SISTEMAS ELECTRÓNICOS PARA LA AUTOMATIZACIÓN

Proyecto de microcontroladores

Grado en Ingeniería Electrónica, Mecatrónica y Robótica

Autores: Haes-Ellis, Richard Mark
Montes Grova, Marco Antonio

Índice

1. Introducción al proyecto

En el proyecto que se muestra a continuacion, se desarrollara un HID(*Human Interface Device*)el control del puntero de un host, en este caso un ordenador, haciendo uso del Boosterpack *BOOSTXL-SENSORS* y el microcontrolador *Tiva TM4C1294*, ambos del fabricante *Texas Instruments*.
En primer lugar, se realizara una introduccion al hardware empleado en el mismo para, posteriormente, abordar el software. Tras ello, en los siguientes puntos del proyecto, se trataran aspectos como el funcionamiento a alto nivel del proyecto como los codigos implementados en propio microcontrolador.
En un ultimo apartado se trataran futuras o posibles mejoras y sus posibles aplicaciones a la automatizacion industrial.

1.1. Descripcion del hardware empleado

Se ha empleado como placa de desarrollo del proyecto el modelo TM9C1294 de la serie Tiva C de Texas Instrument. Este microntrolador es el que se muestra a continuacion:

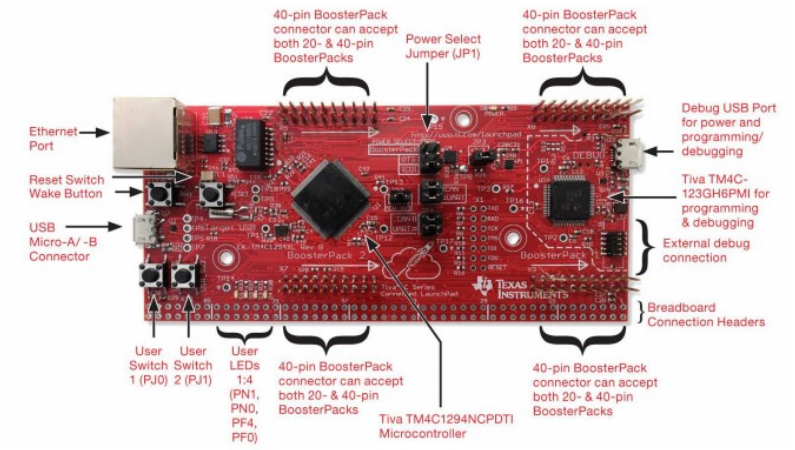

Figura 1: Microcontrolador empleado en el proyecto

Como se puede observar, ademas de poseer una conexion Ethernet, USB y una serie de leds, tendra un microcontrolador integrado, el Tiva TM123GH, conectado a un puerto USB cuya funcionalidad sera programar el microcontrolador principal.
Ademas de ello, dispone de dos pares de ristras de pines destinadas al conexionado de dos Boosterpacks distintos, de tal modo que se puedan ampliar las funcionalidades como pueden ser una pantalla o el boosterpack empleado en este proyecto, BOOSTXL-SENSORS.

Para conocer informacion adiccional sobre el microcontrolador implementado en la placa puede la guia propocionada por el fabricante: http://www.ti.com/lit/ug/spmu365c/spmu365c.pdf.

Se ha optado por el uso de este microcontrolador para este proyecto por su capacidad de ..... lo de poder controladr el raton.....

En cuanto al Boosterpack empleado, BOOSTXL-SENSORS, es el mostrado a continuacion:

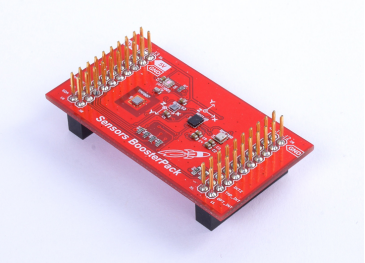

Figura 2: Boosterpack empleado en el proyecto

contiene una gran cantidad de sensores como son una IMU, un magnetometro, sensores ambiente y de luminosidad y sensor de temperatura. En lo que a este proyecto respecta, el principal sensor que se empleara sera la IMU integrada, la *BMI160* de 6 ejes, es decir, se encuentra formada por un acelerometro de 3 ejes y un giroscopio de 3 ejes.

La medida del acelerometro sera dada en *g*, la cual puede ser estimulada modificando la orientacion respecto a la gravedad de la tierra, o cambiando la velocidad a lo largo de un eje. En cuanto a la medida del giroscopio sera dada en grados por segundo, la cual se estimulara girando la placa respecto sus ejes absolutos.
En este proyecto, se emplearan las medidas asociadas a las velocidades angulares en torno a los ejes X y Z, ya que son las unicas empleadas para posicionarse en el plano 2D que forma la pantalla del computador. Se mostrara a continuacion una imagen en la cual se mostrara el movimento en torno al eje X que generara una variacion de la velocidad angular medida por el giroscpio:

