X correcto OWL”, y para lo contrario la consigna indicará “Pin X KO OWL” .

En la imagen 23 se muestra el formato en que es transmitido el resultado vía serie:



```
VT COM3 - Tera Term VT
File Edit Setup Control Window Help
0. Prueba de LEDs: Visual
1. Prueba de boton A: OK
2. Prueba de boton B: OK
3. Prueba de logo: OK
4. Valores nivel sonido: 7dB
4. Valores nivel sonido: 45dB
5. Prueba altavoz: Escuchar
6. Prueba acelerometro: Llenar matriz de LEDs
7. Prueba brujula. Direccion: Noreste
8. Prueba sensor temperatura: 28(C)
Pin 0 correcto OWH
Pin 1 correcto OWH
Pin 2 correcto OWH
Pin 3 correcto OWH
```

*Ilustración 23. Formato del mensaje vía serie para la prueba de pines.*

Por último, se recibirán por cada uno de los pines de la tarjeta madre valores primeramente en valor alto y posteriormente en valor bajo para comprobar la correcta lectura de la tarjeta hija frente a señales digitales.

### 4.9.2.2 Transmisión de señales analógicas

En una versión 2 de la tarjeta se plantea comprobar la lectura de los pines analógicos de la tarjeta hija. Para ello se debe hacer uso de un conmutador (no incluido en la primera versión) que permita alternar según la prueba la señal digital o analógica generada por la tarjeta probadora.

Los programas cargados en las tarjetas incluirán todo el código mencionado hasta ahora y además, incluirán la siguiente parte:

**Tarjeta madre:** En el caso de haberse comprobado todos los pines mediante escritura y lectura de señales, se llegará a la prueba de entradas analógicas.

El código se encarga de ir modificando los valores establecidos en los pines S0-S3 para transmitir a los pines deseados de la tarjeta puesta a prueba. Y por otro lado, mediante el pin 16 se escribirá una señal analógica, basada en un PWM, filtrada en un filtro RC y que pasará por el conmutador para llegar al pin COM del multiplexor. El paso de esta señal mediante el conmutador lo proporcionará un pin de la tarjeta probada, tal y como se comentará posteriormente.

Simultáneamente se enviará un código consigna para que en la tarjeta hija se haga una lectura en los pines de las señales analógicas enviadas.

**Tarjeta hija:** Ahora se distinguen dos funciones no incluidas en el anterior código y son la función “Pruebas analógicas” y la función “Pruebas Digitales”. Estas se encargan de establecer un valor digital en el pin 16 de la tarjeta hija, el cual irá conectado al pin de selección del conmutador, dando paso a la señal analógica o digital en función del valor establecido. Un valor alto en este pin dará paso a la señal analógica y un valor bajo a la digital.

Los valores consigna comentados en el código de la tarjeta madre provocarán la lectura de los pines puestos a prueba. Vía serie se transmitirá el valor de la señal en Voltios, los cuales podrán ser regulados por un potenciómetro. El formato enviado en el mensaje del informe será: “Valor de tensión leído PX: X V”

Los pines sobre los que se requerirá hacer una lectura analógica son los pines P0-P4 y P10, ya que son los que tienen esta funcionalidad habilitada.





# **CAPITULO 5.**

## **DISEÑO DE LA PLACA DE CIRCUITO IMPRESO**

Una vez ha quedado definido y probado el software necesario para probar los pines de la tarjeta, se puede plantear el diseño de la placa PCB que realice la integración de todas las conexiones necesarias.

En este proyecto se trabaja con dos versiones:

1. Una versión v1 prototipo de placa, la cual ha sido mandada a construir y ha servido para detectar todos los fallos y mejoras implementables en futuras versiones.
2. Una versión v2 que corrige todos los fallos y deficiencias encontradas en la versión v1 al construir la placa.

## 5.1. Esquema general de conexiones versión v1

La tarjeta es lo suficientemente grande para alojar la Microbit en un módulo de acoplo, y es por eso por lo que las dimensiones finales de la placa, incluyendo todos sus componentes es de 11,7x10,7 cm.

Las conexiones, de manera genérica, vienen reflejadas en el esquema gráfico de la ilustración 24.

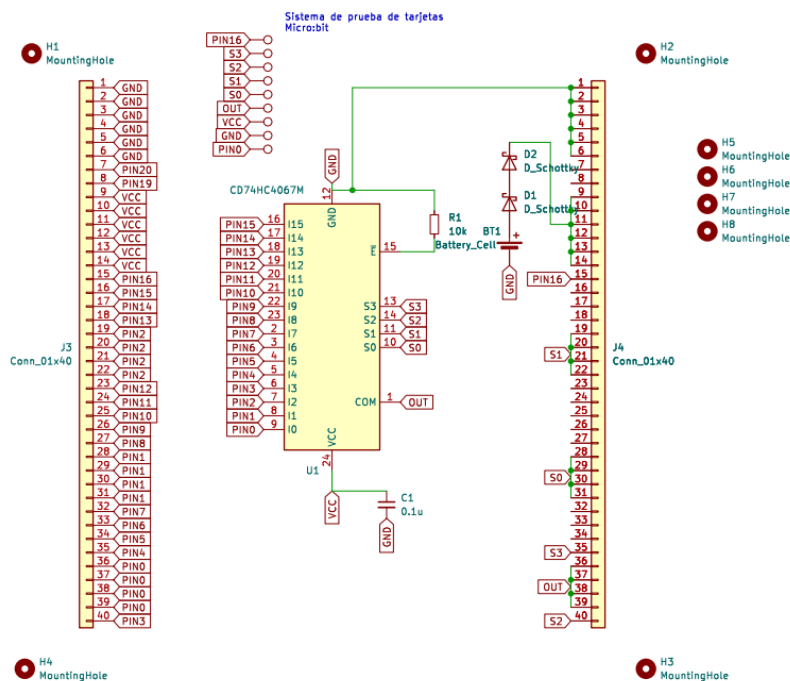


Ilustración 24. Esquemático del PCB v1

La tarjeta hija tiene los 16 pins I/O conectados a los pines del multiplexor transmitiendo sobre ellos la señal deseada.. Este modelo alternativo que abierto como una opción adicional en caso de un deteriorado del chip de radio.

Por otra parte, los pines de selección del multiplexor irán conectados a los puertos P0-4 de la tarjeta madre para hacer lectura de la señal recibida y seleccionar la señal a dar paso. La tarjeta madre podrá ser conectada vía USB o mediante una batería externa con los pines habilitados sobre la PCB para alimentar la tarjeta (VCC-GND)

### **5.1.1. Diseño de la placa PCB**

Para el diseño de la placa se ha hecho uso del software abierto KiCad. Este nos proporciona las herramientas necesarias para el diseño electrónico de la placa de circuito impreso, permitiendo la creación del esquemático del circuito y el diseño posterior de la placa.

Mediante su interfaz gráfica, y añadiendo las librerías necesarias, permite añadir los componentes requeridos, conectarlos y etiquetar sus conexiones. Una vez el esquemático haya sido finalizado, KiCad facilita la transformación de este a un diseño de la PCB en el cual se puede configurar el enrutamiento de la forma más eficiente.

Atendiendo a las restricciones de espacio establecidas, los componentes incluidos para diseñar esta primera versión de la placa son:

- Módulos que permitan intercambiar las tarjetas y se introduzcan de forma vertical [58].
- Una resistencia de 10kOhm para conectar el pin EN con GND [64].
- Un condensador de acoplo entre la alimentación del multiplexor y GND para mejorar la estabilidad de la comunicación [63].
- Un compartimento de dos pilas AA para alimentar la tarjeta madre [56].
- Pistas para conectar todas las partes necesarias manteniendo el máximo espacio posible aprovechando el tamaño de la placa.
- Test Points en puntos clave para comprobar tensiones o visualizar señales mediante un osciloscopio, un polímetro o cualquier analizador lógico.
- Circuito de protección en la alimentación de la tarjeta madre basado en diodos Schottky [59].
- El multiplexor CD74HCT4067M [62].

En este caso se ha diseñado una PCB de doble capa, permitiendo así diseñar el enrutado del circuito de la manera más sencilla posible.

Los pasos seguidos para diseñar el PCB de la versión v1 son los siguientes:

1. Creación de un nuevo proyecto en KiCad
2. Creación del esquemático del proyecto. Para ello habrá que tener en cuenta los elementos añadidos en las librerías del programa. En este proyecto las librerías predefinidas fueron suficientes para diseñar la placa.

El esquemático se ha configurado sin incluir los enrutados, de forma que mediante etiquetas se tenga en cuenta la conexión de los diferentes puntos.

Una vez diseñado el esquemático, se deben dar valores a los componentes pasivos incorporados e incluir un “footprint” a cada componente, dando forma a cada elemento añadido.

3. Editor de placas. Una vez diseñado al completo el esquemático e incluido los “footprints” a cada componente, a partir del editor de placas se puede extrapolar este modelo inicial que permitirá realizar el enrutado de cada parte conectada.

Como los enrutados son muchos y los pines de los chips no están siempre en un orden lógico, se hará uso de dos capas (una frontal y otra trasera) para diseñar la PCB. Para ello se hará uso de vías pasantes.

El resultado final se muestra en la ilustración 26.

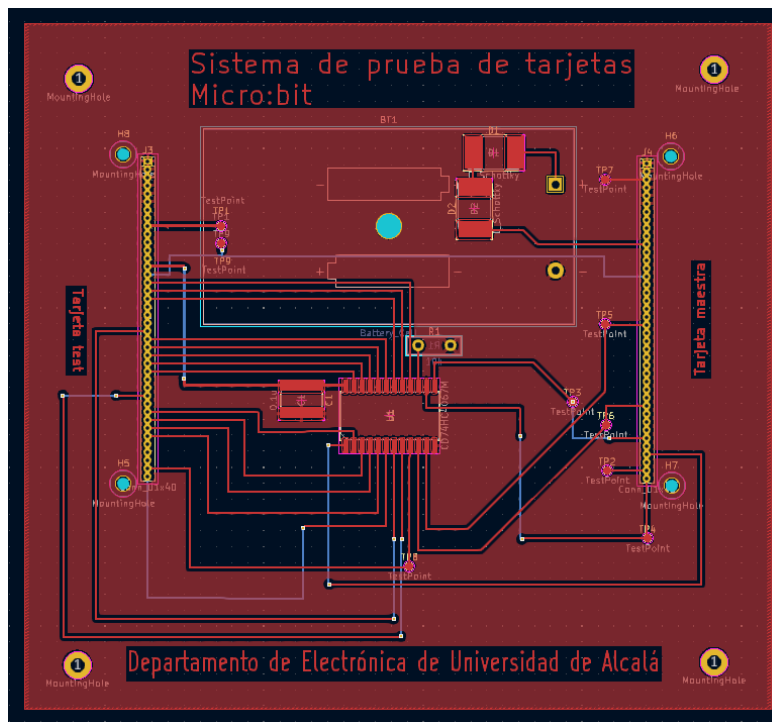
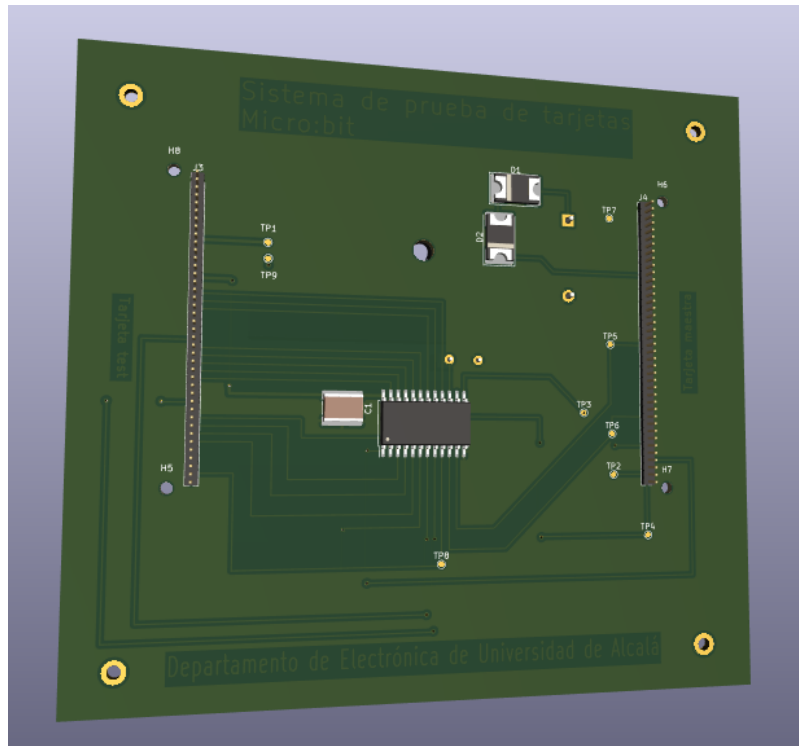


Ilustración 25. Diseño del PCB v1

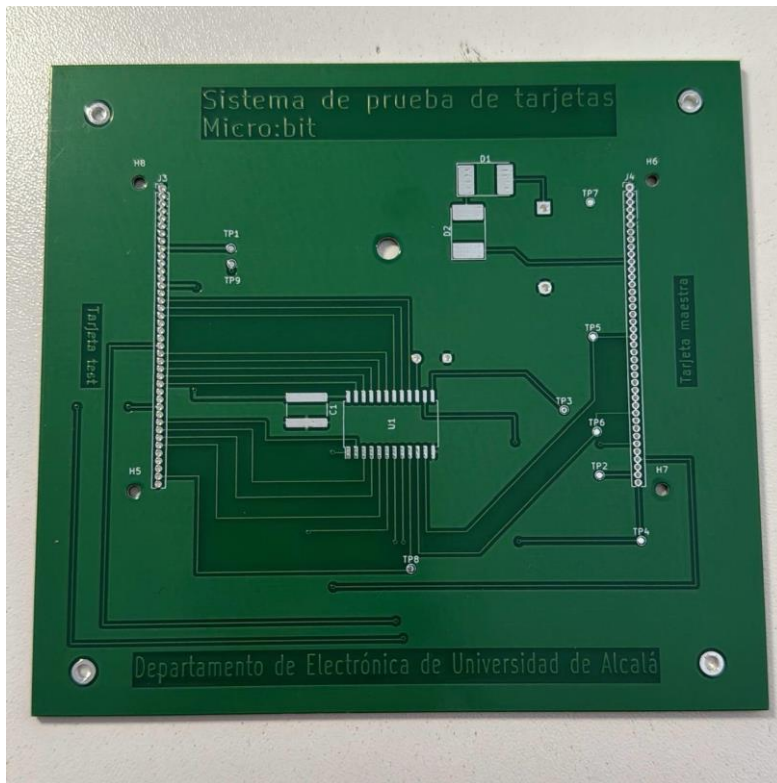
Una vez diseñada la placa, se procede a establecer los límites (y por tanto dimensiones de la placa). Además, como es habitual en la construcción de PCB's añade un plano de cobre sobre el circuito de masa.

En un diseño 3D como el que se muestra en la imagen 26 se diseña una simulación de cómo sería en la realidad la placa.



*Ilustración 26. Simulación 3D de la placa en versión prototipo v1.*

Se muestra sobre la imagen 27 el resultado obtenido tras la construcción.



*Ilustración 27. Tarjeta construida en v1.*

## 5.2. Esquema general de conexiones versión v2

Una vez la placa ha sido construida, se han comprobado varios puntos mejorables. Es por ello que se ha decidido rediseñar la primera versión incluyendo las siguientes modificaciones:

1. Un dimensionamiento más ajustado para el encapsulado de los diodos Schottky [57] . Los actuales son demasiado grandes para las necesidades de protección de la placa.
2. Rediseñar el encapsulado del condensador [66], al igual que en el punto anterior, para un encapsulado superficial de estas necesidades se puede hacer uso de un componente más pequeño.
3. Colocar la resistencia de 10k [68] en el plano visible de la tarjeta y no en la cara posterior.
4. Incluir agujeros con recubrimiento de estaño para anclar físicamente las ranuras de la micro:bit. En la versión v1 no se incluían.
5. Colocar las ranuras de la microbit de cara al usuario que va a hacer uso del sistema.
6. Incluir un jumper [69] que permita la alimentación de ambas tarjetas a la vez habiendo conectado únicamente la tarjeta que debe comunicar vía serie USB.
7. Incluir un interruptor [71] para la alimentación del módulo por pilas.

8. Incluir el módulo de baterías en la cara frontal, no necesitando así de colocar patas tan largas.
9. Cambio del encapsulado del Multiplexor. El modelo seleccionado en la versión 1 se encuentra descatalogado y su rotación actual en mercado es muy baja. El modelo elegido en esta versión es el MUX506IPWR [61].
10. Incluir un switch para alternar entre el paso de señales digitales y analógicas [72].
11. Incluir un potenciómetro de 10kohm para modular el voltaje de la señal analógica [77].
12. En el pin 16 de la tarjeta madre, el cual genera la señal analógica, incluir un filtro RC paso bajo.

El filtro se encargará de impedir el caso de señales de alta frecuencia. Como en el código se ha definido un periodo para la señal analógica de 1kHz, se filtrarán las frecuencias superiores a estas. Por ello el filtro diseñado incluye una resistencia de 1000kohm [74] y un capacitor de 100pF [76].

$$f_c = \frac{1}{2\pi RC} \approx 1600 \text{ kHz}$$

En el esquemático de la versión 2, mostrado en la imagen 28, se muestran las modificaciones por la inclusión de nuevas funcionalidades.

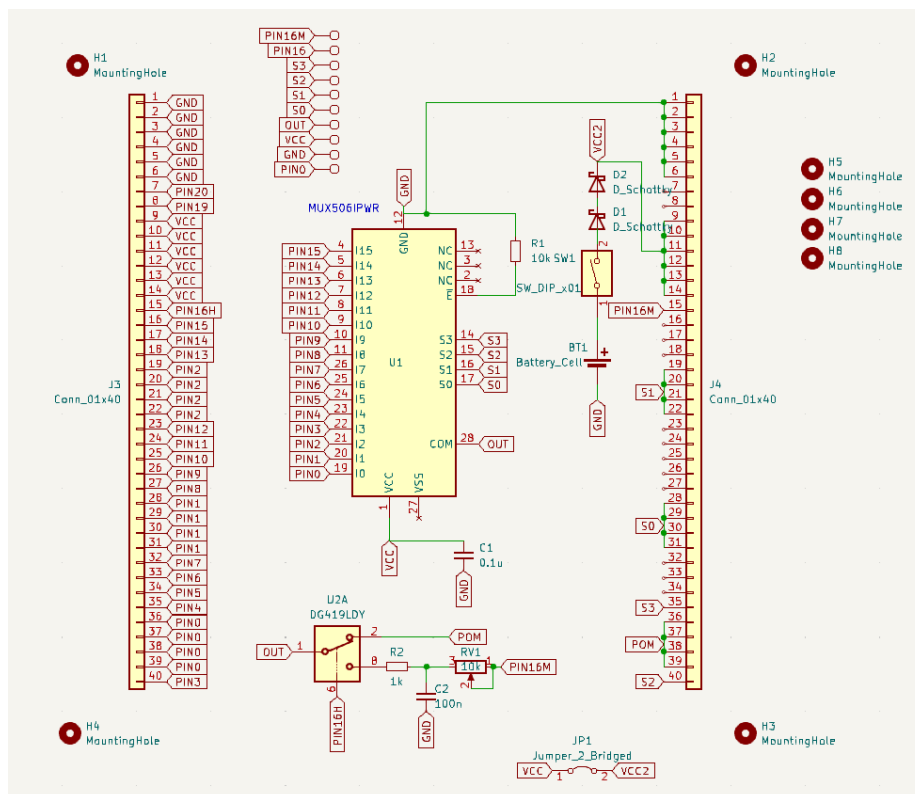


Ilustración 28. Esquemático PCB v3.



La reestructuración de los componentes ya contemplados, la adición de otros y la elección de encapsulados más apropiados, llevarían la versión 2 al diseño mostrado en la ilustración 29.

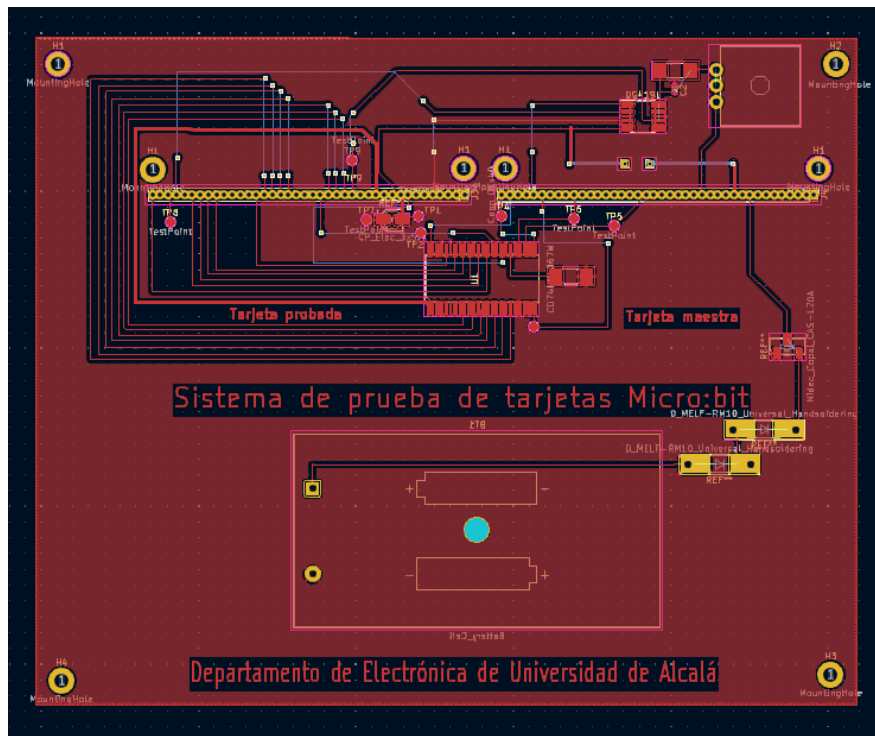


Ilustración 29. Diseño PCB v2.

