ACELERÓMETRO CON MMA7260QT

Aplicaciones en Comunicaciones y Académicas

ROLÓN, Román Emanuel

romanrolon@gmail.com

Universidad Tecnológica Nacional - Facultad Regional Paraná

Argentina – Entre Ríos - Paraná

Palabras claves: serie - acelerómetro - instrumento - interfaz - Freescale – Matlab - CodeWarrior.

Resumen

El objetivo principal del presente documento es proporcionar información sobre la utilización del módulo serie para la transmisión de datos con el microcontrolador (MCU) MC68HC908QB8 perteneciente a la firma ***"Freescale Semiconductor"***.

Particularmente, se lo ha utilizado para la realización de un Instrumento Virtual, el cual permite la medición de niveles de aceleración en el espacio, o sea en cualquier dirección XYZ de un cuerpo. Este instrumento tiene la capacidad y versatilidad de tomar mediciones de niveles de aceleración con signo, esto significa que discrimina la dirección y el sentido de ésta.

Abstract

The main objective of this paper is to provide information on using the module for serial data transmission with the microcontroller (MCU) MC68HC908QB8 belonging to the firm "Freescale Semiconductor."

Particularly, it has been used to implement a Virtual Instrument, which allows the measurement of levels of acceleration in space, or in any direction of an XYZ body. This instrument has the capability and versatility to take measurements of levels of acceleration sign, it means that discriminates the direction and sense of it.

1. Descripción general del Instrumento

Está compuesto básicamente por una placa de transducción y adaptación de la señal obtenida a través de un sensor (acelerómetro) MMA7260QT, una placa de adquisición, almacenamiento y transmisión serie de los datos basada en el MCU MC68HC908QB8 y como interfaz gráfica se ha desarrollado un software de PC en MATLAB utilizando el GUI "Interfaz Gráfica de Usuario". Estas etapas se las puede observar claramente en el siguiente diagrama en bloques general del instrumento:

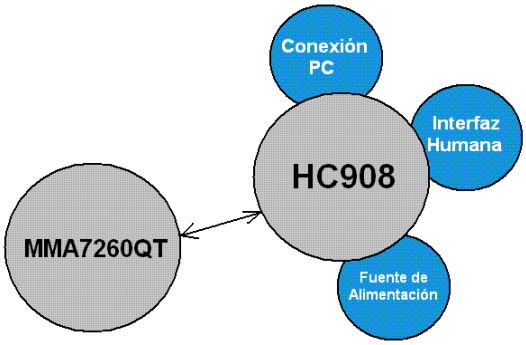

figura 1

2. Acelerómetro: MMA7260QT

Características constructivas:

Los acelerómetros capacitivos operan con una técnica donde la capacitancia del elemento sensor interno varía en función de la aceleración aplicada, como se puede observar en la *figura 2*.

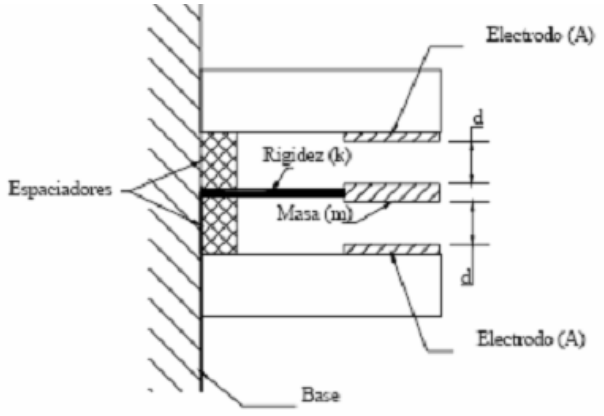

figura 2

En la *figura 2* se representa el elemento sensor, que consiste en dos placas conductoras paralelas tipo electrodo con área de exposición *A* y una masa *m* suspendida por medio de un elemento con rigidez *k*. Entre la masa y los electrodos existe una distancia base *d* simétrica, que se controla con precisión, por lo que el aire que existe en el hueco entre cada electrodo y la masa sísmica forma un *"capacitor mecánicamente variable"*.

El cambio en la distancia *"d"* corresponde a los cambios en la capacitancia. Estos acelerómetros incorporan circuitos micro-eléctricos que usan puentes capacitivos para convertir el cambio de capacitancia a una señal de voltaje útil proporcional a la aceleración.

