Laboratorio 4: STM32: GPIO, ADC, timer, comunicaciones, Iot

1st Ronny Granados
Escuela de Ingeniería Eléctrica
Universidad de Costa Rica.
San José, Costa Rica
C03505
ronny.granadosperez@ucr.ac.cr

I. Introducción/Resumen

Este laboratorio tenía como objetivo explorar las funcionalidades y modos de operación del dispositivo STM32f429-Discovery Kit, el cual fue obtenido en forma física mediante la bodega de la escuela. Se exploraron GPIOs, modificación de registros, protocolo de comunicación SPI y USART y servidores remotos para la implementación de la aplicación en IOT.

II. Nota Teórica

II-A. STM32f429

Se tienen las características eléctricas del dispositivo [2]:

Table 14. Voltage characteristics

Symbol	Ratings	Min	Max	Unit
V_{DD}–V_{SS}	External main supply voltage (including V_{DDA}, V_{DD} and VBAT)(1)	– 0.3	4.0	V
V_{IN}	Input voltage on FT pins(2)	V_{SS} – 0.3	V_{DD}+4.0	
	Input voltage on TTa pins	V_{SS} – 0.3	4.0	
	Input voltage on any other pin	V_{SS} – 0.3	4.0	
	Input voltage on BOOT0 pin	V_{SS}	9.0	
$\lvert\Delta V_{DD}\rvert$	Variations between different V_{DD} power pins	-	50	mV
$\lvert V_{SSX} - V_{SS}\rvert$	Variations between all the different ground pins including V_{REF-}	-	50	
$V_{ESD(HBM)}$	Electrostatic discharge voltage (human body model)	see *Section 6.3.15: Absolute maximum ratings (electrical sensitivity)*		

Figura 1. Características eléctricas del STM32f429.

Y sus propiedades de tolerancia de corriente:

Table 15. Current characteristics

Symbol	Ratings	Max.	Unit
ΣI_{VDD}	Total current into sum of all V_{DD_x} power lines (source)(1)	270	mA
ΣI_{VSS}	Total current out of sum of all V_{SS_x} ground lines (sink)(1)	– 270	
I_{VDD}	Maximum current into each V_{DD_x} power line (source)(1)	100	
I_{VSS}	Maximum current out of each V_{SS_x} ground line (sink)(1)	– 100	
I_{IO}	Output current sunk by any I/O and control pin	25	
	Output current sourced by any I/Os and control pin	– 25	
ΣI_{IO}	Total output current sunk by sum of all I/O and control pins (2)	120	
	Total output current sourced by sum of all I/Os and control pins(2)	– 120	
$I_{INJ(PIN)}$ (3)	Injected current on FT pins (4)	– 5/+0	
	Injected current on NRST and BOOT0 pins (4)		
	Injected current on TTa pins(5)	±5	
$\Sigma I_{INJ(PIN)}$(5)	Total injected current (sum of all I/O and control pins)(6)	±25	

Figura 2. Características de corriente del STM32f429.

Esta placa es muy completa y tiene una arquitectura ARM de 32 bits Cortex-M4 con FPU (RISC), opera a 180 MHz, 3 convertidores analógico/digital y digital/analógico de 12 bits cada uno, dos botones: uno para el usuario y otro de reset, sensor de movimiento I3G4250D, giroscópio de 3 ejes, LEDs, pantalla LCD de 2.4´´, entre otras cosas que lo convierten en un dispositivo muy versátil.

La placa se pudo programar mediante la librería LibOpenCM3, una librería de C que permite, de manera más sencilla, la programación de la placa al utilizar funciones y ofreciendo muchos ejemplos en donde se explotan las distintas funcionalidades del dispositivo.

Además, para poder cargar los programas al MCU se utilizó ST-Link, el cual puede ser instalado para Linux y Windows y facilita mucho la comunicación con la placa.

II-B. ILI4391

Corresponde a la pantalla con la que viene incorporado el MCU, esta tiene una resolución de 240x320 píxeles y con la cual el STM puede conectarse gracias a protocolos SPI y/o I2C. La placa posee varios puertos SPI, en el caso de la pantalla, utiliza el SPI5 para poder comunicarse, así que las instrucciones que se utilicen dentro del código para la pantalla deben de tener esto en cuenta al momento de su implementación. También hace uso de la librería LibOpenCM3, de hecho que varios ejemplos se enfocan en la realización de distintas acciones en la pantalla, así como escribir texto, crear figuras, cambiar tamaños de texto, entre un sinfín de acciones que se pueden realizar.

II-C. L3GD20

Este dispositivo corresponde al giroscopio que incorpora la placa, es configurado por medio de instrucciones SPI, puede hacer lecturas en los 3 ejes angulares (x, y, z) y además es capaz de medir la temperatura actual de la habitación.