En la imagen 30 se muestra cual sería la simulación de la siguiente versión v2:

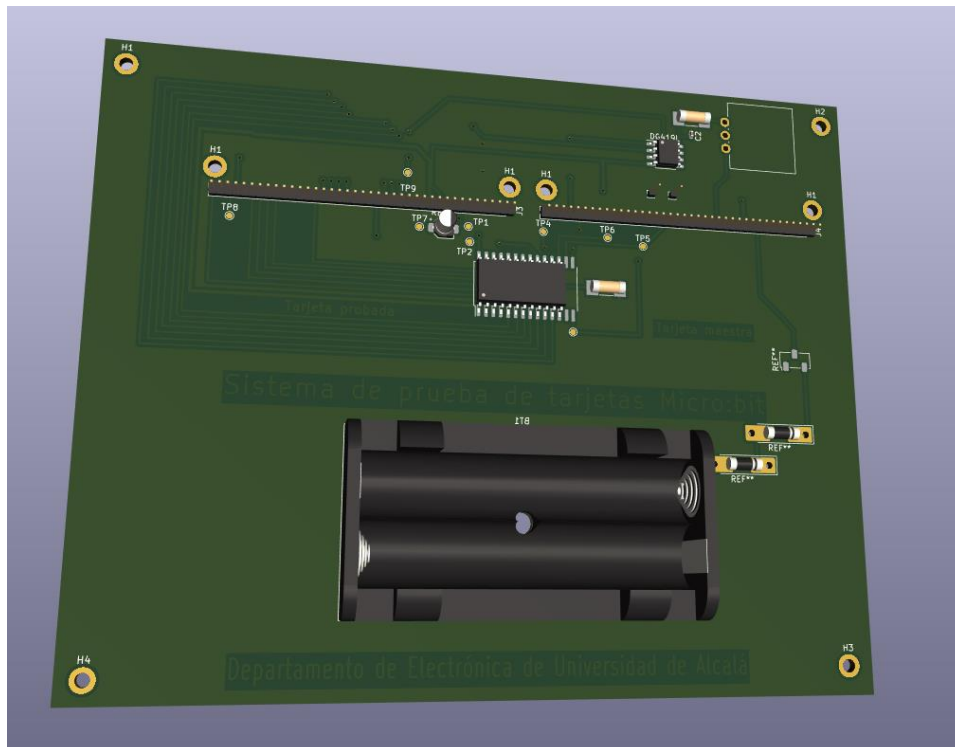


Ilustración 30. Simulación 3D de la placa en v2.

### 5.3. Listado de componentes de ambas versiones

En este apartado se incluyen todos los componentes que adquiridos para cada versión de PCB una vez ha sido construido. Se hace referencia a la bibliografía, donde se incluye en enlace en el que se puede adquirir el componente.

Versión 1:

- Módulos tarjetas microbit [43].
- Una resistencia de 10kOhm [51].
- Un condensador de acoplo de 0,1 uF [50].
- Un compartimento de dos pilas AA [41].
- Dos diodos Schottky 45V 60A [44].
- El multiplexor CD74HCT4067M [40].

Versión 2:

- Módulos tarjetas microbit [43].
- Un compartimento de dos pilas AA [41].
- Dos diodos Schottky 10V 1A [60].
- Condensador acoplo 0,1 uF [65].
- Resistencia de 10k [67].
- Jumper para la alimentación [54].
- Interruptor alimentación [70].
- Multiplexor MUX506IPWR [47].
- Switch de señales [48].
- Filtro RC [73]/[75].
- Potenciómetro 10kOhm [77].



# **CAPITULO 6:**

## **CONCLUSIONES Y TRABAJOS FUTUROS**

En este proyecto se ha podido demostrar cual es el potencial de las tarjetas micro:bit y su importancia dentro del mundo académico. Es por ello por lo que la existencia de un trabajo como este permitirá al futuro del departamento asegurar una enseñanza del más alto nivel optimizando costes y recursos.

Además, al haberse asegurado que el sistema de pruebas es accesible a cualquier usuario, ya tenga conocimientos de electrónica o no, cualquier persona es capaz de comprobar el correcto funcionamiento de todas las tarjetas micro:bit que tenga a su alcance.

En lo que respecta al trabajo, el diseño software ha sido planteado con MakeCode, a pesar del retraso en la creación de código, con el objetivo de que cualquier persona con mínimos conocimientos sea capaz de modificar cualquiera de los dos programas sin necesidad de ser programador.

Por otro lado, cabe mencionar la dificultad para encontrar versiones definitivas dentro del desarrollo hardware, ya que es complicado ajustar todo el diseño a la perfección para que el PCB construido cumpla al completo con los requisitos necesarios.

El proyecto presentado contiene un alto potencial debido a que el margen de mejora de este sistema es muy amplio. Es por ello por lo que se han planteado futuras versiones para conseguir un modelo perfeccionado.

Las áreas de mejora sobre los que proseguir en un futuro son las mencionadas a continuación:

- **Optimización de los programas.** En este proyecto, el código empleado para realizar las pruebas de funcionalidad de pines puede ser rediseñado para minimizar la memoria ocupada por el programa y reducir el número de interacciones vía serie USB.
- **Revisión versión 2 hardware de la PCB.** El diseño construido en este trabajo, aunque cumple con las funciones básicas, podría obtener un mejor resultado si se seleccionan los encapsulados de los componentes de forma que se ajusten más y se coloca cada uno de ellos en una mejor disposición, de forma que permita al usuario manejar la placa más cómodamente. Por otro lado se plantea añadir un conmutador para alternar entre señales digitales y permitir la lectura/escritura de señales analógicas en la tarjeta probada. El siguiente punto sería construir la versión 2 y valorar la finalidad del proyecto.

# CAPITULO 7:

## PRESUPUESTO

En este capítulo se expone un desglose del coste económico del proyecto:

## 6.1. Tarjeta de pruebas

Concepto	Precio	Amortización	Uso	Coste
<b>5 ud. Placa PCB</b>	60.26€	4 años	6 meses	7.53€
<b>Soporte pilas 2xAAA</b>	1.00€	4 años	6 meses	0.12€
<b>Diodo Schottky</b>	1.00€	4 años	6 meses	0.12€
<b>2 ud. Ranura micro:bit</b>	5.48€	4 años	6 meses	0.69€
<b>Multiplexor CD74HC4067M96</b>	0.81€	4 años	6 meses	0.11€
<b>Resistencia 10kΩ</b>	0€	4 años	6 meses	0.00€
<b>Condensador 1uF</b>	0€	4 años	6 meses	0.00€

*Tabla 2. Coste Tarjeta de pruebas.*

## 6.2. Material hardware

Concepto	Precio	Amortización	Uso	Coste
<b>Asus Zenbook 14</b>	799€	4 años	6 meses	99.87€
<b>2 ud. Monitor Benq 24''</b>	160€	4 años	6 meses	20.00€
<b>Teclado Logitech</b>	15€	4 años	6 meses	1.87€
<b>Ratón Logitech</b>	15€	4 años	6 meses	1.87€
<b>3 ud. Tarjetas Micro:bit</b>	45€	4 años	6 meses	5.63€

*Tabla 3. Coste material hardware*

## 6.3. Material software

Concepto	Precio	Amortización	Uso	Coste
<b>Microsoft MakeCode</b>	0€	4 años	6 meses	0.00€
<b>KiCad 7.0</b>	0€	4 años	6 meses	0.00€
<b>Tinker Cad</b>	0€	4 años	6 meses	0.00€

*Tabla 4. Coste Software.*

#### 6.4. Mano de obra

Concepto	Coste
Soldadura de componentes	20.00€

*Tabla 5. Coste mano de obra.*

#### 6.5. Presupuesto total

Concepto	Coste
Tarjeta de pruebas	8.57€
Material Hardware	129.23€
Material Software	0.00€
Mano de obra	20.00€
<b>TOTAL</b>	<b>157.80€</b>

*Tabla 6. Presupuesto total.*





# CAPITULO 8:

## BIBLIOGRAFIA

- [1] *Micro:Bit software*. (s. f.). <https://tech.microbit.org/software/> (Último acceso 12/09/2023)
- [2] *Qué es Micro:Bit - Microes.org - Comunidad Micro:bit en España*. (s. f.)  
<http://microes.org/que-es-microbit.php> (Último acceso 12/09/2023)
- [3] *Microsoft MakeCode for Micro:Bit*. (s. f.). Microsoft MakeCode for micro:bit.  
<https://makecode.microbit.org/#editor> (Último acceso 27/01/2024)
- [4] *Micro:Bit Sound meter*. (s. f.). <https://microbit.org/projects/make-it-code-it/sound-meter/#:~:text=The%20new%20micro%3Abit%27s%20microphone%20measures%20soun%20levels%20in,sounds%20measured%2C%20the%20higher%20the%20bar%20graph%20gets.> (Último acceso 10/01/2024)
- [5] Admin. (2021, 22 junio). *MÓDULO MICRÓFONO – Micro:bit*. <https://microbit.microlab.com/modulo-microfono/> (Último acceso 14/09/2023)
- [6] Erick. (2022, 23 septiembre). *Lenguajes de programación interpretados vs compilados: ¿Cuál es la diferencia?* freeCodeCamp.org.  
<https://www.freecodecamp.org/espanol/news/lenguajes-compilados-vs-interpretados/>  
(Último acceso 18/09/2023)
- [7] *Características y funcionalidades de la tarjeta programable Micro:Bit - Microes.org*. (s. f.-b). <http://microes.org/caracteristicas.php> (Último acceso 10/09/2023)
- [8] *¿No conoces Micro:Bit?* (s. f.). <https://blogmigueteconologia.blogspot.com/2023/01/no-conoces-microbit.html> (Último acceso 14/11/2023)
- [9] Colaboradores de Wikipedia. (2023, 14 noviembre). *Micro bit*. Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Micro\\_Bit](https://es.wikipedia.org/wiki/Micro_Bit) (Último acceso 15/09/2023)
- [10] *Micro:Bit software*. (s. f.-b). <https://tech.microbit.org/software/> (Último acceso 15/09/2023)
- [11] *BBC Micro:Bit overview*. (s. f.). <https://microbit.org/get-started/features/overview/> (Último acceso 15/09/2023)
- [12] *Raspberry Pi Pico vs Micro:Bit V2 vs Arduino UNO R3*. (2021, 3 septiembre).  
<https://solectroshop.com/es/blog/raspberry-pi-pico-vs-microbit-v2-vs-arduino-uno-r3-n77> (Último acceso 15/09/2023)

- [13] Colaboradores de Wikipedia. (2023a, octubre 26). *Arquitectura ARM*. Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Arquitectura\\_ARM](https://es.wikipedia.org/wiki/Arquitectura_ARM) (Último acceso 21/01/2024)
- [14] *Bluetooth pairing*. (s. f.). Microsoft MakeCode. <https://makecode.microbit.org/reference/bluetooth/bluetooth-pairing> (Último acceso 09/10/2023)
- [15] *Cortex-M0*. (s. f.). <https://developer.arm.com/Processors/Cortex-M0> (Último acceso 08/09/2023)
- [16] Colaboradores de Wikipedia. (2023a, noviembre 11). *Arduino uno*. Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Arduino\\_Uno](https://es.wikipedia.org/wiki/Arduino_Uno) (Último acceso 08/09/2023)
- [17] *Micro:bit mobile apps*. (s. f.). <https://microbit.org/get-started/user-guide/mobile/> (Último acceso 08/11/2023)
- [18] *Pines GPIO en microbit con Python*. (2019, 26 octubre). Asociación Programa Ergo Sum. <https://www.programoergosum.es/tutoriales/pines-gpio-en-microbit-con-python/> (Último acceso 08/12/2023)
- [19] Macías, J. T. (2022, 13 octubre). *Acelerómetro – micro:bit*. Robotimáticas. <https://blogsaverroes.juntadeandalucia.es/robotimaticas/acelerometro-microbit/> (Último acceso 16/10/2023)
- [21] *Download PuTTY: Latest Release (0.79)*. (s. f.). <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> (Último acceso 31/11/2023)
- [22] Microbit-foundation. (s. f.). *Microbit-V2-hardware/V2.00/MicroBit\_V2.0.0\_S\_schematic.PDF at main · Microbit-foundation/microbit-V2-hardware*. GitHub. [https://github.com/microbit-foundation/microbit-v2-hardware/blob/main/V2.00/MicroBit\\_V2.0.0\\_S\\_schematic.PDF](https://github.com/microbit-foundation/microbit-v2-hardware/blob/main/V2.00/MicroBit_V2.0.0_S_schematic.PDF) (Último acceso 20/09/2023)

- [23] Bbcmicrobit. (s. f.). *Hardware/V1.3B/SCH\_BBC-Microbit\_V1.3B.pdf at Master* · *BBCMirobit/Hardware*. GitHub.  
[https://github.com/bbcmicrobit/hardware/blob/master/V1.3B/SCH\\_BBC-Microbit\\_V1.3B.pdf](https://github.com/bbcmicrobit/hardware/blob/master/V1.3B/SCH_BBC-Microbit_V1.3B.pdf) (Último acceso 20/12/2023)
- [24] KiCad EDA. (s. f.). *Download*. KiCad Website. <https://www.kicad.org/download/> (Último acceso 15/10/2023)
- [25] *TinkerCad | From mind to design in minutes*. (s. f.). Tinkercad.  
<https://www.tinkercad.com/> (Último acceso 22/10/2023).
- [26] Llamas, L. (2016, 19 noviembre). *Más salidas/entradas en Arduino con Multiplexor CD74HC4067*. Luis Llamas. <https://www.luisllamas.es/mas-salidas-y-entradas-en-arduino-con-multiplexor-cd74hc4067/> (Último acceso 22/11/202).
- [27] *Input/Output PINs — BBC Micro:Bit MicroPython 1.1.1 Documentation*. (s. f.).  
<https://microbit-micropython.readthedocs.io/en/latest/pin.html> (Último acceso 21/10/2023)
- [28] *Texas Instruments CD74HC4040E*. (s. f.). Mouser Electronics.  
[https://www.mouser.es/ProductDetail/Texas-Instruments/CD74HC4040E?qs=st7IWvL5%2FimNOdB5oPThA%3D%3D&utm\\_source=bing&utm\\_medium=cpc&utm\\_campaign=DSAs+Other+Suppliers+Spain+English&utm\\_term=%2FProductDetail%2F&utm\\_content=Other+Suppliers+DSAs%3A+PDP](https://www.mouser.es/ProductDetail/Texas-Instruments/CD74HC4040E?qs=st7IWvL5%2FimNOdB5oPThA%3D%3D&utm_source=bing&utm_medium=cpc&utm_campaign=DSAs+Other+Suppliers+Spain+English&utm_term=%2FProductDetail%2F&utm_content=Other+Suppliers+DSAs%3A+PDP) (Último acceso 12/12/2023).
- [29] Novatech. (2020, 15 julio). *Multiplexor CD74HC4067 + Ejemplos de uso* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=eaMhqu91M3A> (Último acceso 21/12/2023).
- [30] CarrasTech. (2022, 20 mayo). *Kicad tutorial en 10 minutos* [Vídeo]. YouTube.  
[https://www.youtube.com/watch?v=vtUj1Ba\\_ELk](https://www.youtube.com/watch?v=vtUj1Ba_ELk) (Último acceso 29/12/2023).
- [31] *Schematics*. (s. f.). <https://tech.microbit.org/hardware/schematic/#edge-connector> (Último acceso 29/12/2023).

- [32] Ohms. (2021, 7 septiembre). *Tutorial #1 Tarjeta de expansión para Micro:Bit: Conociendo la tarjeta*. 330ohms. <https://blog.330ohms.com/2021/08/27/tutorial-1-tarjeta-exapansion-microbit/> (Último acceso 18/12/2024).
- [33] Colaboradores de Wikipedia. (2023a, marzo 10). *Ranura de expansión*. Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Ranura\\_de\\_expansi%C3%B3n#ISA\\_8\\_\(XT\)](https://es.wikipedia.org/wiki/Ranura_de_expansi%C3%B3n#ISA_8_(XT)) (Último acceso 10/12/2023).
- [34] YahboomTechnology. (s. f.). *GitHub - YahboomTechnology/Microbit-Professional-Expansion-Board: An professional expansion board is designed specifically for Micro:Bit*. GitHub. <https://github.com/YahboomTechnology/Microbit-professional-expansion-board/tree/master> (Último acceso 22/12/2023).
- [35] Gracey, K. (2020, 15 octubre). *P2 Edge Mini Breakout Board - Schematic and pictures*. Parallax Forums. <https://forums.parallax.com/discussion/172290/p2-edge-mini-breakout-board-schematic-and-pictures> (Último acceso 22/12/2023).
- [36] KittenBot. (s. f.). *Www-kittenbot/products/2018-1--robotbit/index.md at en · KittenBot/www-kittenbot*. GitHub. <https://github.com/KittenBot/www-kittenbot/blob/en/products/2018-1--robotbit/index.md> (Último acceso 22/12/2023).
- [37] Chris. (2023, 4 enero). *¿Cuáles son los diferentes tipos de capas de PCB?* Jamindo PCBA. <https://jamindopcba.com/es/diferentes-capas-de-pcb/> (Último acceso 22/12/2023).
- [38] *Hardware Micro:bit*. (s. f.). <https://tech.microbit.org/hardware/> (Último acceso 15/01/2024).
- [39] *Edge connector and pinout*. (s. f.). <https://tech.microbit.org/hardware/edgeconnector/> (Último acceso 20/01/2024).
- [40] *Texas Instruments CD74HCT4067M*. (s. f.). Mouser Electronics. <https://www.mouser.es/ProductDetail/Texas-Instruments/CD74HCT4067M?qs=vul0MIC%2Fa1fex%2F7NLRWpgQ%3D%3D> (Último acceso 20/01/2023)
- [41] *Keystone Electronics 2468*. (s. f.). Mouser Electronics. <https://www.mouser.es/ProductDetail/Keystone->

- [Electronics/2468?qs=3CbvriavsLAoISGi3MywrQ%3D%3D](#) (Último acceso 20/01/2023)
- [42] *DIODE SCHOTTKY 10V 1A DO204AL*. (s. f.). Digikey.es.  
<https://www.digikey.es/en/products/detail/vishay-general-semiconductor-diodes-division/SB110-E3-54/2146184> (Último acceso 20/01/2024)
- [43] *Adafruit 3888*. (s. f.). Mouser Electronics.  
<https://www.mouser.es/ProductDetail/Adafruit/3888?qs=w%2Fv1CP2dgqpGHzosVmV CRA%3D%3D> (Último acceso 25/01/2024)
- [44] *Vishay V60DM45CHM3/I*. (s. f.). Mouser Electronics.  
<https://www.mouser.es/ProductDetail/Vishay-General-Semiconductor/V60DM45CHM3-I?qs=F5EMLAvA7IDVrCYntmpBPg%3D%3D>  
(Último acceso 20/01/2023)
- [45] *Fabricación de PCB - Prototipos de PCB de forma sencilla*. (s. f.).  
<https://www.pcbway.es/> (Último acceso 28/01/2024)
- [46] *Documentation – Arm Developer*. (s. f.).  
<https://developer.arm.com/documentation/102834/latest/> (Último acceso 31/01/2024)
- [47] *Texas Instruments MUX506IPWR*. (s. f.). Mouser Electronics.  
<https://www.mouser.es/ProductDetail/Texas-Instruments/MUX506IPWR?qs=0C8XhJW8e4qYfw%2FdTmN3Og%3D%3D> (Último acceso 05/02/2024)
- [48] *Texas Instruments DG419LDY+*. (s. f.). Mouser Electronics.  
[DG419LDY+ Analog Devices / Maxim Integrated | Mouser España](#) (Último acceso 10/02/2024)
- [49] *Filtro paso bajo- explicado*. (s. f.).  
<https://www.learningaboutelectronics.com/Articulos/Filtro-paso-bajo.php> (Último acceso 10/02/2024)
- [50] *Texas Instruments Murata Electronics*. (s. f.). Mouser Electronics.  
<https://www.mouser.es/ProductDetail/Murata-Electronics/935153521310?qs=BJlw7L4Cy78oLBt0WSy9zA%3D%3D>

(Último acceso 10/02/2024)

[51] *Vishay / Beyschlag MCT0603MG1003BP500*. (s. f.). Mouser Electronics.

[MCT0603MG1003BP500 Vishay / Beyschlag | Mouser España](#) (Último acceso 10/02/2024)

[52] *Vishay / Beyschlag V3NM153-M3/I*. (s. f.). Mouser Electronics.

[V3NM153-M3/I Vishay Semiconductors | Mouser España](#) (Último acceso 10/02/2024)

[53] *Murata 935156424610*. (s. f.). Mouser Electronics.

