

Laboratorio 2

Camilo Andrés Maldonado Lopez¹, Kevin Jesgreg Cañon Alvarez¹ y Yesika Biviana Ramirez Duran¹

¹ Escuela Colombiana de Ingeniería Julio Garavito, Bogotá D.C, Colombia
camilo.maldonado-1, kevin.canon,
yesika.ramirez@mail.escuelaing.edu.co

Resumen. Este documento presenta el proceso de implementación del sistema realizado para el laboratorio 2, en la tarjeta STM32 Núcleo F429ZI.

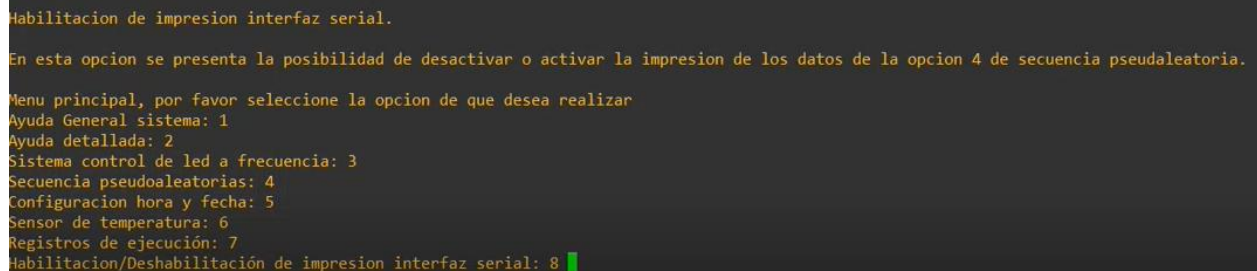
Palabras clave: RTC, I2C, EEPROM, sensor de temperatura, LFSR.

1 Desarrollo

El sistema inicia con un menú el cual muestra al usuario, a través de la interfaz PUTTY, el cual le asigna una opción numérica a cada uno de los comando requeridos y para acceder a cada una de las características el usuario debe ingresar el numero correspondiente a la opción, a continuación, se muestra una lista de la opciones y una muestra de cómo se ve el menú dentro de la interfaz:

Menú principal:

1. Ayuda General
2. Ayuda Detallada
3. Sistema de control de LED a frecuencia
4. Secuencia pseudoaleatoria
5. Configuración hora y fecha
6. Sensor de temperatura
7. Registros de ejecución
8. Habilidad o deshabilitación de impresión en la interfaz serial



```
Habilitacion de impresion interfaz serial.  
  
En esta opcion se presenta la posibilidad de desactivar o activar la impresion de los datos de la opcion 4 de secuencia pseudaleatoria.  
  
Menu principal, por favor seleccione la opcion de que desea realizar  
Ayuda General sistema: 1  
Ayuda detallada: 2  
Sistema control de led a frecuencia: 3  
Secuencia pseudoaleatorias: 4  
Configuracion hora y fecha: 5  
Sensor de temperatura: 6  
Registros de ejecucion: 7  
Habilitacion/Deshabilitacion de impresion interfaz serial: 8 █
```

Fig 1. Menú principal.

Para la implementación de este menú se utiliza una función case la cual dependiendo la opción seleccionada abrirá otro menú o ejecutará una acción además al terminar la misma o en caso de error se retornará de manera inmediata al este menú. Ahora se hablará un poco a cerca de la implementación de cada una de las opción incluidas en este menú.

1.1 Ayuda General

En esta sección se muestra una ayuda muy sencilla cuando se selecciona esta opción y luego se retorna al menú principal

1.2 Ayuda Detallada

En esta sección luego del menú principal se despliega otro menú el cual permite al usuario elegir cual es la función sobre la que quiere conocer más y muestra una descripción corta acerca de cada una de ellas y su uso. Parte de su implementación se muestra en la siguiente figura:

```
band_menu=1;
}
if(menu==2)
{
    printf("\n[2J");
    printf("\nSeleccione de cual opción quiere tener mas información ");
    printf("\nSistema control de led a frecuencia: 1 ");
    printf("\nSecuencia pseudoaleatorias: 2 ");
    printf("\nConfiguración hora y fecha: 3 ");
    printf("\nSensor de temperatura: 4 ");
    printf("\nRegistros de ejecución: 5 ");
    printf("\nRehabilitación/deshabilitación de impresión interfaz serial: 6 ");
    scanf("%s",opcion);
    int help = atoi(opcion);
    if(help==1)
    {
        printf("\n[2J");
        printf("\nSistema control de led a frecuencia\n");
        printf("\nEn esta opción tenemos unas frecuencias predeterminadas para escoger, aunque tambien se encuentra una opción en la que podemos"
            " escribir un valor de frecuencia diferente a las que nos da el sistema, la única condición es que se encuentre en el rango de 0.1 a 10 Hz"
            " de lo contrario el sistema arrojará un error y nos dirigirá al menú principal.\n");
        scanf("%s",next);
        band_menu=1;
    }
    if(help==2)
    {
```

Fig 2. Implementación ayuda detallada.