La forma de trabajar con este dispositivo es muy poco intuitiva al inicio pues se debe de configurar el reloj, los puertos a utilizar para que se active el sensor y realizar las lecturas y escrituras de manera correcta, siguiendo el protocolo SPI. Posee registros de interés de los cuales se puede leer información importante. Mencionando los utilizados en este laboratorio y su principal función: registro **WHO_AM_I**, es de utilidad para poder confirmar que las lecturas se están realizando correctamente, es un identificador único; **CTRL_REG1**,

permite habilitar los ejes, el modo de poder y la selección del ancho de banda; **CTRL_REG2**, permite la configuración del filtro pasa altos; **CTRL_REG4**, mediante este registro se configura el DPS y el modo SPI, además de registros para leer los ejes, los cuales son de 8 bits cada uno y se componen de dos partes, por ejemplo, para el eje x se tendrían los registros **OUT_X_L** y **OUT_X_H**, y se realiza lo mismo con los demás ejes [1].

III. Desarrollo/Análisis de Resultados

Primeramente se trabajó en conocer la placa, comprender cómo cargar los archivos binarios a esta y poder familiarizarse con el entorno en general. Lo primero que se realizó para el laboratorio fue poder tener dominio de la pantalla LCD, poder escribir en esta y tener información que cambia si es necesario y que la pantalla sea refrescada cada cierto tiempo. Para esto se utilizó uno de los ejemplos que utilizaba header files con funciones útiles para poder escribir a la pantalla.

Dentro de la función **main()** debía de inicializarse el reloj y demás configuraciones que venían incluidas en el header correspondiente a la pantalla, por lo que se deben de inicializar mediante funciones, tal que:

```
int main(){
    clock_setup();
    gpio_setup();
    clock_setup();
    spi_setup();
    console_setup(115200);
    sdram_init();
    lcd_spi_init();
    .
}
```

hecho esto, dentro del loop es donde se enviaban datos a la pantalla. Una escritura típica en este display se realizaba mediante la siguiente proción de código en donde había que definir el tamaño del texto, al ubicación, el color y el color del fondo, entre otros parámetros para obtener el resultado deseado:

```
gfx_setTextSize(2);
gfx_setTextColor(LCD_WHITE, LCD_GREY);
gfx_setCursor(15, 131);
gfx_puts("USB/Serial:");
```

Al final, después de todas las configuraciones realizadas, la pantalla luce de la siguiente forma:

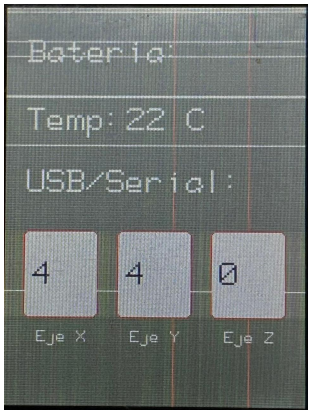

Figura 3. Diseño final de la pantalla LCD.

Poniendo la pantalla en funcionamiento y con completo control sobre esta, se puede proceder a trabajar con el sensor L3GD20, correspondiente al giroscópio. Este fue el dispositivo que más requirió trabajo inicializar pues tiene muchos registros que influyen en la lectura de los datos, autenticidad de estos y la comunicación, así como muchas instrucciones que deben de ser utilizadas para poder leer información de este sensor. Puesto a que se configura mediante comunicaciones SPI haciendo uso del SPI5 de la placa, las instrucciones del código deben hacerse basados en esto, también los puertos correspondientes a los parámetros SPI, representados en la siguiente tabla:

Cuadro I
Configuración del protocolo SPI con los registros del STM32F4.

SCK	GPIOF7
MISO	GPIOF8
MOSI	GPIOF9
SC	GPIOC1

Por lo tanto, todos esos registros y puertos deben ser configurados debidamente en la función **main()**. No solamente los pines debían ser establecidos, sino también los registros en donde se habilitaran las lecturas y especificaciones de ancho de banda, y frecuencia de corte.

III-A. Ancho de Banda y Frecuencia de Corte

El ancho de banda fue establecido mediante el registro **CTRL_REG1** al igual que la frecuencia de corte, dicho registro cuenta con 8 bits y con el cual se puede, entre otras cosas, habilitar los ejes para lectura. Según las especificaciones de la hoja de datos, la forma de obtener una frecuencia de corte baja (12,5 Hz) y una frecuencia de muestreo (190 Hz), fue mediante el siguiente código:

```
gpio_clear(GPIOC, GPIO1); // CS = 0
spi_send(SPI5, CTRL_REG1);
spi_read(SPI5);
spi_send(SPI5, PD | XEN | YEN | ZEN | (1 << 6));
spi_read(SPI5);
gpio_set(GPIOC, GPIO1); // CS = 1
```

La configuración SPI se realizó mediante un registro y de la misma forma que con los ejes en el código anterior. En este caso se desea que la comunicación SPI sea mediante 4 cables, que es la configuración por defecto y un DPS (degrees per second) de 500, esto con el fin de regular la sensibilidad de las lecturas y que no cambie muy abruptamente el sensor con cambios muy pequeños, por lo que se realizó con el siguiente código:

```
gpio_clear(GPIOC, GPIO1); // CS = 0
spi_send(SPI5, CTRL_REG4);
spi_read(SPI5);
spi_send(SPI5, 1 << 4);
spi_read(SPI5);
gpio_set(GPIOC, GPIO1); // CS = 1
```