<https://www.mouser.es/ProductDetail/Murata-Electronics/935156424610?qs=rkhjVJ6%2F3EJBJ%2FhNkzdO1g%3D%3D> (Último acceso 10/02/2024)

[54] *Amphenol 67996-200HLF*. (s. f.). Mouser Electronics.

<https://www.mouser.es/ProductDetail/Amphenol-FCI/67996-200HLF?qs=N11avnkqzksPIMpVNsng%3D%3D> (Último acceso 10/02/2024)

[55] *KOA Speer RN73H1JTTD1002F25*. (s. f.). Mouser Electronics.

[RN73H1JTTD1002F25 KOA Speer | Mouser España](#) (Último acceso 10/02/2024)

[56] *Datasheet Keystone Electronics 2468*. (s. f.). Mouser.es.

<https://www.mouser.es/datasheet/2/215/2468-741064.pdf> (Último acceso 11/02/2024)

[57] *Datasheet DIODE SCHOTTKY 10V 1A DO204AL*. (s. f.-b).

<https://www.vishay.com/docs/88715/sb110.pdf> (Último acceso 11/02/2024)

[58] *Datasheet Adafruit 3888*. (s. f.). [https://cdn-shop.adafruit.com/product-](https://cdn-shop.adafruit.com/product-files/3888/3888_datasheet.pdf)

[files/3888/3888\\_datasheet.pdf](https://cdn-shop.adafruit.com/product-files/3888/3888_datasheet.pdf) (Último acceso 11/02/2024)

[59] *Datasheet V60DM45CHM3/I*. (s. f.).

<https://www.vishay.com/docs/87588/v60dm45c.pdf> (Último acceso 11/02/2024)

[60] *Mouser - DIODE SCHOTTKY 10V 1A DO204AL*. (s. f.). Mouser.es.

<https://www.digikey.es/en/products/detail/vishay-general-semiconductor-diodes-division/SB110-E3-54/2146184> (Último acceso 11/02/2024)



[61] *Datasheet TI MUX506IPWR*. (s. f.). Ti.com.

<https://www.ti.com/lit/ds/symlink/mux506.pdf?ts=1707610538083> (Último acceso 11/02/2024)

[62] *Datasheet CD74HCT4067M*. (s. f.). Ti.com.

[https://www.ti.com/lit/ds/symlink/cd74hct4067.pdf?ts=1707597213103&ref\\_url=https%253A%252F%252Fwww.mouser.pl%252F](https://www.ti.com/lit/ds/symlink/cd74hct4067.pdf?ts=1707597213103&ref_url=https%253A%252F%252Fwww.mouser.pl%252F) (Último acceso 11/02/2024)

[63] *Datasheet 935153521310*. (s. f.).

[https://www.mouser.es/datasheet/2/281/Commercial\\_leafletUWSCV13Murata-1660581.pdf](https://www.mouser.es/datasheet/2/281/Commercial_leafletUWSCV13Murata-1660581.pdf) (Último acceso 11/02/2024)

[64] *Datasheet MCT0603MG1003BP500*. (s. f.).

[https://www.vishay.com/docs/28952/mcs040\\_2at-mct0603at-mcu0805at-mca1206at.pdf](https://www.vishay.com/docs/28952/mcs040_2at-mct0603at-mcu0805at-mca1206at.pdf) (Último acceso 11/02/2024)

[65] *Diode Digikey UKL1H0R1MDDANA*. (s. f.). Digikey.es.

<https://www.digikey.es/en/products/detail/nichicon/UKL1H0R1MDDANA/2598447> (Último acceso 11/02/2024)

[66] *Datasheet Nichicon UKL1H0R1MDDANA*. (s. f.).

[https://www.nichicon.co.jp/english/series\\_items/catalog\\_pdf/e-ukl.pdf](https://www.nichicon.co.jp/english/series_items/catalog_pdf/e-ukl.pdf) (Último acceso 11/02/2024)

[67] *TE Connectivity EP10WSS10KJBB*. (s. f.). Mouser Electronics.

<https://www.mouser.es/ProductDetail/TE-Connectivity-Neohm/EP10WSS10KJBB?qs=sGAepiMZZMtTURnxoZnJAPKHF1kdu5MnAEseKUghufXpyRXCP6yccA%3D%3D> (Último acceso 11/02/2024)

[68] *Datasheet EP10WSS10KJBB*. (s. f.).

[https://www.mouser.es/datasheet/2/418/9/ENG\\_CD\\_2176614\\_BA1-3367921.pdf](https://www.mouser.es/datasheet/2/418/9/ENG_CD_2176614_BA1-3367921.pdf) (Último acceso 11/02/2024)

- [69] *Datasheet 67996-200HLF Amphenol FCI*. (s. f.). [https://cdn.amphenol-com/media/wysiwyg/files/documentation/datasheet/boardwiretoboard/bwb\\_bergstik.pdf](https://cdn.amphenol-com/media/wysiwyg/files/documentation/datasheet/boardwiretoboard/bwb_bergstik.pdf) (Último acceso 11/02/2024)
- [70] *SWITCH SLIDE SPDT 100MA 6V*. (s. f.). Digikey.es. <https://www.digikey.es/en/products/detail/nidec-components-corporation/CAS-120A/6821957> (Último acceso 11/02/2024)
- [71] *Datasheet SWITCH SLIDE SPDT 100MA 6V*. (s. f.). nidec-components.com. <https://www.nidec-components.com/e/catalog/switch/cas.pdf> (Último acceso 11/02/2024)
- [72] *Datasheet https://www.mouser.es/datasheet/2/609/DG417L\_DG419L-3122494.pdf*. (s. f.). Mouser.es. [https://www.mouser.es/datasheet/2/609/DG417L\\_DG419L-3122494.pdf](https://www.mouser.es/datasheet/2/609/DG417L_DG419L-3122494.pdf) (Último acceso 11/02/2024)
- [73] *TE Connectivity EP10WSS10KJBB*. (s. f.-b). Mouser Electronics. <https://www.mouser.es/ProductDetail/TE-Connectivity-Neohm/EP10WSS10KJBB?qs=sGAepiMZZMtTURnxoZnJAPKHF1kdu5MnAEseKUghufXpyRXCP6yccA%3D%3D> (Último acceso 11/02/2024)
- [74] *Datasheet EP10WSS10KJBB*. (s. f.-b). . [https://www.mouser.es/datasheet/2/41\\_8/9/ENG\\_CD\\_2176614\\_BA1-3367921.pdf](https://www.mouser.es/datasheet/2/41_8/9/ENG_CD_2176614_BA1-3367921.pdf) (Último acceso 11/02/2024)
- [75] *CAP FILM 100PF 10% 1KVDC RAD*. (s. f.). Digikey.es. <https://www.digikey.es/en/products/detail/wima/FKP2O101001D00KSSD/9370238> (Último acceso 11/02/2024)
- [76] *Datasheet FKP2O101001D00KSSD*. (s. f.). Digikey.es. [https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/749/WIMA\\_FKP\\_2.pdf](https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/749/WIMA_FKP_2.pdf) (Último acceso 11/02/2024)

[77] *Potenciómetro 10kohm 14810FAGJSX10103KA*. (s. f.). Digikey.es.

<https://www.digikey.es/en/products/detail/vishay-spectrol/14810FAGJSX10103KA/10739597>

[78] *Datasheet POT CONDUCTIVE PLASTIC ELEMENT 14810FAGJSX10103KA*.

(s. f.). Vishay.com. <https://www.vishay.com/docs/57040/148149.pdf>



# CAPITULO 9:

## ANEXOS

## 9.1. Proyecto Make It Digital

El proyecto Make It Digital de la BBC nace con el propósito de incentivar el interés por la codificación, programación y la tecnología digital. La campaña se centró en la idea de que la tecnología digital está transformando la forma en que vivimos y trabajamos, y es esencial que las personas tengan las habilidades necesarias para prosperar en esta era digital. El Reino Unido, ante la escasez de profesionales actual en el sector digital, propone este proyecto para empezar a paliar los efectos de la falta de personal que se sufrirán en un futuro próximo. Se prevé que en los próximos cinco años se necesiten 1,4 millones de personas en el mundo para ocupar este tipo de profesiones.

La fundación está inspirada en el proyecto lanzado en 1981 “BBC Micro”, el cual fue uno de los primeros ordenadores al alcance de cualquier usuario y consiguió familiarizar a la población de Gran Bretaña con los primeros ordenadores personales. De igual manera, Make It Digital pretende despertar entre los ciudadanos el entusiasmo por la creatividad digital, entre los más jóvenes a través las aulas y entre los más mayores a partir de contenido en línea, radio y televisión.

Para ello la BBC se asoció con organizaciones, empresas y escuelas para organizar eventos y actividades alrededor de todo el país relacionadas con la tecnología digital. Uno de los eventos más destacados fue el “BBC Make it Digital Tour”, que recorrió varias ciudades del Reino Unido y ofreció talleres y actividades interactivas para jóvenes y familias.

El resultado del proyecto fue de un alto impacto en la conciencia pública sobre la importancia de adquirir sobre las nuevas generaciones las nociones digitales y de programación. Ayudó a impulsar el interés en la programación y creció el número de participaciones en actividades STEM (Science, Technology, Engineering and Mathematics). Dejando así un duradero legado en forma de recursos físicos como la tarjeta y digitales, como múltiples cursos y tutoriales webs disponibles en la página oficial de Micro:bit.

En definitiva, esta ambiciosa campaña destinada a fomentar el interés y las habilidades en el campo de la tecnología digital y programación tuvo un impacto positivo en la conciencia pública que se verá reflejado en el crecimiento de las próximas generaciones.

## 9.1.1 Comparativa con otras tarjetas del mercado

Una vez hemos profundizado en las posibilidades que ofrece la tarjeta a partir de sus especificaciones hardware y software, se va a realizar un estudio comparativo de la tarjeta micro:bit con el resto de las tarjetas de similares que existen en el mercado.

Entre las más conocidas se encuentran la popular Arduino UNO rev3, y la Raspberry Pi Pico, ambas extremadamente extendidas a lo largo de este mundo de creación digital. En este capítulo compararemos las tres tarjetas y deduciremos cual de ellas es la más adecuada para los proyectos que se quieran llevar a cabo.

### 9.1.1.1 Raspberry Pi Pico

A principios del año 2021, Raspberry Pi introdujo su primer microcontrolador propio, el RP2040, junto con la presentación de una placa de desarrollo de bajo coste llamada Raspberry Pi Pico. Este lanzamiento marcó un cambio significativo en su enfoque y buscó atraer a una audiencia aún más amplia a su ecosistema.

Las especificaciones de la tarjeta Raspberry Pi Pico son las siguientes:

<i>Raspberry Pi Pico</i>	
<b>Microcontrolador</b>	RP2040
<b>Memoria RAM</b>	264 Kbyte
<b>Memoria Flash</b>	2 Mbyte
<b>Cores</b>	Dual-Core
<b>Reloj CPU</b>	48 Mhz
<b>GPIO</b>	26 I/O
<b>Sensores</b>	No incorporados
<b>ADC</b>	3 x 12 bits
<b>DAC</b>	No incluido
<b>Comunicación</b>	UART, SPI, I2C, GPIO
<b>Conectividad inalámbrica</b>	No incluido
<b>Salida de vídeo</b>	No incluido
<b>Lenguajes de programación</b>	MicroPython, C/C++
<b>Alimentación</b>	3.3V
<b>Dimensiones</b>	21.0 mm × 51.0 mm
<b>Precio</b>	3.43

*Tabla 7. Especificaciones de la tarjeta Raspberry Pi Pico..*

La Raspberry Pi Pico y la Micro:bit son dos tarjetas de desarrollo con enfoques ligeramente diferentes. La Raspberry Pi Pico tiene un microcontrolador más potente con más memoria RAM y flash, lo que la hace adecuada para proyectos más avanzados que requieran mayor capacidad de procesamiento y almacenamiento. Por otro lado, la Micro:bit está diseñada específicamente para la educación y ofrece una amplia gama de sensores incorporados y conectividad Bluetooth, lo que la convierte en una opción extraordinaria para proyectos educativos y de aprendizaje. La elección entre ambas depende de tus necesidades y objetivos específicos.

### 9.1.1.2 Arduino UNO rev3

La placa Arduino Uno es una placa de microcontrolador de código abierto desarrollada por la empresa Arduino CC, y monta el microchip Atmega328P. El principal objetivo del lanzamiento de esta placa era crear proyectos básicos y de bajo coste para crear herramientas digitales accesibles a la mano de usuarios de cualquier clase.

La tarjeta Arduino Uno presenta las siguientes características:

<i>Arduino UNO rev3</i>	
<b>Microcontrolador</b>	Atmega328
<b>Memoria RAM</b>	2Kbyte
<b>Memoria Flash</b>	32 Kbyte
<b>Cores</b>	Single – Core
<b>Reloj CPU</b>	16 MHz
<b>GPIO</b>	20 I/O
<b>Sensores</b>	No incorporados
<b>ADC</b>	6 x 10 bits
<b>DAC</b>	No incluido
<b>Comunicación</b>	UART, SPI, I2C, GPIO
<b>Conectividad inalámbrica</b>	No incluido
<b>Salida de vídeo</b>	No incluido
<b>Lenguajes de programación</b>	C alike, Arduino IDE
<b>Alimentación</b>	5V
<b>Dimensiones</b>	68.6 mm × 53.4 mm
<b>Precio</b>	18,17 €

*Tabla 8. Especificaciones de la tarjeta Arduino UNO rev3.*



### 9.1.1.3 Conclusiones de la comparativa

En resumen, la Arduino Uno y la Micro:bit son dos tarjetas de desarrollo con enfoques y características diferentes. La Arduino Uno utiliza un microcontrolador AVR de 8 bits y se destaca por su amplia comunidad y facilidad de uso con el IDE de Arduino, mientras que la Micro:bit está diseñada principalmente para la educación y cuenta con sensores incorporados y conectividad Bluetooth para proyectos más orientados a la enseñanza y la experimentación. La elección entre ambas depende de tus necesidades y objetivos específicos de proyecto.

Aunque la Arduino y la Raspberry son opciones respetables, versátiles y muy instauradas en este mercado, la Micro:bit destaca como una poderosa herramienta educativa y creativa que merece una atención especial.

La Micro:bit ha sido diseñada desde cero con el objetivo de implementarse en la educación. Es perfecta para estudiantes y principiantes, ya que ofrece una interfaz intuitiva y una suave curva de aprendizaje. Los bloques visuales de MakeCode y la compatibilidad con MicroPython hacen que la programación sea accesible para todos, permitiendo a los jóvenes programadores dar vida a sus ideas sin dificultades.

Uno de los mayores activos de la Micro:bit son sus sensores incorporados. La Micro:bit permite a los estudiantes explorar conceptos de movimiento, orientación y visualización de datos de una manera divertida y práctica.

Además, es portátil y alimentada por batería, lo que supone que se puede portar a cualquier lugar para poner en práctica los proyectos en cualquier ambiente. Y su conectividad Bluetooth Low Energy (BLE) facilita la interacción con dispositivos móviles y la creación de proyectos de IoT (Internet de las cosas).

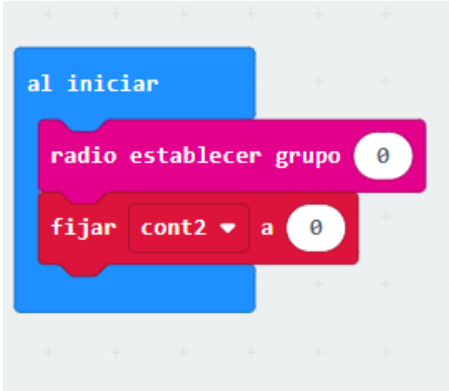
A pesar de ser una tarjeta relativamente joven en comparación con la Arduino Uno, la Micro:bit ha construido una comunidad muy viva y en crecimiento. Esto significa que los recursos educativos, tutoriales y proyectos están fácilmente disponibles en línea, lo que facilita aún más el aprendizaje y la resolución de problemas.

En definitiva, aunque las placas de la competencia son opciones muy aconsejables y en varios aspectos son superiores a la Micro:bit, la experiencia que brinda esta es mucho más enriquecedora y creativa que la del resto. Su accesibilidad, sensores integrados y comunidad comprometida la convierten en una elección excelente para quienes buscan desarrollar a la próxima generación de

creadores digitales aprendiendo manera divertida y efectiva. Es la plataforma perfecta para llevar la imaginación al mundo real y construir un futuro digital inspirador.

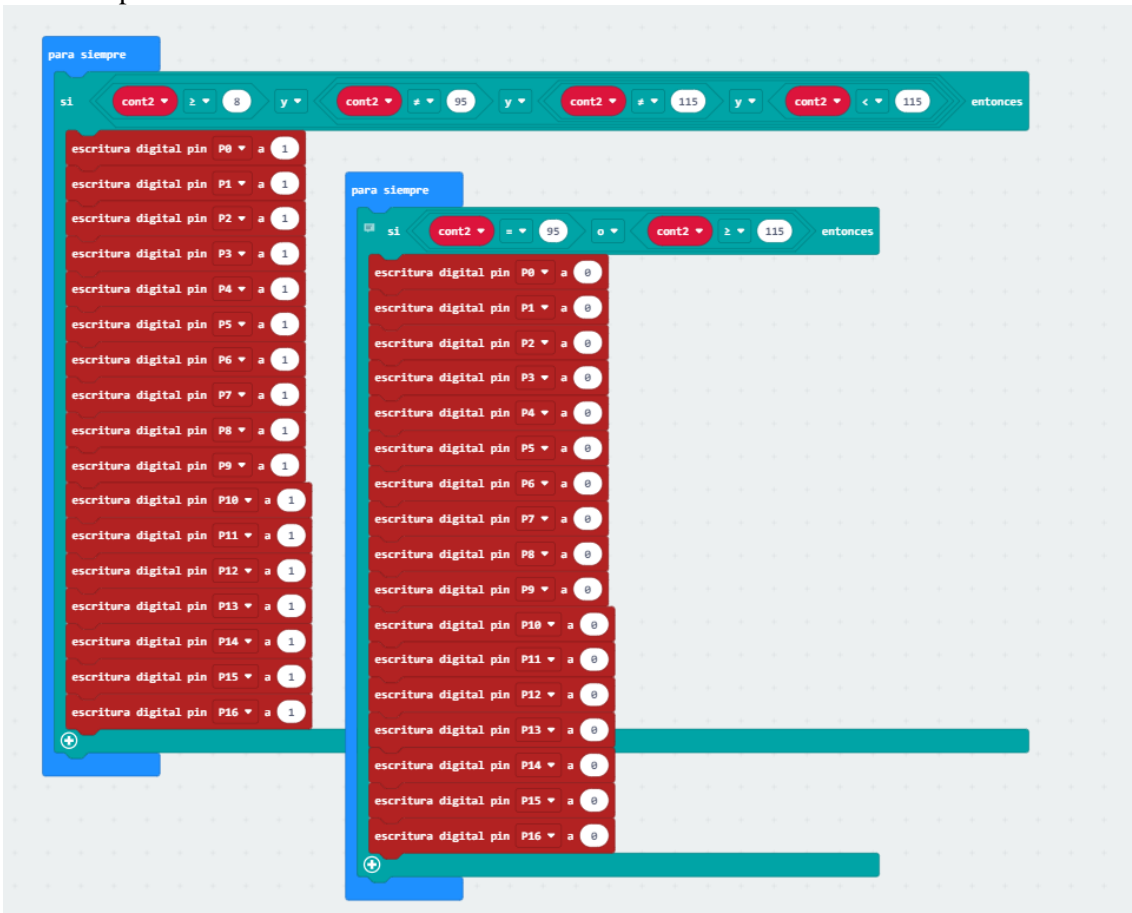
## 9.2 Código cargado en la tarjeta puesta a prueba o hija

Al iniciar:



```
let Nivel_volumen = 0
let cont2 = 0
radio.setGroup(0)
cont2 = 0
```

Para siempre:



```

basic.forever(function () {
  if (cont2 >= 8 && (cont2 != 95 && (cont2 != 115 && cont2 < 115))) {
    pins.digitalWritePin(DigitalPin.P0, 1)
    pins.digitalWritePin(DigitalPin.P1, 1)
    pins.digitalWritePin(DigitalPin.P2, 1)
    pins.digitalWritePin(DigitalPin.P3, 1)
    pins.digitalWritePin(DigitalPin.P4, 1)
    pins.digitalWritePin(DigitalPin.P5, 1)
    pins.digitalWritePin(DigitalPin.P6, 1)
    pins.digitalWritePin(DigitalPin.P7, 1)
    pins.digitalWritePin(DigitalPin.P8, 1)
    pins.digitalWritePin(DigitalPin.P9, 1)
    pins.digitalWritePin(DigitalPin.P10, 1)
    pins.digitalWritePin(DigitalPin.P11, 1)
    pins.digitalWritePin(DigitalPin.P12, 1)
    pins.digitalWritePin(DigitalPin.P13, 1)
    pins.digitalWritePin(DigitalPin.P14, 1)
    pins.digitalWritePin(DigitalPin.P15, 1)
    pins.digitalWritePin(DigitalPin.P16, 1)
  }
})