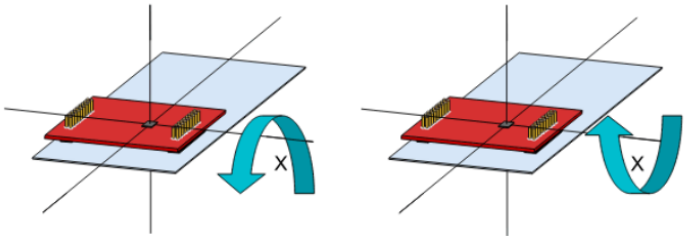

Figura 3: Movimento en torno al eje X que genera una variacion de velocidad angular

La variacion del angulo en torno al eje Z se medira del mismo modo, como se muestra a continuacion:

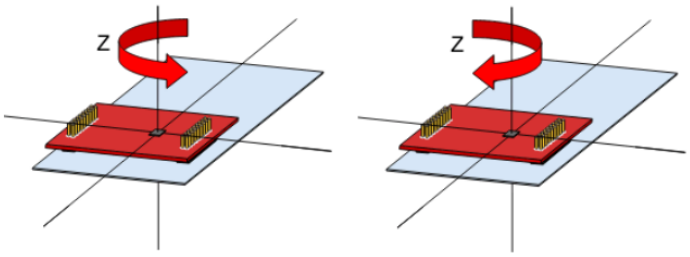

Figura 4: Movimento en torno al eje Z que genera una variacion de velocidad angular

La comunicacion del sensor con otros sensores del boosterpack y con la placa sera por medio de I^2C, el cual es el principal bus serie de datos, empleado para la comunicacion entre elementos de un circuito.

Al igual que antes, se puede consultar datos concretos de cada sensor en la guia proporcionada por el fabricante: http://www.ti.com/lit/ug/slau666b/slau666b.pdf

1.2. Descripcion del software empleado

En cuanto al software empleado se basara en la API proporcionada por el fabricante, *TivaWare* para la familia de microcontoladores TIVA. Esta API contiene los elementos que se muestran en la siguiente imagen:

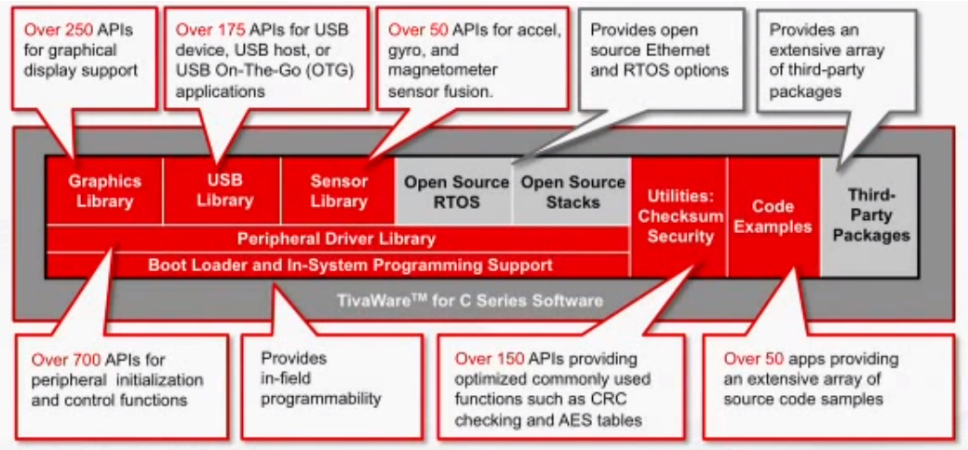

Figura 5: Estructura de la API Tivaware

Principalmente, se emplearan las liberias asociadas al manejo de los perifericos, *Peripheral Driver Library*, al manejo del puerto USB, *USB Library* y al manejo de sensores, *Sensor Library*.
Ademas de la gran ayuda brinda la API, se proporcionan una serie de ejemplos de uso que ayudaran al desarrollo del software propio.

Ademas de ello, ha sido necesario el uso de librerias para el manejo de los GPio de la placa y la comunicacion por el puerto serie de la UART.
Ademas de ello, se han empleado una serie de declaraciones proporciondadas por el profesor para solventar el desajuste de la actualizacion de las librerias, *driverlib2.h* y *sensorlib2.h*.

2. Funcionamiento del proyecto

En este apartado, se desarrollaran las principales funcionalidades implementadas en el proyecto. Las principales funciones son:

- Desarrollo de un HID con el microcontrolador.
- Empleo del Boosterpack para tomar la definir la posicion del puntero en la pantalla.
- Filtrado de las medidas tomadas a nivel de software.

Un HID(*Human Interface Device*) es una arquitectura de comunicacion empleada para comunicar los perifericos de interaccion humana como pueden ser ratones o teclados.
La comunicación entre el dispositivo HID y el host se realiza a través de un conjunto de estructuras de ïnformes"definidas por el dispositivo que el host puede consultar. Los informes se definen tanto para la comunicación de la entrada del dispositivo con el host y para la selección de salidas y funciones del host.

