PROYECTO TD III UTN - FRA 2016

INCLINOMETRO DIGITAL

Adrián Pablo Cafa

e-mail: [adriancafa@gmail.com](mailto:adriancafa@gmail.com) / adriancafa@hotmail.com

**RESUMEN:** *El proyecto realizado consiste en un inclinómetro digital realizado a partir de una IMU (dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giróscopos).*

**PALABRAS CLAVE**: Inclinometro – acelerómetro – giróscopo - IMU

Contenido

[1 INTRODUCCIÓN 3](#_Toc474751332)

[2 DIAGRAMA EN BLOQUES: 5](#_Toc474751333)

[3 DIAGRAMA DE CONEXIONES DETALLADO: 6](#_Toc474751334)

[4 DESCRIPCION DE LOS COMPONENTES DE HARDWARE: 6](#_Toc474751335)

[5 DIAGRAMA ESQUEMATICO Y CIRCUITO IMPRESO: 11](#_Toc474751336)

[6 MODO DE EMPLEO E INTERFAZ CON EL USUARIO 11](#_Toc474751337)

[7 CALIBRACION DEL INSTRUMENTO Y CALCULO DE LA INCERTIDUMBRE 12](#_Toc474751338)

[7.1 Cálculo del intervalo para α1 14](#_Toc474751339)

[7.2 Cálculo del intervalo para α2 14](#_Toc474751340)

[7.3 Cálculo del intervalo para α3 14](#_Toc474751341)

[7.4 Cálculo del intervalo para α4 14](#_Toc474751342)

[7.5 Cálculo del intervalo para α5 15](#_Toc474751343)

[7.6 Cálculo del intervalo para α6 15](#_Toc474751344)

[7.7 Cálculo del intervalo para α7 15](#_Toc474751345)

[7.8 Cálculo del intervalo para α8 15](#_Toc474751346)

[7.9 Cálculo del intervalo para α9 16](#_Toc474751347)

[7.10 Cálculo del intervalo para α10 16](#_Toc474751348)

[8 GUIA DE CODIGO: 17](#_Toc474751349)

[9 ESTUDIO DE MERCADO: 20](#_Toc474751350)

[10 ESPECIFICACIONES DEL EQUIPO: 20](#_Toc474751351)

[11 CONCLUSIONES: 21](#_Toc474751352)

[12 BIBLIOGRAFIA: 21](#_Toc474751353)

# INTRODUCCIÓN

***¿QUÉ ES UN IMU MPU-6050?***

Se trata de un dispositivo capaz de medir la fuerza (aceleración) y la velocidad. Genéricamente consta de un Acelerómetro y un Giroscopio. Por lo tanto una IMU no mide ángulos. Por lo menos no directamente, requiere algunos cálculos.

El MPU-6050 es una IMU de 6DOF (se lee “6 Degrees Of Freedom“). Esto significa que lleva un acelerómetro y un giroscopio, ambos de 3 ejes (3+3 = 6DOF). Hay IMUs de 9DOF, en este caso también llevan un magnetómetro. Otras pueden tener 5DOF, en cuyo caso el giroscopio sólo mide dos ejes, etc.

MPU-6050

El MPU-6050 opera con 3.3 voltios, aunque algunas versiones (como la mía) llevan un regulador que permite conectarla a 5V.

El MPU-6050 utiliza el protocolo de comunicación I2C.

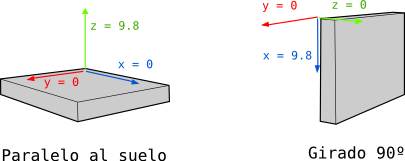
PARTES DE LA IMU:

**El Acelerómetro:**

El **acelerómetro** como su nombre indica mide la aceleración. La aceleración puede expresarse en 3 ejes: X, Y y Z, las tres dimensiones del espacio. Por ejemplo, si se mueve la IMU hacia arriba, el eje Z marcará un cierto valor. Si es hacia delante, marcará el eje X, etc.

Así pues, la IMU también detecta la aceleración de la gravedad terrestre. Gracias a la gravedad terrestre puedes usar las lecturas del acelerómetro para saber cuál es el **ángulo de inclinación** respecto al eje X o eje Y.

Supongamos que la IMU esté perfectamente alineada con el suelo. Entonces, como puedes ver en la imagen, **el eje Z marcará 9.8**, y los otros dos ejes marcarán 0. Ahora supongamos que **giramos la IMU 90 grados**. Ahora es el eje X el que está perpendicular al suelo, por lo tanto marcará la aceleración de la gravedad.

[](http://robologs.net/wp-content/uploads/2014/10/imu_ac_ejemplos.png)

Si sabemos que la gravedad es 9.8 m/s², y sabemos que medición dan los tres ejes del acelerómetro, por trigonometría es posible calcular el ángulo de inclinación de la IMU. Una buena fórmula para calcular el ángulo es:

[CodeCogsEqn(3)](http://robologs.net/wp-content/uploads/2014/10/codecogseqn3.gif)

[CodeCogsEqn(2)](http://robologs.net/wp-content/uploads/2014/10/codecogseqn2.gif)

Ahora, dado que el ángulo se calcula a partir de la gravedad, no es posible calcular el ángulo Z con esta fórmula ni con ninguna otra. Para hacerlo se necesita otro componente: el magnetómetro, que es un tipo de brújula digital. El MPU-6050 no lleva, y por tanto nunca podrá calcular con precisión el ángulo Z. Sin embargo, para la gran mayoría de aplicaciones sólo se necesitan los ejes X e Y.

**El Giroscopio**

En un principio, los giroscopios eléctricos eran unos voluminosos artefactos que valían la mayor parte del presupuesto militar de un estado. Más tarde, durante la segunda guerra mundial se emplearon para dirigir cohetes y torpedos. Por suerte, gracias la revolución digital y la miniaturización de circuitos, hoy en día cualquier aficionado a la electrónica puede permitirse uno. [](http://robologs.net/wp-content/uploads/2014/10/v2b.jpg)

*Los giroscopios eléctricos se utilizaron en los cohetes V-2*

El **giroscopio** mide la **velocidad angular,** la velocidad angular es el número de **grados que se gira en un segundo**.

Si sabemos el **ángulo inicial** de la IMU, podemos sumarle el valor que marca el giroscopio para saber el**nuevo ángulo** a cada momento. Supongamos que iniciamos la IMU a 0º. Si el giroscopio realiza una medida cada segundo, y marca 3 en el eje X, tendremos el ángulo con esta sencilla fórmula:

Dónde **Δ**t es el **tiempo que transcurre** cada vez que se calcula esta fórmula, **AnguloYAnterior** es el **ángulo calculado la última vez** que se llamó esta fórmula y **GiroscopioY** es**la lectura del ángulo Y del giroscopio**.

