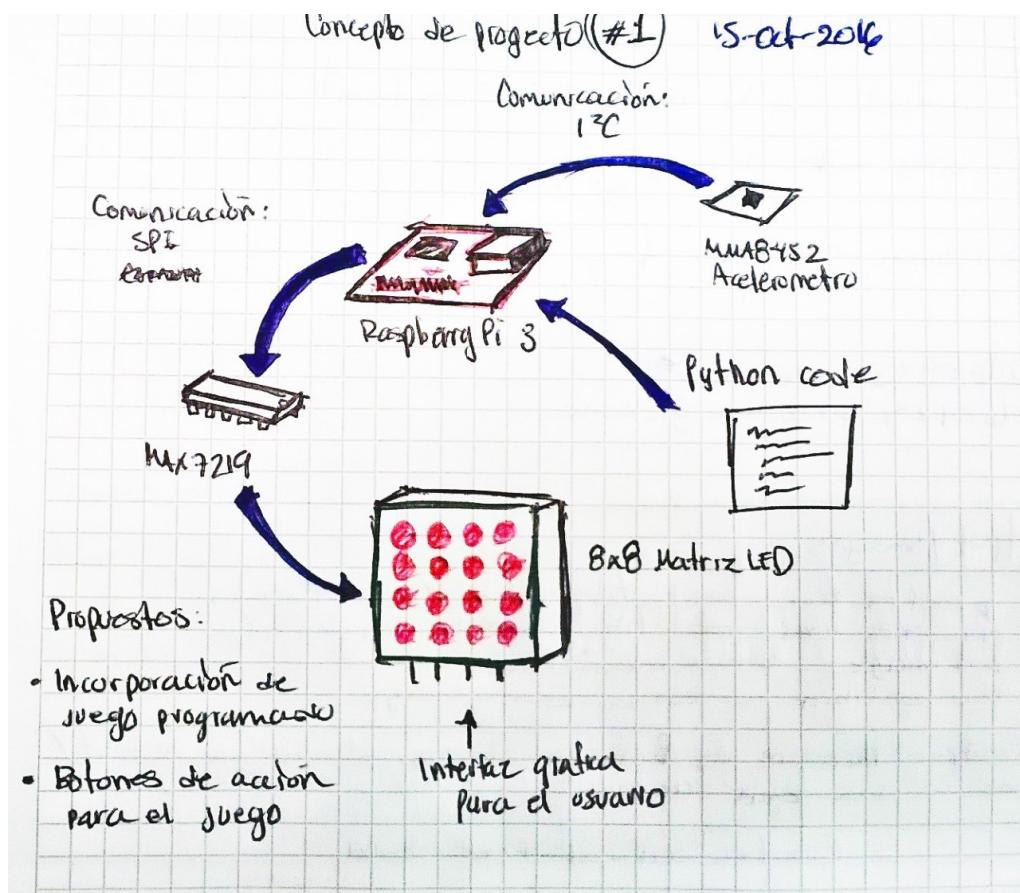


# Proyecto Apollo

El proyecto Apollo de plantea de principio, un pequeño juego de “naves” programado en Python incorporado en una Raspberry Pi 3 (Rpi3) con una distribución de Linux lo mas adaptada posible al juego como un embebido.

Planteamos la interacción con el usuario, utilizando los GPIO's de la Rpi3 donde nuestra interfaz gráfica es una matriz led de 8x8 Led's.

El mando de control de la “nave”, por medio de movimiento físico de la placa, detectando los movimientos por medio de un acelerómetro.



Buscamos agregar mas interacción o aplicación útil con el usuario por medio de botones de acción o con la incorporación de mas sensores y código del juego. Que se discutirán en el equipo y de-acuerdo al tiempo restante para la entrega del proyecto.

