

Construcción de un guante de realidad virtual de bajo costo para su uso en el desarrollo de realidad digital.

M.A. López-Olvera¹

Materia: Microcontroladores

Asesor: Rafael Guadalupe Chávez Moreno²

¹Licenciatura en Tecnología, Escuela Nacional de Estudios Superiores, UNAM, Querétaro, México.

²Laboratorio HiL & SiL, Unidad de Alta Tecnología, UNAM, Querétaro, México.

¹Alexander_105@comunidad.unam.mx

Contenido

| | |
|--|----|
| Contenido | 1 |
| INTRODUCCIÓN | 2 |
| PLANTEAMIENTO DEL PROBLEMA | 2 |
| Preguntas de investigación | 2 |
| Hipótesis | 2 |
| JUSTIFICACIÓN | 2 |
| ANTECEDENTES Y FUNDAMENTACIÓN TEÓRICA | 3 |
| OBJETIVOS | 5 |
| General | 5 |
| Específicos | 5 |
| METODOLOGÍA..... | 6 |
| Materiales | 6 |
| Justificación de los materiales..... | 6 |
| Diagrama de flujo | 7 |
| Código | 7 |
| Prototipo | 12 |
| RESULTADOS | 12 |
| CONCLUSIONES | 15 |
| REFERENCIAS..... | 15 |
| Apéndice A. Opciones en el mercado..... | 16 |
| Apéndice B. Dispositivos de realidad digital..... | 18 |

INTRODUCCIÓN

En este proyecto se construyó un guante de realidad virtual que servirá como base teórica y práctica para la posterior construcción y adaptación de un casco de realidad virtual, el guante desarrollado y presentado en este documento es capaz de comunicarse con los dispositivos móviles por medio de conexión bluetooth y su uso primordial es como un mando de realidad virtual, además, será adaptable para comunicarse con el casco de realidad digital que se desarrollará en futuros proyectos. En el documento actual se puede encontrar el procedimiento seguido para el desarrollo del primer prototipo de guante de RV, los resultados, así como las conclusiones de su desarrollo.

PLANTEAMIENTO DEL PROBLEMA

Preguntas de investigación

¿Cuáles son los dispositivos y materiales necesarios para el desarrollo del proyecto?
¿Cuáles son las adaptaciones y mejoras que se le pueden realizar a los componentes de bajo costo utilizados en este proyecto?

No todos tienen acceso a tecnologías de realidad virtual, ¿qué podemos hacer nosotros para fomentar el acercamiento por parte de las universidades y los estudiantes a este tipo de tecnologías?

Hipótesis

Se logrará desarrollar un dispositivo de realidad virtual de bajo costo con características similares a las de los productos actuales en el mercado, facilitando de esta forma su uso por parte de los estudiantes interesados en el desarrollo de realidad virtual, aumentada y mixta. La correcta construcción y funcionamiento del guante logrará que un mayor número de estudiantes, investigadores y personas en general se sientan atraídos al desarrollo de realidad digital en cualquiera que sea su campo.

JUSTIFICACIÓN

Si bien para el desarrollo de realidad virtual y aumentada es suficiente contar con un computador capaz de procesar softwares como Unity y Blender, para lograr una visualización inmersiva y con mayor interacción por parte del usuario, es necesario contar con cascos de realidad digital. Hoy en día existe una gran variedad de dispositivos en el mercado que van desde muy baja, a muy alta gama, dependiendo de las necesidades del usuario, desafortunadamente en la mayoría de los dispositivos, su costo de adquisición está directamente relacionado con el total de funcionalidades con las que cuente dicho casco, haciendo que sea casi imposible para la mayoría de las personas adquirir un dispositivo de gama media/alta.

En este proyecto se pretende desarrollar un guante de realidad digital y establecer conexión con dispositivos móviles para su uso en la construcción de un casco de gama media para obtener un número mayor de funcionalidades por parte del usuario, haciendo uso de componentes de bajo costo.

ANTECEDENTES Y FUNDAMENTACIÓN TEÓRICA

a. Acelerómetros

Los acelerómetros son dispositivos electromecánicos que detectan las fuerzas de aceleración, ya sea estática o dinámica. Las fuerzas estáticas incluyen la gravedad, mientras que las fuerzas dinámicas pueden incluir vibraciones y movimiento.

Los acelerómetros pueden medir la aceleración en uno, dos o tres ejes. Los de tres ejes son más comunes conforme los costos de producción de estos baja.