```

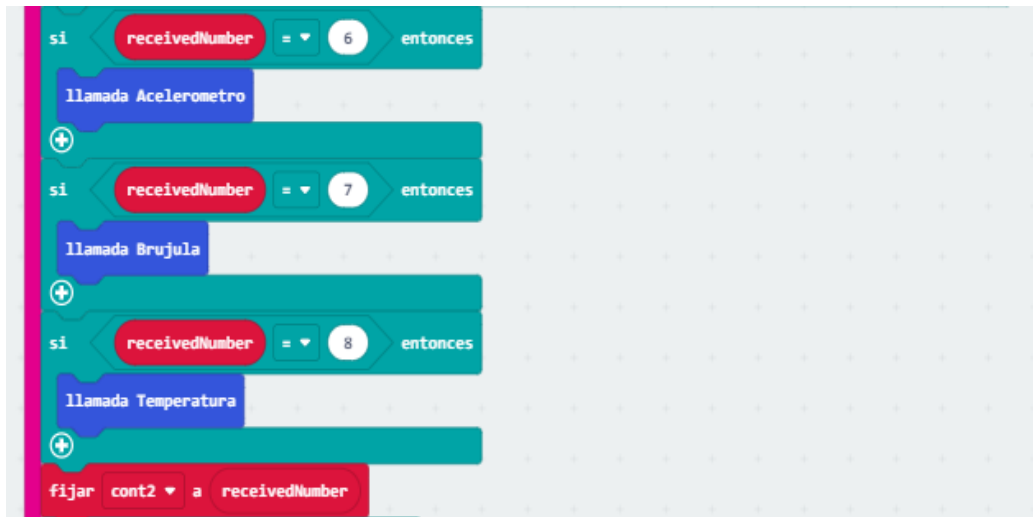
```

basic.forever(function () {
  if (cont2 == 95 || cont2 >= 115) {
    pins.digitalWritePin(DigitalPin.P0, 0)
    pins.digitalWritePin(DigitalPin.P1, 0)
    pins.digitalWritePin(DigitalPin.P2, 0)
    pins.digitalWritePin(DigitalPin.P3, 0)
    pins.digitalWritePin(DigitalPin.P4, 0)
    pins.digitalWritePin(DigitalPin.P5, 0)
    pins.digitalWritePin(DigitalPin.P6, 0)
    pins.digitalWritePin(DigitalPin.P7, 0)
    pins.digitalWritePin(DigitalPin.P8, 0)
    pins.digitalWritePin(DigitalPin.P9, 0)
    pins.digitalWritePin(DigitalPin.P10, 0)
    pins.digitalWritePin(DigitalPin.P11, 0)
    pins.digitalWritePin(DigitalPin.P12, 0)
    pins.digitalWritePin(DigitalPin.P13, 0)
    pins.digitalWritePin(DigitalPin.P14, 0)
    pins.digitalWritePin(DigitalPin.P15, 0)
    pins.digitalWritePin(DigitalPin.P16, 0)
  }
}

```

Al recibir un valor por radio:





```
radio.onReceivedNumber(function (receivedNumber)
{
  if (receivedNumber == 0) {
    LEDs()
  }
  if (receivedNumber == 1) {
    BotonA()
  }
  if (receivedNumber == 2) {
    BotonB()
  }
  if (receivedNumber == 3 && (control.hardware
Version()).includes("2. ")) {
    Logo()
  }
  if (receivedNumber == 3 && (control.hardware
Version()).includes("1. ")) {
    serial.writeLine("3. Prueba de logo no d
isponible en V1")
  }
  if (receivedNumber == 4) {
    Microfono()
  }
}
```

```
if (receivedNumber == 4 && (control.hardware
Version()).includes("1. ")) {
  serial.writeLine("4. Prueba de rújula a
no disponible en V1")
}
if (receivedNumber == 5) {
  Altavoz()
}
if (receivedNumber == 5 && (control.hardware
Version()).includes("1. ")) {
  serial.writeLine("5. Prueba de altavoz n
o disponible en V1")
}
if (receivedNumber == 6) {
  Acelerometro()
}
if (receivedNumber == 7) {
  Brujula()
}
if (receivedNumber == 8) {
  Temperatura()
}
cont2 = receivedNumber
```

```
si < cont2 = 80 > entonces
  serial escribir línea "Pin 0 correcto OWH"
+
si < cont2 = 81 > entonces
  serial escribir línea "Pin 1 correcto OWH"
+
si < cont2 = 82 > entonces
  serial escribir línea "Pin 2 correcto OWH"
+
si < cont2 = 83 > entonces
  serial escribir línea "Pin 3 correcto OWH"
+
si < cont2 = 84 > entonces
  serial escribir línea "Pin 4 correcto OWH"
+
si < cont2 = 85 > entonces
  serial escribir línea "Pin 5 correcto OWH"
+
si < cont2 = 86 > entonces
  serial escribir línea "Pin 6 correcto OWH"
+
si < cont2 = 87 > entonces
  serial escribir línea "Pin 7 correcto OWH"
+
si < cont2 = 88 > entonces
  serial escribir línea "Pin 8 correcto OWH"
+
```

```
si < cont2 = 89 > entonces
  serial escribir línea "Pin 9 correcto OWH"
+
si < cont2 = 90 > entonces
  serial escribir línea "Pin 10 correcto OWH"
+
si < cont2 = 91 > entonces
  serial escribir línea "Pin 11 correcto OWH"
+
si < cont2 = 92 > entonces
  serial escribir línea "Pin 12 correcto OWH"
+
si < cont2 = 93 > entonces
  serial escribir línea "Pin 13 correcto OWH"
+
si < cont2 = 94 > entonces
  serial escribir línea "Pin 14 correcto OWH"
+
si < cont2 = 95 > entonces
  serial escribir línea "Pin 15 correcto OWH"
+
si < cont2 = 100 > entonces
  serial escribir línea "Pin 0 KO OWH"
+
si < cont2 = 101 > entonces
  serial escribir línea "Pin 1 KO OWH"
+
```

```
si cont2 = 102 entonces
  serial escribir línea "Pin 2 KO OWH"
+
si cont2 = 103 entonces
  serial escribir línea "Pin 3 KO OWH"
+
si cont2 = 104 entonces
  serial escribir línea "Pin 4 KO OWH"
+
si cont2 = 105 entonces
  serial escribir línea "Pin 5 KO OWH"
+
si cont2 = 106 entonces
  serial escribir línea "Pin 6 KO OWH"
+
si cont2 = 107 entonces
  serial escribir línea "Pin 7 KO OWH"
+
si cont2 = 108 entonces
  serial escribir línea "Pin 8 KO OWH"
+
si cont2 = 109 entonces
  serial escribir línea "Pin 9 KO OWH"
+
si cont2 = 110 entonces
  serial escribir línea "Pin 10 KO OWH"
+
```

```
si cont2 = 111 entonces
  serial escribir línea "Pin 11 KO OWH"
+
si cont2 = 112 entonces
  serial escribir línea "Pin 12 KO OWH"
+
si cont2 = 113 entonces
  serial escribir línea "Pin 13 KO OWH"
+
si cont2 = 114 entonces
  serial escribir línea "Pin 14 KO OWH"
+
si cont2 = 115 entonces
  serial escribir línea "Pin 15 KO OWH"
+
si cont2 = 120 entonces
  serial escribir línea "Pin 0 correcto OWL"
+
si cont2 = 121 entonces
  serial escribir línea "Pin 1 correcto OWL"
+
si cont2 = 122 entonces
  serial escribir línea "Pin 2 correcto OWL"
+
si cont2 = 123 entonces
  serial escribir línea "Pin 3 correcto OWL"
+
```

```
si <cont2 = 124> entonces
  serial escribir línea "Pin 4 correcto OWL"
+
si <cont2 = 125> entonces
  serial escribir línea "Pin 5 correcto OWL"
+
si <cont2 = 126> entonces
  serial escribir línea "Pin 6 correcto OWL"
+
si <cont2 = 127> entonces
  serial escribir línea "Pin 7 correcto OWL"
+
si <cont2 = 128> entonces
  serial escribir línea "Pin 8 correcto OWL"
+
si <cont2 = 129> entonces
  serial escribir línea "Pin 9 correcto OWL"
+
si <cont2 = 130> entonces
  serial escribir línea "Pin 10 correcto OWL"
+
si <cont2 = 131> entonces
  serial escribir línea "Pin 11 correcto OWL"
+
si <cont2 = 132> entonces
  serial escribir línea "Pin 12 correcto OWL"
+
si <cont2 = 133> entonces
  serial escribir línea "Pin 13 correcto OWL"
+
si <cont2 = 134> entonces
  serial escribir línea "Pin 14 correcto OWL"
+
si <cont2 = 135> entonces
  serial escribir línea "Pin 15 correcto OWL"
+
si <cont2 = 140> entonces
  serial escribir línea "Pin 0 KO OWL"
+
si <cont2 = 141> entonces
  serial escribir línea "Pin 1 KO OWL"
+
si <cont2 = 142> entonces
  serial escribir línea "Pin 2 KO OWL"
+
si <cont2 = 143> entonces
  serial escribir línea "Pin 3 KO OWL"
+
si <cont2 = 144> entonces
  serial escribir línea "Pin 4 KO OWL"
+
si <cont2 = 145> entonces
  serial escribir línea "Pin 5 KO OWL"
+
```

```

si cont2 = 146 entonces
  serial escribir línea "Pin 6 KO OWL"
+
si cont2 = 147 entonces
  serial escribir línea "Pin 7 KO OWL"
+
si cont2 = 148 entonces
  serial escribir línea "Pin 8 KO OWL"
+
si cont2 = 149 entonces
  serial escribir línea "Pin 9 KO OWL"
+
si cont2 = 150 entonces
  serial escribir línea "Pin 10 KO OWL"
+
si cont2 = 151 entonces
  serial escribir línea "Pin 11 KO OWL"
+
si cont2 = 152 entonces
  serial escribir línea "Pin 12 KO OWL"
+
si cont2 = 153 entonces
  serial escribir línea "Pin 13 KO OWL"
+
si cont2 = 154 entonces
  serial escribir línea "Pin 14 KO OWL"
+

```

```

si cont2 = 155 entonces
  serial escribir línea "Pin 15 KO OWL"
+
si receivedNumber = 42 entonces
  pausa (ms) 100
  si lectura digital pin P0 = 1 entonces
    serial escribir línea "Pin 0 correcto ORH"
  si no
    serial escribir línea "Pin 0 KO ORH"
+
si receivedNumber = 43 entonces
  si lectura digital pin P1 = 1 entonces
    serial escribir línea "Pin 1 correcto ORH"
  si no
    serial escribir línea "Pin 1 KO ORH"
+
si receivedNumber = 44 entonces
  si lectura digital pin P2 = 1 entonces
    serial escribir línea "Pin 2 correcto ORH"
  si no
    serial escribir línea "Pin 2 KO ORH"
+

```



```

si receivedNumber = 45 entonces
  si lectura digital pin P3 = 1 entonces
    serial escribir línea "Pin 3 correcto ORH"
  si no
    serial escribir línea "Pin 3 KO ORH"
+
si receivedNumber = 46 entonces
  si lectura digital pin P4 = 1 entonces
    serial escribir línea "Pin 4 correcto ORH"
  si no
    serial escribir línea "Pin 4 KO ORH"
+
si receivedNumber = 47 entonces
  si lectura digital pin P5 = 1 entonces
    serial escribir línea "Pin 5 correcto ORH"
  si no
    serial escribir línea "Pin 5 KO ORH"
+
si receivedNumber = 48 entonces
  si lectura digital pin P6 = 1 entonces
    serial escribir línea "Pin 6 correcto ORH"
  si no
    serial escribir línea "Pin 6 KO ORH"
+

```

```

si receivedNumber = 49 entonces
  si lectura digital pin P7 = 1 entonces
    serial escribir línea "Pin 7 correcto ORH"
  si no
    serial escribir línea "Pin 7 KO ORH"
+
si receivedNumber = 50 entonces
  si lectura digital pin P8 = 1 entonces
    serial escribir línea "Pin 8 correcto ORH"
  si no
    serial escribir línea "Pin 8 KO ORH"
+
si receivedNumber = 51 entonces
  si lectura digital pin P9 = 1 entonces
    serial escribir línea "Pin 9 correcto ORH"
  si no
    serial escribir línea "Pin 9 KO ORH"
+
si receivedNumber = 52 entonces
  si lectura digital pin P10 = 1 entonces
    serial escribir línea "Pin 10 correcto ORH"
  si no
    serial escribir línea "Pin 10 KO ORH"
+

```

```
si receivedNumber = 53 entonces
  si lectura digital pin P11 = 1 entonces
    serial escribir línea "Pin 11 correcto ORH"
  si no
    serial escribir línea "Pin 11 KO ORH"
+
+
si receivedNumber = 54 entonces
  si lectura digital pin P12 = 1 entonces
    serial escribir línea "Pin 12 correcto ORH"
  si no
    serial escribir línea "Pin 12 KO ORH"
+
+
si receivedNumber = 55 entonces
  si lectura digital pin P13 = 1 entonces
    serial escribir línea "Pin 13 correcto ORH"
  si no
    serial escribir línea "Pin 13 KO ORH"
+
+
si receivedNumber = 56 entonces
  si lectura digital pin P14 = 1 entonces
    serial escribir línea "Pin 14 correcto ORH"
  si no
    serial escribir línea "Pin 14 KO ORH"
+
+

```

```
si receivedNumber = 57 entonces
  si lectura digital pin P15 = 1 entonces
    serial escribir línea "Pin 15 correcto ORH"
  si no
    serial escribir línea "Pin 15 KO ORH"
+
+
si receivedNumber = 58 entonces
  si lectura digital pin P0 = 0 entonces
    serial escribir línea "Pin 0 correcto ORL"
  si no
    serial escribir línea "Pin 0 KO ORL"
+
+
si receivedNumber = 59 entonces
  si lectura digital pin P1 = 0 entonces
    serial escribir línea "Pin 1 correcto ORL"
  si no
    serial escribir línea "Pin 1 KO ORL"
+
+
si receivedNumber = 60 entonces
  si lectura digital pin P2 = 0 entonces
    serial escribir línea "Pin 2 correcto ORL"
  si no
    serial escribir línea "Pin 2 KO ROL"
+
+

```

```
si receivedNumber = 61 entonces
si lectura digital pin P3 = 0 entonces
serial escribir línea "Pin 3 correcto ORL"
si no
serial escribir línea "Pin 3 KO ROL"
+
+
si receivedNumber = 62 entonces
si lectura digital pin P4 = 0 entonces
serial escribir línea "Pin 4 correcto ORL"
si no
serial escribir línea "Pin 4 KO ROL"
+
+
si receivedNumber = 63 entonces
si lectura digital pin P5 = 0 entonces
serial escribir línea "Pin 5 correcto ORL"
si no
serial escribir línea "Pin 5 KO ROL"
+
+
si receivedNumber = 64 entonces
si lectura digital pin P6 = 0 entonces
serial escribir línea "Pin 6 correcto ORL"
si no
serial escribir línea "Pin 6 KO ROL"
+
+
```

```
si receivedNumber = 65 entonces
si lectura digital pin P7 = 0 entonces
serial escribir línea "Pin 7 correcto ORL"
si no
serial escribir línea "Pin 7 KO ROL"
+
+
si receivedNumber = 66 entonces
si lectura digital pin P8 = 0 entonces
serial escribir línea "Pin 8 correcto ORL"
si no
serial escribir línea "Pin 8 KO ROL"
+
+
si receivedNumber = 67 entonces
si lectura digital pin P9 = 0 entonces
serial escribir línea "Pin 9 correcto ORL"
si no
serial escribir línea "Pin 9 KO ROL"
+
+
si receivedNumber = 68 entonces
si lectura digital pin P10 = 0 entonces
serial escribir línea "Pin 10 correcto ORL"
si no
serial escribir línea "Pin 10 KO ROL"
+
+
```

```
si receivedNumber = 69 entonces
  si lectura digital pin P11 = 0 entonces
    serial escribir línea "Pin 11 correcto ORL"
  si no
    serial escribir línea "Pin 11 KO ROL"
+
+
si receivedNumber = 70 entonces
  si lectura digital pin P12 = 0 entonces
    serial escribir línea "Pin 12 correcto ORL"
  si no
    serial escribir línea "Pin 12 KO ROL"
+
+
si receivedNumber = 71 entonces
  si lectura digital pin P13 = 0 entonces
    serial escribir línea "Pin 13 correcto ORL"
  si no
    serial escribir línea "Pin 13 KO ROL"
+
+
si receivedNumber = 72 entonces
  si lectura digital pin P14 = 0 entonces
    serial escribir línea "Pin 14 correcto ORL"
  si no
    serial escribir línea "Pin 14 KO ROL"
+
+
si receivedNumber = 73 entonces
  si lectura digital pin P15 = 0 entonces
    serial escribir línea "Pin 15 correcto ORL"
  si no
    serial escribir línea "Pin 15 KO ROL"
+
+
```

```

if (cont2 == 80) {
    serial.writeLine("Pin
0 correcto OWH")
}
if (cont2 == 81) {
    serial.writeLine("Pin
1 correcto OWH")
}
if (cont2 == 82) {
    serial.writeLine("Pin
2 correcto OWH")
}
if (cont2 == 83) {
    serial.writeLine("Pin
3 correcto OWH")
}
if (cont2 == 84) {
    serial.writeLine("Pin
4 correcto OWH")
}
if (cont2 == 85) {
    serial.writeLine("Pin
5 correcto OWH")
}
if (cont2 == 86) {
    serial.writeLine("Pin
6 correcto OWH")
}
if (cont2 == 87) {
    serial.writeLine("Pin
7 correcto OWH")
}
if (cont2 == 88) {
    serial.writeLine("Pin
8 correcto OWH")
}
if (cont2 == 89) {
    serial.writeLine("Pin
9 correcto OWH")
}
if (cont2 == 90) {
    serial.writeLine("Pin
10 correcto OWH")
}
if (cont2 == 91) {
    serial.writeLine("Pin
11 correcto OWH")
}
if (cont2 == 92) {
    serial.writeLine("Pin
12 correcto OWH")
}
if (cont2 == 93) {
    serial.writeLine("Pin
13 correcto OWH")
}
if (cont2 == 94) {
    serial.writeLine("Pin
14 correcto OWH")
}
if (cont2 == 95) {
    serial.writeLine("Pin
15 correcto OWH")
}
}
if (cont2 == 100) {
    serial.writeLine("Pin
0 KO OWH")
}
if (cont2 == 101) {
    serial.writeLine("Pin
1 KO OWH")
}
if (cont2 == 102) {
    serial.writeLine("Pin
2 KO OWH")
}
if (cont2 == 103) {
    serial.writeLine("Pin
3 KO OWH")
}
if (cont2 == 104) {
    serial.writeLine("Pin
4 KO OWH")
}
if (cont2 == 105) {
    serial.writeLine("Pin
5 KO OWH")
}
if (cont2 == 106) {
        serial.writeLine("Pin
6 KO OWH")
    }
    if (cont2 == 107) {
        serial.writeLine("Pin
7 KO OWH")
    }
    if (cont2 == 108) {
        serial.writeLine("Pin
8 KO OWH")
    }
    if (cont2 == 109) {
        serial.writeLine("Pin
9 KO OWH")
    }
    if (cont2 == 110) {
        serial.writeLine("Pin
10 KO OWH")
    }
    if (cont2 == 111) {
        serial.writeLine("Pin
11 KO OWH")
    }
    if (cont2 == 112) {
        serial.writeLine("Pin
12 KO OWH")
    }
    if (cont2 == 113) {
        serial.writeLine("Pin
13 KO OWH")
    }
    if (cont2 == 114) {
        serial.writeLine("Pin
14 KO OWH")
    }
    if (cont2 == 115) {
        serial.writeLine("Pin
15 KO OWH")
    }
    if (cont2 == 120) {
        serial.writeLine("Pin
0 correcto OWL")
    }
    if (cont2 == 121) {
        serial.writeLine("Pin
1 correcto OWL")
    }
    if (cont2 == 122) {
        serial.writeLine("Pin
2 correcto OWL")
    }
    if (cont2 == 123) {
        serial.writeLine("Pin
3 correcto OWL")
    }
    if (cont2 == 124) {
        serial.writeLine("Pin
4 correcto OWL")
    }
    if (cont2 == 125) {
        serial.writeLine("Pin
5 correcto OWL")
    }
    if (cont2 == 126) {
        serial.writeLine("Pin
6 correcto OWL")
    }
    if (cont2 == 127) {
        serial.writeLine("Pin
7 correcto OWL")
    }
    if (cont2 == 128) {
        serial.writeLine("Pin
8 correcto OWL")
    }
    if (cont2 == 129) {
        serial.writeLine("Pin
9 correcto OWL")
    }
    if (cont2 == 130) {
        serial.writeLine("Pin
10 correcto OWL")
    }
    if (cont2 == 131) {
        serial.writeLine("Pin
11 correcto OWL")
    }
    if (cont2 == 132) {
        serial.writeLine("Pin
12 correcto OWL")
    }
}
if (cont2 == 133) {
    serial.writeLine("Pin
13 correcto OWL")
}
if (cont2 == 134) {
    serial.writeLine("Pin
14 correcto OWL")
}
if (cont2 == 135) {
    serial.writeLine("Pin
15 correcto OWL")
}
if (cont2 == 140) {
    serial.writeLine("Pin
0 KO OWL")
}
if (cont2 == 141) {
    serial.writeLine("Pin
1 KO OWL")
}
if (cont2 == 142) {
    serial.writeLine("Pin
2 KO OWL")
}
if (cont2 == 143) {
    serial.writeLine("Pin
3 KO OWL")
}
if (cont2 == 144) {
    serial.writeLine("Pin
4 KO OWL")
}
if (cont2 == 145) {
    serial.writeLine("Pin
5 KO OWL")
}
if (cont2 == 146) {
    serial.writeLine("Pin
6 KO OWL")
}
if (cont2 == 147) {
    serial.writeLine("Pin
7 KO OWL")
}
if (cont2 == 148) {
    serial.writeLine("Pin
8 KO OWL")
}
if (cont2 == 149) {
    serial.writeLine("Pin
9 KO OWL")
}
if (cont2 == 150) {
    serial.writeLine("Pin
10 KO OWL")
}
if (cont2 == 151) {
    serial.writeLine("Pin
11 KO OWL")
}
if (cont2 == 152) {
    serial.writeLine("Pin
12 KO OWL")
}
if (cont2 == 153) {
    serial.writeLine("Pin
13 KO OWL")
}
if (cont2 == 154) {
    serial.writeLine("Pin
14 KO OWL")
}
if (cont2 == 155) {
    serial.writeLine("Pin
15 KO OWL")
}
}
if (receivedNumber == 42)
{
    basic.pause(100)
    if (pins.digitalReadP
in(DigitalPin.P0) == 1) {
        serial.writeLine(
"Pin 0 correcto ORH")
    } else {
        serial.writeLine(
"Pin 0 KO ORH")
    }
}
}