Para la facilidad de avance de proyecto, incorporamos elementos de hardware que ya teníamos:

- Matriz 8x8 leds
- Max7219

- Acelerometro MMA8452

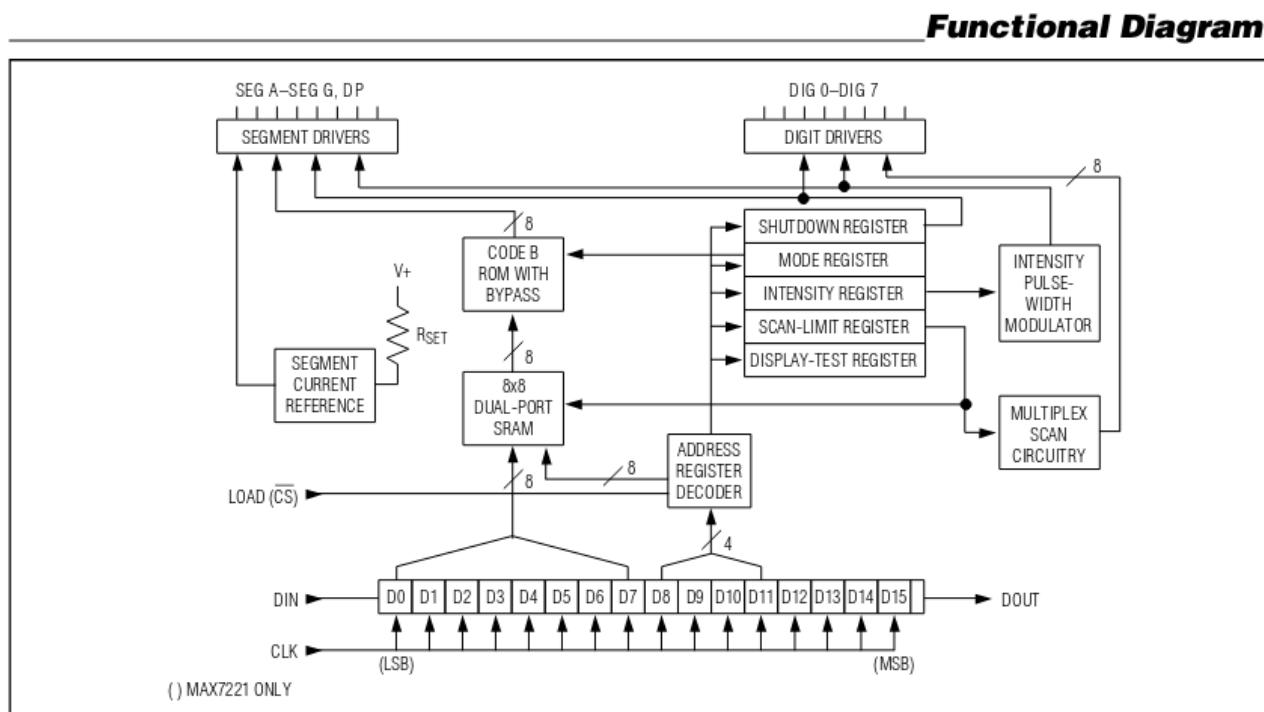
Buscando incorporar mas tarde:

- Botones de acción
- Audio
- ....

Comenzamos por analizar como funciona el driver de la matriz led MAX72219 para poder configurarla y echarla a andar con nuestra placa Rpi3. (2)

## MAX7219

El circuito integrado Max7219 es un driver compacto, de entrada y salida serial (input/output) para displays numéricos de 7 segmentos con hasta 8 dígitos o 64 leds individuales. Es compatible con los protocolos de comunicación SPI, QSPI, y MICROWIRE.



El utilizar el circuito integrado Max7219 nos facilita una configuración sencilla de control de una matriz de 8x8 leds, dando parámetros a los registros de la memoria interna. Y una buena comunicación por medio del protocolo SPI.

## Digit and Control Registers

**Table 1. Serial-Data Format (16 Bits)**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	X	X					ADDRESS	MSB				DATA		LSB

El serial data input (DIN), envia en paquetes de 16-bit al registro interno de desplazamiento con cada flanco ascendente del reloj (CLK).