Generalmente, los acelerómetros contienen placas capacitivas internamente. Algunos de estos son fijos, mientras que otros están unidos a resortes minúsculos que se mueven internamente conforme las fuerzas de aceleración actúan sobre el sensor. Como estas placas se mueven en relación el uno al otro, la capacitancia entre ellos cambia. A partir de estos cambios en la capacitancia, la aceleración se puede determinar.(Arun Faisal et al., 2019)

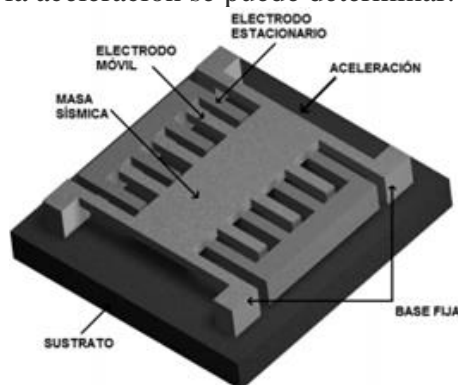


Fig. 1. Sistema microelectromecánico para la aceleración en 1 eje.

Para la mayoría de los acelerómetros, las conexiones básicas que se requieren para la operación son el poder y las líneas de comunicación.

Los acelerómetros se comunican a través de un convertidor analógico, digital, o interfaz de conexión modulada por ancho de pulsos (PWM).

Los acelerómetros con una interfaz analógica entregan un voltaje proporcional a la aceleración en cada uno de sus ejes (hablando de uno de 3 ejes) que normalmente fluctúan entre tierra y el valor de alimentación V_{cc} . En el caso de los que cuentan con una interfaz digital pueden comunicarse a través de los protocolos de comunicación de SPI o I2C. Estos tienden a tener más funcionalidad y ser menos susceptibles al ruido que acelerómetros analógicos.

Un acelerómetro también se puede utilizar para detectar el movimiento. Por ejemplo, un acelerómetro en el WiiMote de Nintendo se puede utilizar para detectar golpes de derecha y de revés emulando los golpes de una raqueta de tenis, o el efecto que se le pone a una bola

de boliche. Por último, un acelerómetro también se puede usar para detectar si un dispositivo está en un estado de caída libre. Esta característica se implementa en varias unidades de disco duro: si esto se detecta manda la señal al Disco Duro para guardar datos y bloquearse para evitar que el golpe dañe los datos.(Anonimo, 2021)

El acelerómetro de 3 ejes del MPU-60X0 utiliza masas de prueba separadas para cada eje. La aceleración a lo largo de un eje particular induce el desplazamiento de la masa de prueba correspondiente y los sensores capacitivos detectan el desplazamiento de forma diferencial. La arquitectura del MPU-60X0 reduce la susceptibilidad de los acelerómetros a las variaciones de fabricación, así como a la deriva térmica.

Cuando el dispositivo se coloca sobre una superficie plana (horizontal), medirá 0 g en los ejes X e Y y + 1 g en el eje Z. El factor de escala de los acelerómetros se calibra en la fábrica y es nominalmente independiente de la tensión de alimentación. Cada sensor tiene un ADC sigma-delta dedicado para proporcionar salidas digitales. El rango de escala completa de la salida digital se puede ajustar a $\pm 2g$, $\pm 4g$, $\pm 8g$ o $\pm 16g$. Tenga en cuenta que: + 1g es igual a $9,8 \text{ m/s}^2$, que es la aceleración gravitacional.(PANAMAHITEK, 2020)

b. Giroscopio

El giroscopio es un componente electrónico que permite medir el ángulo de rotación de un dispositivo en todos sus ejes. Este sensor de movimiento ayudará al acelerómetro a mantener la orientación del casco y guante de realidad virtual.

La utilización de este elemento es muy común en juegos donde al inclinar algún dispositivo, controlas la acción. En la realidad aumentada se basa en la proyección de una imagen falsa sobre una escena real, mientras que, en la realidad virtual, los smartphones o gafas de RV pueden construir una realidad paralela en las que el giroscopio juega un papel vital. En este caso podremos apreciar el auténtico valor de este elemento, ya que a través de la realidad virtual es posible recrear un mundo totalmente inventado en el cual nos podremos desplazar gracias a la ayuda del giroscopio.

El MPU-60X0 consta de tres giroscopios de velocidad MEMS vibratorios independientes, que detectan la rotación alrededor de los ejes X, Y y Z. Cuando los giroscopios giran alrededor de cualquiera de los ejes de detección, el efecto Coriolis provoca una vibración que es detectada por un pickoff capacitivo. La señal resultante se amplifica, demodula y filtra para producir un voltaje que es proporcional a la velocidad angular. Este voltaje se digitaliza mediante convertidores analógicos a digitales (ADC) de 16 bits integrados en el chip para muestrear cada eje.(Cholakian, 1979)

El rango de escala completa de los sensores giroscópicos se puede programar digitalmente a ± 250 , ± 500 , ± 1000 o ± 2000 grados por segundo (dps). La frecuencia de muestreo del ADC se puede programar desde 8.000 muestras por segundo hasta 3,9 muestras por segundo, y los filtros de pasa baja seleccionables por el usuario permiten una amplia gama de frecuencias de corte.(Arun Faisal et al., 2019)

c. Magnetómetro

Se les dice magnetómetros a los dispositivos que sirven para cuantificar en fuerza o dirección la señal magnética de una muestra.