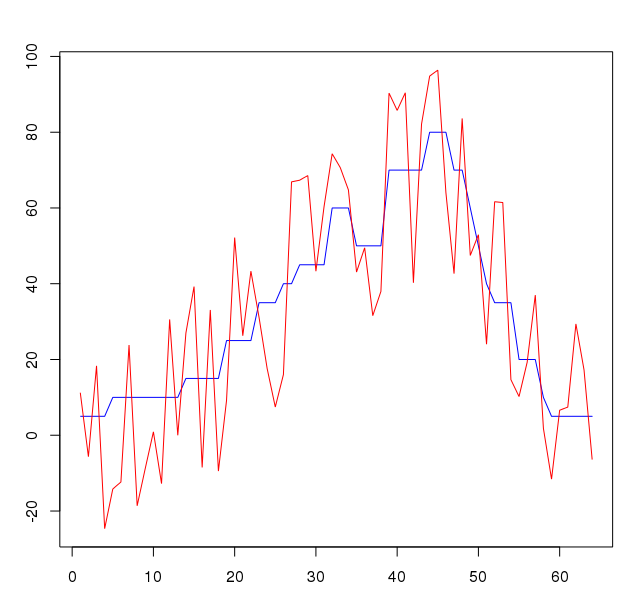
Y lo mismo pasa con los ejes X, Z. Sólo que se suele ignorar el eje Z, puesto que al no poder calcular un ángulo Z con el Acelerómetro, no se puede aplicar un Filtro Complementario para el eje Z (posteriormente voy a explicar que es el filtro complementario y para que lo utilicé en este proyecto).

Ya tenemos las lecturas ahora procederemos a procesarlas para obtener inclinaciones.

Error en las medidas:

Hay dos problemas muy importantes: el **ruido** y los **errores.**

El ruido son todas aquellas interferencias que afectan a los dispositivos electrónicos. El acelerómetro es capaz de medir cualquier ángulo, sin embargo sus lecturas son ruidosas y tienen un cierto margen de error.

Si dibujamos por ejemplo un típico gráfico de las medidas de un acelerómetro digital en función del tiempo, veremos algo como lo siguiente:[](http://robologs.net/wp-content/uploads/2014/10/ruido_imu.png)El ángulo real (ideal) está marcado en azul, y las medidas reales están en rojo.

Además de esto, el acelerómetro también detecta cualquier aceleración que no sea la de la gravedad. Por tanto, si movemos la IMU sin girarla, al aplicar una aceleración en otro eje, la IMU lo detectará como un cambio de rotación.

Por otra parte tenemos el giroscopio. A diferencia del acelerómetro, da las medidas con mucha precisión. Pero al realizar los cálculos del ángulo es inevitable que se produzca un pequeño error, que con el tiempo va acumulándose, esto es el drift.

Esto no significa para nada que no se logre obtener inclinaciones de precisión. Hay varias formas de combinar los datos del acelerómetro y el giroscopio para así obtener medidas precisas y aquí es donde entra en juego el **filtro complementario.**

El filtro:

Hay que conseguir eliminar el **ruido**, el **drift** y conseguir que el acelerómetro no cambie de ángulo al detectar otra fuerza que no sea la gravedad.

Hay distintos algoritmos, llamados **filtros**, que hacen esta tarea. Uno de los mejores es el famoso **Filtro de Kálman**. Se utiliza en los **aviones**, **cohetes** y **satélites** geoestacionarios.

El filtro de Kálman es sorprendente. Es considerado uno de los mayores hallazgos del siglo pasado, y con razón. Es capaz de calcular el error de cada medida a partir de las medidas anteriores, eliminarlo y dar el valor real del ángulo. En cierto modo es un algoritmo que aprende en cada iteración.

Sin embargo tiene dos problemas:

1. Tiene un coste de procesamiento algo elevado
2. Es muy complicado de entender

Así pues, tenemos otros filtros a nuestra disposición. El que utilicé en mi proyecto es conocido como **Filtro Complementario**. Es fácil de utilizar, bajo coste de procesamiento y con una precisión muy buena.

El **Filtro Complementario** es en realidad una unión de dos filtros diferentes: un **High-pass Filter** para el giroscopio y un **Low-pass Filter** para el Acelerómetro. El primero deja pasar únicamente los valores por encima de un cierto límite, al contrario que el Low-pass filter, que sólo permite a los que están por debajo.

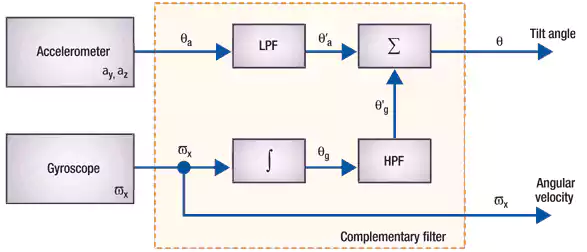
La fórmula resultante de combinar (**complementar**, de aquí el nombre) los dos filtros es:

lo = 0,98\*(Angulo + AnguloGyro \* Δt) + 0,02\*AnguloAccel)

Dónde **AnguloGyro** es el ángulo del Giroscopio que hemos calculado previamente, y **AnguloAccel** con el ángulo del Acelerómetro calculado con la fórmula de la tangente. **Δ**t es **el tiempo que paso** (en segundos) **desde la última vez que calculo el filtro**. Esta fórmula es la misma para el eje X, Y.

NOTA: También se puede cambiar el valor de 0.98 y 0.02 por un valor personalizado. Eso sí, ambos tienen que sumar 1.

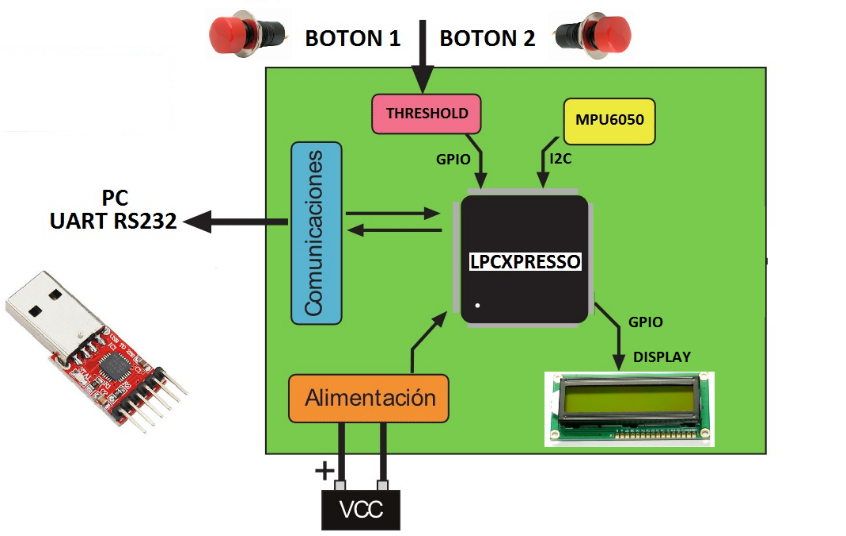
Diagrama en bloques del filtro complementario:



# DIAGRAMA EN BLOQUES:

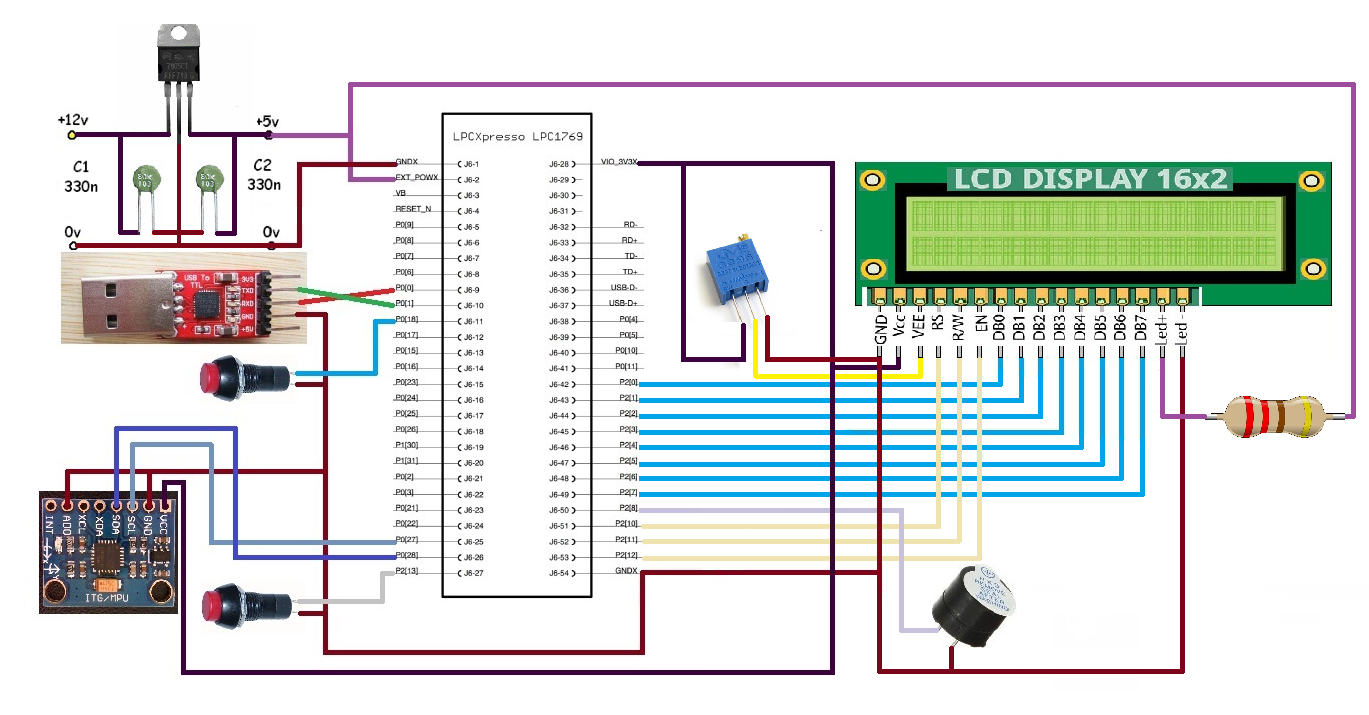
Se detalla a continuación el diagrama en bloques del proyecto utilizado. En el proyecto enviamos constantemente los datos de la IMU al microcontrolador y estos los debugueamos por la UART a la pc para tener un registro y también lo visualizamos en el display. Al mismo tiempo el usuario puede seleccionar el threshold (por defecto esta en 45º) con los 2 botones que están a disposición del usuario.

El microcontrolador se comunica con el sensor por I2C.



# DIAGRAMA DE CONEXIONES DETALLADO:

Se esquematiza en la figura siguiente el diagrama de conexiones completo de todo el hardware utilizado en el proyecto.



# DESCRIPCION DE LOS COMPONENTES DE HARDWARE:

**SENSOR MPU6050:**

El sensor escogido, el IMU MPU6050 del fabricante Invensense, presenta

las siguientes características:

Giróscopo:

* Sistema de giróscopo en 3 ejes fabricado con tecnología MEMS.
* Puede ser ajustado a valores de ±250, ±500, ± 1000, y ±2000 °/s
* Integra 3 conversores analógico digital

de 16 bits, uno por cada eje

* Dispone de un filtro paso bajo programable
* Consume 3.6mA.
* Dispone de self-test programable.

Acelerómetro:

* Sistema de acelerómetro

en 3 ejes fabricado con tecnología MEMS.

* Ajustable a valores de

Sensibilidad de ±2g, ±4g, ±8g

y ±16g

* Integra 3 conversores analógico digital

de 16 bits, uno por cada eje

* Consumo de 500 μA
* Dispone de interrupciones programables.
* Detección de caída libre, vibraciones y movimiento 0.
* Dispone de self- test programable



Otras:

* Posibilidad de ser ampliado con un magnetómetro a un sistema de 9 grados de libertad.
* Master auxiliar para comunicación I2C.
* Regulador de tensión propio.
* Buffer o memoria de 1024 bytes FIFO.
* Sensor de temperatura.
* Filtros programables por el usuario para los acelerómetros, los giróscopos y el sensor de temperatura.
* Dispone de un DMP (Digital Motion Processor), un procesador disponible para el usuario que permitiría descargar de trabajo al microcontrolador principal.
* Interfaces de comunicación SPI e I2C primaria y secundaria.

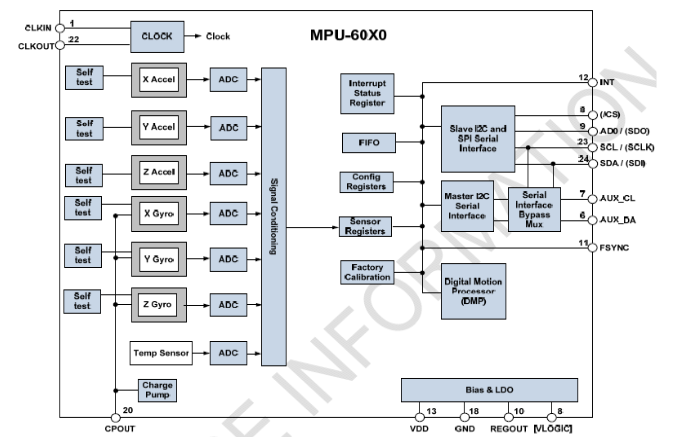


Diagrama de bloques MPU 6050.

(Fuente: MPU6050 Datasheet).

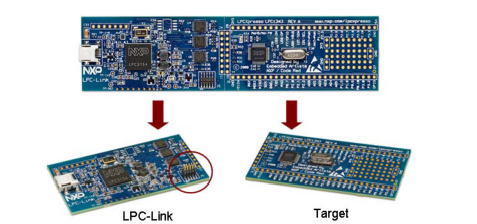
**LPCXpresso LPC1769:**

Es una placa de desarrollo desarrollada por el fabricante **NXP**, basada en un microcontrolador ARM Cortex-M3 de 32 bits con 64kb de memoria y 512 kb de memoria Flash, que trabaja a 120 MHz de velocidad de reloj, además de disponer soporte USB 2.0 (host y device) y Ethernet.

(Ver especificaciones de la “disposición de pines” y la “arquitectura del LPCxPresso” en Anexos)

Otras características importantes del LPC1769 son:

* 4 puertos UART
* 1 controlador DMA de 8 canales de propósito general
* 2 canales CAN
* 2 controladores SSP
* 1 interfaz SPI
* 8 canales de 12 bits ADC
* DAC de 10 bits
* 70 pines GPIO de propósito general.



**Modulo LCD**

Nuestro proyecto dispone de un módulo LCD tipo 16x2, el Winstar modelo WH1602A.

Las pantallas de cristal líquido LCD o display LCD para mensajes (Liquid Cristal Display) tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma sencilla.

La pantalla consta de una matriz de caracteres (normalmente de 5x7 o 5x8 puntos) distribuidos en dos líneas de 16 caracteres cada línea.