### Shutdown Mode.-

**Table 3. Shutdown Register Format (Address (Hex) = 0xC)**

MODE	ADDRESS CODE (HEX)	REGISTER DATA							
		D7	D6	D5	D4	D3	D2	D1	D0
Shutdown Mode	0XC	X	X	X	X	X	X	X	0
Normal Operation	0XC	X	X	X	X	X	X	X	1

Configuramos nuestro registro para operar de manera normal (0x0C----- → 0x01).

### Decode-Mode Register.-

“The decode-mode register sets BCD code B (0-9, E, H, L, P, and -) or no-decode operation for each digit. Each bit in the register corresponds to one digit.”

**Table 4. Decode-Mode Register Examples (Address (Hex) = 0X9)**

DECODE MODE	REGISTER DATA								HEX CODE
	D7	D6	D5	D4	D3	D2	D1	D0	
No decode for digits 7-0	0	0	0	0	0	0	0	0	0x00
Code B decode for digit 0 No decode for digits 7-1	0	0	0	0	0	0	0	1	0x01
Code B decode for digits 3-0 No decode for digits 7-4	0	0	0	0	1	1	1	1	0x0F
Code B decode for digits 7-0	1	1	1	1	1	1	1	1	0xFF

Configuramos nuestro registro para no decodificar (0x9----- → 0x00).

### Display-Test Register.-

“Display-test mode turns all LEDs on by overriding, but not altering, all controls and digit registers (including the shutdown register).”

Configuramos nuestro registro para operar de manera normal (0xF----- → 0x00).

### Scan-Limit Register.-

“The scan-limit register sets how many digits are displayed, from 1 to 8.“

Configuramos nuestro registro para operar con el maximo (0xB----- → 0x07).

## Intensity Control.-

“Display brightness can be controlled digitally by using the intensity register.”

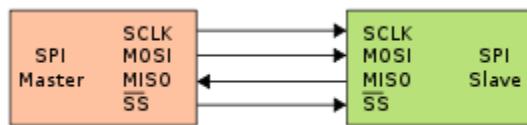
**Table 7. Intensity Register Format (Address (Hex) = 0xXA)**

DUTY CYCLE MAX7219	DUTY CYCLE MAX7221	D7	D6	D5	D4	D3	D2	D1	D0	HEX CODE
1/32 (min on)	1/16 (min on)	X	X	X	X	0	0	0	0	0xX0
3/32	2/16	X	X	X	X	0	0	0	1	0xX1
5/32	3/16	X	X	X	X	0	0	1	0	0xX2
7/32	4/16	X	X	X	X	0	0	1	1	0xX3
9/32	5/16	X	X	X	X	0	1	0	0	0xX4
11/32	6/16	X	X	X	X	0	1	0	1	0xX5
13/32	7/16	X	X	X	X	0	1	1	0	0xX6
15/32	8/16	X	X	X	X	0	1	1	1	0xX7
17/32	9/16	X	X	X	X	1	0	0	0	0xX8
19/32	10/16	X	X	X	X	1	0	0	1	0xX9
21/32	11/16	X	X	X	X	1	0	1	0	0xXA
23/32	12/16	X	X	X	X	1	0	1	1	0xXB
25/32	13/16	X	X	X	X	1	1	0	0	0xXC
27/32	14/16	X	X	X	X	1	1	0	1	0xXD
29/32	15/16	X	X	X	X	1	1	1	0	0xXE
31/32	15/16 (max on)	X	X	X	X	1	1	1	1	0XF

## Comunicación SPI

El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj (comunicación sincrónica).

Incluye una línea de reloj, dato entrante, dato saliente y un pin de chip select, que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse. De esta forma, este estándar permite multiplexar las líneas de reloj.