El magnetómetro sirve para que el teléfono inteligente o la tablet se enciendan o apaguen, al acercársele o alejarle un imán, para así encender y apagar automáticamente la pantalla del dispositivo cuando se abra o se cierre su funda, sin necesidad de pulsar ningún botón. Por otro lado, la mayoría de los relojes inteligentes, teléfonos inteligentes y tabletas tienen un sensor integrado que mide el campo magnético de la Tierra y pueden situar dónde está el norte, como si fueran una brújula (geo magnetómetro).

Además de esta función, algunas aplicaciones pueden convertir el magnetómetro de un reloj inteligente o móvil en un detector de metales, Pueden hacer que el magnetómetro responda ante metales como el níquel, el hierro y el acero, y podría ser útil para determinar si hay cables detrás de una pared o incluso para encontrar objetos perdidos detrás de un lugar de difícil acceso.

d. Sensor de proximidad

Los HMD (Head-Mounted display) más avanzados se venden acompañados de unos dispositivos conocidos como sensores de posición que, colocados en la habitación, permiten al sistema determinar la ubicación del HMD y de otros periféricos que pueda portar el usuario, dándole así a este la posibilidad de moverse libremente en el espacio virtual a escala.

Entre los más conocidos están el Lighthouse utilizado por las gafas HTC Vive, o el Constellation usado por las Oculus Rift. Compatible con otros sistemas es Nolo VR, un sistema de seguimiento de posición para visores de móvil que se compone de una estación base, un marcador para el visor, y dos mandos, y es compatible con juegos de Steam VR (permite el uso de juegos de realidad virtual en un equipo personal a partir de su plataforma).

OBJETIVOS

General

Adaptar un casco de realidad digital de gama media haciendo uso de componentes de bajo costo, para obtener un número mayor de funcionalidades por parte del usuario y utilizarlo en el desarrollo de realidad digital.

Específicos

- Comprender la programación y los protocolos de comunicación de microcontroladores, comenzando con el desarrollo de un mando de realidad virtual para su comunicación con un teléfono móvil y su interacción en aplicaciones de realidad digital.

METODOLOGÍA

Materiales

- Acelerómetro
- Giroscopio
- 2 sensores infrarrojos FC-51
- Módulo Bluetooth
- PIC16f877A
- Regulador de voltaje
- Batería de 9V recargable

Justificación de los materiales

Seguimiento de la cabeza

Las gafas de realidad virtual necesitan algún tipo de seguimiento para poder funcionar. Y es que esa es precisamente la idea; que la imagen se adapte en función de cómo nos movamos. Para esto se requiere de un chip capaz de rastrear la rotación. Contamos con dos estilos de seguimiento: el rastreo de la cabeza y el de la posición. Para realizar el rastreo de la cabeza se utilizan sensores comunes como el giroscopio, el acelerómetro o el magnetómetro. Algunas formas de detectar el posicionamiento de cabeza y brazos se realizan mediante sistemas de seguimiento con el uso de leds (como lo hacen Oculus Rift y Playstation) o con el uso de cajas colocadas en la pared formando un ángulo de $90^{\circ}2$, en donde las cajas contienen leds y emisores láser (como lo hace Vive con su sistema llamado Lighthouse).

Rastreo de movimiento

Control que consistirá en un guante de realidad virtual para dar la sensación al usuario de que se están sujetando los objetos visualizados. Se deberá utilizar acelerómetro y giroscopio para detectar los cambios de posición y de giro del mando.

Controles

Se usarán dos sensores infrarrojos que detecten los movimientos del dedo índice y central por parte del usuario, dichos controles se encargarán de realizar distintas acciones dentro de las aplicaciones de realidad virtual a desarrollar en futuros proyectos.