Con esto listo ya se puede hacer lectura de los ejes y de la temperatura medidos por el sensor. Para realizar lectura, según el protocolo SPI, se debe de hacer algo muy similar a las escrituras anteriores. A continuación se muestra una porción del código donde se realiza la lectura de la temperatura y se imprime en consola y se guarda en una variable:

```
// Leo temperatura
gpio_clear(GPIOC, GPIO1);
spi_send(SPI5, OUT_TEMP | (1 << 7));
spi_read(SPI5);
spi_send(SPI5, 0);
temp = spi_read(SPI5);
console_puts("Temperatura:");
print_int(temp);
console_puts("\n");
gpio_set(GPIOC, GPIO1);
```

Una vez que se obtienen estas lecturas ya se pueden mostrar los valores en la pantalla LCD, tal como se pudo anticipar en la figura 3.

III-B. Mediciones de ejes X, Y y Z

Estas mediciones tuvieron que ser ajustadas tanto con la sensibilidad DPS de la sección anterior, como con un parámetro de ajuste que fue encontrado en un repositorio de GitHub donde trabajaban con el mismo sensor. La siguiente tabla muestra los parámetros de ajuste según el valor de DPS que se haya escogido:

Cuadro II
PARÁMETROS DE AJUSTE PARA MEDICIONES DE LOS EJES SEGÚN EL VALOR DE DPS.

DPS	Valor de Ajuste
250	0.00875
500	0.0175
2000	0.070

Este factor debía ser multiplicado por la lectura del eje antes de ser enviada la consola y la pantalla para poder ser efectiva. Se observó una gran diferencia antes y después de ajustar sensibilidad y de aplicar ese factor de ajuste, las mediciones se ven más controladas y menos aleatorias y tienen más sentido físicamente que cuando no se tenía nada de esto configurado, sin embargo aún se tiene un comportamiento extraño en donde no se logra un valor estable para ningún eje, con un desfase de $\pm$ 1 aproximadamente.

Por otra parte, la lectura de la temperatura fue realizada de la misma manera que la de los ejes, sin embargo, este parámetro tiene un comportamiento opuesto al esperado. Se estabiliza en un valor que tiene sentido (aproximadamente 19 °C) pero cuando se expone a ambientes más fríos la temperatura sube en vez de bajar y viceversa. No se pudo encontrar ningún factor en la hoja de datos que pudiera hacer que se comportara de la manera correcta, por lo que se decidió trabajar con la temperatura de esa manera.

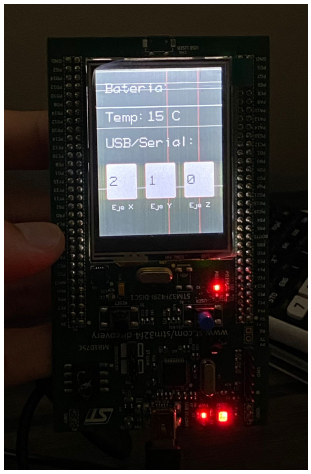

Figura 4. Imagen Final de la Placa.

IV. CONCLUSIONES Y RECOMENDACIONES

Debido a problemas con el funcionamiento de la placa y su debida conexión con la máquina virtual corriendo en mi computadora, el tiempo que tuve para realizar este laboratorio fue de pocos días. Días en los que tuve que comprender el funcionamiento de toda la placa y los sensores desde cero, sin embargo, gran parte de las tareas por realizar no fueron conseguidas. Se intentó, mediante un script de Python realizar la conexión con ThingsBoard pero no fue exitosa.

Agradecimiento especial al compañero Roger Piovet quien ha brindado su ayuda y conocimientos para la realización de este laboratorio, así como al profesor, quien se mostró muy comprensivo con toda la situación.

Como recomendaciones están iniciar el laboratorio con tiempo, probando que todo funcione y asistir a consulta para poder comprender conceptos que queden vagos. Existe documentación tanto para la placa como para el sensor y la librería, pero todo se encuentra de manera separada, por lo que muchas veces se puede tomar un día o más en relacionar un concepto con el otro. Esta placa tiene un nivel más elevado a las ya vistas en el curso por lo que sí se recomienda un tiempo para poder familiarizarse con esta, llevar un plan de trabajo

ordenado y romper el laboratorio en partes o bloques para ir saliendo con todo de manera eficiente.

REFERENCIAS

[1] ST. L3GD20. https://www.st.com/resource/en/application_note/an4505-l3gd20-3axis-digital-output-gyroscope-stmicroelectronics.pdf, 2013.

[2] ST. STM32F427xx STM32F429xx. https://www.st.com/resource/en/datasheet/stm32f427vg.pdf, 2018.