Además de la flexibilidad que ofrece la arquitectura básica, los dispositivos HID también se benefician de una gran universalidad entre sistemas operativos, lo que significa que no es necesario desarrollar un driver, sobre todo en el caso de dispositivos estándar como teclados y joysticks.
A pesar de estas ventajas, el uso de HID tiene un inconveniente. La tasa de datos que pueden transferirse esta limitada a un máximo de 64 KB/s.

Se empleara comunicacion mediante USB. El puerto USB de la familia Tiva TM4C de microcontroladores soporta 3 modos de funcionamiento:

- **Host mode**: Permite conectar un teclado o un raton al microcontrolador.
- **Device mode**:Establece una comunicacion con el PC a traves del USB.
- **On-The-Go mode**: Permite multiplexar el USB entre hosts y dispositivos.

En este proyecto se empleara el USB en modo *Device* o dispositivo, ya que se busca controlar el puntero del ordenador con el microcontrolador.
Durante el desarrollo del codigo de programacion se detallara el modo en el que se establece esta comunicacion entre el ordenador y el microcontrolador por medio del USB para crear el HID buscado.

Por otro lado, para controlar la posicion del puntero en la pantalla, se hara uso del giroscopio que, empleando la velocidad angular obtenida por el mismo, se podra estimar la posicion del puntero en el plano de la pantalla a partir de la definicion de un sistema inicial.
Como se mostro cuando se explico el boosterpack empleado que contiene el sensor, se tomaran las medidas de la velocidad angular en torno a los ejes X y Z.

3. Código de programación desarrollado

En primer lugar, es conveniente mistrar un esquema de las librerias y funciones empleadas en el proyecto. Ese esquema se muestra a continuacion:

Em el codigo principal, es decir, en *usb_ dev_ mouse.c*, en primer lugar se hara la declaracion de librerias. Las primeras librerias que se declararan son las asociadas a las funciones logicas del microntrolador.

```

```

Listing 1: Variables y defines del codigo

```
int i=1;
```

Listing 2: Declaración de librerías

```

```

Listing 3: Funciones empleadas

```

```

Listing 4: Programa principal

4. Posibles mejoras del proyecto

En cuando a la principal mejora del proyecto, se basara en la implementacion de una comunicacion inalambrica en el mismo. Durante el desarrollo del proyecto se plantearon diversas vias posibles de implementacion:

- Implementacion de una comunicacion basada en radio-frecuencia. Esta via se planteo empleando el boosterpack del fabricante *Texas Instrument*, CC110L, el cual emplea el protocolo de comunicacion SimpliciTI. Sin embargo, este modulo de radio frecuencia, ha sido disenado para su utilizacion con el microcontrolador MSP430, y es poco compatible con otros microcontroladores, aunque sea del mismo fabricante.

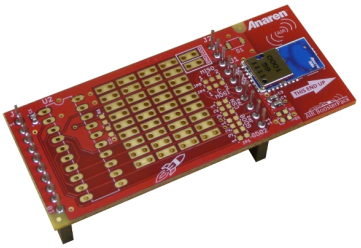

Figura 6: 430BOOST-CC110L Boosterpack

- Implementacion de una comunicacion Wi-Fi. Para la implementacion de este modo de comunicacion, se haria uso del modulo ESP01, el cual es un modulo Wi-Fi de bajo coste, que puede ser configurado como punto de acceso o como cliente y enviar mensajes TCP entre varios para comunicarse entre ellos.

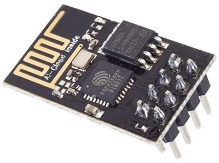

Figura 7: Modulo WiFi ESP01

Esta ultima via de comunicacion, es la que se ha considerado mas factible ya que los modulos ESP01 emplean comandos AT, es decir, el conjunto de comandos Hayes, los cuales son un conjunto de comandos empleados para configurar y parametrizar los modems.
La comunicacion del modulo ESP01 es mediante UART(*Universal Asynchronous Receiver-Transmitter*), y debido a que el microntrolador posee 8 puertos UART, es una aplicación bastante factible. Empleando las funciones para enviar datos por la UART en función de la dirección base de la misma, las cuales se encuentran ya diseñadas en el directorio raiz de este proyecto, seria posible establecer una comunicacion serial con el modulo Wi-Fi. Sin embargo, una vez establecida dicha comunicación serial, la cual conlleva consigo una sincronizacion de relojes entre las UART y la comunicación USB para implementar el dispositivo.
Una vez establecida esta comunicación, es necesario trazar un entramado de conexiones de red para crear un servidor TCP en el ESP01 a modo de punto de acceso al cual se pueda conectar el otro, el cual se encuentra en el otro microcontrolador, de tal modo que le envíe los datos por tramas TCP asociados a la IMU y la pulsación de los botones.

No se ha optado por implementarlo en este proyecto, debido a la necesidad del uso de los objetos inherentes al lenguaje de programacion C++ y sus clases para establecer un buen entramado de conexiones de red. Además de ello presentó una notable carga de tiempo de trabajo la sincronización de los relojes.

5. Anexos

Listing 1: Declaración e inicialización de variables