Características físicas:

El sensor tiene un encapsulado tipo QFN con 16 pines, sus dimensiones son de 6[mm] x 6[mm] x 1,45[mm], en la *figura 3* se muestra la vista desde abajo *"Bottom View"*. [1]

figura 3

Una vista desde arriba *"Top View"* con la distribución de pines del sensor se puede apreciar en la *figura 4*.

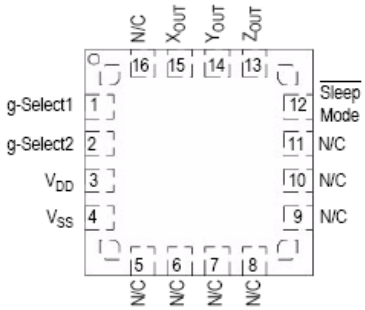

figura 4

Características funcionales:

Las características principales de este dispositivo electrónico son:

- Selección del nivel de sensibilidad mediante la conexión de los pines g-Select1 y g-Select2 (pines 1 y 2) como lo muestra la siguiente tabla:

g-Select2	g-Select1	g-Range	Sensivility
0	0	1.5[g]	800[mV/g]
0	1	2[g]	600[mV/g]
1	0	4[g]	300[mV/g]
1	1	6[g]	200[mV/g]

- Su voltaje de operación es de 2.2[V] - 3.6[V]
- Posee un bajo consumo de corriente el cual es de 500[µA]
- Posee modo "sleep" o bajo consumo de 3[µA]
- Viene en encapsulado QFN de 6[mm] x 6[mm] x 1.45[mm]
- Cuenta con un filtro pasa bajo integrado
- Posee un diseño robusto resistente a los golpes
- Posee un bajo costo
- No contiene plomo en los terminales

3. Microcontrolador: MC68HC908QB8

El MCU utilizado pertenece a la firma Freescale Semiconductor, el mismo es miembro de la familia de ocho bits de bajo costo y alta performance MC68HC908. En la siguiente figura se observa su arquitectura interna.

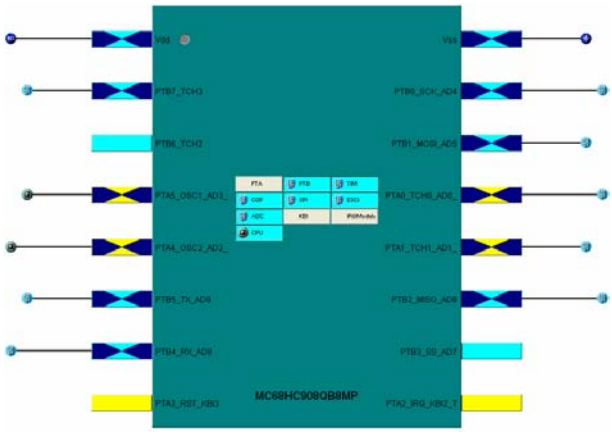

figura 5

Este MCU posee 8K bytes de memoria FLASH y 256 bytes de memoria RAM. Además cuenta con 16 pines y como se observa en la *figura 5* la mayoría de ellos son utilizados. Internamente se divide mediante módulos y los que se encuentran de color turquesa son aquellos que se han utilizado para esta aplicación. [2]

Los módulos PTA y PTB hacen referencia a los puertos de entrada/salida.

- El PTA consta de 6 bits, en el cual se utiliza el PTA5 (OSC1) como entrada de clock de 9,8304[MHz]. el PTA0 (AD0) y el PTA1 (AD1) son utilizados como canales del ADC recibiendo las señales analógicas del eje X e Y respectivamente.
- El PTB consta de 8 bits, se utiliza el PTB4 (TX) y PTB5 (RX), que corresponden a la transmisión y recepción de datos serie hacia y desde la PC, mediante el módulo SCI (Serial Communications Interface Module). Los puertos PTB0 y PTB1 son configurados

como salida y corresponden a los pines del sensor g-Select1 y g-Select2 respectivamente. El puerto PTB2 (AD6) es utilizado como canal del ADC y recibe la señal analógica del eje Z.

- El módulo TIM es un contador/temporizador, utilizado para realizar diferentes retardos dentro del programa.