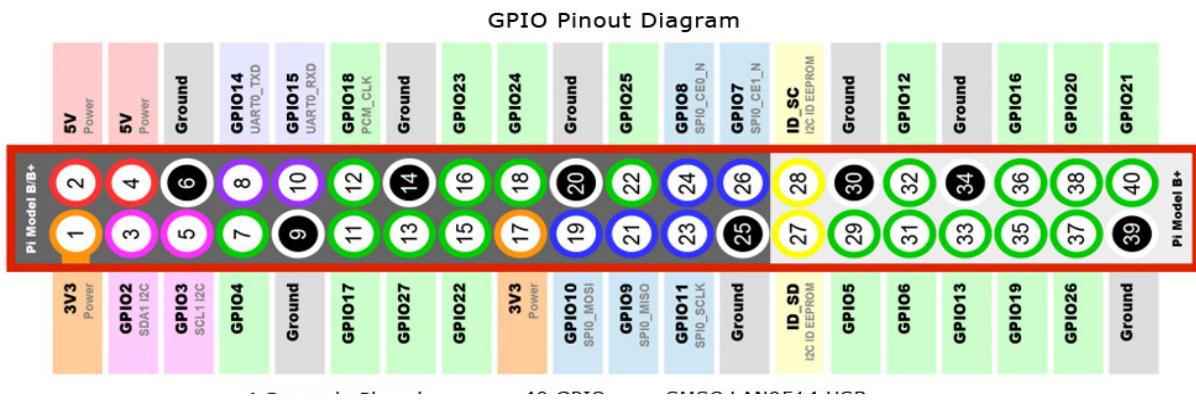


El SPI es un protocolo síncrono. La sincronización y la transmisión de datos se realiza por medio de 4 señales:

- SCLK (Clock): Es el pulso que marca la sincronización. Con cada pulso de este reloj, se lee o se envía un bit. También llamado TAKT (en Alemán).
- MOSI (Master Output Slave Input): Salida de datos del Master y entrada de datos al Slave. También llamada SIMO.
- MISO (Master Input Slave Output): Salida de datos del Slave y entrada al Master. También conocida por SOMI.
- SS/Select: Para seleccionar un Slave, o para que el Master le diga al Slave que se active. También llamada SSTE.

La Cadena de bits es enviada de manera síncrona con los pulsos del reloj, es decir con cada pulso, el Master envía un bit. (2)

Rpi3 esta equipada con un bus de SPI con 2 chip selects.



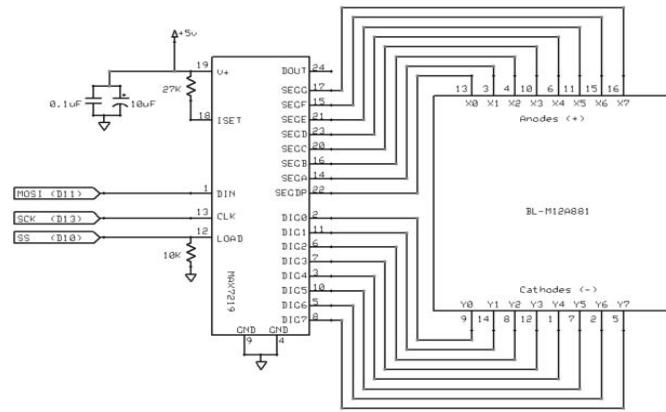
SPI_MOSI	GPIO 10	PIN 19	DATA
SPI_MISO	GPIO 21	PIN 21	
SPI_CE0_N	GPIO 08	PIN 24	CHIP SELECT
SPI_CE1_N	GPIO 07	PIN 26	
SPI_CK	GPIO 11	PIN 23	CLOCK

Utilizaremos los GPIO's en amarillo para conectar nuestra matriz de 8x8 leds, mas dos terminales extra, una para VCC (GPIO 2) y una para GND (GPIO 6).

VCC	PIN 2	5V
GND	PIN 6	TIERRA

ref:

- (1) <https://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf>
- (2) [https://es.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](https://es.wikipedia.org/wiki/Serial_Peripheral_Interface)
- (3) <https://www.raspberrypi.org/documentation/hardware/raspberrypi/spi/README.md>