El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi 44780 el modelo de controlador más utilizado.

*Características generales de un módulo LCD 16x2*:

* Consumo muy reducido, del orden de 7.5mW
* Pantalla de caracteres ASCII, además de los caracteres japoneses Kanji, caracteres griegos y símbolos matemáticos
* Desplazamiento de los caracteres hacia la izquierda o a la derecha
* Memoria de 40 caracteres por línea de pantalla, visualizándose 16 caracteres por línea
* Movimiento del cursor y cambio de su aspecto
* Permite que el usuario pueda programar 8 caracteres
* Pueden ser gobernados de 2 formas principales
  + Conexión con bus de 4 bits
  + Conexión con bus de 8 bits

*Descripción de los pines*

A continuación se presenta la descripción de señales empleadas por el módulo LCD así como el número de pin a la que corresponden.

|  |  |  |
| --- | --- | --- |
| **PIN Nº** | **SIMBOLO** | **DESCRIPCION** |
| 1 | *Vss* | *Patilla de tierra de alimentación* |
| 2 | *Vdd* | *Patilla de alimentación de 5V* |
| 3 | *Vo* | *Patilla de contraste del cristal líquido. Normalmente se conecta a un potenciómetro* |
|  |  | *a través del cual se aplica una tencion variable entre 0 y +5V* |
| 4 | *RS* | *Selección del registro de control* |
|  |  | *RS=0 Selección del registro de control* |
|  |  | *RS=1 Selección del registro de datos* |
| 5 | *R/W* | *Señal de lectura/escritura* |
|  |  | *R/W=0 El módulo LCD es escrito* |
|  |  | *R/W=1 El módulo LCD es leído* |
| 6 | *E* | *Señal de activación del módulo LCD* |
|  |  | *E=0 Módulo desconectado* |
|  |  | *E=1 Módulo conectado* |
| 7 a 14 | *D0-D7* | *Bus de datos bi-direccional. A través de estas líneas se realiza la transferencia* |
|  |  | *de información entre el módulo LCD y el sistema informático que lo gestiona* |
| 15 | *A* | *LED Backlight (-)* |
| 16 | *K* | *LED Backlight (+)* |



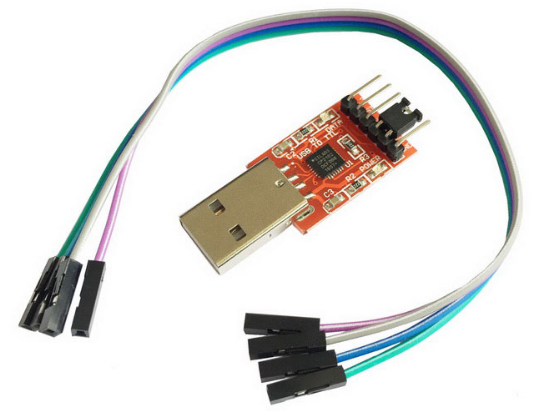
Dicho LCD tiene una tensión de alimentación típica de 5V, el cual concuerda con la alimentación del microcontrolador. El LCD tiene una comunicación en paralelo, es decir, posee 8 pines de datos. Junto con 1 Enable para habilitar el módulo LCD y los respectivos pines de RW (Read/Write) y RS (Comando o Dato).

***Características de alimentación del display:***



**Módulo CP2102 USB/UART:**

Este módulo basado en el chip CP2102 fabricado por Silicon Labs, permite la comunicación bidireccional entre la PC (USB) y un dispositivo que disponga de puertos UART, en nuestro caso la placa LPC1769.



MODULO CP2102

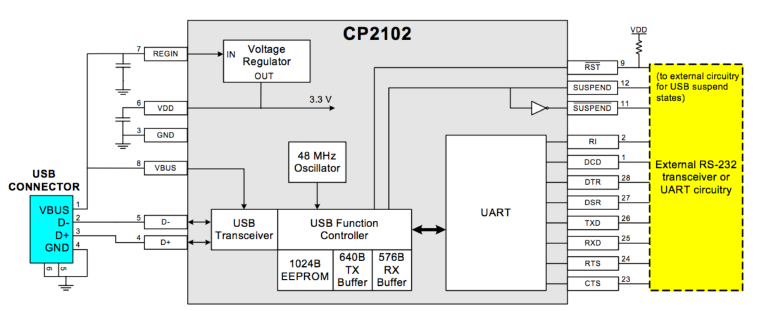
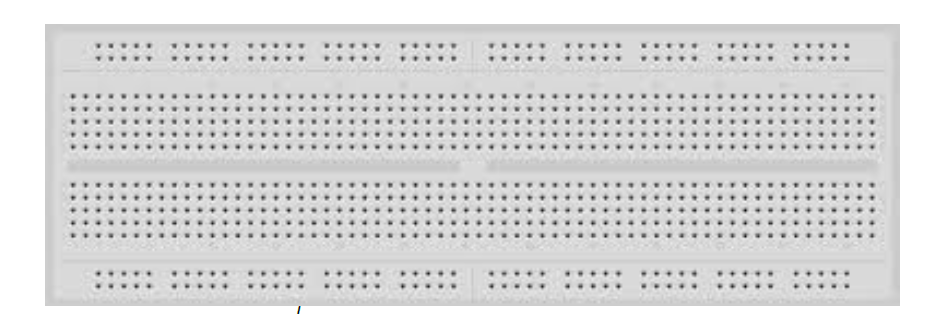


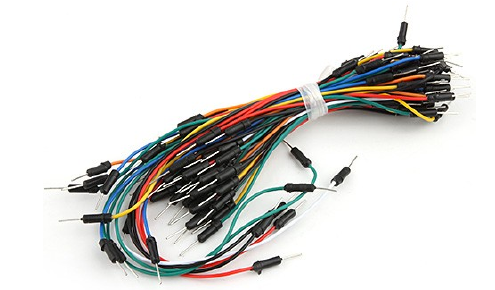
DIAGRAMA EN BLOQUES DEL CP2102

**Placa de prueba y cableado:**

Para facilitar el montaje rápido del prototipo se utilizó una protoboard.



Para la interconexión de los distintos componentes se utilizaron cables especiales para protoboard de diferentes tamaños.



CABLES DE PROTOBOARD

**ZUMBADOR ACTIVO 5V – 2,3 kHz:**

Al conectarse, dentro del rango de voltaje de entrada de alimentación, genera un pitido de 2,3Khz.