Este MCU posee otros módulos que no se han utilizado como por ejemplo SPI (que permite comunicar dispositivos por otro protocolo serie), IRQ (estas son interrupciones externas) y el modulo KBI (interrupciones de teclado).

4. Módulo serie (ESCI)

Para programar el microcontrolador se utiliza el software "CodeWarrior v6.2", este software se puede descargar de forma gratuita en la página oficial de freescale. [5]

Para la configuración del módulo serie se utiliza el *"device initialization"* (DI), el cual es una herramienta gráfica y muy fácil de utilizar. Esta herramienta permite al usuario configurar los diferentes parámetros, en este caso, del módulo serie. En la *figura 5* se puede observar el aspecto físico del MCU en el DI.

Para comenzar con la configuración del módulo serie hacemos clic sobre el módulo etiquetado como ESCI "Enhanced Serial Comunications Interface Module" y se abre una ventana como muestra la siguiente figura:

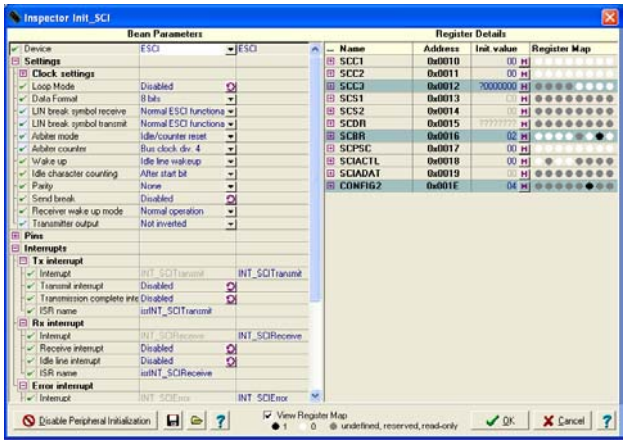

figura 6

Como se observa en la *figura 6*, la ventana tiene dos columnas, la de la izquierda hace referencia a todos los parámetros configurables del módulo, y la de la derecha muestra los detalles de los registros internos del MCU que se modifican. Haciendo una ampliación de la columna de la izquierda:

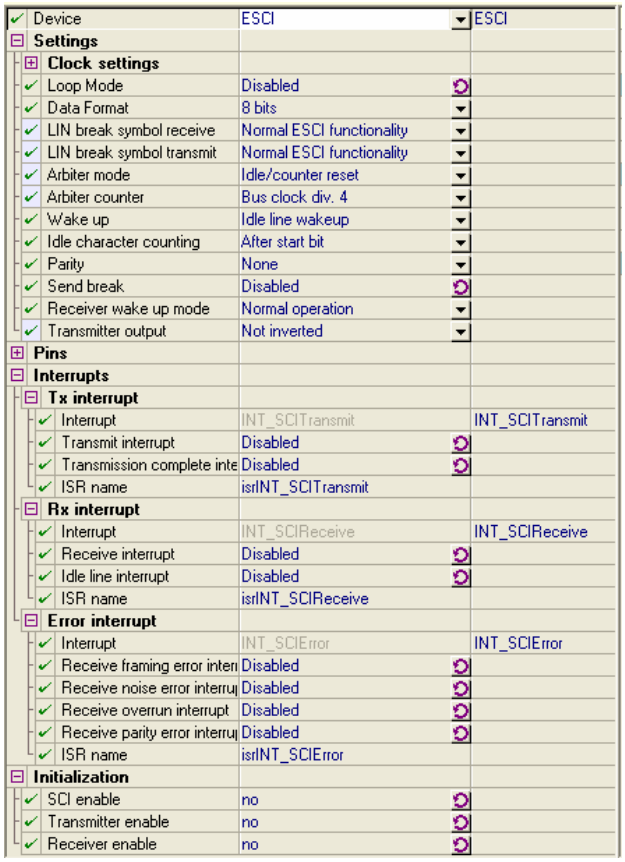

figura 7

Como se puede observar hay cuatro solapas generales:

- ❖ Settings
- ❖ Pins
- ❖ Interrupts
- ❖ Initialization

En la solapa *Settings* se configuran los parámetros principales referidos a la forma en que se transmiten los datos, como por ejemplo:

✓ Baud rate: determina la velocidad de transmisión, esta puede ser 2400 Baud, 9600 Baud, 19200 Baud, etc.

✓ Data Format: determina la longitud de los caracteres, 8 o 9.