```





```

    } else {
        serial.writeLine(
"Pin 9 KO ROL")
    }
    if (receivedNumber == 68)
    {
        if (pins.digitalReadP
in(DigitalPin.P10) == 0) {
            serial.writeLine(
"Pin 10 correcto ORL")
        } else {
            serial.writeLine(
"Pin 10 KO ROL")
        }
    }
    if (receivedNumber == 69)
    {
        if (pins.digitalReadP
in(DigitalPin.P11) == 0) {
            serial.writeLine(
"Pin 11 correcto ORL")
        } else {
            serial.writeLine(
"Pin 11 KO ROL")
        }
    }
}
}
}

```

```

    }
    if (receivedNumber == 70)
    {
        if (pins.digitalReadP
in(DigitalPin.P12) == 0) {
            serial.writeLine(
"Pin 12 correcto ORL")
        } else {
            serial.writeLine(
"Pin 12 KO ROL")
        }
    }
    if (receivedNumber == 71)
    {
        if (pins.digitalReadP
in(DigitalPin.P13) == 0) {
            serial.writeLine(
"Pin 13 correcto ORL")
        } else {
            serial.writeLine(
"Pin 13 KO ROL")
        }
    }
}
}
}

```

```

    if (receivedNumber == 72)
    {
        if (pins.digitalReadP
in(DigitalPin.P14) == 0) {
            serial.writeLine(
"Pin 14 correcto ORL")
        } else {
            serial.writeLine(
"Pin 14 KO ROL")
        }
    }
    if (receivedNumber == 73)
    {
        if (pins.digitalReadP
in(DigitalPin.P15) == 0) {
            serial.writeLine(
"Pin 15 correcto ORL")
        } else {
            serial.writeLine(
"Pin 15 KO ROL")
        }
    }
}
}
}

```



```

if (receivedNumber >= 8 && receivedNumber <=
155) {
    Pruebas_Digitales()
}

```

```

if (receivedNumber >= 160 && receivedNumber
<= 165) {
    Pruebas_Analogicas()
}
if (receivedNumber == 160) {

```

```

        serial.writeLine("Valor rújula rúju P0
:" + pins.analogReadPin(AnalogPin.P0) * (3.3 / 1
024) + "V")
    }
    if (receivedNumber == 161) {
        serial.writeLine("Valor rújula rúju P0
:" + pins.analogReadPin(AnalogPin.P1) * (3.3 / 1
024) + "V")
    }
    if (receivedNumber == 162) {
        serial.writeLine("Valor rújula rúju P0
:" + pins.analogReadPin(AnalogPin.P2) * (3.3 / 1
024) + "V")
    }
    if (receivedNumber == 163) {

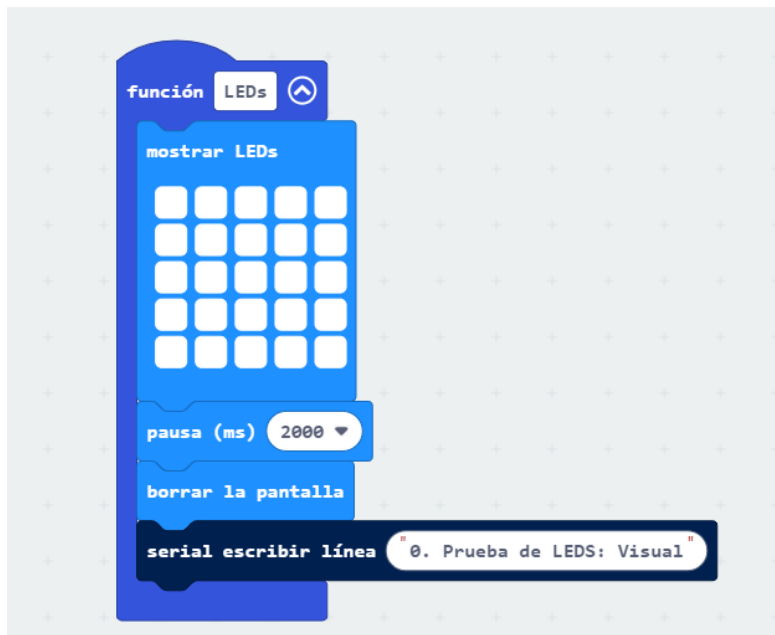
```

```

        serial.writeLine("Valor rújula rúju P0
:" + pins.analogReadPin(AnalogPin.P3) * (3.3 / 1
024) + "V")
    }
    if (receivedNumber == 164) {
        serial.writeLine("Valor rújula rúju P0
:" + pins.analogReadPin(AnalogPin.P4) * (3.3 / 1
024) + "V")
    }
    if (receivedNumber == 165) {
        serial.writeLine("Valor rújula rúju P0
:" + pins.analogReadPin(AnalogPin.P10) * (3.3 /
1024) + "V")
    }
}
})

```

## 9.2.1 Matriz de LEDs (0)



El código empleado en JavaScript es el siguiente:

```

if (receivedNumber == 0) {
    basic.showLeds(`
        # # # #
        # # # #
        # # # #
        # # # #
        # # # #
    `)
    basic.pause(2000)
    basic.clearScreen()
    serial.writeLine("Prueba de LEDs: Visual")
}

```



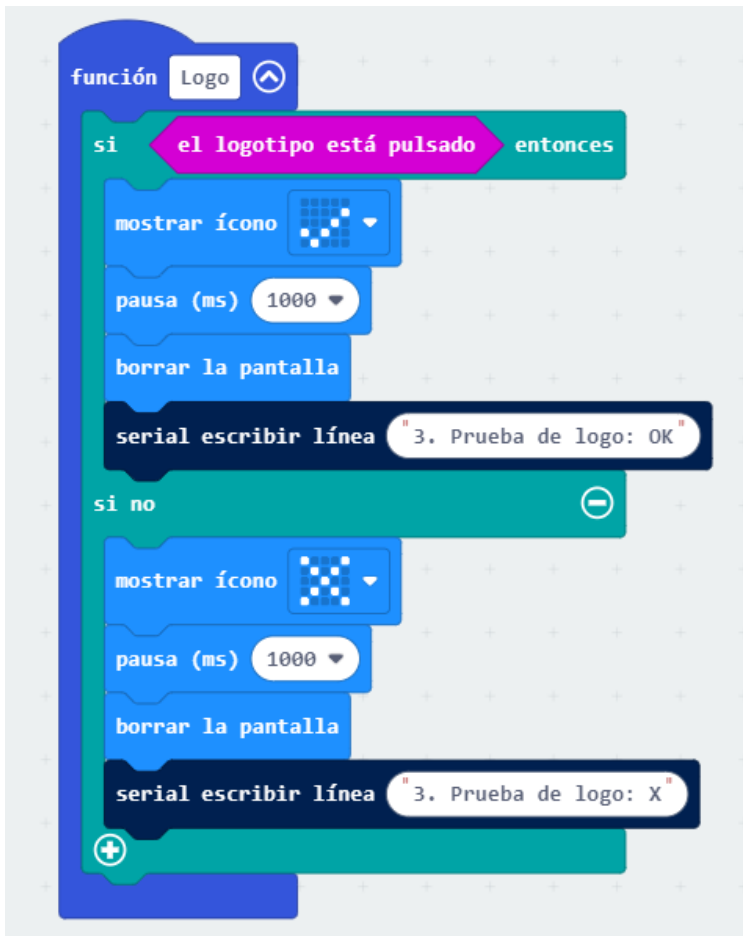
## 9.2.2 Botones programables (1 y 2)



```
function BotonB () {
  basic.pause(1500)
  if (input.buttonIsPressed(Button.B)) {
    basic.showIcon(IconNames.SmallDiamond)
    basic.pause(500)
    basic.clearScreen()
    if (input.buttonIsPressed(Button.B)) {
      basic.showIcon(IconNames.No)
      basic.pause(500)
      basic.clearScreen()
      serial.writeLine("2. Prueba de botón B: X")
    } else {
      basic.showIcon(IconNames.Yes)
      basic.pause(500)
      basic.clearScreen()
      serial.writeLine("2. Prueba de botón B: OK")
    }
  } else {
    basic.showIcon(IconNames.No)
    basic.pause(500)
    basic.clearScreen()
    serial.writeLine("2. Prueba de botón B: X")
  }
}
```

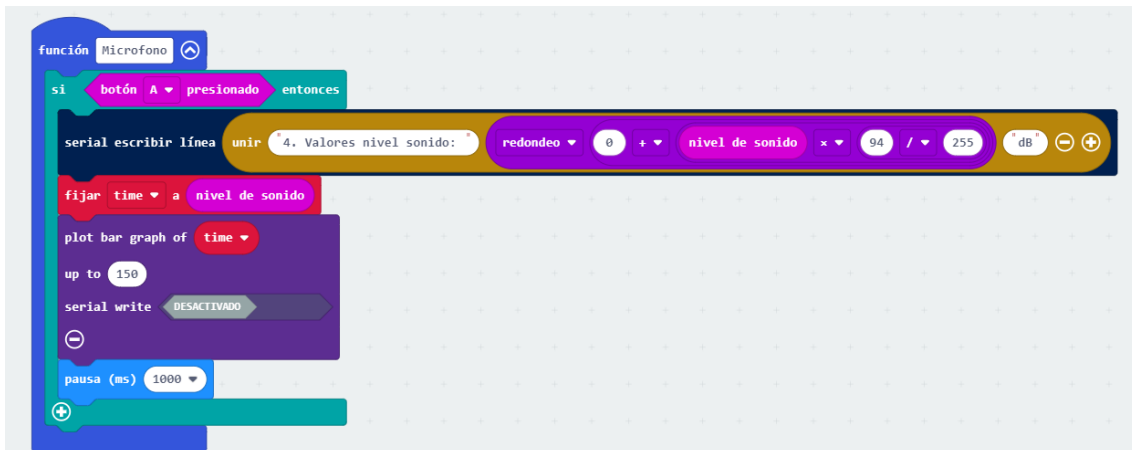
```
function BotonA () {
  basic.pause(1500)
  if (input.buttonIsPressed(Button.A)) {
    basic.showIcon(IconNames.SmallDiamond)
    basic.pause(500)
    basic.clearScreen()
    if (input.buttonIsPressed(Button.A)) {
      basic.showIcon(IconNames.No)
      basic.pause(500)
      basic.clearScreen()
      serial.writeLine("1. Prueba de botón A: X")
    } else {
      basic.showIcon(IconNames.Yes)
      basic.pause(500)
      basic.clearScreen()
      serial.writeLine("1. Prueba de botón A: OK")
    }
  } else {
    basic.showIcon(IconNames.No)
    basic.pause(500)
    basic.clearScreen()
    serial.writeLine("1. Prueba de botón A: X")
  }
}
```

### 9.2.3 Logo sensible al contacto (3)



```
function Logo () {  
  if (input.logoIsPressed()) {  
    basic.showIcon(IconNames.Yes)  
    basic.pause(1000)  
    basic.clearScreen()  
    serial.writeline("3. Prueba de logo: OK")  
  } else {  
    basic.showIcon(IconNames.No)  
    basic.pause(1000)  
    basic.clearScreen()  
    serial.writeline("3. Prueba de logo: X")  
  }  
}
```

## Micrófono (4)



Código JavaScript:

```
function Microfono () {  
  if (input.buttonIsPressed(Button.A)) {  
    serial.writeLine("4. rújula a a sonido: " + Math.round(0 + input.soundLevel() * (94 / 255)) + "  
dB")  
    time = input.soundLevel()  
    led.plotBarGraph(  
      time,  
      150,  
      false  
    )  
    basic.pause(1000)  
  }  
}
```

## 9.2.4 Altavoz (5)



Código JavaScript:

```
function Altavoz () {  
  serial.writeLine("5. Prueba altavoz: Escuchar")  
  basic.showIcon(IconNames.EighthNote)  
  music.play(music.stringPlayable("B A G A G F A C5 ", 125), music.PlaybackMode.UntilDone)  
}
```

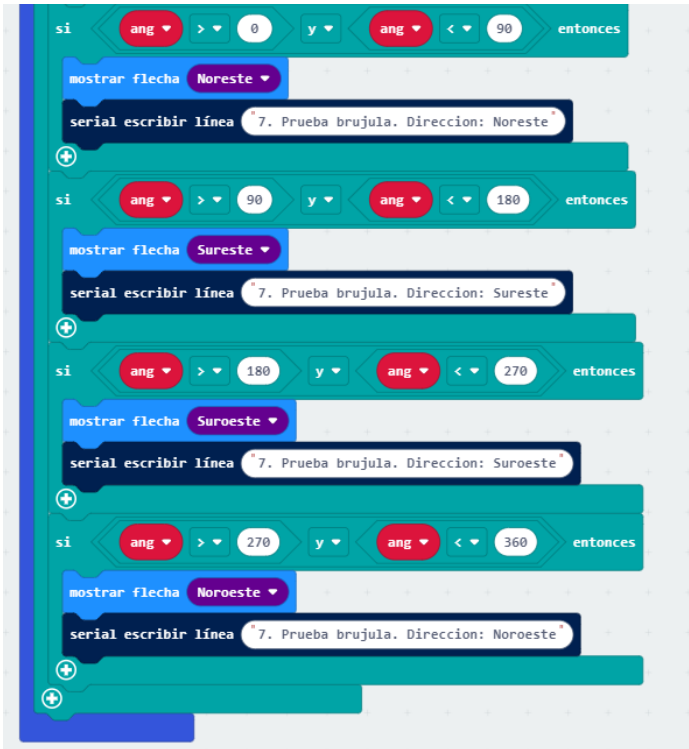
## 9.2.5 Acelerómetro (6)



```
function Acelerometro () {  
  if (input.buttonIsPressed(Button.A)) {  
    for (let index = 0; index < 1; index++) {  
      serial.writeLine("6. Prueba rújula a a : Llenar matriz de LEDs")  
      input.calibrateCompass()  
    }  
  }  
}
```

## 9.2.6 Brújula (7)

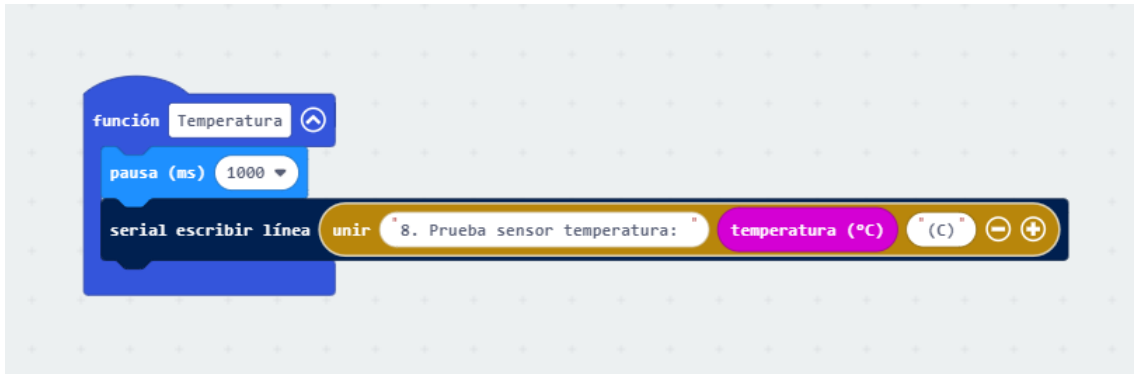




```
function Brujula () {
  if (input.buttonIsPressed(Button.A)) {
    ang = input.compassHeading()
    if (ang == 0) {
      basic.showArrow(ArrowNames.North)
      serial.writeLine("7. Prueba rújula.
Direccion:Norte")
    }
    if (ang == 90) {
      basic.showArrow(ArrowNames.East)
      serial.writeLine("7. Prueba rújula.
Direccion:Este")
    }
    if (ang == 180) {
      basic.showArrow(ArrowNames.South)
      serial.writeLine("7. Prueba rújula.
Direccion:Sur")
    }
    if (ang == 270) {
      basic.showArrow(ArrowNames.West)
      serial.writeLine("7. Prueba rújula.
Direccion: Oeste")
    }
    if (ang > 0 && ang < 90) {
      basic.showArrow(ArrowNames.NorthEast)
    }
  }
}
```

```
serial.writeLine("7. Prueba rújula.
Direccion: Noreste")
}
if (ang > 90 && ang < 180) {
  basic.showArrow(ArrowNames.SouthEast)
}
serial.writeLine("7. Prueba rújula.
Direccion: Sureste")
}
if (ang > 180 && ang < 270) {
  basic.showArrow(ArrowNames.SouthWest)
}
serial.writeLine("7. Prueba rújula.
Direccion: Suroeste")
}
if (ang > 270 && ang < 360) {
  basic.showArrow(ArrowNames.NorthWest)
}
serial.writeLine("7. Prueba rújula.
Direccion: Noroeste")
}
}
let ang = 0
```

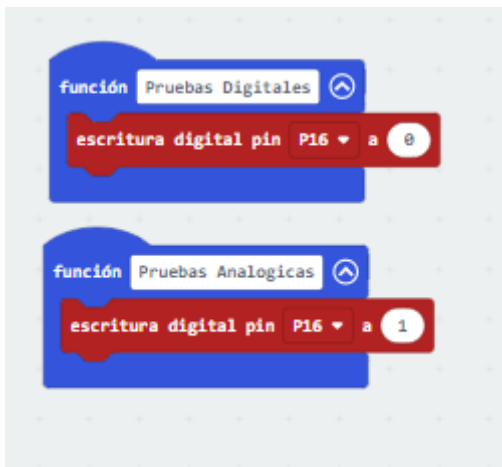
## 9.2.7 Sensor de temperatura (8)



Código JavaScript:

```
function Temperatura () {  
  basic.pause(1000)  
  serial.writeLine("8. Prueba sensor temperatura: " + input.temperature() + "°C")  
}
```

## 9.2.8 Habilitación conmutador



```
function Pruebas_Digitales () {  
  pins.digitalWritePin(DigitalPin.P16, 0)}  
function Pruebas_Analogicas () {  
  pins.digitalWritePin(DigitalPin.P16, 1)}
```

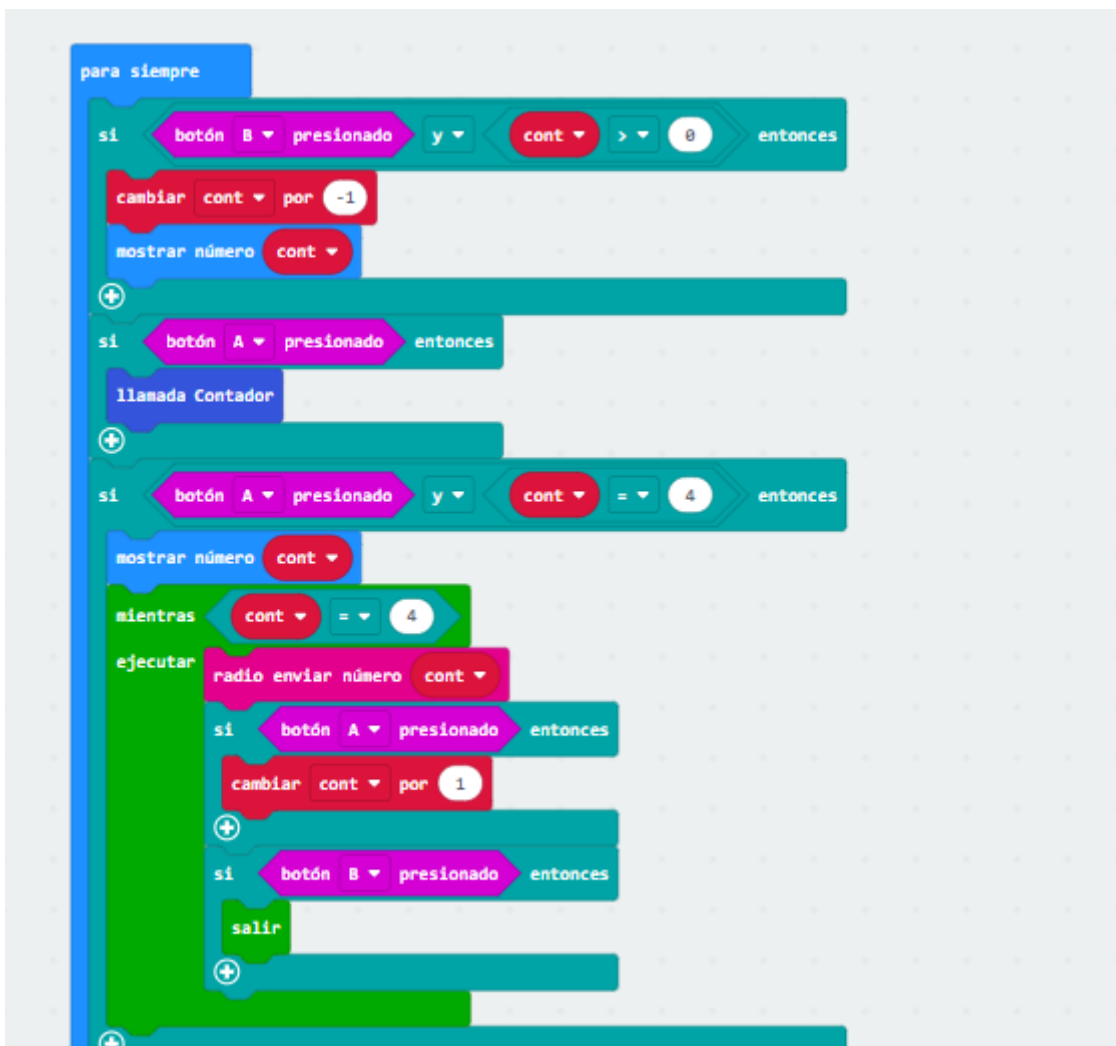
### 9.3 Código cargado en tarjeta referencia o madre