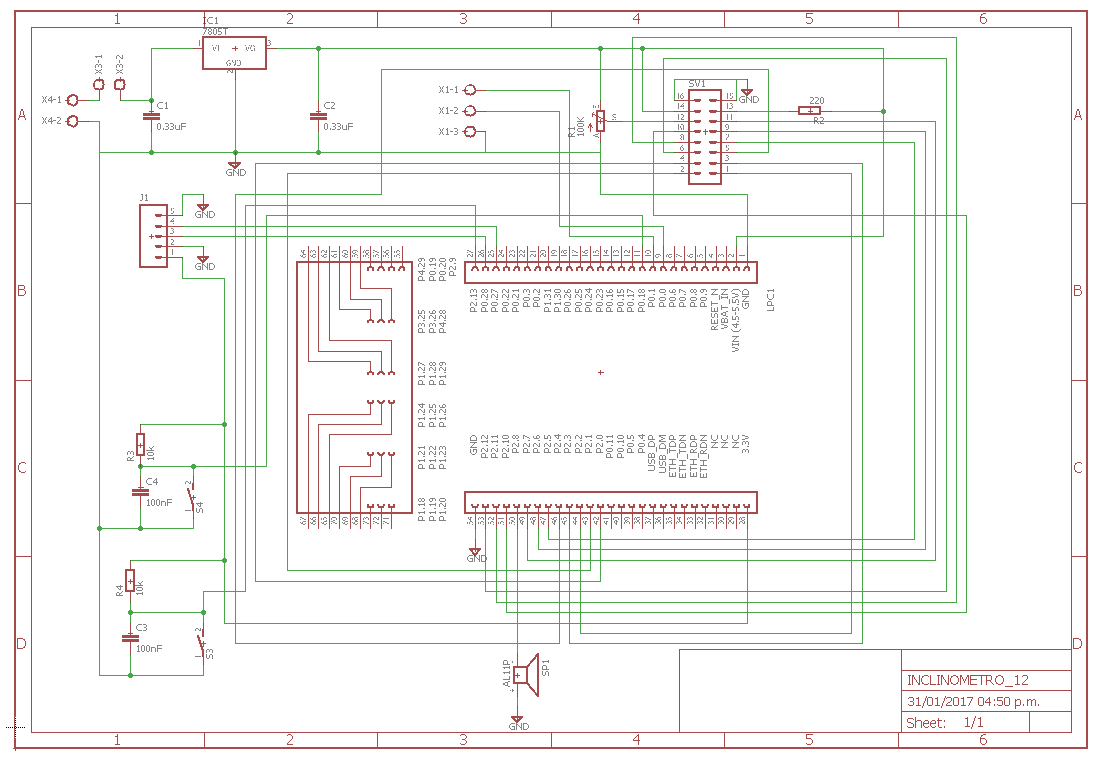
Voltage :3.5-5 .5 V

Current: <25mA

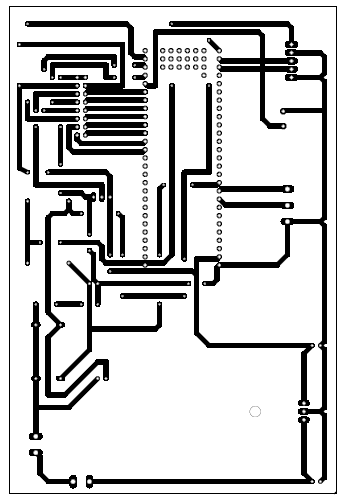
Frecuencia: 2300 ± 500



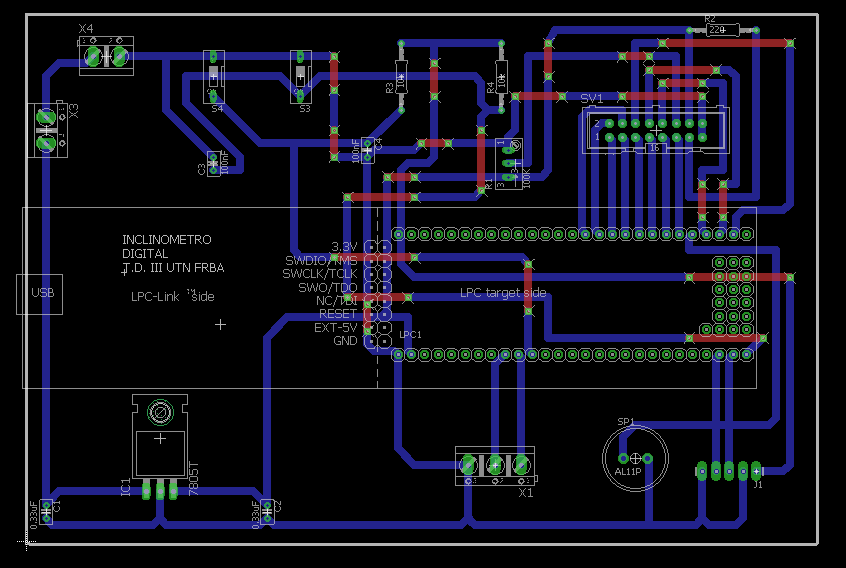
# DIAGRAMA ESQUEMATICO Y CIRCUITO IMPRESO:



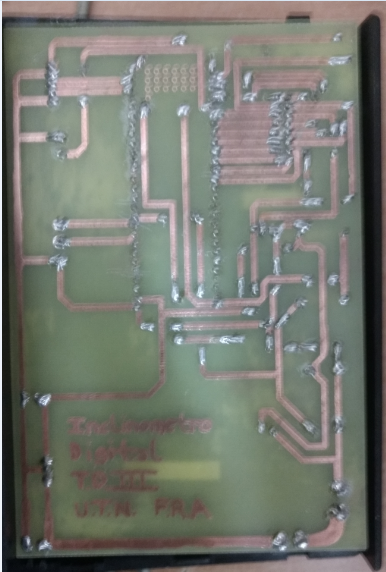
ESQUEMATICO



CIRCUITO IMPRESO



CIRCUITO IMPRESO CON VISTA DE LOS COMPONENTES



FOTOGRAFIA DEL PCB ARMADO

# MODO DE EMPLEO E INTERFAZ CON EL USUARIO

Se adjunta un MANUAL DE USUARIO con detalles de fotografías, solo a modo informativo citamos los puntos más relevantes para el empleo del dispositivo.

**Modo de empleo:**

1.- Enchufe el aparato con una fuente de energía mayor a 5V de corriente continua, luego encienda el aparato presionando el switch de encendido, ubicado en la parte derecha, hasta alcanzar el otro extremo

2.- El aparato está listo para usarse. La lectura debe realizarse de la siguiente forma: en la pantalla del display, en el primer renglón aparecerá el nivel de ángulo threshold (45º inicialmente y por defecto). En el segundo renglón se visualiza la medición de la inclinación del dispositivo (ROLL) con un intervalo de muestro de 200 ms. Si se supera el ángulo threshold girando en sentido horario u antihorario sonara la alarma. Ademas en el segundo renglón se especifica el angulo de referencia (0° inicialmente) el cual lo variamos presionando ambos botones al mismo tiempo.

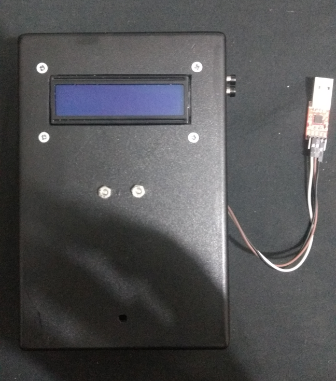
3.- Si el usuario lo desea puede setear con los botones el ángulo de threshold deseado. En el mismo display se podrá visualizar este ángulo.

A su vez como se indico anteriormente el usuario al presionar los dos botones al mismo tiempo puede actualizar el ángulo de referencia

4.- El usuario también dispone de la posibilidad de conectarse por medio del CP2102 montado en equipo a algún puerto libre de la pc con el fin de visualizar por una terminal (Putty, Xshell, etc) las distintas inclinaciones. Cuenta gracias a esto la posibilidad de ir registrando en un archivo las mismas.