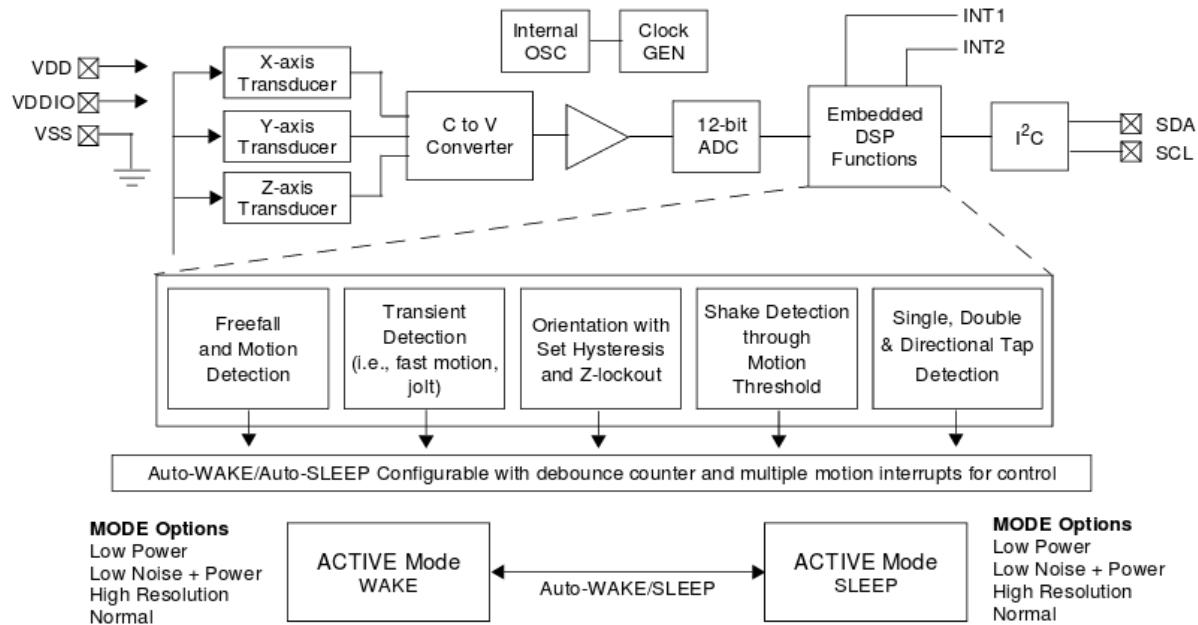


## Acelerometro MMA8452

El MMA8452Q es un acelerómetro de bajo consumo, de tres ejes, capacitivos con 12 bits de resolución. Este acelerómetro está repleto de funciones integradas con opciones flexibles de usuario programables, configurables a dos pines de interrupción. Las funciones embebidas de interrupción permiten el ahorro de energía en general para aliviar el procesador central de datos de la encuesta continua.

El MMA8452Q tiene seleccionables por el usuario escalas completas de  $\pm 2g$  /  $\pm 4g$  /  $\pm 8g$  con datos filtrados de paso alto así como los no filtrados de datos disponibles en tiempo real. Su salida serie por bus I<sup>2</sup>C.

### 1 Block Diagram and Pin Description



El utilizar el acelerometro MMA8452Q es muy sencillo obtener los valores de aceleración en los ejes X, Y, y Z. Ademas de una buena comunicación por medio del protocolo I<sup>2</sup>C.

## 6.7 Control Registers

**Note:** Except for STANDBY mode selection, the device must be in STANDBY mode to change any of the fields within CTRL\_REG1 (0X2A).

### 0x2A: CTRL\_REG1 System Control 1 Register

0x2A CTRL\_REG1 Register (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ASLP_RATE1	ASLP_RATE0	DR2	DR1	DR0	LNOISE	F_READ	ACTIVE

**Table 53. CTRL\_REG1 Description**

ASLP_RATE[1:0]	Configures the Auto-WAKE sample frequency when the device is in SLEEP Mode. Default value: 00. See <a href="#">Table 54</a> for more information.
DR[2:0]	Data rate selection. Default value: 000. See <a href="#">Table 55</a> for more information.
LNOISE	Reduced noise reduced Maximum range mode. Default value: 0. (0: Normal mode; 1: Reduced Noise mode)
F_READ	Fast Read mode: Data format limited to single Byte Default value: 0. (0: Normal mode 1: Fast Read Mode)
ACTIVE	Full Scale selection. Default value: 00. (0: STANDBY mode; 1: ACTIVE mode)

Configuramos nuestro registro para operar de manera normal (ACTIVE) (0x2A----- → 0x01).