✓ Parity: determina la paridad de los datos, esta puede ser par, impar o nula.

✓ Transmiter output: determina el protocolo de transmisión de salida, puede ser invertido o no invertido.

En la solapa *Pinns* se observa solamente el pin del MCU correspondiente a TxD "Transmitter Data" y a RxD "Receiver Data".

En la solapa *Interrupts* se pueden configurar tres tipos de interrupciones:

✓ Tx Interrupt: aquí se puede habilitar la interrupción por transmisión serie, también se le puede colocar el nombre de dicha interrupción.

✓ Rx Interrupt: Interrupción por recepción serie.

✓ Error Interrupt: esta interrupción se puede habilitar también por si existe un error en la transmisión o recepción serie.

Por último, en la solapa *Initialization* se habilitan las interrupciones.

Luego de haber configurado todos los parámetros del módulo serie, se debe generar el código, para esto se debe hacer clic en *"Generate Code"*.

5. Metodología de trabajo

El sensor genera una señal analógica de salida en cada uno de los ejes proporcional a la aceleración sometida, esta señal analógica se acondiciona mediante la implementación de filtros pasa bajos pasivos de primer orden del tipo RC. En la *figura 8* se puede observar la implementación de estos filtros, es necesario aclarar que esta configuración ha sido extraída directamente de la hoja de datos del sensor.

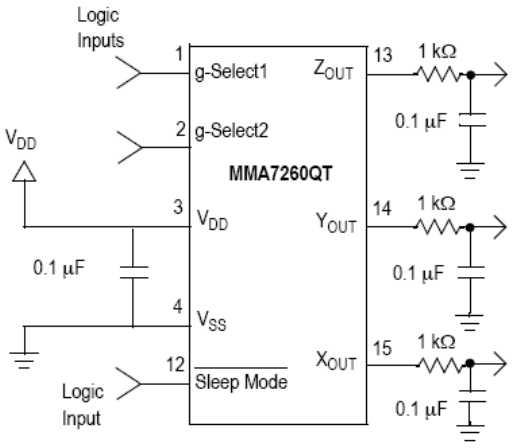

figura 8

Como el sensor y el MCU pertenecen a la misma firma, la misma hoja de datos del sensor tiene una aplicación típica "typical Aplication" en la cual nos muestra como se realiza la conexión entre ambos, esto se puede apreciar en la *figura 9*.

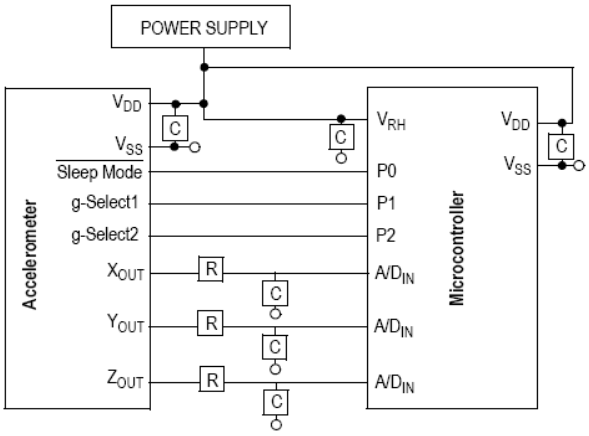

figura 9

Como se observa en la figura anterior las salidas del sensor se conectan directamente a tres entradas del ADC del MCU. Los pines de configuración de la sensibilidad del sensor son conectados a dos pines configurados como salida del MCU.

En esta aplicación no se controla el pin 12 (Sleep Mode) desde el MCU, se conecta directamente a VDD (3,3[V]), lo que significa que nunca se encuentra en modo bajo consumo, debido a que para esta aplicación el consumo de corriente no es tan significativo.

El software implementado en el MCU se lo puede describir conceptualmente a través de un diagrama de flujo como se puede observar en la *figura 10*.

En el documento ANEXO se encuentra el código implementado en el MCU.