Diagrama de flujo

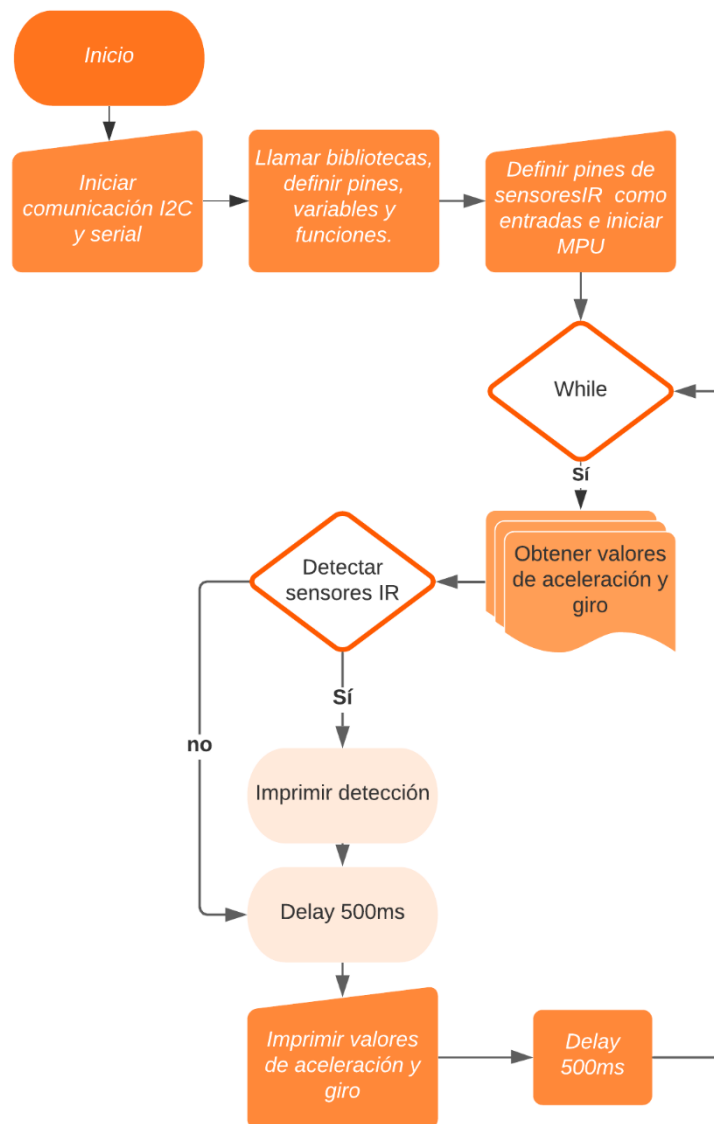


Fig. 2 Diagrama de flujo para la programación del microcontrolador del guante de realidad virtual.

Código

- Archivo de registros del MPU6050 (MPU_REGISTERS.h)

En este archivo se encuentran algunos de los registros que se ocuparon para la configuración y extracción de datos del MPU6050, lo importante que hay que notar de los registros es existen dos bits de para cada valor de registro interno, HIGH y LOW. Además, en este documento agregamos variables con las direcciones para la escritura y lectura en la comunicación I2C. Debemos notar que la dirección en 7 bits para el MPU6050 es 0x68, pero al añadir un bit de escritura/lectura, se vuelven 0xD0 y 0xD1, respectivamente.

| | | |
|----|---------------|------|
| 27 | #define w_Add | 0xD0 |
| 28 | #define r_Add | 0xD1 |

Fig. 3. Variables de escritura/lectura para el mpu6050.

- Archivo de funciones del MPU6050 (MPU_FUNCTIONS.c)

En este archivo es necesario comenzar llamando a los registros previamente definidos. Una vez hecho eso procedemos creando funciones para cada una de las acciones que queramos realizar en nuestro código. Se definieron un total de 10 funciones, pero solo se describirán un total de 4, pues las demás funciones se basan en la misma idea a una que será descrita en las siguientes líneas:

a. mpu_write

Esta función recibe 2 valores, la dirección a la que queremos escribir y el dato que queremos escribir en esa dirección.

Comienza iniciando el protocolo i2c con la función i2c_start (), después usamos la función i2c_write() 3 veces, una para llamar a la dirección de escritura del mpu, otra para agregar a la dirección deseada por el usuario y la otra para escribir el dato que el usuario ingresó, por último, usa la función i2c_stop() para parar la comunicación i2c.

```
void mpu_write(int add, int data){
    i2c_start();
    i2c_write(w_Add);
    i2c_write(add);
    i2c_write(data);
    i2c_stop();
}
```

Fig. 4. Función mpu_write en el archivo de funciones.

b. mpu_init

En esta función se comienza reiniciando el mpu con el uso de la función mpu_write, definida previamente, posteriormente se selecciona el oscilador interno del mpu como fuente de reloj, se prosigue seleccionando el valor deseado para el filtro pasa baja digital con ayuda del registro CONFIG, por último, se selecciona el rango de escalación del giroscopio con el registro GYRO_CONFIG.


```

void mpu_init(){
    mpu_write(PWR_MGMT_1, 0x80); //reseteamos el dispositivo
    delay_ms(100);
    mpu_write(PWR_MGMT_1, 0x00); //Internal 8MHz oscillator
    delay_ms(100);
    mpu_write(CONFIG, 0x01); //Frame Synchro=input data rate
    delay_ms(10);
    mpu_write(GYRO_CONFIG, 0x00); //+/- 250°/s
}

```

Fig. 5. Función `mpu_init` en el archivo de funciones, esta función se llamará a comienzo del archivo `main` para establecer las configuraciones iniciales del mpu.