### 0x0E: XYZ\_DATA\_CFG Register

The XYZ\_DATA\_CFG register sets the dynamic range and sets the high pass filter for the output data. When the HPF\_OUT bit is set. The data registers 0x01 - 0x06 will contain high pass filtered data when this bit is set.

0x0E: XYZ\_DATA\_CFG (Read/Write)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	HPF_OUT	0	0	FS1	FS0

**Table 15. XYZ Data Configuration Descriptions**

HPF_OUT	Enable High pass output data 1 = output data high pass filtered. Default value: 0
FS[1:0]	Output buffer data format full scale. Default value: 00 (2g).

The default full scale value range is 2g and the high pass filter is disabled.

**Table 16. Full Scale Range**

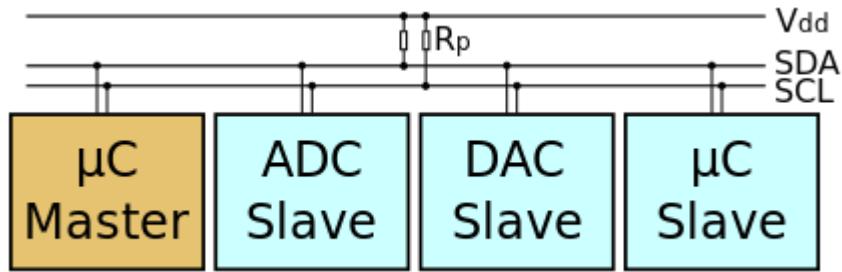
FS1	FS0	Full Scale Range
0	0	2
0	1	4
1	0	8
1	1	Reserved

Configuramos nuestro registro para operar en la escala mas baja ([+/-]2g) (0x0E----- → 0x00)

## I2C

El I<sup>2</sup>C está diseñado como un bus maestro-esclavo. La transferencia de datos es siempre inicializada por un maestro; el esclavo reacciona. Es posible tener varios maestros mediante un modo multimaestro, en el que se pueden comunicar dos maestros entre si, de modo que uno de ellos trabaja como esclavo.

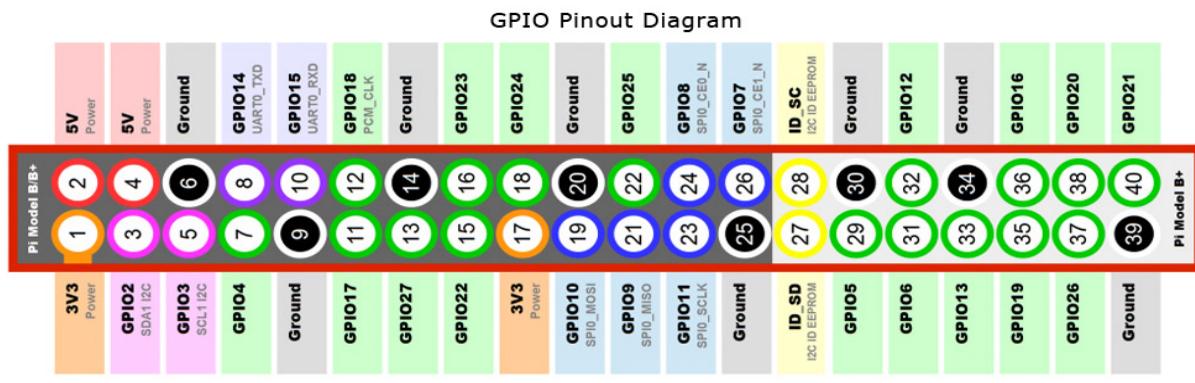
El arbitraje (control de acceso en el bus) se rige por las especificaciones, de este modo los maestros pueden ir turnándose.