1.3 Sistema de control de LED a frecuencia deseada

Luego de seleccionar esta opción se muestra al usuario otro menú el cual permite hacer el encendido y apagado del LED, además permite que el led vaya a diferentes frecuencias de oscilación entre 0.1 y 10 Hz.

Dentro del menú, que se muestra a continuación, se encuentran algunas opciones predeterminadas como: 0.1 Hz, 0.5 Hz, 1 Hz, 3Hz, 5Hz, 7Hz, 9 Hz y 10 Hz las cuales pueden ser seleccionadas por el usuario de manera rápida seleccionando la opción y además se encuentra la opción otras la cual permite que el usuario ingrese un número dentro del rango permitido en caso de no elegir ninguna de las opciones.

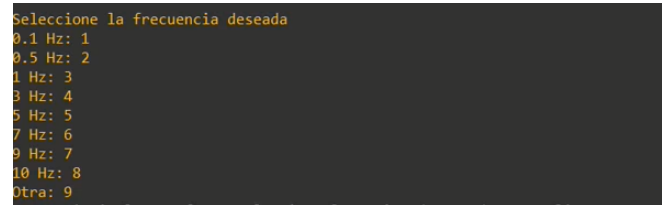


Fig 3. Menú para el sistema de control de LED.

Este sistema funciona a través de una serie de retardos, en el caso de las opciones predeterminadas se encuentra un retardo establecido, por ejemplo, en el caso de 1 Hz el led, que para este caso es el LED número 2, durará 1 segundo encendido y un segundo apagado, y para el caso de las opciones predeterminadas se usa un retardo el cual se determina de la siguiente manera.

```

if(dig==9)
{
    printf("\r\nValor frecuencia entre 0.1 a 10 Hz");
    scanf("%s",fre);
    int f = atoi(fre);
    var = (1/f)*1000;
    while(cont<<101)
    {
        HAL_GPIO_TogglePin(LD2_GPIO_Port,LD2_Pin);
        HAL_Delay(var);
        cont++;
    }
}

```

Fig 4. Parte del código para frecuencias diferentes a las opciones.

1.4 Secuencia pseudoaleatoria

En este caso se implementó la secuencia a partir de una compuerta XOR, la cual acepta un tamaño de 7 hasta 16 bits. Su implementación y visualización se muestran a continuación:

```

int cont=0,cant=0,max=4096,p_max=0,pos=0;
uint16_t lfsr1=atoi(semilla);
uint16_t seed=0xFFFF;
uint16_t bit1;
printf("\r\n#bits término de realimentación %s\r\n", tags);
printf("\r\nValor inicial(semilla) %s\r\n", semilla);
printf("\r\n#Valores esperados %d\r\n", max);
while(cont<4096)
{
    bit1 = ((lfsr1 >> 0) ^ ((lfsr1 >> 1) & 1);
    lfsr1 = (lfsr1 >> 1) | (bit1 << 11);
    cont++;
    if(hand_impression==1)
    {
        printf("\r\nData: %x\r\n", lfsr1);
    }
    HAL_GPIO_TogglePin(LD2_GPIO_Port,LD2_Pin);
    HAL_Delay(200);
    cant++;
    if(atoi(semilla)==lfsr1)
    {
        cont=4097;
        printf("\r\nCantidad: %d\r\n", cant);
    }
    if(0xFFFF ==lfsr1)
    {
        pos=cant;
        p_max=1;
    }
}

```

Fig 5. Implementación de la secuencia.

```

Ingrese el valor semilla:
El valor semilla ingresado: 1

Ingrese el tamaño para la realimentacion
El tamaño: 7

#bits término de realimentación 7
Valor inicial(semilla) 1
#Valores esperados 128
Cantidad: 127
#bits término de realimentación 7
Valor inicial(semilla) 1
Numero de valores calculados 127
Tuvo periodo maximo en la secuencia 121

Menu principal, por favor seleccione la opcion de que desea realizar
Ayuda General sistema: 1
Ayuda detallada: 2
Sistema control de led a frecuencia: 3
Secuencia pseudoaleatorias: 4
Configuracion hora y fecha: 5
Sensor de temperatura: 6
Registros de ejecución: 7
Habilitacion/Deshabilitación de impresion interfaz serial: 8

```

Fig 6. Visualización con la deshabilitación de impresión en la interfaz.

```

data: 29
data: 54
data: 2a
data: 55
data: 6a
data: 75
data: 7a
data: 7d
data: 7e
data: 7f
data: 3f
data: 1f
data: f
data: 7
data: 3
data: 1
Cantidad: 127
#bits término de realimentación 7
Valor inicial(semilla) 1
Numero de valores calculados 127
Tuvo periodo maximo en la secuencia 121

```

Fig 7. Visualización con la habilitación de impresión en la interfaz.