c. `mpu_read`

Esta función recibe como argumento la dirección que se desea leer del mpu y se define una variable que será regresada al usuario después de hacer la lectura. Se comienza usando la función `i2c_start()` para iniciar el protocolo i2c, se escribe la dirección de escritura del mpu, luego se escribe la dirección del mpu a leer con `i2c_write(<dirección>)`, posteriormente se inicia de nuevo el protocolo i2c, y se escribe la dirección de lectura del mpu, asignamos el valor leído a la variable “val” y mandamos un bit NACK (en lectura el maestro manda la señal de recibido y decide si quiere más información o no) para posteriormente terminar comunicación con un `i2c_stop()`.

```

int16 mpu_read(int add){
    int val;
    i2c_start();
    i2c_write(w_Add);
    i2c_write(add); //register that we want to read
    i2c_start();
    i2c_write(r_Add);
    val = i2c_read(0); //with NACK
    i2c_stop();
    return val;
}

```

Fig. 6. Función `mpu_read` en el archivo de funciones, con esta función leeremos los registros del mpu y extraeremos las variables deseadas.

d. `mpu_get_Acc_x`

Se define una variable con signo entera de 16 bits inicial “val” y una flotante `Acc_x` que contendrá nuestro valor de aceleración en el eje x. Usamos nuestra función `mpu_read()` con el registro `ACCEL_XOUT_H` como parámetro, recorreremos 8 bits a la izquierda nuestra variable `val` y le agregamos los siguientes 8 bits menos significativos con el registro `ACCEL_ZOUT_L`, por último dividimos nuestra variable “val” por 16384 porque hay que tener en cuenta que la escala del acelerómetro está configurada a 2g y la aceleración se da en una variable con signo de 16 bits (-32768 a +32768), además, la aceleración gravitacional es +1g, por lo que el sensor estará midiendo cerca de 16384 sobre el eje z cuando se encuentra en reposo. Entonces **esa es la razón para dividir por 16384** y

obtener un valor cercano a 1.

| AFS_SEL | Full Scale Range | LSB Sensitivity |
|---------|------------------|-----------------|
| 0 | $\pm 2g$ | 16384 LSB/g |
| 1 | $\pm 4g$ | 8192 LSB/g |
| 2 | $\pm 8g$ | 4096 LSB/g |
| 3 | $\pm 16g$ | 2048 LSB/g |

Fig. 7. Sensibilidades por LSB del acelerómetro para cada rango de escala completa.

```
float mpu_get_Acc_x(){
    signed int16 val; //we are going to obtain values from -32768 to +32768
    float Acc_x;
    val = mpu_read(ACCEL_XOUT_H);
    val = (val << 8) + mpu_read(ACCEL_XOUT_L);
    Acc_x = (float)val/(float)16384;
    return Acc_x;
}
```

Fig. 8. Función `mpu_get_Acc_x` en el archivo de funciones, con esta función extraemos el valor para la aceleración en el eje X.

e. `mpu_get_Gyr_x`

Se define una variable con signo entera de 16 bits inicial “val” y una flotante Gyr_x que contendrá nuestro valor de rotación en el eje x. Usamos nuestra función `mpu_read()` con el registro `GYRO_XOUT_H` como parámetro, recorremos 8 bits a la izquierda nuestra variable `val` y le agregamos los siguientes 8 bits menos significativos con el registro `GYRO_XOUT_L`, por último dividimos nuestra variable “val” por 131 porque hay que tener en cuenta que seleccionamos el rango de escala completa de ± 250 °/s y eso nos dirige a una sensibilidad de 131 bits menos significativos por grado por segundos o 131 LSB/°/s.

| FS_SEL | Full Scale Range | LSB Sensitivity |
|--------|------------------|-----------------|
| 0 | ± 250 °/s | 131 LSB/°/s |
| 1 | ± 500 °/s | 65.5 LSB/°/s |
| 2 | ± 1000 °/s | 32.8 LSB/°/s |
| 3 | ± 2000 °/s | 16.4 LSB/°/s |

Fig. 9. Sensibilidades por LSB del giroscopio para cada rango de escala completa.

```
float mpu_get_Gyr_x(){
    signed int16 val;
    float Gyr_x;
    val = mpu_read(GYRO_XOUT_H);
    val = (val << 8) + mpu_read(GYRO_XOUT_L);
    Gyr_x = (float)val/(float)131; //the LSB S
    return Gyr_x;
}
```

Fig. 10. Función `mpu_get_Gyr_x` en el archivo de funciones, con esta función extraemos el valor para el giro en el eje X.

- Archivo principal (`main.c`).