Una de las propiedades del I<sup>2</sup>C es el hecho de que un microcontrolador puede controlar toda una red de circuitos integrados con sólo dos I/O-Pins (Input/Output) y un software muy simple. Los buses de este tipo fueron realizados ya que una proporción significativa del precio de un circuito integrado y la placa de circuito depende del tamaño de la carcasa y del número de pines. Una carcasa grande tiene más pines, necesita más espacio en la placa de circuito y tiene más conexiones que podrían fallar. Todo esto aumenta los costes de desarrollo, producción y pruebas.

Aunque es más lento que los sistemas de bus más nuevos, I<sup>2</sup>C es beneficioso (debido al bajo coste) para los sistemas periféricos que no necesitan ser rápidos. A menudo es usado para la transmisión de datos de control y configuración, por ejemplo para control de volumen, conversor de señal analógica-digital o digital-analógica con baja tasa de frecuencia de muestreo, relojes a tiempo real, pequeños espacios de memoria o commutadores bidireccionales y multiplexores. Incluso los sensores electrónicos integran con frecuencia un convertidor analógico-digital con un I<sup>2</sup>C.(5)

Rpi3 esta equipada con un bus de I2C.



ref:

(4) <http://www.electronicaembajadores.com/datos/pdf2/ss/ssac/ssac002.pdf>

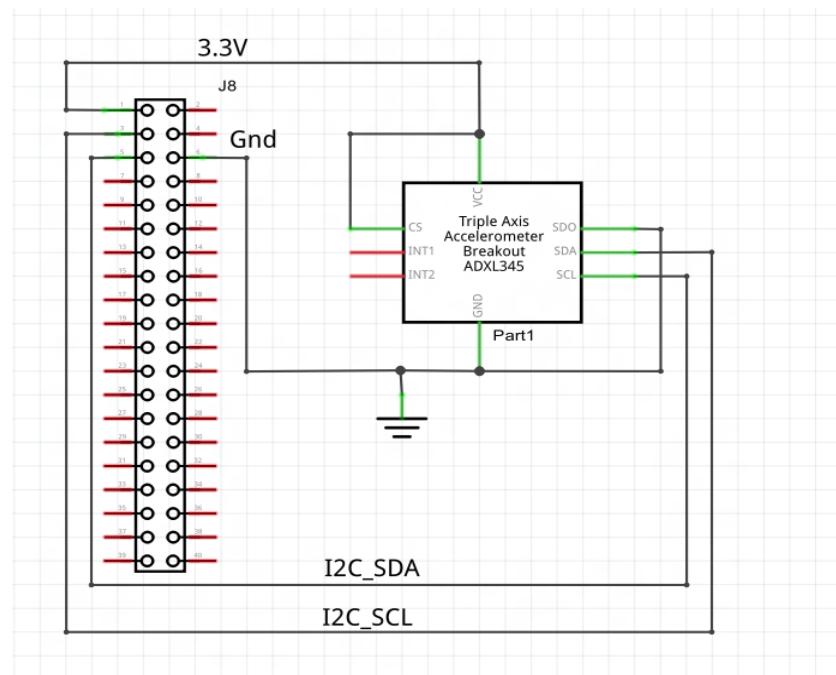
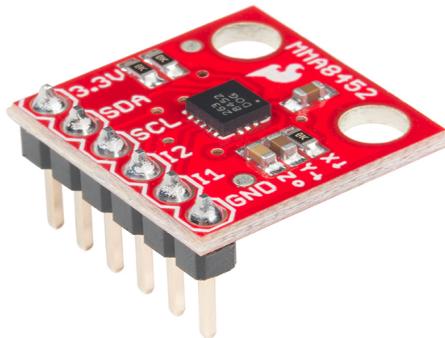
(5) <https://es.wikipedia.org/wiki/I%C2%BCB2C>

SDA I2C	GPIO 2	PIN 3
SCL I2C	GPIO 3	PIN 5

Utilizaremos los GPIO's mas dos terminales extra, una para VCC (PIN 1) y una para GND (PIN 9).

VCC	PIN 1	3.3V
GND	PIN 9	TIERRA

Diagrama de conexión con pi3.



## Esquematico general de conexiones