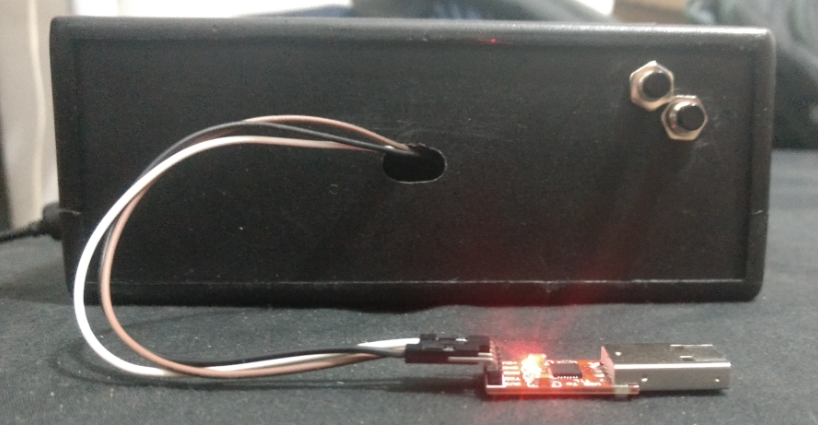
**Interfaz con el usuario:**

**VISTA FRONTAL:**



**VISTAS LATERALES:**





# CALIBRACION DEL INSTRUMENTO Y CALCULO DE LA INCERTIDUMBRE

Para poder conocer el grado de error que posee el inclinómetro, se calculó la incertidumbre del mismo para distintos valores de inclinación mediante un sistema de medición. Para ello se construyó un sistema de plano inclinado cuya inclinación es variable. Dicho sistema tiene incorporado un goniómetro para poder medir la inclinación del mismo.

El proceso de medición consistió en colocar el inclinómetro en el sistema de plano inclinado para distintos grados de inclinación, obteniendo de esta manera mediciones realizadas con el inclinómetro y con el goniómetro(referencia). Finalmente, con las mediciones realizadas se calcularon los intervalos de incertidumbre para distintos grados de inclinación. Se opto limitar el uso del equipo hasta los 90° que es hasta donde lo podemos calibrar de forma simple. Ademas hasta 90° cubre las expectativas de los posibles usos del equipo

Intervalos de incertidumbre:

Para el cálculo del intervalo de incertidumbre se realizaron 8 mediciones con el inclinómetro digital para cada uno de los ángulos seleccionados que se detallan a continuación:

|  |  |
| --- | --- |
| Ángulo | Valor |
| α1 | 0° |
| α2 | 10° |
| α3 | 20° |
| α4 | 30° |
| α5 | 40° |
| α6 | 50° |
| α7 | 60° |
| α8 | 70° |
| α9 | 80° |
| α10 | 90° |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



Vista lateral sistema de medición

Vista superior sistema de medición

Para el cálculo directo se utiliza la siguiente expresión:

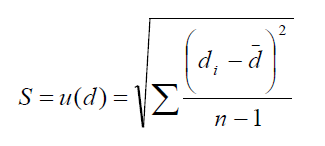
“y” es la incertidumbre de la magnitud a medir

“yr” es la incertidumbre de la magnitud de referencia.

“d” es la incertidumbre de la diferencia medida

La incertidumbre yr viene especificada en el instrumento patrón, para nuestro caso se tomará la que corresponda al goniómetro.

La incertidumbre d se calcula de la siguiente manera:



Para calcular la incertidumbre final del inclinómetro digital se emplea la siguiente expresión:



Si tomamos un k=2 (95% de confianza) el intervalo de incertidumbre (expandida) estará dado por



Para todos los ángulos se emplea yr de 0,5º

## 7.1 Cálculo del intervalo para α1

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α1-1 | 0 |
| α1-2 | -1 |
| α1-3 | 0 |
| α1-4 | -1 |
| α1-5 | 0 |
| α1-6 | -1 |
| α1-7 | 0 |
| α1-8 | 0 |
| α1-9 | 0 |
| α1-10 | 0 |

Valor medio: = -0.25

Incertidumbre estándar: = 0,2142857143

u(y) = = 0,6813

U = 2u(y) = ± 2 . 0.6813 = ± 1.3627º

## Cálculo del intervalo para α2

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α2-1 | 10 |
| α2-2 | 10 |
| α2-3 | 10 |
| α2-4 | 10 |
| α2-5 | 10 |
| α2-6 | 9 |
| α2-7 | 10 |
| α2-8 | 9 |
| α2-9 | 10 |
| α2-10 | 10 |

Valor medio: = 9.8

Incertidumbre estándar: = 0.1778

u(y) = = 0.6540

U = 2u(y) = ± 2 . = ± 1.3080º

## Cálculo del intervalo para α3

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α3-1 | 20 |
| α3-2 | 20 |
| α3-3 | 20 |
| α3-4 | 21 |
| α3-5 | 21 |
| α3-6 | 20 |
| α3-7 | 20 |
| α3-8 | 20 |
| α3-9 | 21 |
| α3-10 | 20 |

Valor medio: = 20,3

Incertidumbre estándar: = 0.2333

u(y) = = 0.6952

U = 2u(y) = ± 2 . = ± 1.3904º

## Cálculo del intervalo para α4

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α4-1 | 31 |
| α4-2 | 31 |
| α4-3 | 31 |
| α4-4 | 31 |
| α4-5 | 30 |
| α4-6 | 31 |
| α4-7 | 31 |
| α4-8 | 31 |
| α4-9 | 30 |
| α4-10 | 31 |

Valor medio: = 30.8

Incertidumbre estándar: = 0.1778

u(y) = = 0.6540

U = 2u(y) = ± 2 . = ± 1.3081º

## Cálculo del intervalo para α5

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α5-1 | 40 |
| α5-2 | 40 |
| α5-3 | 41 |
| α5-4 | 41 |
| α5-5 | 41 |
| α5-6 | 41 |
| α5-7 | 40 |
| α5-8 | 40 |
| α5-9 | 40 |
| α5-10 | 41 |

Valor medio: = 40.5

Incertidumbre estándar: = 0.2778

u(y) = = 0.7265

U = 2u(y) = ± 2 . = ± 1.4530º

## Cálculo del intervalo para α6

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α6-1 | 51 |
| α6-2 | 51 |
| α6-3 | 51 |
| α6-4 | 51 |
| α6-5 | 51 |
| α6-6 | 51 |
| α6-7 | 50 |
| α6-8 | 51 |
| α6-9 | 50 |
| α6-10 | 51 |

Valor medio: = 50.8

Incertidumbre estándar: = 0.1778

u(y) = = 0.6540

U = 2u(y) = ± 2 . = ± 1.3081º

## Cálculo del intervalo para α7

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α7-1 | 61 |
| α7-2 | 61 |
| α7-3 | 61 |
| α7-4 | 61 |
| α7-5 | 61 |
| α7-6 | 60 |
| α7-7 | 60 |
| α7-8 | 61 |
| α7-9 | 61 |
| α7-10 | 61 |

Valor medio: = 60.8

Incertidumbre estándar: = 0.1778

u(y) = = 0.6540

U = 2u(y) = ± 2 . = ± 1.3081º

## Cálculo del intervalo para α8

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α8-1 | 71 |
| α8-2 | 71 |
| α8-3 | 71 |
| α8-4 | 70 |
| α8-5 | 70 |
| α8-6 | 71 |
| α8-7 | 71 |
| α8-8 | 71 |
| α8-9 | 70 |
| α8-10 | 71 |