Al iniciar:



```
let cont2 = 0
let cont = 0
radio.setGroup(0)
cont = 0
```

En un bucle:



The image shows two identical Scratch code blocks. Each block starts with a 'si' (if) block: 'si botón A presionado y cont = 6 entonces'. This is followed by a 'mostrar número' block with 'cont'. Then a 'mientras' (while) loop: 'mientras cont = 7'. Inside the loop, there is an 'ejecutar' block containing: 'radio enviar número cont', 'si botón A presionado entonces' (followed by 'cambiar cont por 1'), and 'si botón B presionado entonces' (followed by 'salir'). The code blocks are connected by plus signs, indicating they are part of a larger script.



```

si [botón A] presionado y [cont] = 9 entonces
  mostrar número [cont]
  escritura digital pin P1 a 0
  escritura digital pin P2 a 0
  escritura digital pin P3 a 0
  escritura digital pin P4 a 0
  fijar [cont2] a 80
  pausa (ms) 1000
  repetir 1 veces
  ejecutar
  si [lectura digital pin P0] = 1 entonces
    radio enviar número [cont2]
  +
  si [lectura digital pin P0] = 0 entonces
    radio enviar número 100
  +
  cambiar [cont] por 1
  pausa (ms) 1000
+
si [cont] = 10 entonces
  escritura digital pin P1 a 0
  escritura digital pin P2 a 0
  escritura digital pin P3 a 0
  escritura digital pin P4 a 1
  fijar [cont2] a 81
  pausa (ms) 1000
  repetir 1 veces
  ejecutar
  si [lectura digital pin P0] = 1 entonces
    radio enviar número [cont2]
  +
  si [lectura digital pin P0] = 0 entonces
    radio enviar número 101
  +
  cambiar [cont] por 1
  pausa (ms) 1000
+

```

Código progresivo de lectura a nivel alto desde el pin 0 hasta el pin 15.

```

si cont = 25 entonces
  escritura digital pin P1 a 0
  escritura digital pin P2 a 0
  escritura digital pin P3 a 0
  escritura digital pin P4 a 0
  fijar cont2 a 120
  pausa (ms) 1000
  repetir 1 veces
  ejecutar
  si lectura digital pin P0 = 0 entonces
    radio enviar número cont2
  +
  si lectura digital pin P0 = 1 entonces
    radio enviar número 140
  +
  cambiar cont por 1
  pausa (ms) 1000

```

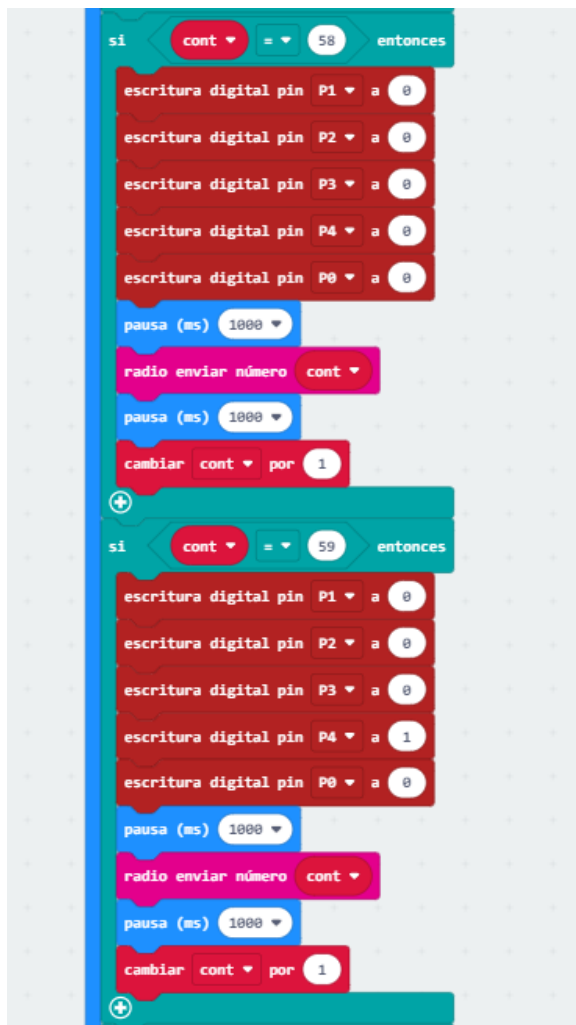
Código progresivo de lectura a nivel bajo desde el pin 0 hasta el pin 15.

```

si cont = 42 entonces
  mostrar número cont
  escritura digital pin P1 a 0
  escritura digital pin P2 a 0
  escritura digital pin P3 a 0
  escritura digital pin P4 a 0
  escritura digital pin P0 a 1
  pausa (ms) 1000
  radio enviar número cont
  pausa (ms) 1000
  cambiar cont por 1
  +
  si cont = 43 entonces
    escritura digital pin P1 a 0
    escritura digital pin P2 a 0
    escritura digital pin P3 a 0
    escritura digital pin P4 a 1
    escritura digital pin P0 a 1
    pausa (ms) 1000
    radio enviar número cont
    pausa (ms) 1000
    cambiar cont por 1
  +

```

Código progresivo de escritura a nivel alto desde el pin 0 hasta el pin 15.



Código progresivo de escritura a nivel bajo desde el pin 0 hasta el pin 15.

```

Basic.forever(function () {
  if (input.buttonIsPressed
(Button.B) && cont > 0) {
    cont += -1
    basic.showNumber(cont)
  }
  if (input.buttonIsPressed
(Button.A)) {
    Contador()
    if (input.buttonIsPressed
(Button.A) && cont == 4) {
      basic.showNumber(cont)
    }
    while (cont == 4) {
      radio.sendNumber(
cont)
      if (input.buttonI
sPressed(Button.A)) {
        cont += 1
      }
      if (input.buttonI
sPressed(Button.B)) {
        break;
      }
    }
    if (input.buttonIsPressed
(Button.A) && cont == 6) {
      basic.showNumber(cont)
      while (cont == 6) {
        radio.sendNumber(
cont)
        if (input.buttonI
sPressed(Button.A)) {
          cont += 1
        }
        if (input.buttonI
sPressed(Button.B)) {
          break;
        }
      }
      if (input.buttonIsPressed
(Button.A) && cont == 7) {
        basic.showNumber(cont)
        while (cont == 7) {
          radio.sendNumber(
cont)
          if (input.buttonI
sPressed(Button.A)) {
            cont += 1
          }
          if (input.buttonI
sPressed(Button.B)) {
            break;
          }
        }
        if (input.buttonIsPressed
(Button.A) && cont == 9) {
          basic.showNumber(cont)
          pins.digitalWritePin(
DigitalPin.P1, 0)
          pins.digitalWritePin(
DigitalPin.P2, 0)
          pins.digitalWritePin(
DigitalPin.P3, 0)
          pins.digitalWritePin(
DigitalPin.P4, 0)
          cont2 = 80
          basic.pause(1000)
          for (let index = 0; i
ndex < 1; index++) {
            if (pins.digitalR
eadPin(DigitalPin.P0) == 1) {
              radio.sendNum
ber(cont2)
            }
            if (pins.digitalR
eadPin(DigitalPin.P0) == 0) {
              radio.sendNum
ber(100)
            }
          }
          cont += 1
          basic.pause(1000)
        }
      }
    }
  }
}

```















```

    basic.pause(1000)
    cont += 1
  }
  if (cont == 72) {
    pins.digitalWritePin(
DigitalPin.P1, 1)
    pins.digitalWritePin(
DigitalPin.P2, 1)
    pins.digitalWritePin(
DigitalPin.P3, 1)
    pins.digitalWritePin(
DigitalPin.P4, 0)
  }

```

```

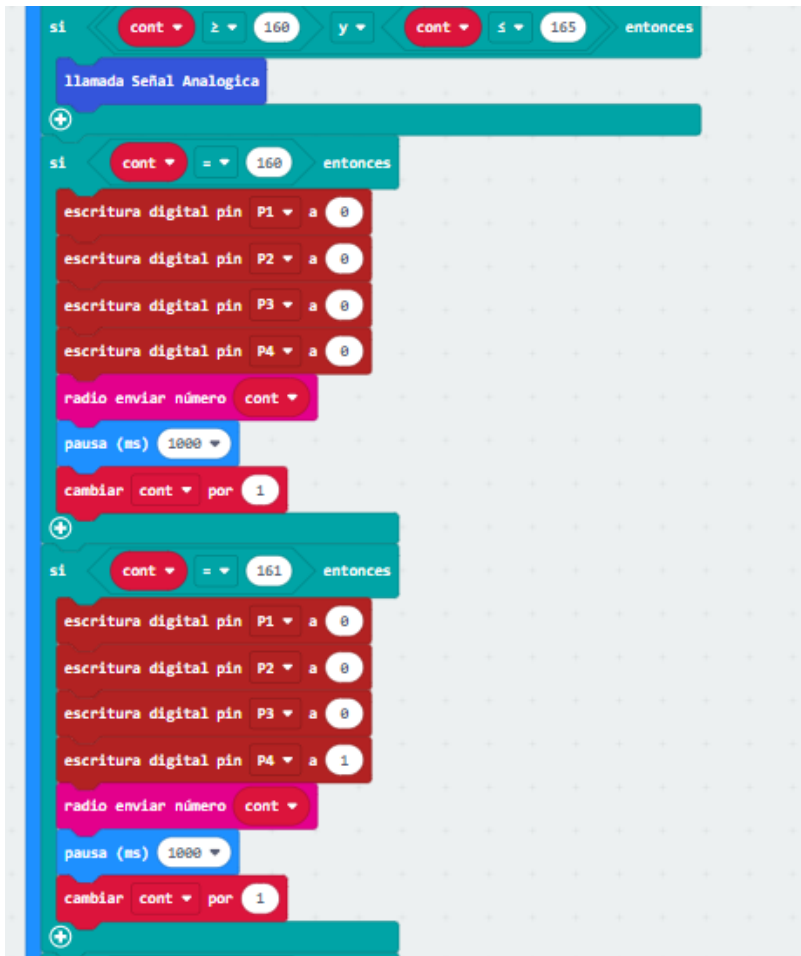
    pins.digitalWritePin(
DigitalPin.P0, 0)
    basic.pause(1000)
    radio.sendNumber(cont)
  }
  basic.pause(1000)
  cont += 1
  if (cont == 73) {
    pins.digitalWritePin(
DigitalPin.P1, 1)
    pins.digitalWritePin(
DigitalPin.P2, 1)
  }

```

```

    pins.digitalWritePin(
DigitalPin.P3, 1)
    pins.digitalWritePin(
DigitalPin.P4, 1)
    pins.digitalWritePin(
DigitalPin.P0, 0)
    basic.pause(1000)
    radio.sendNumber(cont)
  }
  basic.pause(1000)
  cont += 160
}
})

```



```
si cont = 162 entonces
  escritura digital pin P1 a 0
  escritura digital pin P2 a 0
  escritura digital pin P3 a 1
  escritura digital pin P4 a 0
  radio enviar número cont
  pausa (ms) 1000
  cambiar cont por 1
+
si cont = 163 entonces
  escritura digital pin P1 a 0
  escritura digital pin P2 a 0
  escritura digital pin P3 a 1
  escritura digital pin P4 a 1
  radio enviar número cont
  pausa (ms) 1000
  cambiar cont por 1
+
si cont = 164 entonces
  escritura digital pin P1 a 0
  escritura digital pin P2 a 1
  escritura digital pin P3 a 0
  escritura digital pin P4 a 0
  radio enviar número cont
  pausa (ms) 1000
  cambiar cont por 1
+
si cont = 165 entonces
  escritura digital pin P1 a 1
  escritura digital pin P2 a 0
  escritura digital pin P3 a 1
  escritura digital pin P4 a 0
  radio enviar número cont
  pausa (ms) 1000
  cambiar cont por 1
+
```

```

if (cont >= 160 && cont <= 165) {
  Señal_Analogica()
}
if (cont == 160) {
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 0)
  pins.digitalWritePin(DigitalPin.P3, 0)
  pins.digitalWritePin(DigitalPin.P4, 0)
  radio.sendNumber(cont)
  basic.pause(1000)
  cont += 1
}
if (cont == 161) {
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 0)
  pins.digitalWritePin(DigitalPin.P3, 0)
  pins.digitalWritePin(DigitalPin.P4, 1)
  radio.sendNumber(cont)
  basic.pause(1000)
  cont += 1
}
if (cont == 162) {
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 0)
  pins.digitalWritePin(DigitalPin.P3, 1)
  pins.digitalWritePin(DigitalPin.P4, 0)
  radio.sendNumber(cont)
  basic.pause(1000)
  cont += 1
}
}

if (cont == 163) {
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 0)
  pins.digitalWritePin(DigitalPin.P3, 1)
  pins.digitalWritePin(DigitalPin.P4, 1)
  radio.sendNumber(cont)
  basic.pause(1000)
  cont += 1
}
if (cont == 164) {
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 1)
  pins.digitalWritePin(DigitalPin.P3, 0)
  pins.digitalWritePin(DigitalPin.P4, 0)
  radio.sendNumber(cont)
  basic.pause(1000)
  cont += 1
}
if (cont == 165) {
  pins.digitalWritePin(DigitalPin.P1, 1)
  pins.digitalWritePin(DigitalPin.P2, 0)
  pins.digitalWritePin(DigitalPin.P3, 1)
  pins.digitalWritePin(DigitalPin.P4, 0)
  radio.sendNumber(cont)
  basic.pause(1000)
  cont += 1
}
}
}

```



```

function Señal_Analogica () {
  duty = 1024
  pins.analogWritePin(AnalogPin.P0, duty)
  pins.analogSetPeriod(AnalogPin.P0, 1000)
}

```

Función de avance de prueba genérica:



```

function Contador () {
  if (cont >= 0 && cont <= 3 || cont == 5) {
    basic.showNumber(cont)
    radio.sendNumber(cont)
    cont += 1
  }
  if (cont == 8) {
    basic.showNumber(cont)
    radio.sendNumber(cont)
    cont += 1}}
}

```

## 9.4 Modulo Bluetooth

La tarjeta Micro:bit viene equipada con un módulo Bluetooth integrado que permite la comunicación inalámbrica con otros dispositivos, como teléfonos móviles, tablets y otras tarjetas Micro:bit. Este módulo permite crear una gran gama de aplicaciones, desde control remoto hasta intercambio de datos y comunicación entre dispositivos.

En nuestro caso, para comprobar el correcto funcionamiento del módulo Bluetooth de la tarjeta, se realizará la conexión de esta con un dispositivo iPhone. Para ello, será necesario hacer uso de la aplicación gratuita de micro:bit disponible tanto en iOS como en Android.

Para realizar el emparejamiento de la tarjeta mediante bluetooth, habrá que seguir los siguientes pasos [17]:

1. Se abre la aplicación de micro:bit y se pincha el botón “Manage Conections”. Despues se elige añadir un nuevo dispositivo.
2. Se pulsa la combinacion de botones indicada en la micro:bit.

En la imagen 31 se muestra la secuencia:

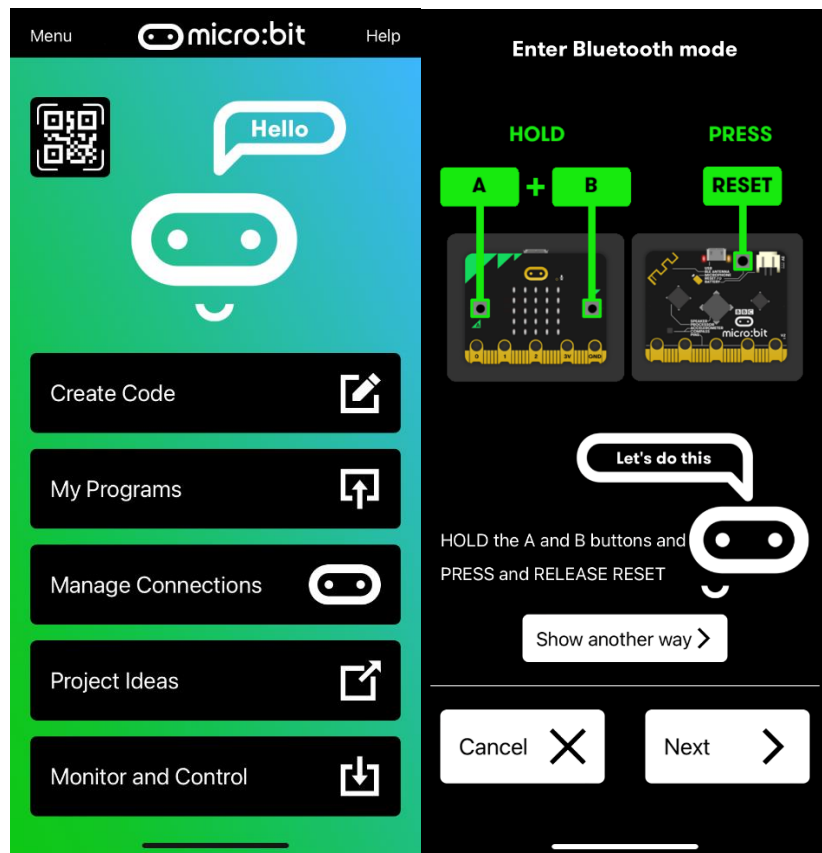


Ilustración 31. Menú general APP móvil

3. Se introduce la señal definida por la tarjeta en la aplicación. En la imagen 32 se representa:

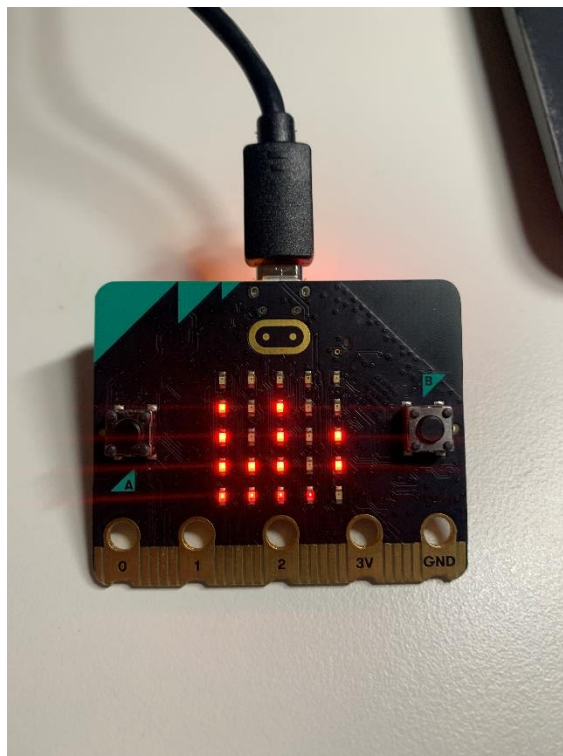
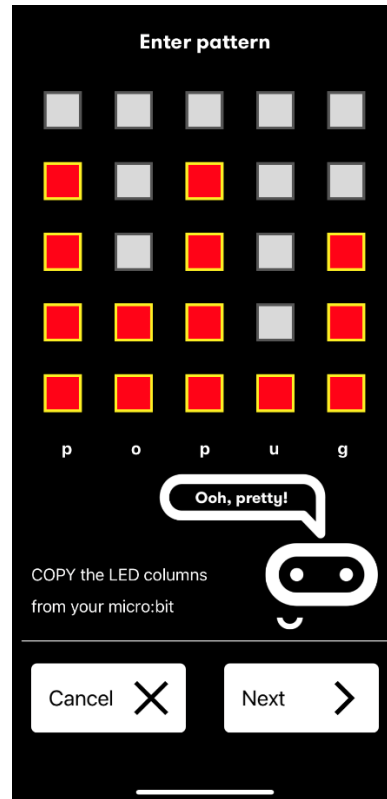


Ilustración 32. Código de conexión Bluetooth.