Por último, el archivo principal `main.c` comienza haciendo uso de las bibliotecas y protocolos de comunicación, en este caso, se usará el protocolo I2C para comunicarse con el esclavo (`mpu6050`) y el protocolo UART con ayuda del controlador de línea RS-232 (para resistir interferencias) para la transmisión y recepción de datos a través de la terminal serial. Es importante recordar llamar a la biblioteca de las funciones del MPU y definir una función que se encargará de detectar los movimientos de los dedos índice y medio. Los siguientes pasos en este archivo son intuitivos y ya se incluyó el diagrama de flujo previamente, por lo que se omitirá la descripción de cada paso y simplemente se añadirán capturas de pantalla del código omitiendo las impresiones de las medidas.

```
void Finger_Detection();
void accel_and_gyro_detections();
float Acc_x, Acc_y, Acc_z, Gyr_x, Gyr_y, Gyr_z;

void main(void){
    SET_TRIS_B(0XC0); //IR PINS LIKE INPUTS
    //lcd_init();
    mpu_init();
    while(1){
        Acc_x = mpu_get_Acc_x();
        Acc_y = mpu_get_Acc_y();
        Acc_z = mpu_get_Acc_z();
        Gyr_x = mpu_get_Gyr_x();
        Gyr_y = mpu_get_Gyr_y();
        Gyr_z = mpu_get_Gyr_z();
```

Fig. 11. Primera parte del archivo `main.c`, omitiendo la llamada de los protocolos y bibliotecas.

```
void Finger_Detection(){
    //lcd_putc('\f');
    if(input(IR_1) == 0){
        //lcd_gotoxy(1,1);
        //printf(lcd_putc, "ForefingerActive");
        printf("\n\rForefinger Active");
    }
    if(input(IR_2) == 0){
        //lcd_gotoxy(1,2);
        //printf(lcd_putc, "MidfingerActive");
        printf("\n\rMiddle Finger Active");
    }
    delay_ms(500);
}
```

Fig. 12. Función encargada de detectar el movimiento de los dedos índice y medio, su estructura es básica y únicamente evalúa con condicionales “if()” si el usuario ha movido sus dedos a una posición en la cuál la luz infrarroja emitida por el mismo sensor IR ha sido detectada.

Además, se creó una función extra para imprimir detecciones de aceleración y giro, no se adjunta por ser de muy baja complejidad, pero se puede encontrar en el archivo `main.c` como “`accel_and_gyro_detections`”.

Prototipo

Una vez implementado el código en c para la comunicación del microcontrolador con el MPU6050 y el envío de los datos por la terminal serial, se comenzó a realizar el primer prototipo del proyecto, el cuál fue realizado de forma rústica y con el uso de los componentes disponibles fuera de un laboratorio, los pasos en el desarrollo del prototipo se pueden enumerar en el siguiente orden:

- 1) Prueba del funcionamiento y envío de los datos.
- 2) Cortado de la protoboard a la medida exacta para la conexión de los componentes y su correcta adhesión al guante.
- 3) Conexión de los componentes a la protoboard.
- 4) Pegado de la protoboard al guante de realidad virtual
- 5) Adhesión de los sensores infrarrojos al guante de RV.
- 6) Conexión del regulador de voltaje a 5V.
- 7) Integración de la fuente de voltaje (batería de 9V) al circuito.
- 8) Pruebas de funcionamiento del prototipo completo
- 9) Corrección de fallos

RESULTADOS

- Prototipo



Fig. 13. Primer prototipo de guante de realidad virtual.

- Extracción de datos

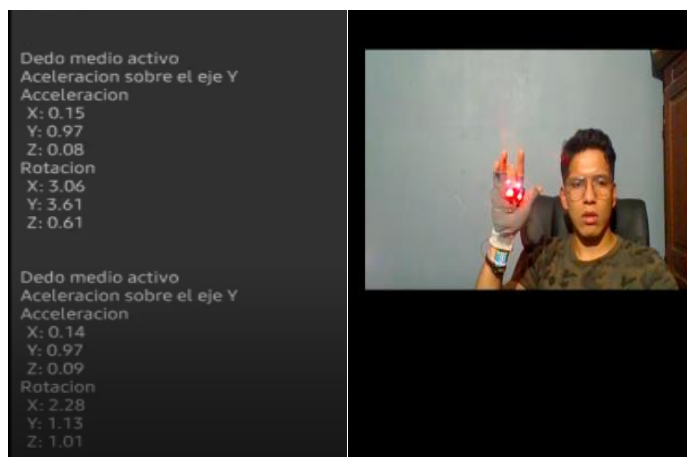


Fig. 14. Detección del dedo medio.

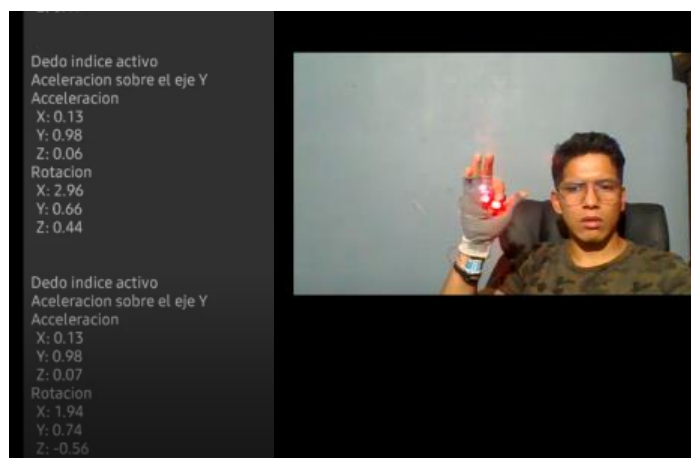


Fig. 15. Detección del dedo índice.