1.5 Configuración hora y fecha

Usando el modulo DS3231, el cual es un reloj en tiempo real que cuanta con un oscilador de cristal con compensación de temperatura (TCXO). El cual se comunica con la tarjeta Núcleo F429ZI por medio de I2C. el montaje usado se muestra a continuación:

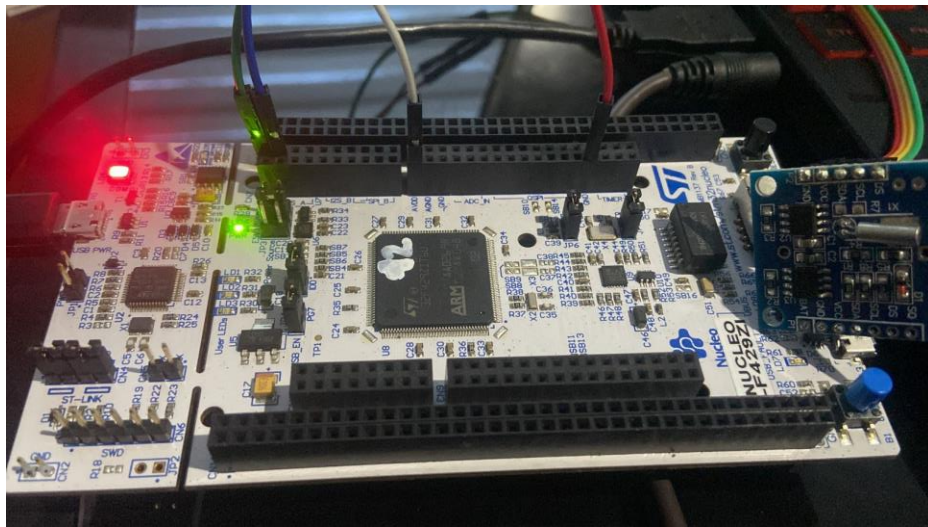


Fig 8. Montaje modulo DS3231 y Tarjeta Núcleo F429ZI.

La configuración del Set-time se realiza desde el código y luego se tiene en muestra en la interfaz la fecha y hora y se retorna al menú principal. Como se muestra en la fig 5:

```

Time: 00:76:102   Date: 00/00/2000

Menu principal, por favor seleccione la opcion de que desea realizar
Ayuda General sistema: 1
Ayuda detallada: 2
Sistema control de led a frecuencia: 3
Frecuencia pseudoaleatorias: 4
Configuración hora y fecha: 5
Sensor de temperatura: 6
Registros de ejecución: 7
Habilitación/Deshabilitación de impresion interfaz serial: 8

```

Fig 9. Visualización de hora y fecha en la interfaz.

Para llevar a cabo esta función se realiza usa la función de `get_time` dentro del bucle principal para que así la hora se actualice automáticamente luego de configurarla al inicio. La función de `get_time` hace el llamado a otra función la cual se encarga de

convertir la hora de decimal a bcd. La implementación de la función get time se muestra en la figura 6.

```

111
112=void Get_Time(void)
113 {
114     uint8_t get_time[7];
115     HAL_I2C_Mem_Read(&hi2c1, DS3231_ADDRESS, 0x00, 1, get_time, 7, 1000);
116     time.seconds = bcdToDec(get_time[0]);
117     time.minutes = bcdToDec(get_time[1]);
118     time.hour = bcdToDec(get_time[2]);
119     time.dayofweek = bcdToDec(get_time[3]);
120     time.dayofmonth = bcdToDec(get_time[4]);
121     time.month = bcdToDec(get_time[5]);
122     time.year = bcdToDec(get_time[6]);
123 }

```

Fig 10. Función get_time.

```

70 uint8_t decToBcd(int val)
71 {
72     return(uint8_t)((val/10*16)+(val%10));
73 }
74 int bcdToDec(uint8_t val)
75 {
76     return (int)((val/16*10)+(val%16));
77 }
78

```

Fig 11. Funciones de conversión de decimal a bcd y bcd a decimal.

1.6 Habilitación o deshabilitación de impresión en la interfaz serial

Esta opción básicamente se encarga de mostrar u ocultar las secuencias pseudoaleatorias, por defecto están ocultas o deshabilitadas y cada vez que se selecciona esta opción pasa de un estado a otro, su visualización en la interfaz se puede ver en las figuras 6 y 7.

Referencias

1. Contollers Tech. (2020, 4 de enero). *DS3231 RTC Module and STM32 // cubeide // I2C-LCD* [Video]. YouTube. <https://www.youtube.com/watch?v=LBD7noDBnlc>
2. (s.f.). Mixed-signal and digital signal processing ICs | Analog Devices. <http://www.analog.com/media/en/technical-documentation/data-sheets/ds1307.pdf>
3. Apuntes de clase DMIC-2022-2