4. Se siguen las ultimas indicaciones y se enlaza el dispositivo. En la ilustración 33 se muestra la secuencia.

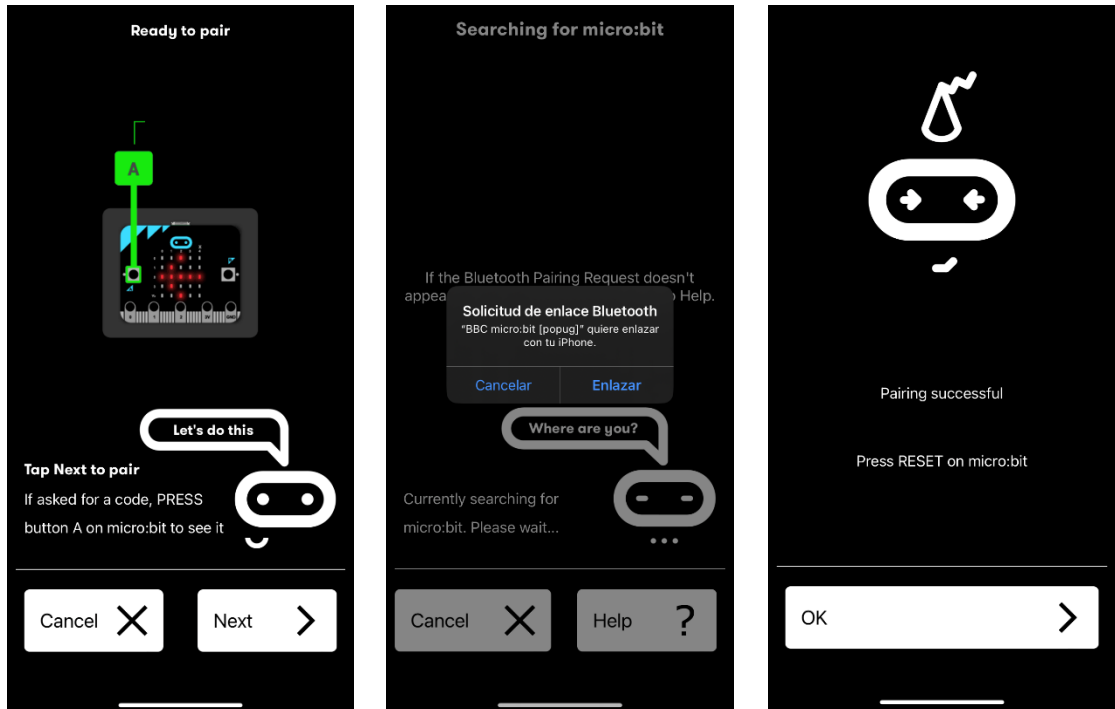
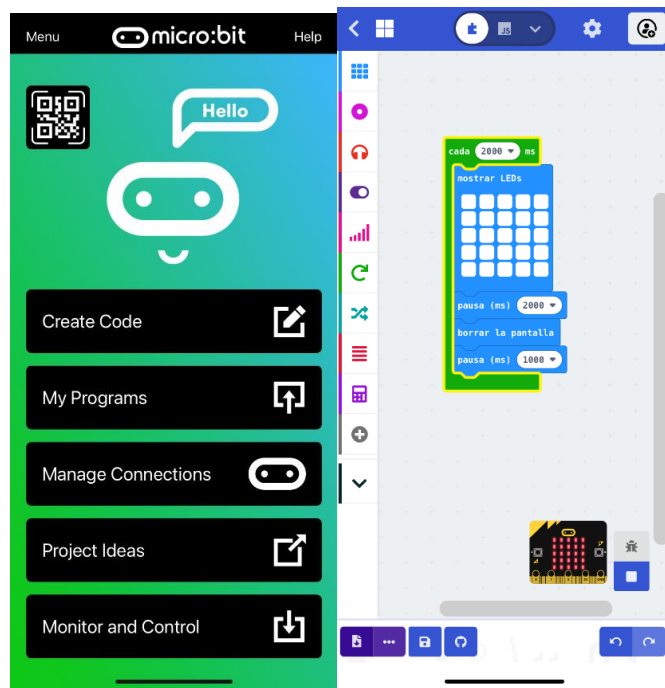


Ilustración 33. Secuencia de enlace dispositivo.

Una vez conectada la tarjeta al dispositivo móvil, se prueba la conexión ejecutando un programa de los ya planteados anteriormente.

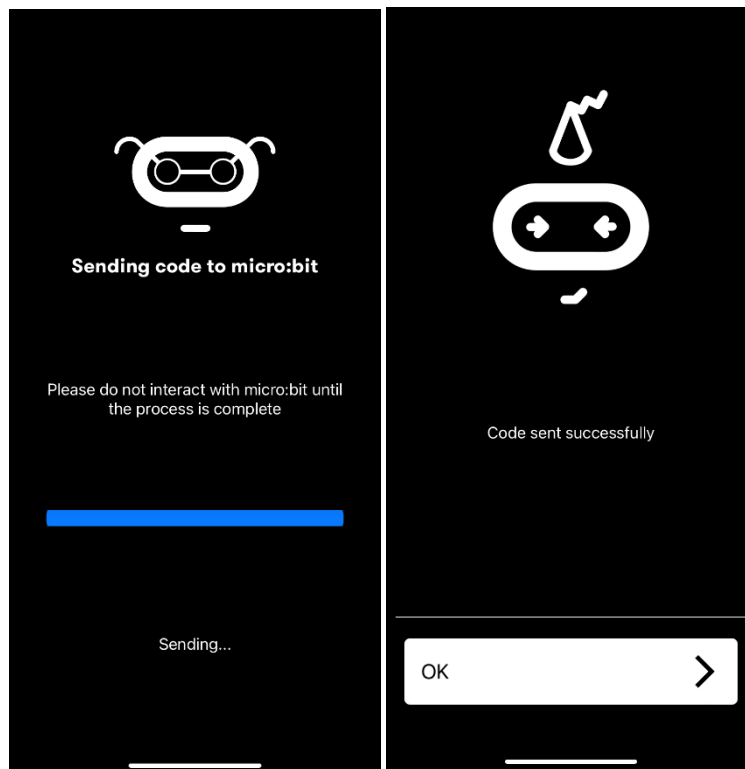
Los pasos para comprobar el correcto envío del programa son los siguientes:

1. Se pulsa sobre el módulo “My programs” tal y como se muestra en la imagen 34. Se elige el programa ya precargado dentro de la aplicación o se accede a la plataforma MakeCode para crear un código.



*Ilustración 34. MakeCode en APP*

2. Se envía el programa a través de la aplicación a la tarjeta para ser ejecutada de la misma forma que si se realizase desde un equipo PC. La secuencia se muestra en la imagen 35.



*Ilustración 35. Transmisión de un programa de la APP a la tarjeta*

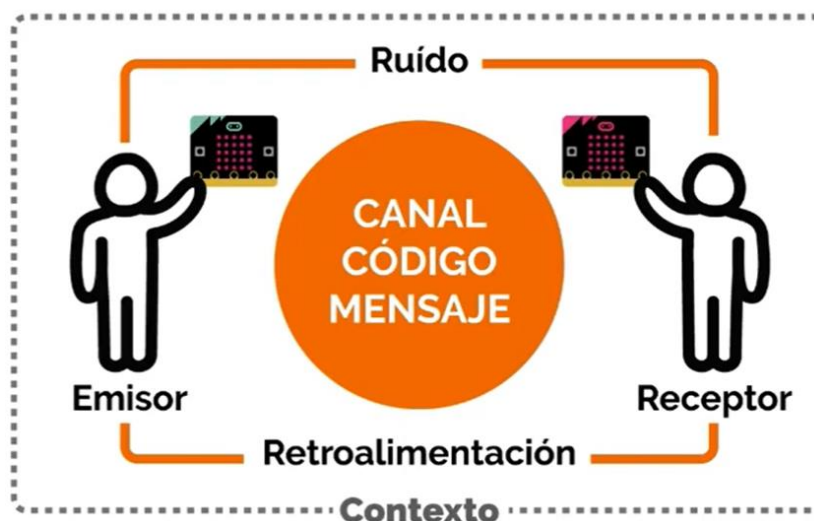


## 9.5 Comunicación radio

En este apartado de pruebas funcionales, se va a estudiar la comunicación inalámbrica entre las tarjetas a partir del módulo de radio incorporado, un punto indispensable para la prueba realizadas sobre la tarjeta hija. De esta forma, se comprobará la operatividad del módulo de la tarjeta expuesta a la prueba de calidad.

Es importante conocer el esquema de comunicación que tendrán ambas placas. Existirá un emisor (placa emisora) que transmitirá un mensaje en un determinado código a través de un canal, en este caso el aire. Y en la otra parte, habrá un receptor que recibirá el mensaje, lo decodificará y tomará acciones al respecto. En el caso de que el receptor emita un mensaje de vuelta, existirá retroalimentación.

En la imagen 36 se muestra un esquema de conexiones:



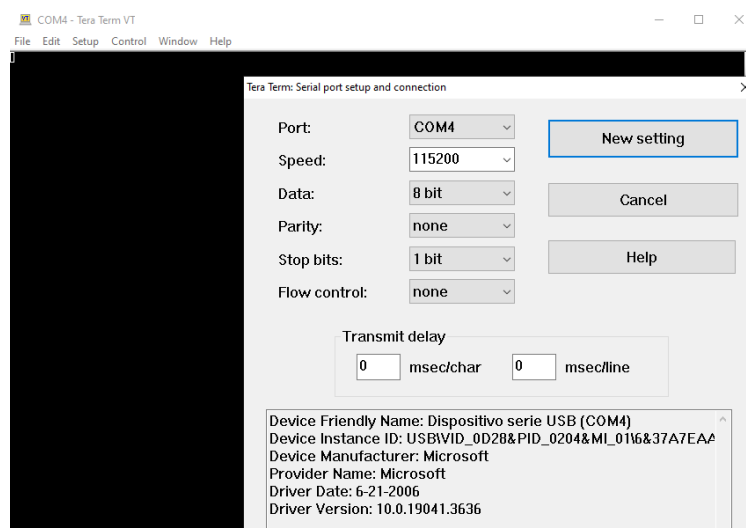
*Ilustración 36. Esquema módulo Bluetooth.*

Para conectar dos tarjetas se hará acopio de dos micro:bit, y se cargará un programa emisor en la primera y un programa receptor en la segunda. Las tarjetas deben estar relativamente cerca, generalmente a unos pocos metros, para que la comunicación por radio funcione de manera confiable. Además, se pueden utilizar múltiples tarjetas Micro:bit en el mismo entorno, pero cada grupo de tarjetas debe configurarse en un canal de radio único.

Los programas cargados en ambas tarjetas para las pruebas a realizar serán pautados por varios contadores que servirán de consigna para dar paso a la prueba a realizar. Cuando está consigna enviada por la tarjeta madre a la tarjeta hija sea recibida, se dará paso automáticamente a una prueba específica de funcionalidad cuyos resultados serán transmitidos vía USB de la tarjeta hija en forma de informe.

## 9.6 Manual de usuario sistema de prueba de tarjetas Micro:bit

1. En primer lugar, asegúrese de tener correctamente conectada a la placa de circuito impreso la tarjeta madre en su correspondiente sitio. Esta tarjeta madre debe ser completamente operativa, ya que servirá como referencia para poner a prueba al resto de tarjetas examinadas.
2. Alimente la placa madre introduciendo las pilas AAA de 1.5V cada una en el reverso de la placa. La alimentación alternativa vía USB es una opción alternativa.
3. Una vez seguidos estos puntos, conecte la tarjeta a examinar en la ranura destinada a esta misma, se identifica dentro de la placa con el rotulo “Tarjeta test”. El sistema hardware está listo para comenzar las pruebas.
4. Cargue el programa “Programa total hija” en la tarjeta a examinar y el “Programa total madre” en la placa referencia. Para ello deberá conectar las placas vías usb al PC para descargar el programa. La vía más simple es copiando el fichero .HEX al conectar la tarjeta madre al ordenador.
5. Una vez cargados los programas en ambas tarjetas, se debe conectar la tarjeta a prueba a un programa que permita mostrar los datos que serán transmitidos vía USB serie. Para este proyecto se ha trabajado sobre *Tera Term* con una velocidad de transmisión de 115200 baudios, 8 bits sin paridad y 1 stop-bit, tal y como se muestra en la imagen 37.



*Ilustración 37. Configuración del puerto serie*

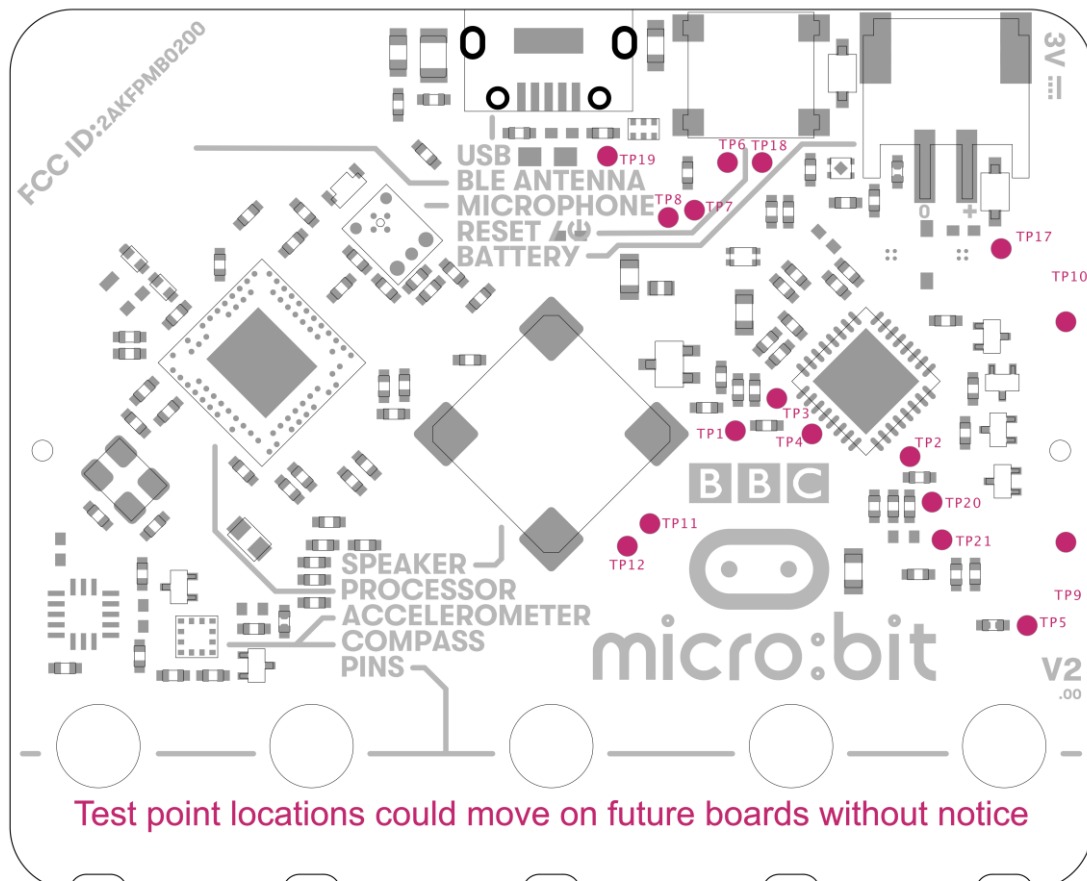
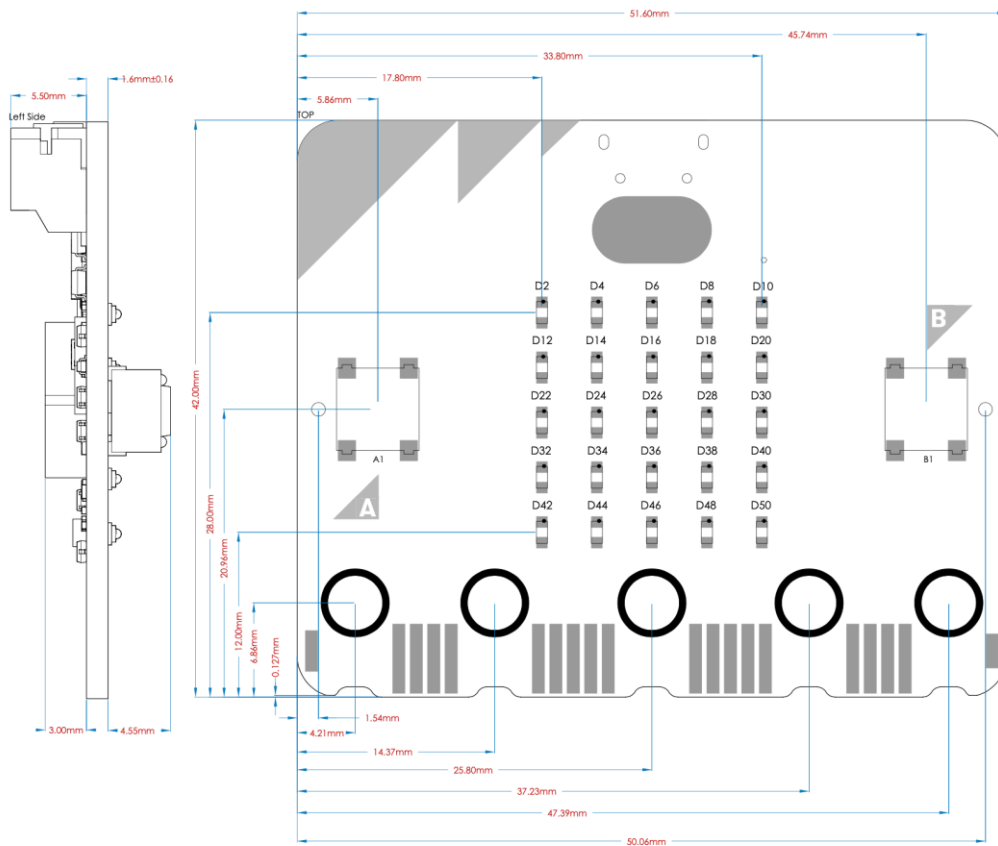
6. Se puede iniciar la secuencia de pruebas. Para comenzar, pulse sobre el botón “A” de la tarjeta madre. Se ha de tener en cuenta que hay periféricos que se incluyen en la versión V2 de la micro:bit pero no en las versiones V1, por lo que en el caso de no incluirse se

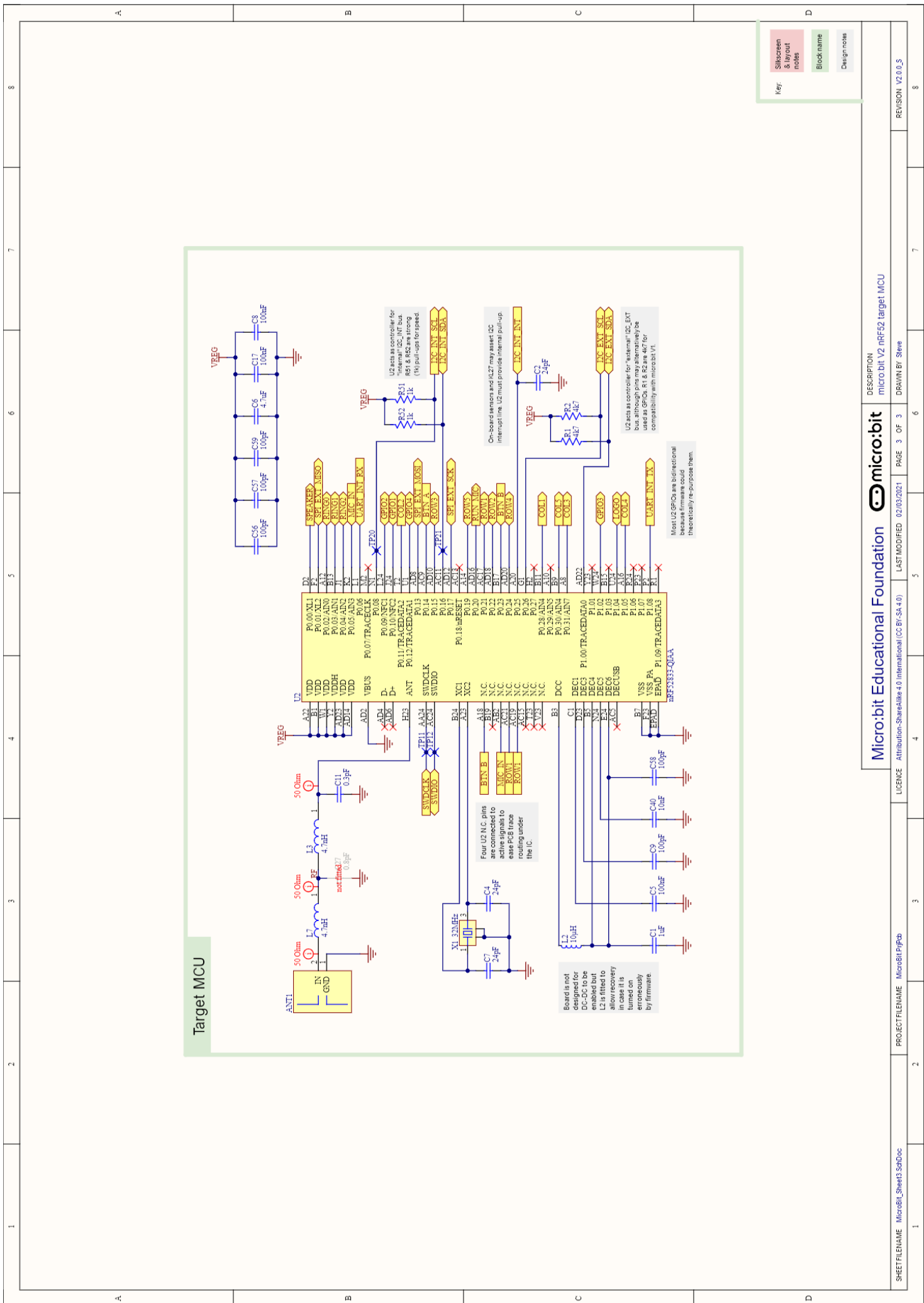
recibirá un mensaje informativo. En cualquier momento, pulsando sobre el botón B se podrá retroceder entre las pruebas.