Fig. 16. Aceleración sobre el eje Z ≈ 1 (gravedad).



Fig. 16. Desaceleración sobre el eje Z ≈ -1 (gravedad).

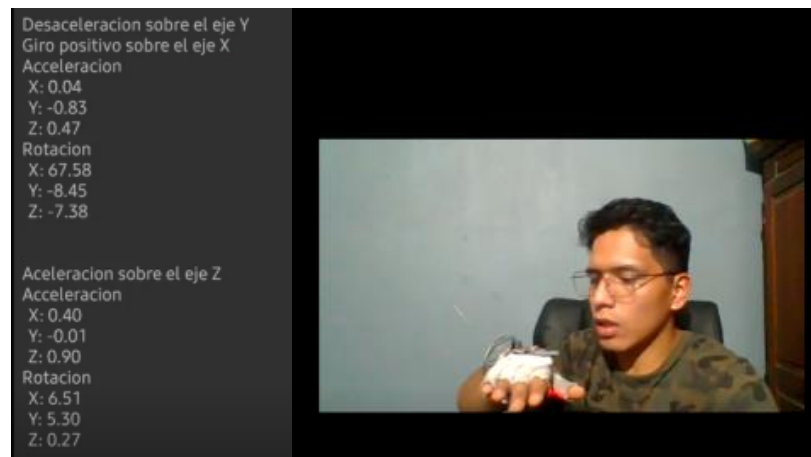


Fig. 17. Detección de giro positivo sobre el eje X.

Las demás medidas son realizadas de la misma forma sobre todos los ejes y para poder apreciar los giros será necesario observarlo en video.

La aceleración que afecta al eje z es la aceleración gravitacional = $+1g = 9.81m / s^2$ o en el mundo digital (en cuanto a la salida del sensor) es ± 16384 .

CONCLUSIONES

Es necesario que el estudiante disponga de bases teóricas durante su formación, pero también es indispensable aplicar sus conocimientos en las diferentes áreas del conocimiento, ese es el poder que tiene la materia de microcontroladores, ya que puede ser aplicada prácticamente a cualquier área siempre y cuando el estudiante sea capaz de generar ideas innovadoras.

En este proyecto se ha logrado construir un guante de realidad virtual que se usará en el desarrollo de realidad digital y que es capaz de comunicarse con dispositivos móviles, así como cualquier otro dispositivo que cuente con bluetooth, lo cual es indispensable para su uso en conjunto con el casco de realidad virtual que se pretende desarrollar en proyectos venideros.


Se espera que la construcción del guante de realidad virtual desarrollado en este proyecto, en conjunto con la adaptación del casco que se realizará en un proyecto futuro pueda ser reproducido y empleado por universidades en donde las posibilidades de adquisición de productos de alto costo no son las mejores, así como también para que un mayor número de estudiantes, investigadores y personas en general se sientan atraídos al desarrollo de realidad digital en cualquiera que sea su área de interés.

REFERENCIAS

- Anonimo. (2021). *¿Qué es y cómo funciona un acelerómetro?* <https://comofuncionahoy.com/que-es-un-acelerometro/>
- Arun Faisal, I., Waluyo Purboyo, T., & Siswo Raharjo Ansori, A. (2019). A Review of Accelerometer Sensor and Gyroscope Sensor in IMU Sensors on Motion Capture. *Journal of Engineering and Applied Sciences*, 15(3), 826–829. <https://doi.org/10.36478/jeasci.2020.826.829>
- Cholakian, R. C. (1979). MPU-6000 and MPU-6050 Product Specification Revision 3.4. *French Studies*, 33(1), 1–12. <https://doi.org/10.1093/fs/XXXIII.1.1>
- PANAMAHITEK. (2020). *Acelerómetros de 3 ejes, lo que necesitas saber*. <http://panamahitek.com/acelerometros-de-3-ejes-lo-que-necesitas-saber/>

Apéndice A. Opciones en el mercado

Modulo Bmx055


| | |
|-------------------------|---|
| Características clave |  |
| 3 sensores | Giroscopio de 16 bits Acelerómetro de 12 bits Sensor geomagnético |
| Paquete | 20 pines 3.0x4.5mm, 0.95mm alto |
| Voltaje común | 2.4V a 3.6V |
| Interfaz digital | I^2C |
| Operación e integración | Individual, 9 ejes |
| Precio | \$175- \$319 |