Valor medio: = 70.7

Incertidumbre estándar: = 0.2333

u(y) = = 0.6952

U = 2u(y) = ± 2 . = ± 1.3904º

## Cálculo del intervalo para α9

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α9-1 | 81 |
| α9-2 | 81 |
| α9-3 | 80 |
| α9-4 | 81 |
| α9-5 | 80 |
| α9-6 | 81 |
| α9-7 | 81 |
| α9-8 | 80 |
| α9-9 | 81 |
| α9-10 | 81 |

Valor medio: = 80.7

Incertidumbre estándar: = 0.2333

u(y) = = 0.6952

U = 2u(y) = ± 2 . = ± 1.3904º

## Cálculo del intervalo para α10

Tabla de valores obtenidos con el inclinómetro:

|  |  |
| --- | --- |
| Angulo patrón | Angulo medido |
| α10-1 | 90 |
| α10-2 | 90 |
| α10-3 | 90 |
| α10-4 | 91 |
| α10-5 | 91 |
| α10-6 | 90 |
| α10-7 | 90 |
| α10-8 | 91 |
| α10-9 | 90 |
| α10-10 | 91 |

Valor medio: = 90.4

Incertidumbre estándar: = 0.2667

u(y) = = 0.7187

U = 2u(y) = ± 2 . = ± 1.4376º

Analisis de las mediciones realizadas:

Realizando un análisis de los intervalos de incertidumbre obtenidos se pudo observar que los mismos tienen un grado incertidumbre elevado, en el peor caso fue de +-1.4376º.

Se pudo observar también que en la mayoría de las mediciones realizadas, la diferencia entre lo medido por el inclinómetro y el goniómetro fue de 1º, con lo cual es muy probable que el inclinómetro tenga un error de Offset. Este error se podría corregir por software o calibrando la posición del sensor. Luego, realizando esta corrección es muy probable que volviendo a realizar las mediciones se obtengan grados de incertidumbre menores.

Otra de las causas de que se hayan obtenido valores de incertidumbre elevados es la utilización del goniómetro que posee una mínima división de un grado, con lo cual no es un instrumento muy adecuado para tomarlo como referencia. Se optó por la opción del goniómetro dado a que era el único instrumento accesible económicamente.

# 8 GUIA DE CODIGO:

FUNCIONES DEL DISPLAY (Display.c):

**void inicializacion\_lcd(void):**

No devuelve nada ni tampoco recibe argumentos. Es la encargada de inicializar el LCD, en modo 8 bits.

**void limpiarpantalla (void):**

No devuelve nada ni tampoco recibe argumentos. Función que se ocupa de borrar la pantalla del LCD

**void goto\_xy(unsigned char fila, unsigned char columna):**

No devuelve nada y recibe como argumentos dos unsigned char. Esta función se utiliza para posicionarse en el display según la fila y columna que se le pase como argumentos.

**void instruccion(unsigned int inst, unsigned int tiempo):**

No devuelve nada y recibe como argumentos dos enteros. El primer argumento es la instrucción a pasar a la memoria del LCD y el 2do argumento es el tiempo de demora que tendrá la instrucción a ejecutar

**void escribestring(char \* p):**

No devuelve nada y recibe como argumento un puntero con la dirección de inicio de una cadena de caracteres. Se utiliza para imprimir un mensaje en el display. Utiliza la función imprimecaracter(char \*) para imprimir de a un carácter la cadena.

**void imprimecaracter(char \* z):**

No devuelve nada y recibe como argumento un puntero a carácter. Se utiliza para imprimir un carácter en el LCD.

FUNCIONES Y TAREAS DEL PROGRAMA PRINCIPAL (GPIO.c)

**main():**

Funcion principal, la ejecución de la aplicación comienza con la inicialización de los componentes hardware: uart, gpio, i2C y los componentes software: tareas, semáforos. A continuación el planificador gestiona las tareas para que lleven a cabo su función. Estas se describen a continuación.

**void vENTRADA( void \*pvParameters):**

Esta tarea tarea se ocupa de revisar los botones para ver si hay que actualizar el Threshold. Se vale de la función botones().

**static** **void** **vINCLINOMETRO**(**void** \*pvParameters):

Esta tarea se ocupa de mostrar la lectura de los registros de la IMU (comunicándose por I2C). Se vale de la ayuda del Filtro Complementario. El dato del ángulo actual se envía por puerto serie para mostrarlo en una terminal y además se muestra en el display.

**void presentacion\_threshold(void):**

No devuelve nada ni tampoco recibe argumentos. Se utiliza para escribir las palabras “THRESHOID =X” (X=45 es el valor inicial por default) en el primer renglón. Cuando el usuario presiono uno de los dos botones para variar el threshold este valor de X se actualiza. Para esto se vale de la función imprimecaracter(char \*)

**void delay\_ms\_init(uint32\_t):**

Timer inicial para inicializar el display (ya que luego los tiempos los manejo con el timertick con la función propia de FreeRtos vTaskDelay()). No devuelve nada y recibe un entero como argumento. Es la función encargada de realizar un retardo al ser invocada pasándole como argumento el tiempo que se desee.

**void presentacion\_angulo(char \*Cadena\_angulo);**

No devuelve nada y recibe como argumentos un puntero a la dirección de inicio a una cadena de caracteres. Esta función es la encargada de imprimir en el segundo renglón constantemente el ángulo de inclinación indicado como “ROLL=X”, siendo X el valor de inclinación que se va actualizando constantemente.

**void** **presentacion\_referencia**(**void**):

No devuelve nada ni tampoco recibe argumentos. Esta función es la encargada de imprimir en el segundo renglón constantemente el ángulo de referencia que se toma por cada inclinación que se produce, indicado como “REF=X”, siendo X el valor de inclinación de referencia que se va actualizando constantemente. Este valor como se especifica en el manual de usuario se actualiza cuando el usuario presiona ambos botones al mismo tiempo.

**void getAres(void):**

No devuelve nada recibe nada, calcula la resolución de la aceleración.

**void getGres(void):**

No devuelve ni recibe nada, calcula la resolución del Giroscopo.

**void complementaryFilter(float\* pitch, float\* roll)**

No devuelve y recibe como argumentos dos punteros float destinados a almacenar los angulos, se utiliza para aplicar un filtro complementario de las mediciones de los giróscopos y acelerómetros, tomados de los registros de la IMU, los valores del giro de pitch como de roll son almacenados en dos punteros float.

**void calibrate(float\* dest1, float\* dest2)**

No devielve nada y recibe como argumentos dos punteros float destinados a albergar los registros de bias, estos son dos valores que se restaran constantemente y se obtienen gracias a esta calibración inicial de la IMU.

**void** **readAccelData**(int16\_t\* dest):

No devuelve y recibe como argumento un puntero a entero destinado a almacenar la lectura de los tres registros de los acelerómetros AX,AY y AZ.

**void** **readGyroData**(int16\_t\* dest)

Igual que la function de lectura de los acelerometros pero en este caso para los giroscopios GX,GY y GZ.

