

Comunicación Serial Matlab-STM

Juan David Sandoval

Juandsandoval8@gmail.com, Ingeniería Electrónica

Resumen—La comunicación serie o comunicación secuencial, en telecomunicaciones e informática, es el proceso de envío de datos de un bit a la vez, de forma secuencial, sobre un canal de comunicación o un bus de datos.

La ventaja de la comunicación serie es que necesita un número más pequeño de líneas de transmisión que una comunicación paralela que transmita la misma información. Por otra parte, surgen una serie de problemas en la transmisión de un gran número de bits en paralelo, como los problemas de interferencia o desincronización.

A la misma frecuencia de transmisión, la comunicación paralela tiene un mayor rendimiento.

Abstract— Serial communication or sequential communication, in telecommunications and information technology, is the process of sending data one bit at a time, sequentially, over a communication channel or a data bus.

The advantage of serial communication is that you need a smaller number of transmission lines than a parallel communication that transmits the same information. On the other hand, a series of problems arise in the transmission of a large number of bits in parallel, such as interference or desynchronization problems.

At the same transmission frequency, parallel communication has higher performance

1. INTRODUCCIÓN

En este informe se buscará establecer una conexión del tipo serial entre Matlab y STM, en donde nuestra interfaz de Matlab mostrará como un “hyper-terminal” los valores de conversión de un módulo joystick provenientes de nuestra tarjeta STM

2. METODOLOGÍA

A. PROBLEMA

En clase se planteó una parte del proyecto de grado que consistiría en establecer una comunicación serial entre Matlab y Mbed, donde esta vez lo único que haremos es enviar datos y recibir en nuestro Matlab.

B. MATERIALES

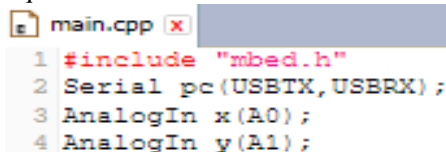
*STM32F411RE y Matlab

*Modulo joystick

C. COMO USARLOS

- STM32F411RE: Como ya vimos en los anteriores

informes y guías usaremos nuestro Mbed para programar nuestra tarjeta STM, una vez creado nuestro proyecto trabajaremos nuestro ADC pues será lo único que se enviará a Matlab.



```
1 #include "mbed.h"
2 Serial pc(USBTX, USBRX);
3 AnalogIn x(A0);
4 AnalogIn y(A1);
```

- Lo primero que se hace es declarar nuestros puertos seriales que vendrían siendo “USBTX Y USBRX”, definimos las entradas analógicas que corresponden a las de los potenciómetros del módulo joystick.

```
5 int main()
6 {
7
8     while(1) {
9         pc.printf("X value = %.0f\n\r", x.read()*3300);
10        pc.printf("Y value = %.0f\n\r", y.read()*3300);
11        wait_ms(100);
12    }
13 }
```

- En nuestro código principal vendrá nuestro ciclo infinito o “while(1)”, donde pondremos lo que se enviará a Matlab con un “printf”, dentro de este comando vendrá un string para mostrar a que conversión nos referimos y de una vez ingresamos en la misma línea de código de nuestras entradas análogas y su respectiva multiplicación por 3300.

```

ejemplos.m*  +
1 - clear all;
2 - clc;
3 - delete(instrfind({'port'}, {'COM4'}));
4 - s=serial('COM4');
5 - s.BaudRate=9600;
6 - warning('off', 'MATLAB:serial:fscanf:unsuccessfulRead');
7 - fopen(s);
8 - while(1)
9 -     ADC=fscanf(s, '%d');
10 -     ADC
11 - end
12 - fclose(s);
13 - delete(s);
14 - clear all;

```

- En nuestro Matlab, los comandos nuevos que usaremos son:
 - delete(instrfind({'port'}, {'COM4'})): Limpia todos los puertos para solo usar el puerto que requerimos.
 - s=serial('COM4'): nombraremos a COM4 como nuestro puerto a usar.
 - s.BaudRate=9600: configura la velocidad a la cual se enviarán los datos esta debe ser la misma de la STM por eso 9600.
 - warning: apaga todas las alertas generadas en los sprinf
 - fopen(s): abrimos el puerto serial para empezar a enviar y/o recibir datos.
 - fscanf(s, '%d'): recibe datos enviados por el serial y los recoge en formato entero.
 - fclose: cierra el puerto una vez terminado el código.
 - delete(s): borra todo lo que había en el puerto para dejarlo listo para usar y que no queden residuos
- En esta parte del código recibiremos el dato mediante un fscanf y recibiremos todo lo que viene del serial es decir nuestra variable "s" y serán variables del tipo %d (enteros).

```

Command Window

,
Y value = 3298
,

ADC =

,
X value = 1638
,

ADC =

,
Y value = 3298
,
.

ADC =

,
X value = 1638
,
fx

```

- Una vez corremos el código y movemos nuestro módulo joystick esto veremos en nuestro "CommandWindow", como vemos es muy parecido a lo que vemos en nuestro hyperterminal de "Arduino" o "RealTerm", veremos cómo lee datos cada 100mS ya que esa es la velocidad de transmisión que le dimos a nuestra tarjeta STM para enviar por el puerto serial.

3. CONCLUSIONES

Podemos concluir que muchas veces tenemos un déficit grande en cuanto a la programación con esta tarjeta de desarrollo aun así cuantas con muchas facilidades que esperamos tener la facilidad de usar después con la experiencia adquirida en esta clase, se logró el objetivo que es lo importante pero falta mejorar.

4. CONCLUSIONES

Podemos concluir que muchas veces tenemos un déficit grande en cuanto a la programación con esta tarjeta de desarrollo aun así cuantas con muchas facilidades que esperamos tener la facilidad de usar después con la experiencia adquirida en esta clase, se logró el objetivo que es lo importante pero falta mejorar.

5. BIBLIOGRAFIA

- Docs.zephyrproject.org. (2019). ST Nucleo F411RE — Zephyr Project Documentation. [online] Available at: https://docs.zephyrproject.org/latest/boards/arm/nucleo._f411re/doc/index.html [Accessed 29 Nov. 2019].
- Arm Mbed. (2019). Home | Mbed. [online] Available at: <https://www.mbed.com/en/> [Accessed 29 Nov. 2019].
- La.mathworks.com. (2019). MATLAB - El lenguaje del cálculo técnico. [online] Available at: <https://la.mathworks.com/products/matlab.html> [Accessed 29 Nov. 2019].