Modulo Mpu9250

| | |
|-----------------------|--|
| Características clave |  |
| 3 sensores | Giroscopio de 3 ejes Acelerómetro de 16 bits, 3 ejes brújula digital de 3 ejes Además, incluye un procesador de movimiento digital (DMP), que descarga el cálculo de los algoritmos de procesamiento de movimiento desde el procesador host |
| Paquete | 20 pines |

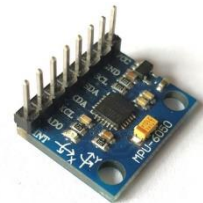
| | |
|-------------------------|--------------------|
| | 3x3x1mm |
| Voltaje común | 2.4V a 3.6V |
| Interfaz digital | I^2C o SPI |
| Operación e integración | Individual, 9 ejes |
| Precio | \$134- \$476 |

Módulo BMX160

| | |
|-------------------------|--|
| Características clave |  |
| 3 sensores | Giroscopio de 16 bits Acelerómetro de 16 bits Sensor geomagnético Además, cuenta con interruptor para detección de movimiento autónoma mejorada |
| Paquete | 20 pines 2.5x3.0mm, 0.95mm alto |
| Voltaje común | 1.71V a 3.6V |
| Interfaz digital | I^2C o SPI |
| Operación e integración | Individual, 9 ejes |
| Precio | \$164, difícil de encontrar |

Módulo Mpu6050

- Valores combinados del acelerómetro MEMS 3-axes y del giroscopio de 3 ejes
- Fuente de alimentación: 3-5 V
- Comunicación: Protocolo I2C
- El ADC de 16 bits incorporado proporciona alta precisión
- El DMP incorporado proporciona alta potencia computacional
- Se puede utilizar para interactuar con otros dispositivos IIC como magnetómetro



- Dirección IIC configurable
- Sensor de temperatura incorporado

Apéndice B. Dispositivos de realidad digital

Oculus Rift, realidad virtual.



Fig. 3 Oculus Rift (Oculus, 2020).

- Sensores: Acelerómetro, giroscopio, magnetómetro, seguimiento posicional de 360 grados
- Un conjunto de visor para VR con campo de visión de 110 grados.
- Par de controladores touch con seguimiento para una presencia intuitiva de las manos en RV.
- Resolución por ojo: 1080 x 1200
- Dos sensores de Rift que monitorizan constelaciones de LED infrarrojos para transformar los movimientos reales en virtuales.
- Precio: 260 €

Epson MOVERIO BT-300, realidad aumentada.



Fig. 4 Epson MOVERIO BT-300 (EPSON, 2020).

- El nuevo estándar en AR: pantalla Si-OLED, que permite una experiencia transparente completa y capacidad AR 3D

- Calidad de imagen asombrosa: la pantalla HD (720p) y el alto brillo garantizan una imagen nítida y colores vivos
- Cámara frontal HD: cámara frontal de 5 MP para tomar fotos y videos POV con calidad HD
- Rendimiento de vanguardia: CPU de cuatro núcleos a 1,44 GHz, 2 GB de RAM y una duración de la batería de hasta 6 horas
- Plataforma flexible: impulsada por Moverio OS de código abierto
- Precio: 849 €

Acer Mixed Reality, realidad mixta inmersiva.



Fig. 5 Acer Mixer Reality (Acer, 2020).

- Tamaño de pantalla: LCD 2.89" x 2.
- Densidad de pixel: 706 ppi.
- Resolución: 1440 x 1440 para cada ojo.
- Refresco: 90Hz usando HDMI 2.0, 60Hz usando HDMI 1.4.
- Sensores de seguimiento: acelerómetro, giroscopio, sensor de proximidad y magnetómetro.
- Cámara de seguimiento: Inside Out B+W VGA Camera.
- Conectividad: 1 cable 4 metros HDMI 1.4/2.0, 1 x USB 3.0, bluetooth 4.0.
- Campo de visión: 100 grados.
- Peso: 440g.
- Precio: 360 €.

Microsoft Hololens 2, realidad mixta holográfica.



Fig. 6 Microsoft Hololens 2 (Microsoft, 2020).

- Lentes transparentes holográficos.
- Resolución holográfica: 2,5 M puntos de luz.
- Interfaz de gestos, voz y mirada.
- Procesador Qualcomm Snapdragon 850.
- Sensores: acelerómetro, giroscopio, magnetómetro, 4 Cámaras de seguimiento, 1 cámara de profundidad y 1 cámara de foto 8MP y vídeo 1080p a 30 FPS, 5 micrófonos y altavoces integrados.
- Conectividad: Wi-Fi 802.11ac, USB tipo C, Bluetooth 5.0.
- Seguimiento 6DoF, mapeo espacial y captura de hologramas.
- Memoria Flash de 64 GB y 4 GB de RAM.
- Campo de visión: hasta 120 grados.
- Peso 566g.
- Precio: 3.500 \$ para empresas.