**void** **reset**(void)

No devuelve ni recibe nada y su función como indica es resetear la IMU.

**void** **init**(void)

No devuelve ni recibe nada, su función es cargar la configuración inicial de la IMU como ser despertar clareando el bit de sleep, tomar una base de tiempo estable cargar los registros de los acelerómetros y giróscopos para una frecuencia de sampling y ancho de banda determinados.

**void** **whoAmI**(void):

No devuelve ni recibe nada, es un testeo para verificar que se recibe 0x68 al consultar a la IMU y ver que la IMU esta online para operar.

**void** **writeByte**(uint8\_t address, uint8\_t subAddress, uint8\_t data):

No devuelve y recibe la dirección de la IMU (MPU6050\_ADDRESS) o dispositivo a trabajar, el registro a usar y la mascara, su función es escribir los registros de la IMU.

**void** **readByte**(uint8\_t address, uint8\_t subAddress, uint8\_t data):

ídem anterior pero para realizar una lectura de los registros.

**void readBytes(uint8\_t address, uint8\_t subAddress, uint8\_t byteNum, uint8\_t\* dest):**

No devuelve nada y recibe como argumentos, la direccion del dispositivo, el registro a trabajar, un entero y un puntero a entero. El propósito de esta función es leer datos de la IMU para su posterior promediado y parseo para saber los datos de cada registro como ser AX, AY,AZ, GX,GY y GZ.

**int** **botones**(**void**):

Rutina que se encarga de muestrear los botones B3 y B4.

Retorna:

0 -> no hay botón

1 -> se pulsó B3

2 -> se pulsó B4

3 -> se pulsó B3 y B4

# 9 ESTUDIO DE MERCADO:

Existen 3 niveles de equipos en lo que respecta a instrumentos digitales:

- Equipos de gama baja.

- Equipos de gama media.

- Equipos de gama alta.

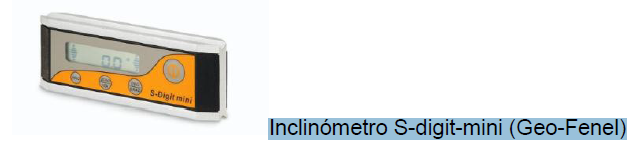
**Equipos de gama baja:**

Son equipos de bajo costo y baja precisión, son por lo general utilizados para obtener soluciones rápidas dentro de algún ambiente de taller. Se pudo observar que en el mercado no hay disponibles inclinómetros de esta gama o son difíciles de conseguir.

**Equipos de gama media:**

Son equipos de un costo intermedio y su campo de aplicación es en talleres, pequeños laboratorios y ámbitos académicos.

Ej: S-digit-mini y EL821 de la firma Geo-Fennel.





**Equipos de gama alta:**

Son equipos de costo elevado y alta precisión de marcas mundialmente conocidas (ej. Bosch). Son utilizados en grandes laboratorios, empresas y universidades. Su costo es elevado y se los puede conseguir en el mercado. Ej. Dnm 60l de la firma Bosch.



# 10 ESPECIFICACIONES DEL EQUIPO:

Este equipo utiliza un redondeo del ángulo adquirido de la IMU de la siguiente forma:

- Si la parte decimal es > 0.5º: Ej.: 15.7º, entonces tomamos como ángulo resultante: 16º,

- Si la parte decimal es < 0.5º: Ej.: 15.3º, entonces tomamos como ángulo resultante: 15º

Esto en el código lo especificamos de la siguiente manera (rollAngle es el ángulo obtenido por la IMU luego de aplicarle el filtro complementario)

angulo\_ENTERO=(**int**)rollAngle;

angulo\_partedecimal=(**float**)(rollAngle-angulo\_ENTERO);

**if** (fabs(angulo\_partedecimal>0.5))

angulo\_ENTERO = angulo\_ENTERO + 1;

Características del equipo:

* Dimensiones: 165 mm (L), 107 mm (W), 65 mm (D)
* Alimentación: fuente externa 12-20Vcc (es posible alimentarlo también a batería).
* Peso aproximado: 500g
* Rango angular: 0° - 90° (se puede medir un anguo mayor a los 90° pero no se obtuvieron buenos resultados superado este rango por lo que no se aconseja superar los 90° para tareas de mayor precisión)
* Incertidumbre: ± 1.4376º (el equipo redondea a un valor entero pero esta es la incertidumbre máxima del equipo para el rango detallado)
* Conectividad: USB a Serie
* Display: LCD 16 caracteres x 2 líneas
* Temperatura de operación: 0ºC – 50ºC

# 11 CONCLUSIONES:

En base a todo lo expuesto en los puntos anteriores, nuestro instrumento podría a llegar a incorporarse en el mercado argentino como un instrumento de gama baja debido a su bajo costo y precisión. Sin embargo realizado pequeñas modificaciones en el software se podría mejorar la resolución del mismo existiendo la posibilidad de incorporarlo al mercado como un instrumento de gama media sin variar los costos (se deben realizar las mediciones correspondientes para corroborar este punto).

NOTA IMPORTANTE CON RESPECTO A LA RESOLUCION QUE INDICA EL EQUIPO:

Con respecto a la resolución, en el código, notar que seleccione por default lo siguiente para el acelerómetro y para el Giroscopio:

**int** Ascale = *AFS\_2G*;

**int** Gscale = *GFS\_250DPS*;

con lo cual Al inicializar el sensor, los RANGOS por defecto serán:

- acelerómetro -2g a +2g

- giroscopio: -250°/sec a +250°/sec

Teniendo en cuenta que la RESOLUCION de las lecturas es de 16 bits (valor fijo especificado por el fabricante) por lo que el rango de lectura es de -32768 a 32767.

Cada lectura que hago del ángulo roll (rollAngle), luego de aplicado el filtro complementario, posteriormente la redondeo a un valor entero tal como indique en especificaciones del equipo (pag.20).

Es posible sacar este redondeo para tener una precisión aun mayor, de punto flotante, si el usuario así lo desea sacando las líneas de código siguientes, que se encuentran en la tarea:

“**static** **void** **vINCLINOMETRO**(**void** \*pvParameters)”

PARTE DEL CODIGO QUE SE PUEDE BORRAR PARA NO REDONDEAR A ENTERO:

angulo\_ENTERO=(**int**) rollAngle;

angulo\_partedecimal=(**float**)rollAngle-angulo\_ENTERO;

**if** (fabs(angulo\_partedecimal>0.5))

angulo\_ENTERO = angulo\_ENTERO + 1;

# 12 BIBLIOGRAFIA:

Display Winstar:

http://pdf.datasheetcatalog.com/datasheets2/29/295556\_1.pdf

Manual de usuario LPC17xx - Rev.3.1 — 20140402

LPCXpresso LPC17xx schematic Rev.B 2011-02-11

LPCXpresso - Getting started Rev.12 2013-04-17

LPCOpen Software Development Platform LPC17XX

http://www.nxp.com/products/software-and-tools/hardware-development-tools/lpcxpresso-boards/lpcopen-software-development-platform-lpc17xx:LPCOPEN-SOFTWARE-FOR-LPC17XX?tab=Design\_Tools\_Tab

IMU MPU6050:

<https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>

Libreria de Kris Winer para MPU6050

<https://github.com/kriswiner/MPU-6050/blob/master/MPU6050IMU.ino>