- 6.0. Visualizará un 0 sobre la matriz de LEDs de la tarjeta madre y en la tarjeta hija se encenderá durante varios segundos la matriz completa. En caso de encenderse toda la matriz, esta estaría en correctas condiciones. Vía serie recibirá el mensaje: “Prueba de Leds: Visual”.
- 6.1. y 6.2. Al pulsar de nuevo sobre el botón “A” de la tarjeta madre (la secuencia para pasar a la siguiente comprobación será siempre la misma), se pasará a la siguiente prueba de botones. Una vez haya sido pulsado, sobre la tarjeta hija pulse el botón “A”. En caso de ser correctos, se mostrará un tick sobre la matriz de LEDs de la tarjeta hija y vía serie se comunicará el mensaje “Prueba de botón A: OK”. Al pulsar de nuevo sobre la tarjeta madre se iniciará la misma prueba sobre el botón “B”.
- 9.7 La siguiente prueba se realiza sobre el botón táctil de micro:bit. Al mostrarse el numero 3 en la tarjeta madre, ponga un dedo sobre el sensor táctil y se mostrará un tick sobre la matriz de LEDs de la tarjeta hija mientras vía serie se comunicará el mensaje “Prueba de logo: OK”.
- 9.8 Al volver a pulsar, se iniciará la prueba del micrófono. En este caso, para captar únicamente valores cuando el usuario lo permita y no de forma constante, si se pulsa sobre el botón “A” de la tarjeta puesta a prueba se registrarán valores vía serie con el mensaje “Valores nivel de sonido: X dB”  
Se captarán valores entre 100 Hz y 80 kHz representados en este caso en una escala de 0 a 255. Es aconsejable probar este punto con algún audio variable , así como una canción a la cual se le vaya modificando el volumen.
- 9.9 La siguiente prueba consiste en una prueba sobre el altavoz que incluye la Micro:bit, simplemente se reproducirá una melodía a 125bpm durante unos segundos para verificar su correcto funcionamiento. Vía serie se recibirá el mensaje “Prueba altavoz: Escuchar”.
- 9.10 A continuación, al pulsar sobre el botón de la tarjeta madre, se iniciará una validación del acelerómetro. Cuando se quiera iniciar la prueba se debe pulsar el botón “A” de la tarjeta hija. La prueba consistirá en completar la matriz de LEDs. En caso de ser completada se mostrará en la matriz una cara sonriente.
- 9.11 La siguiente prueba examinará la brújula que incluye la microbit. Cada vez que se pulse el botón “A” de la tarjeta hija se recibirá el mensaje “Prueba brújula. Direccion:X” siendo X la dirección entre a los que apunta la parte frontal de la microbit.
- 9.12 La penúltima validación pondrá a prueba el termómetro de la tarjeta, vía serie se recibirá “Prueba sensor temperatura : X ©” siendo este el valor en grados centígrados percibidos por el sensor.

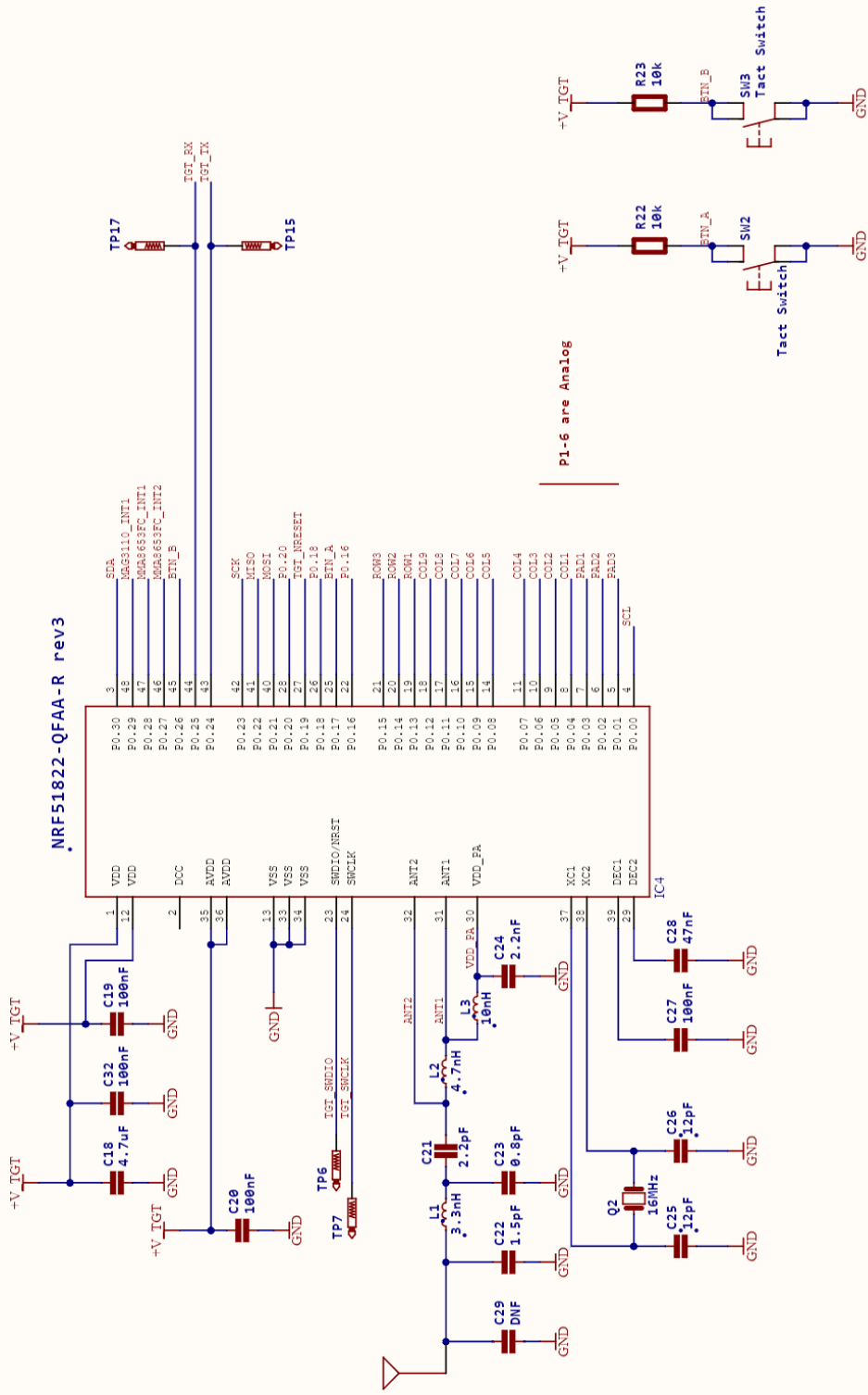
- 9.13 Esta última prueba pone a prueba todos y cada uno de los 16 pines I/O que incluye la tarjeta. Únicamente se pulsará el botón “A” de la tarjeta madre y vía serie se irá recibiendo el diagnóstico de cada uno de los pines en forma secuencial. Primeramente, se probará la escritura digital a nivel alto de cada uno de los pines y posteriormente la escritura a nivel bajo. Posteriormente comenzará el diagnóstico para la lectura de los pines analógicos. Para ello, mediante el uso del potenciómetro se establecerá el rango de tensiones que se esperan recibir, desde 0 a 3,3V
- 10 Una vez se finalice esta, el sistema de pruebas habrá finalizado su secuencia y se tendrán todos los resultados en el informe reflejado en el programa *Tera Term*.
- 11 Ahora es responsabilidad del usuario hacer un balance de los resultados y determinar si la tarjeta es aún operativa para ser funcional entre los requerimientos del departamento de electrónica. En caso de recibir puntos KO, se tiene que valorar la reparación de los periféricos dañados o la destrucción de la placa.

## 9.7. Tarjeta Micro:bit y versiones PCB





# Nordic nRF51822



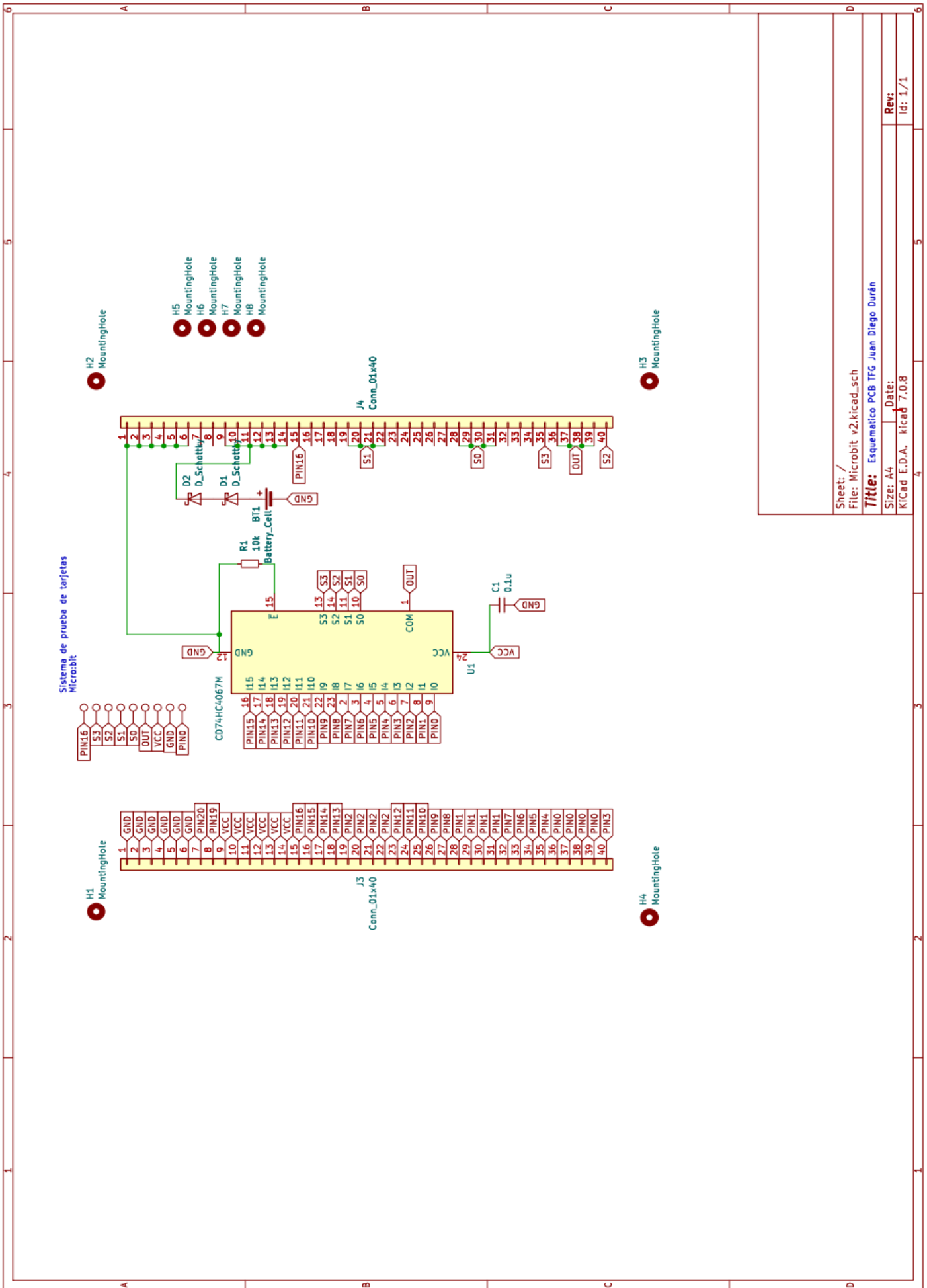
nRF51822-QFAA-R rev3

Changes from V1.04:  
 C19 = DNF  
 C23 = 0.8 pF

P1-6 are Analog

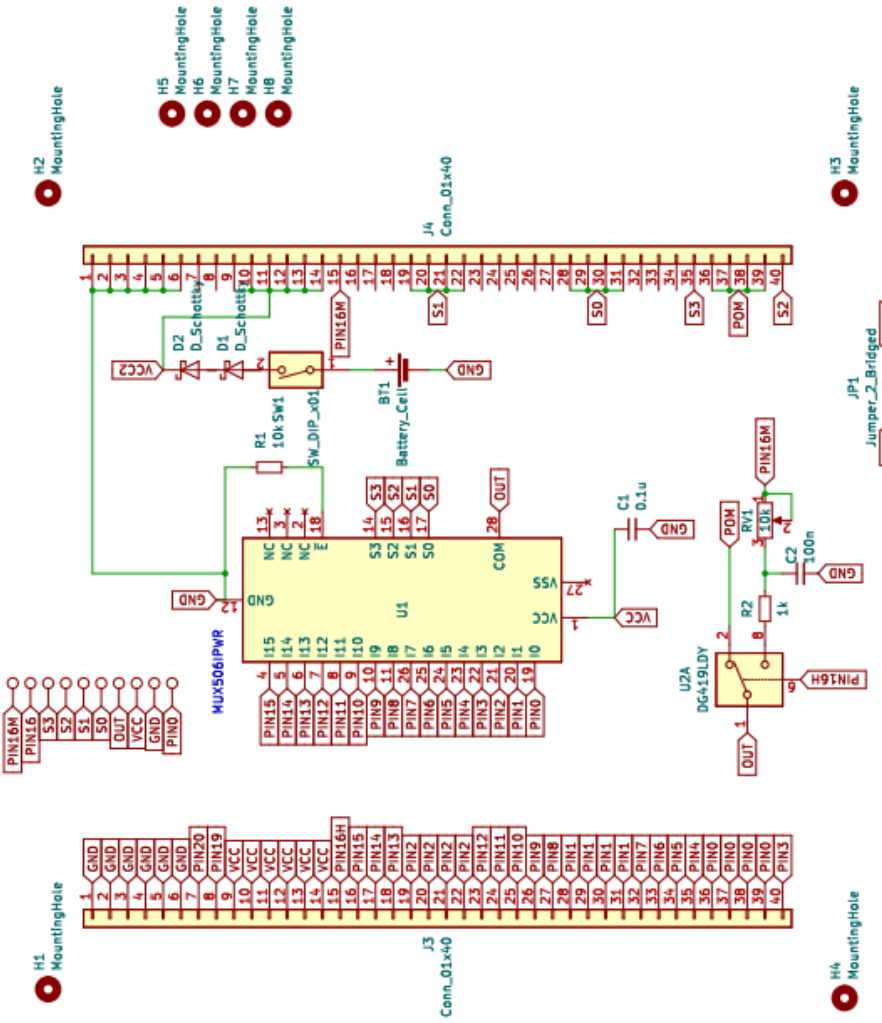
## BBC micro:bit

BBC micro:bit V1.3B  
 2016/12/9 14:26:31  
 SHEET 5 of 6



Sheet: /
File: Microbit v2.kicad_sch
<b>Title: Esquemático PCB TFG Juan Diego Durán</b>
Size: A4
Date:
KiCad E.D.A. kicad 7.0.8
Rev: 1/1
Id: 1/1





H2 MountingHole

H5 MountingHole  
H6 MountingHole  
H7 MountingHole  
H8 MountingHole

H3 MountingHole

PIN16M  
PIN16  
S3  
S2  
S1  
S0  
OUT  
VCC  
GND  
PIN0

MUX506IPWR

J3 Conn\_01x40

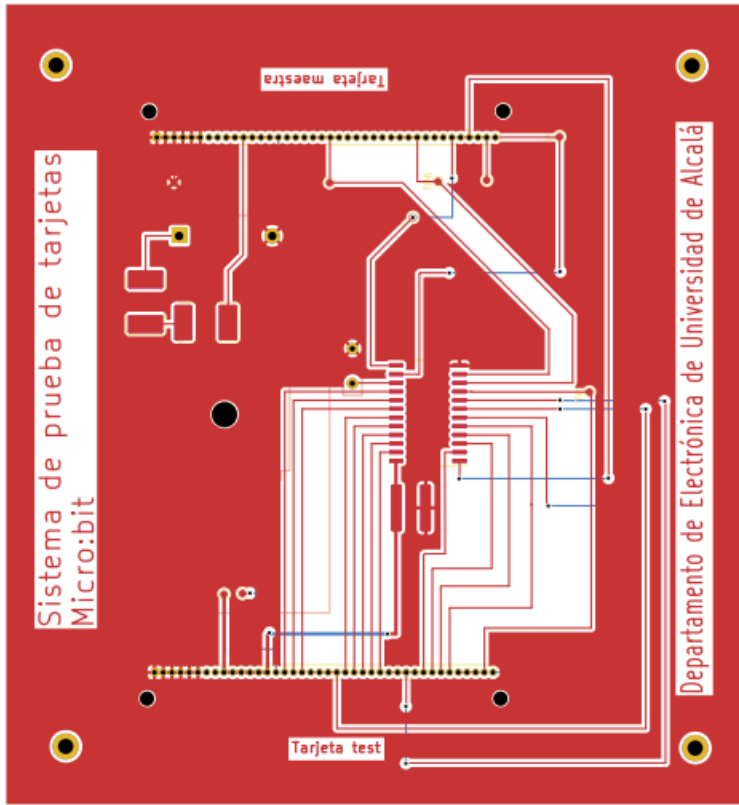
J4 Conn\_01x40

JP1 Jumper\_2\_Bridged

H4 MountingHole

Sistema de prueba de tarjetas Microbit

Sheet: /	File: Microbit v2.kicad_sch
Title: Esquemico PCB TFG Juan Diego Durán	
Size: A4	Date:
KiCad E.D.A. eschema 7.0.8	Rev: 1/1
Id: 1/1	



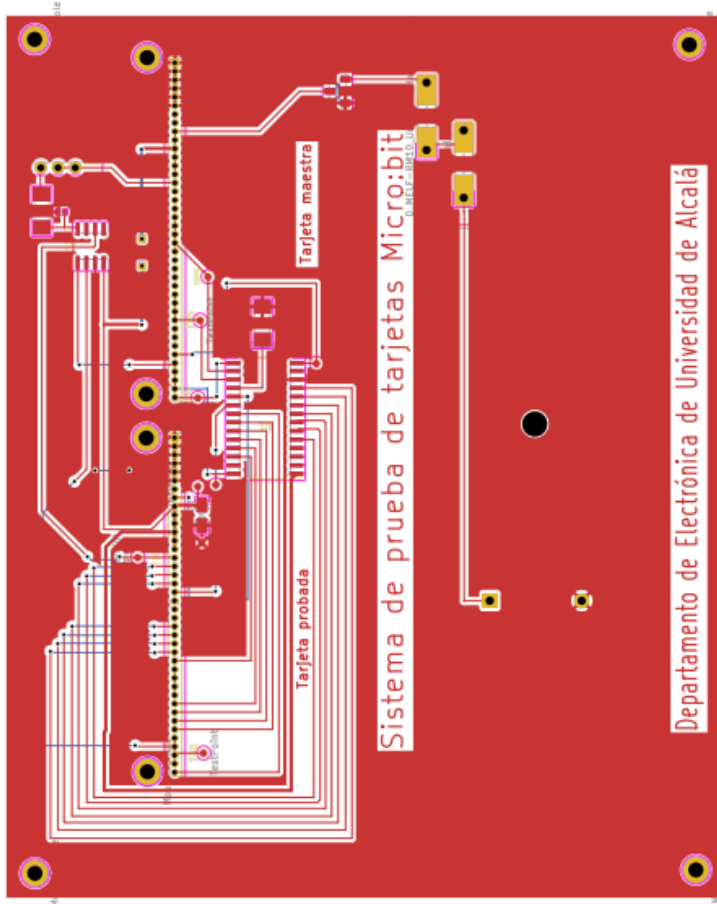
Sheet:  
File: Microbit v1.kicad\_pcb

Title: Placa TFG Juan Diego Durán

Size: A4 Date:

KiCad E.D.A. pcbnew 7.0.8

Rev:  
Id: 1/1



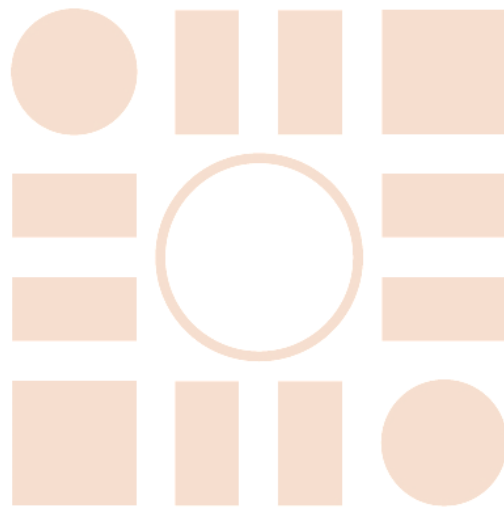
Sheet:  
File: Microbit v2.kicad\_pcb

Title: Placa TFG Juan Diego Durán

Size: A4 Date: Rev: 1/1

KiCad E.D.A. pcbnew 7.0.8 Id: 1/1

Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá