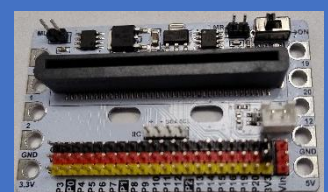
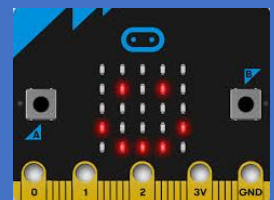




# PROYECTO: ONE ON ONE BASKETBALL



Licencia Creative Commons



Carlos Lorente Jiménez

## Micro:bit y Huskylens



## Contenido

MATERIALES NECESARIOS .....	2
CONEXIÓN HUSKYLENS .....	3
CONFIGURANDO HUSKY LENS .....	4
PROGRAMAMOS HUSKYLENS .....	5
CONEXIÓN PANTALLA 1602 I2C.....	6
PROGRAMAMOS PANTALLA 1602 I2C .....	8
SUMANDO PUNTOS .....	10
CONEXIÓN DISPLAY 7 SEGMENTOS.....	11
CRÉDITOS .....	12

***Agradecimientos*** a Alan Lorente Díaz que, sin lugar a dudas, hace mejores fotografías que quien escribe este manual.





## MATERIALES NECESARIOS

Placa programable Micro:bit v2

Placa de expansión Micro:bit

HuskyLens

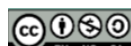
Display LCD

Display 7 segmentos

Pinzas de cocodrilo

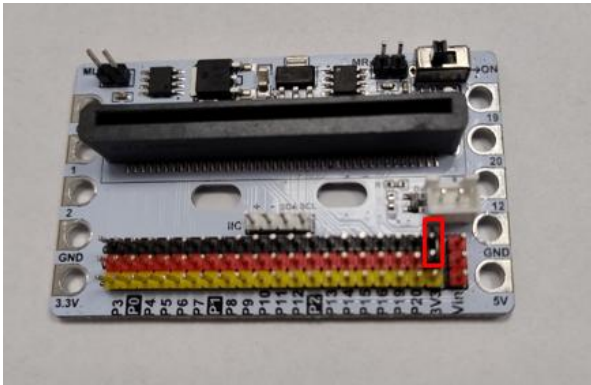
Papel de aluminio o papel de cobre adhesivo

Cartón





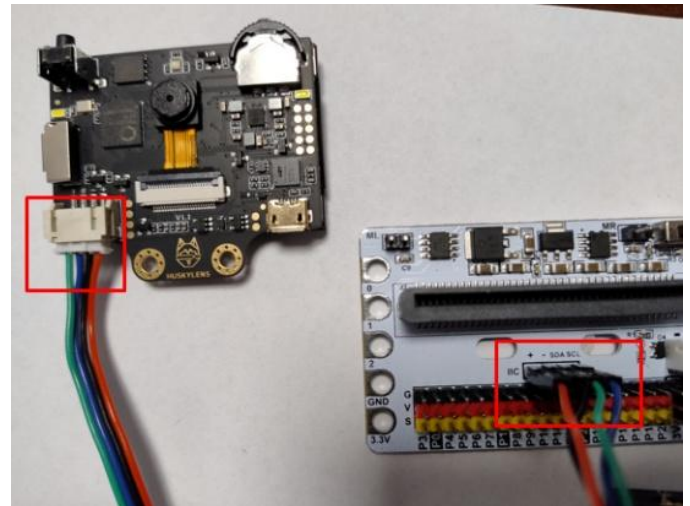
## CONEXIÓN HUSKYLENS



Utilizamos la placa de expansión de Micro:bit, poniendo el **jumper en 5V** para poder proveer a la Huskylens, el LCD y el display de 7 segmentos del voltaje óptimo de funcionamiento.

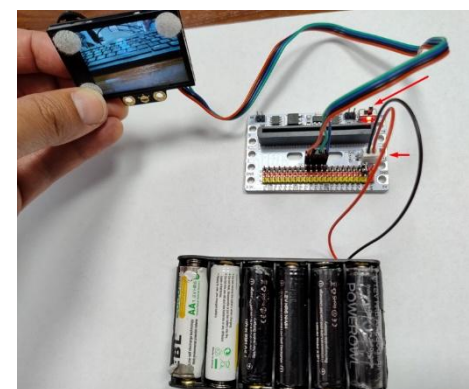
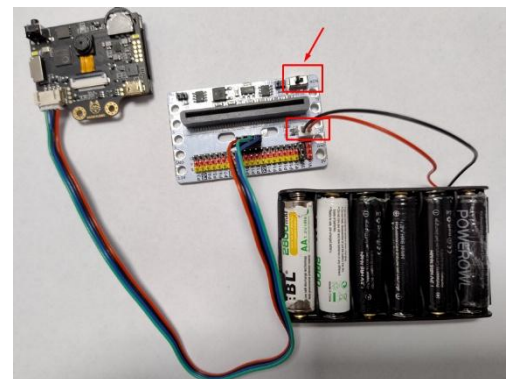
Para conectar la **Huskylens** tenemos en cuenta que tenemos un + (cable rojo) y un – (cable negro) que darán alimentación al dispositivo. Por otra parte utilizaremos el puerto de comunicación I2C de la placa que tiene su parte de línea de datos (**SDA – Serial Data**) y su línea de reloj (**SCL – Serial Clock**).

Conectamos el **cable verde** al **SDA** (datos) y el **cable azul** al **SCL** (reloj).



En la fotografía se muestra el portapilas conectado. Siempre que estemos conectando más dispositivos, es conveniente tener el interruptor en OFF o desconectar el portapilas. Lo pondremos en ON solo para comprobar que les llega alimentación.

Podemos cerrar el interruptor (ON) y comprobar que la HuskyLens se enciende. También comprobamos que se enciende una luz roja en la parte del interruptor.



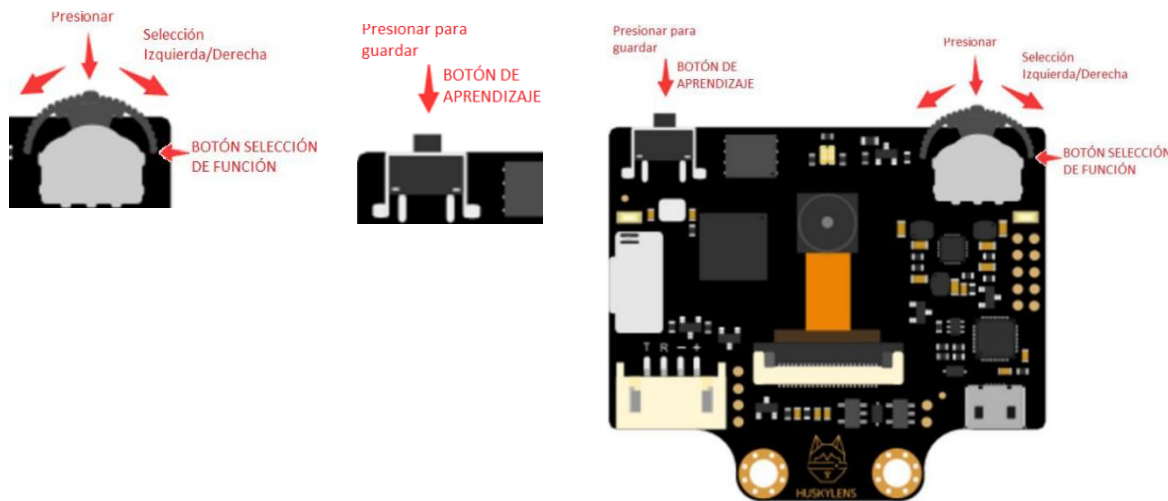




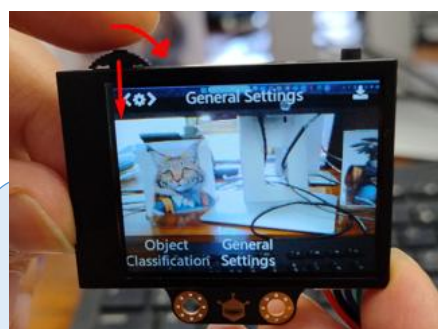
## CONFIGURANDO HUSKY LENS

Al cerrar el interruptor de la placa de expansión le llegará alimentación a la HuskyLens y lo primero que nos saldrá en pantalla será el logo del dispositivo.

Manejaremos los botones y ruleta de la HuskyLens que se muestran en las siguientes imágenes:



Imágenes extraídas de la página [DFRobot](https://dfrobot.com/)



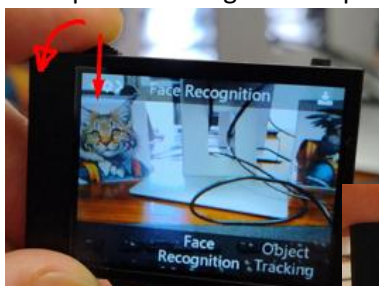
1.Podemos navegar por el menú girando la ruleta que se muestra en pantalla.

Al llegar a “General Settings” **pulsamos** el botón de función.

2.Dentro de la sección “General Settings” seleccionamos “Protocol Type” e “I2C” con el mismo proceso (girar ruleta y pulsar botón de función).

Acto seguido, giramos la ruleta hasta llegar a “Save & Return”, pulsamos el botón de función y seleccionamos “Yes”.

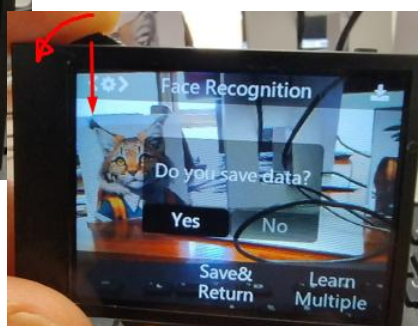
3.Después nos dirigimos al apartado “Face Recognition”.



Para acceder a esta sección es importante que mantengamos el botón de función pulsado hasta que nos aparezca el menú correspondiente.



Nos dirigimos a “Learn Multiple” y lo activamos pulsando el botón de función y girando la ruleta. El icono debe quedar de la siguiente manera:

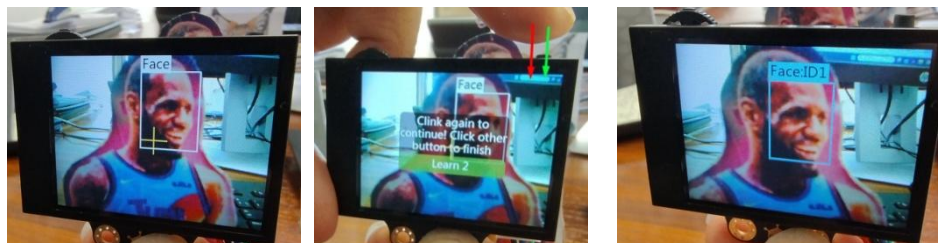


Terminamos dirigiéndonos a “Save & Return” y guardando los cambios seleccionando “Yes”.





4. Ahora tenemos la posibilidad de grabar los rostros que queremos que reconozca nuestra cámara. Al pasar la cámara por una cara nos saldrá enmarcado "Face", pulsamos el **botón de aprendizaje** una primera vez y nos saldrá el mensaje "Click again to continue! Click other button to finish" Pulsamos una segunda vez el botón de aprendizaje, quedará grabado este rostro y aparecerá **Face: ID1**.



Con la siguiente cara procedemos de la misma manera. Si queremos terminar el proceso, pulsamos el botón de función. Como podemos ver en la imagen, la cámara ha dejado registrado **Face:ID2**.



## PROGRAMAMOS HUSKYLENS

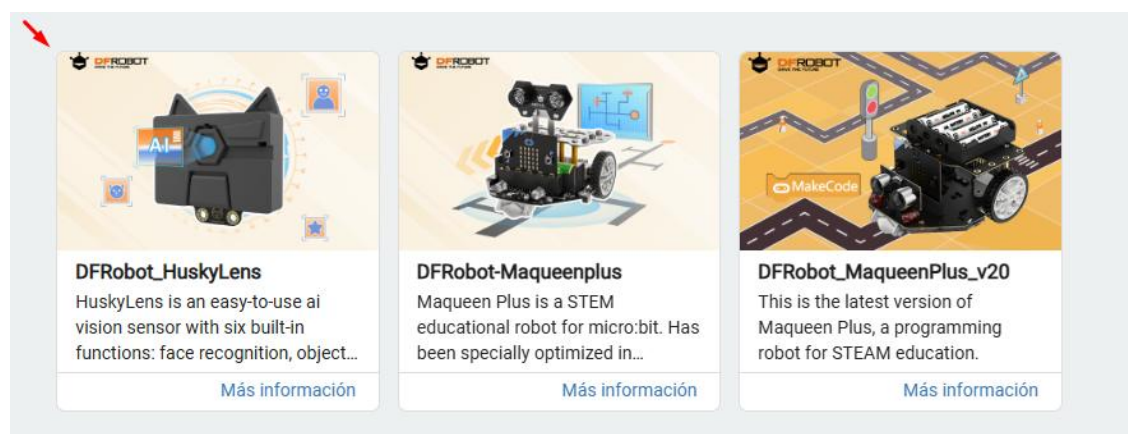
Utilizamos el entorno de programación de Makecode: [makecode.microbit.org](https://makecode.microbit.org)

Vamos a plantear un ejercicio en el que al reconocer al jugador 1 por cámara (Face: ID1) mostraremos por la pantalla de la Micro:bit el número 1 y reproduciremos un sonido.

Para poder programar la HuskyLens en Makecode necesitamos añadir su extensión.

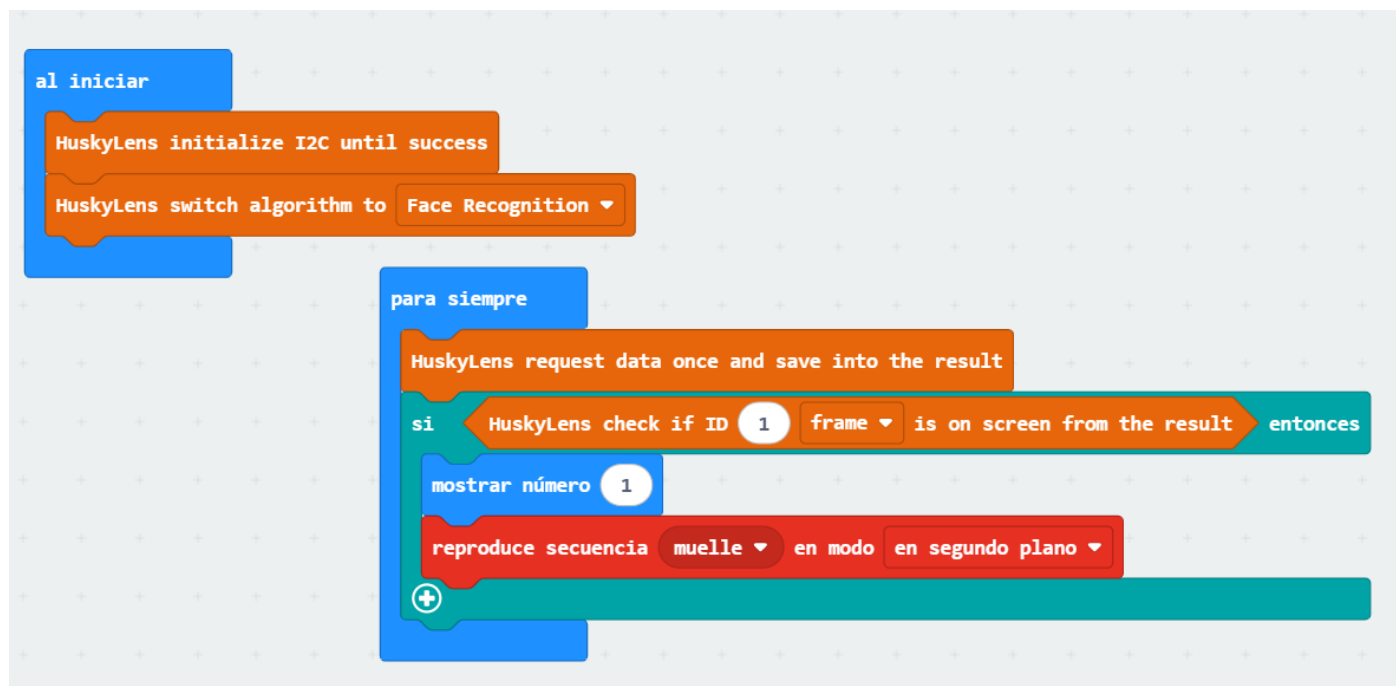


De los tres resultados que obtenemos seleccionamos DFRobot\_HuskyLens.





Y la programación que tenemos que realizar quedaría de la siguiente manera:



Vemos que cuando iniciamos el programa se inicializa el protocolo de comunicación I2C con la placa de expansión e indicamos que vamos a utilizar el Reconocimiento de Rostros (Face Recognition).

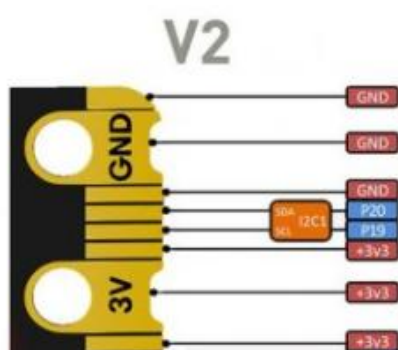
En el bucle recogemos el resultado de la cámara y, si el ID = 1, significa que la cámara está reconociendo al jugador 1. Se mostrará el número 1 en la pantalla de la Micro:bit y se reproducirá el sonido “muelle”.

Carga el programa en tu Microbit y conéctala a tu placa de expansión para comprobar que funciona.

**¿Eres capaz de ampliar el código para que reconozca al jugador 2, muestre en la pantalla de la Micro:bit el número 2 y reproduzca otro sonido?**

## CONEXIÓN PANTALLA 1602 I2C

Para conectar el **módulo de pantalla LCD 1602 I2C** (en adelante LCD 1602) tendremos en consideración que utiliza también el protocolo de comunicación I2C. Como ya tenemos ocupado el conexionado I2C de la placa con la Huskylens, utilizamos los pines 19 y 20 de la Micro:bit que funcionan como SCL (P19) y SDA (P20) del protocolo de comunicación I2C.

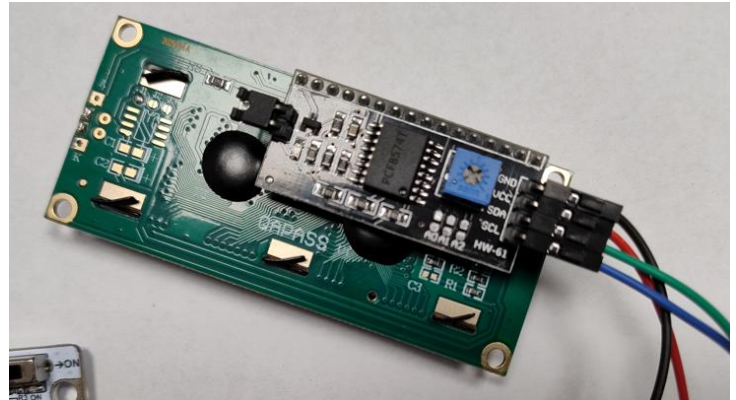




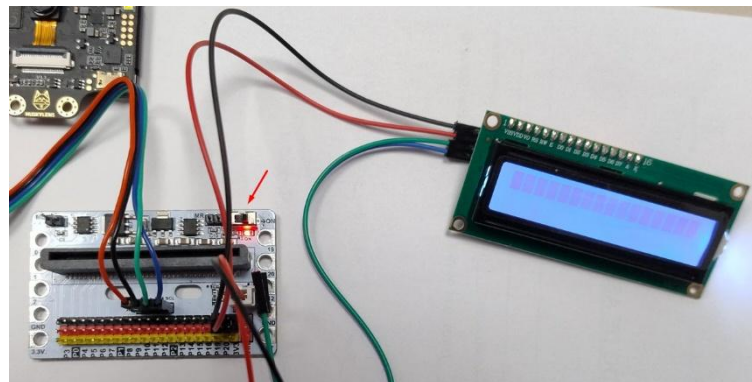
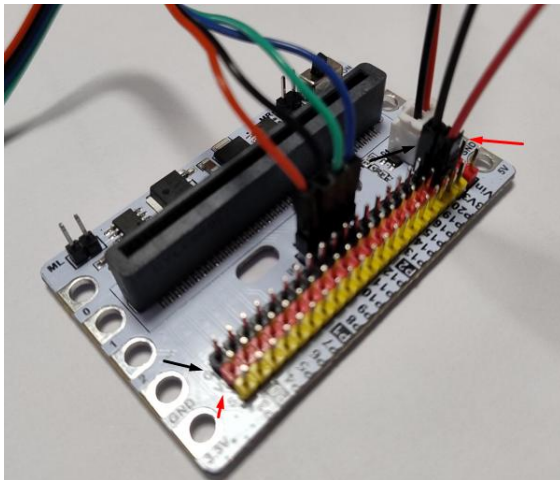


Para facilitar el conexionado y que sea similar al de la HuskyLens pondremos en el LCD 1602 un cable hembra-hembra **verde** al **SDA** (datos) y uno **azul** al **SCL** (reloj). Para la alimentación, cable hembra-hembra **rojo** al VCC y cable **negro** al GND.

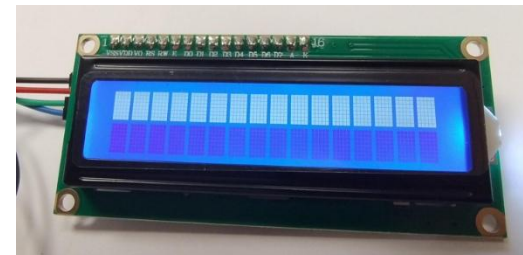
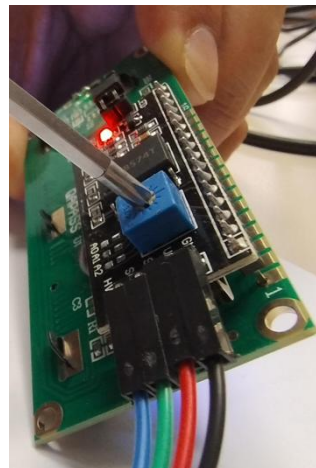
Acto seguido, conectamos el cable **rojo** a cualquier pin de la fila V (**fila roja**) de nuestra placa de expansión y el cable **negro** a la fila G (**fila negra**).



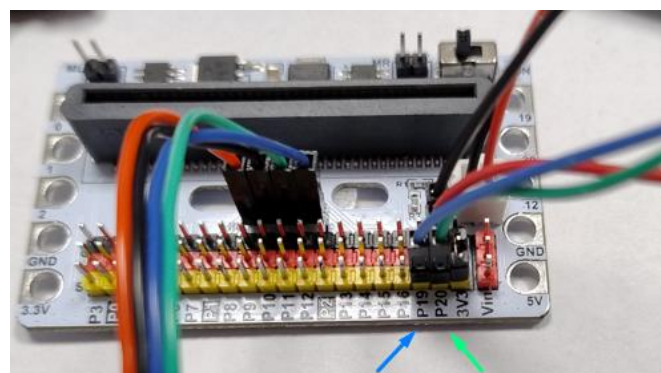
En este momento, si conectamos la placa de expansión, debería llegarle alimentación al LCD y encenderse la pantalla.



Podemos configurar el contraste de la pantalla manipulando con un destornillador el potenciómetro que encontramos en su parte trasera.



Una vez que hemos comprobado que le llega alimentación ponemos el interruptor de la placa de expansión en OFF por seguridad. Posteriormente conectamos el cable **verde** al **P20** y el cable **azul** al **P19**.



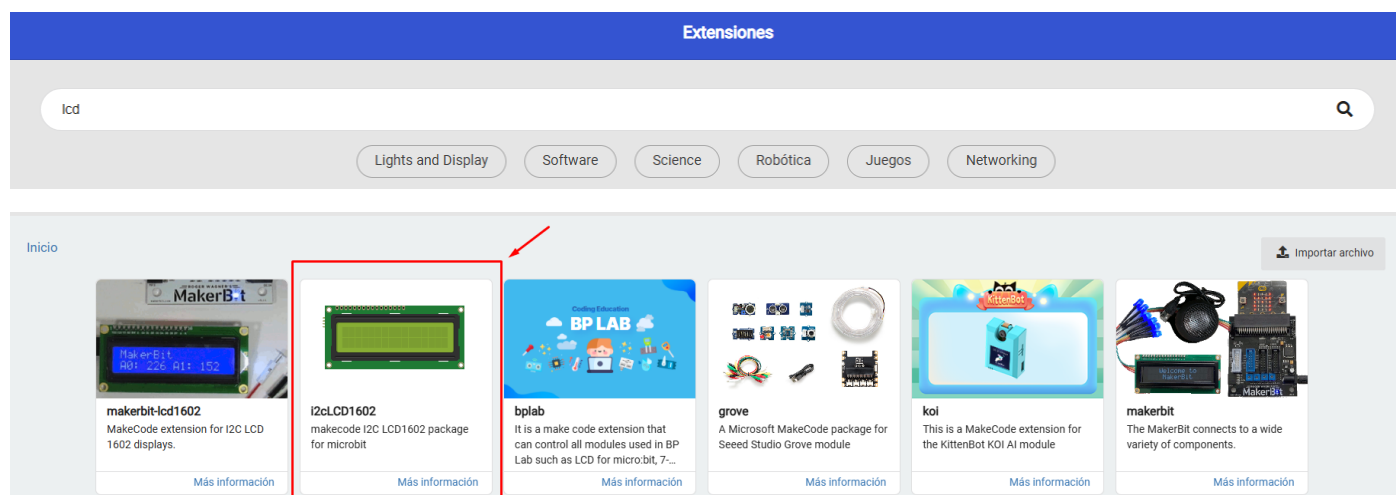




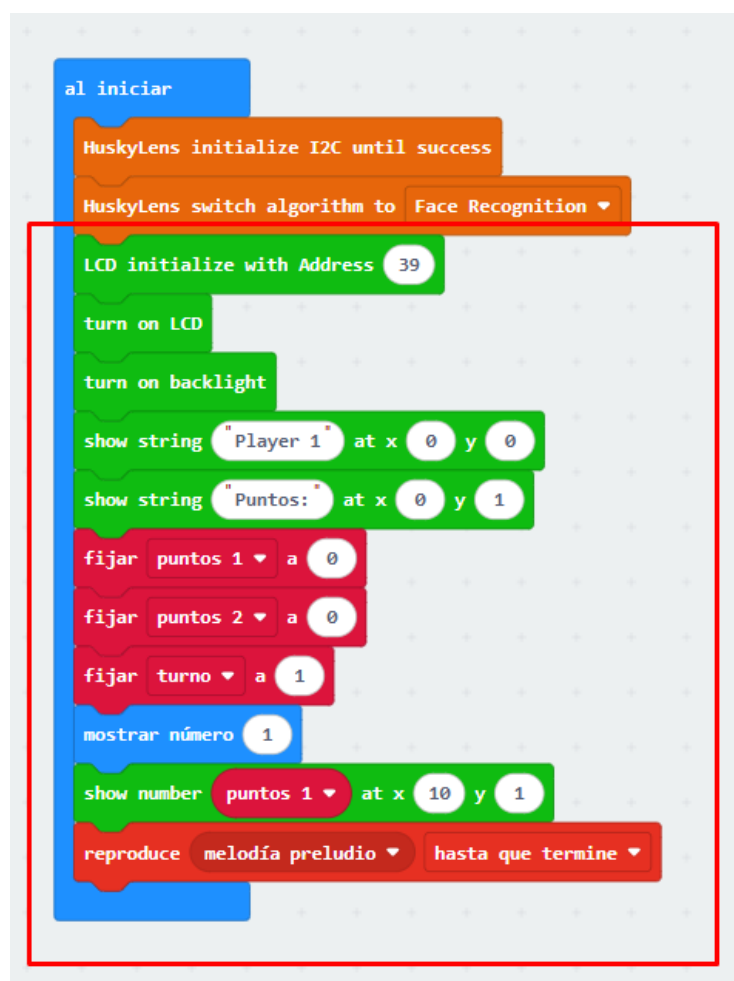
## PROGRAMAMOS PANTALLA 1602 I2C

Incorporamos más código a nuestro proyecto. Añadimos en este caso la puesta en marcha del LCD 1602.

Empezamos añadiendo la extensión del componente LCD.



E incorporamos el código resaltado en rojo al que ya tenemos de la HuskyLens:



- **Inicializamos** el LCD con dirección 39. De esta manera, estamos indicando que vamos a utilizar el protocolo I2C de comunicación.
- **Encendemos** la pantalla.
- Le damos un **contraste**.
- Mostramos la cadena de texto **Player 1** en la posición (x,y) = (0, 0) que corresponde a la primera fila (y=0) y el inicio de la fila (x=0)
- Mostramos la cadena de texto **Puntos** en la posición (x,y) = (0, 1). Segunda fila (y=1) e inicio de dicha fila (x = 0).
- **Inicializamos las variables** “puntos 1”, “puntos 2” y “turno” (debes crearlas en su apartado correspondiente de Makecode). Estas variables nos servirán para mostrar información de puntos y cambiar el turno entre jugadores.
- **Mostramos el número 1** en la pantalla Micro:bit ya que es el jugador 1 quien comienza la partida.
- **Mostramos los puntos** que tiene el jugador 1 en la posición del LCD (x,y) = (10, 1).
- Reproducimos **una melodía de inicio de partida**.





Incorporamos el siguiente código al bloque “para siempre” que ya teníamos:

```
para siempre
  HuskyLens request data once and save into the result
  si HuskyLens check if ID 1 frame is on screen from the result entonces
    mostrar número 1
    reproduce secuencia muelle en modo en segundo plano
    fijar turno a 1
    repetir 16 veces
      ejecutar Shift Right
      pausa (ms) 100
    clear LCD
    show string "Player 1" at x 0 y 0
    show string "Puntos:" at x 0 y 1
    show number puntos 1 at x 10 y 1
```

- Ponemos el valor a la variable **turno = 1** para controlar el turno del jugador 1.
- **Movemos 16 posiciones a la derecha** el texto de la pantalla LCD.
- **Borramos el texto** de la pantalla.
- Mostramos los correspondientes textos del jugador que está jugando y los puntos que lleva acumulados.



Recuerda que en el anterior apartado se te pidió añadir código para que la HuskyLens cambiará al jugador 2. Por lo tanto, ahora debes intentar incorporar el código necesario para que se cambie la variable turno al valor 2, se mueva el texto en pantalla y se muestre el texto Player 2 y su correspondiente puntuación (en el siguiente apartado veremos cómo sumar puntos).

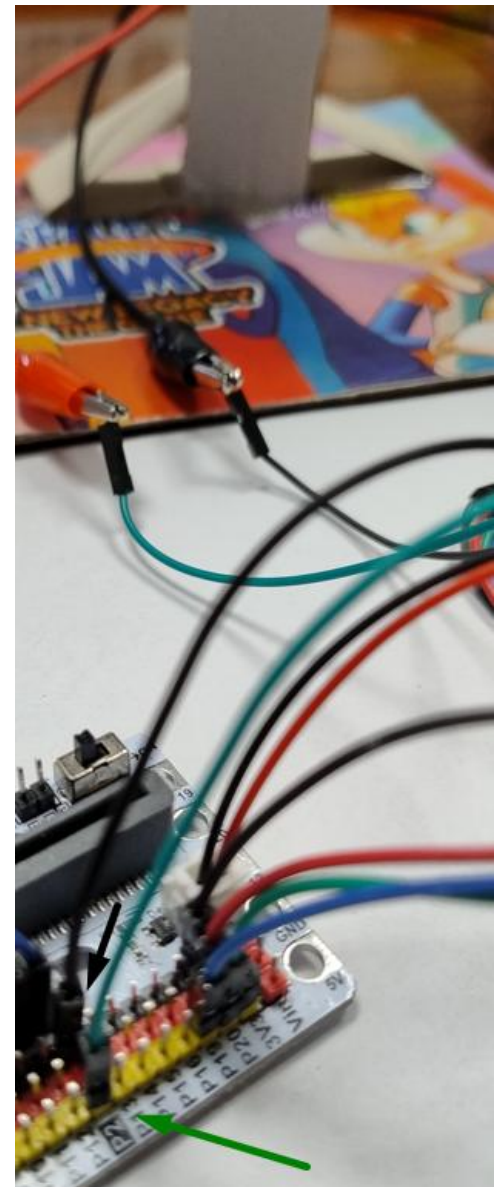
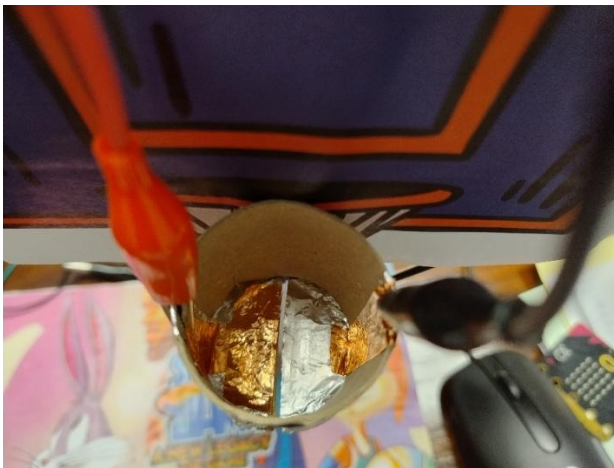




## SUMANDO PUNTOS

El objetivo es que cada vez que encestemos sumemos un punto a ese jugador. Podemos hacer la canasta, por ejemplo, reciclando el tubo de cartón de papel higiénico. Lo cortamos por la mitad y utilizamos el sobrante para poner una base en nuestra canasta. Recubrimos la base con papel de aluminio o con cobre adhesivo. Lo importante es dejar una pequeña abertura.

Con pinzas de cocodrilo conectamos cada parte del circuito abierto en un extremo y en el otro a cable macho-hembra. La zona hembra la conexiaremos al pin 2 (P2) y a un pin negativo (fila negra).



En la programación añadimos el bloque que aparece en la imagen a lo que ya tenemos:

- Al cerrarse el circuito con una pelota de aluminio, el pin2 (P2) quedará presionado y sumaremos un punto al jugador 1 si estamos en su turno.

¿Eres capaz de añadir un bloque similar para que sume puntos en el turno del jugador 2?







## CONEXIÓN DISPLAY 7 SEGMENTOS

Seguiremos ampliando el manual con más dispositivos.





## CRÉDITOS

Manual elaborado por Carlos Lorente Jiménez.

Algunas imágenes extraídas de la página [DFRobot](https://www.dfrobot.com/).

Idea inicial del proyecto basado en la actividad del canal STEAM Thinking:

<https://www.youtube.com/watch?v=OBMDUiSbwEY>

Personajes del setup basados en la película de WarnerBros Space Jam.