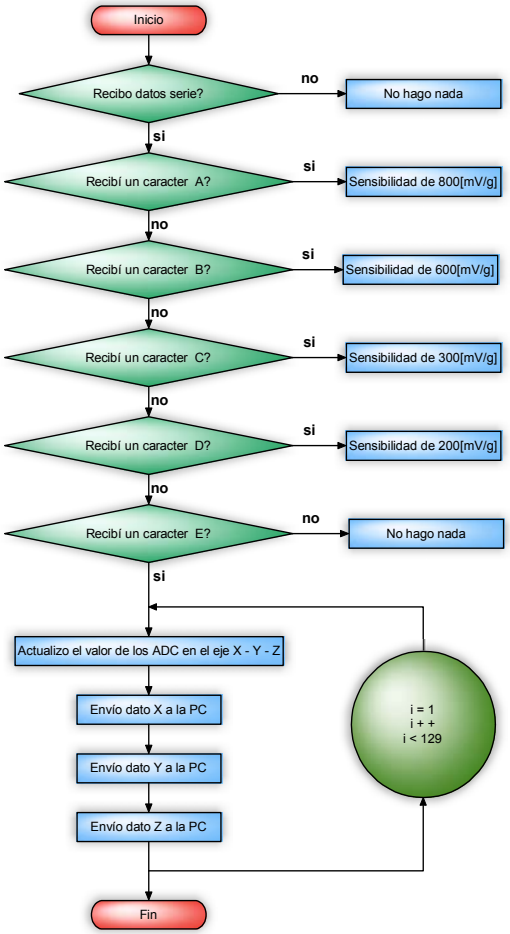

figura 10

El sensor se encuentra activo en todo momento y el MCU actúa en función de lo que le indica el software implementado en Matlab, ya sea para configurar la sensibilidad o para tomar datos del sensor y transmitiros a la PC.

6. Interfaz Humana

Como ya se ha mencionado con anterioridad, la interfaz humana esta basada en un software implementado con la interfaz gráfica de Matlab 7.6 *"GUI"*. [3] [4]

Las principales prestaciones de este instrumento virtual se pueden rotular de la siguiente manera:

- Presenta en tres gráficas temporales y en tiempo real de los niveles de aceleración teniendo en cuenta su signo.
- Posee sensibilidad ajustable.
- Muestra los valores de tensión actuales en los pines de los tres ejes del sensor.
- Muestra los valores de aceleración actuales en función de g.
- Muestra el valor máximo y mínimo de aceleración registrado.
- Presenta una gráfica en tres dimensiones que representa una medida de la orientación del sensor.

A continuación se muestra la interfaz gráfica:

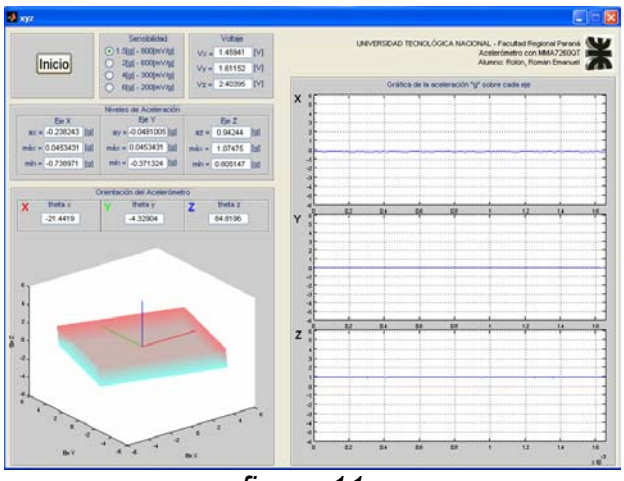

figura 11

7. Bibliografía

[1] Hoja de datos del acelerómetro. (citado en junio de 2009). Se encuentra disponible en: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMA7260QT&nodeId=01126911184209

[2] Hoja de datos del microcontrolador. (citado en junio de 2009). Se encuentra disponible en: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?tab=Buy_Parametric_Tab&code=HC08Q&fromSearch=false&nodeId=01624684497663

[3] Uso de la ayuda online de Matlab, (citado en junio de 2009). Se encuentra disponible en: www.mathworks.com

[4] Uso de comunicación serie con Matlab, (citado en junio de 2009). Se encuentra disponible en: http://www.matpic.com/MICROCHIP/MICROCHIP_COMSERIAL_MATLAB_PIC.html

[5] Software CodeWarrior v6.2, (citado en junio de 2009). Se encuentra disponible en: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=CW-MICROCONTROLLERS&fpsp=1&tab=Design_Tools_Tab

ACELERÓMETRO CON MMA7260QT
ANEXO

Código implementado en el MCU

```
#include <hidef.h>
#include "derivative.h"
#include "variables.h"

/*definición de funciones*/
void MCU_init(void);
void retardo_100ms(word dato);
void update_ADC(void);

void main(void) {
MCU_init();
PTB_PTB0=0; //pin g1 del acelerómetro
PTB_PTB1=0; //pin g2 del acelerómetro
PTB_PTB7=0; //led que indica que me encuentro en el programa principal
SCC1_ENSCI=1;   //enciendo SCI
SCC2_RE=1;      //enciendo receptor
SCC2_SCRIE= 1;  //habilito interrupcion de Rx serie
loop:
PTB_PTB7=1; //led que indica que me encuentro en el programa principal
retardo_100ms(2); //hago un retardo de 200[mseg]
PTB_PTB7=0; //led que indica que me encuentro en el programa principal
retardo_100ms(2); //hago un retardo de 200[mseg]
goto loop; //salto a loop
for(;;) {}
} //fin main (programa principal)

//*********************************************************************
//Función retardo_100ms
//*********************************************************************

void retardo_100ms(word dato){
word i;
TMODH=0xEF;     //en decimal es 61439 -- hace q desborde cada 100ms
TMODL=0xFF;
for(i=0;i<dato;i++){
  /* T1SC: TOF=0,TOIE=0,TSTOP=1,TRST=0,PS2=0,PS1=1,PS0=0 */
  TSC = 0x22;
  TSC_TSTOP=0; //activo contador
  while(TSC_TOF==0){}
  TSC=0x30; //stop y reset de contador
} //fin for
} //fin función

//*********************************************************************
//Función update_ADC Fs= 76.8[KHz]
//*********************************************************************

void update_ADC(void){
  ADSCR=0x20; //Enciendo el canal 0 del ADC
  while(!ADSCR_COCO){ }
  ADX=ADRL;
  ADSCR=0x3F; //apago el ADC
```

```
  ADSCR=0x21; //Enciendo el canal 1 del ADC
  while(!ADSCR_COCO){ }
  ADY=ADRL;
  ADSCR=0x3F; //apago el ADC
  ADSCR=0x26; //Enciendo el canal 6 del ADC
  while(!ADSCR_COCO){ }
  ADZ=ADRL;
  ADSCR=0x3F; //apago el ADC
} //fin función
```

En la función de interrupción serie se escribe el siguiente código:

```
__interrupt void isrINT_SCIReceive(void)
{
word i; //para el for
PTB_PTB7=0;
while(SCS1_SCRF==0){}
SCC2_RE=0; //apago receptor
SCC2_SCRIE=0; //deshabilito interrupcion de Rx serie
if(SCDR=='A'){
  PTB_PTB0=0; //g1=0
  PTB_PTB1=0; //g2=0
} else if(SCDR=='B'){
  PTB_PTB0=1; //g1=1
  PTB_PTB1=0; //g2=0
} else if(SCDR=='C'){
  PTB_PTB0=0; //g1=0
  PTB_PTB1=1; //g2=1
} else if(SCDR=='D'){
  PTB_PTB0=1; //g1=1
  PTB_PTB1=1; //g2=1
} else if(SCDR=='E'){
  SCC2_TE=1; //enciendo transmisor
  for(i=0;i<128;i++){
     update_ADC(); //actualizo ADX, ADY y ADZ
     SCS1_SCTE=0; //borro bit de transmision
     SCDR=ADX; //envío dato x a la pc
     while(SCS1_SCTE==0){} //espero q termine de transmitir
     SCS1_SCTE=0; //borro bit de transmision
     SCDR=ADY; //envío dato y a la pc
     while(SCS1_SCTE==0){} //espero q termine de transmitir
     SCS1_SCTE=0; //borro bit de transmision
     SCDR=ADZ; //envío dato z a la pc
     while(SCS1_SCTE==0){} //espero q termine de transmitir
  } //fin for
SCC2_TE=0; //apago el transimisor
} //fin del último else if
SCC2_RE=1;      //enciendo receptor
SCC2_SCRIE=1;  //habilito interrupcion de Rx serie
} //fin interrupción
```

Software MATLAB

No se ha anexado el código implementado en Matlab debido a que es demasiado extenso, solamente se muestra una imagen de la interfaz gráfica.