Ejemplo N 8:

Comunicación serie asíncrona: Script de Matlab

```
%-----------------------------------------------------------------+
%                  Serial Communication                             +
%-----------------------------------------------------------------+
clear all;close all;clc;
delete(instrfind({'Port'},{'COM3'}));
dataType = 'string'; %'string' , 'number'
buff=14;
s = serial('COM3');
set(s,'BaudRate',9600);
set(s,'Terminator','CR/LF');
set(s,'InputBufferSize',buff);
fopen(s);

if strcmp(dataType,'string')
   charTx = input('Ingrese el caracter de eco: ','s');
   fprintf(s,charTx); %Write text to device
   salida = fscanf(s); %Read ASCII data from device, and format as text
   display(salida)
elseif strcmp(dataType,'number')
   valTx = input('Ingrese el valor de eco: ');
   fprintf(s,valTx); %Write text to device
   salida = fread(s); %Read binary data from device
   plot(salida);grid on
end
fclose(s)
delete(s)
clear s

%-----------------------------------------------------------------------------------------------
```

Comunicación serie asíncrona: Código C.

```
#define AddrSCS 0x400FC1A0
#define AddrCCLKCFG  0x400FC104
#define AddrPCLKSEL0 0x400FC1A8
#define AddrPCLKSEL1 0x400FC1AC
#define AddrCLKSRCSEL 0x400FC10C
#define AddrPLL0CFG 0x400FC084
#define AddrPLL0FEED 0x400FC08C
#define AddrPLL0CON 0x400FC080
#define AddrPLL0STAT 0x400FC088

#define AddrPCONP 0x400FC0C4
#define AddrPINSEL0 0x4002C000
#define AddrU2LCR 0x4009800C
#define AddrU2DLL 0x40098000
#define AddrU2DLM 0x40098004
#define AddrU2IER 0x40098004
#define AddrISER0 0xE000E100
#define AddrU2THR 0x40098000
```

```
#define AddrU2LSR 0x40098014
#define AddrU2RBR 0x40098000

unsigned int volatile * const SCS = (unsigned int *) AddrSCS;
unsigned int volatile * const CCLKCFG = (unsigned int *)AddrCCLKCFG;
unsigned int volatile * const PCLKSEL0 =  (unsigned int *)AddrPCLKSEL0;
unsigned int volatile * const PCLKSEL1 =  (unsigned int *)AddrPCLKSEL1;
unsigned int volatile * const CLKSRCSEL = (unsigned int *)AddrCLKSRCSEL;
unsigned int volatile * const PLL0CFG = (unsigned int *)AddrPLL0CFG;
unsigned int volatile * const PLL0FEED = (unsigned int *)AddrPLL0FEED;
unsigned int volatile * const PLL0CON = (unsigned int *) AddrPLL0CON;
unsigned int volatile * const PLL0STAT = (unsigned int *)AddrPLL0STAT;

unsigned int volatile * const PCONP = (unsigned int *) AddrPCONP;
unsigned int volatile * const PINSEL0 = (unsigned int *) AddrPINSEL0;
unsigned int volatile * const U2LCR = (unsigned int *) AddrU2LCR;
unsigned int volatile * const U2DLL = (unsigned int *) AddrU2DLL;
unsigned int volatile * const U2DLM = (unsigned int *) AddrU2DLM;
unsigned int volatile * const U2IER = (unsigned int *) AddrU2IER;
unsigned int volatile * const ISER0 = (unsigned int *) AddrISER0;
unsigned int volatile * const U2THR = (unsigned int *) AddrU2THR;
unsigned int volatile * const U2LSR = (unsigned int *) AddrU2LSR;
unsigned int volatile * const U2RBR= (unsigned int *) AddrU2RBR;

void clockConfig(void);
void uartConfig(void);
void enviar(char);
int main(void) {
        clockConfig();
        uartConfig();

        while(1){}
        return 0;
}

void uartConfig(void){
        *PCONP |= 1<<24; //UART 2 power/clock control bit.
        *PCLKSEL1 &= !(3<<16); //Peripheral clock selection for UART2: CCLK/4
        *U2LCR |= 3; // Word Length Select: 8-bit character length, Stop Bit Select:1 stop bit,
        // Parity Enable: Disable parity generation and checking, Break Control:Disable break transmission
        *U2LCR |= (1<<7); // Enable access to Divisor Latches
        *U2DLL = 163;  // The UARTn Divisor Latch LSB Register, along with the UnDLM
        *U2DLM = 0;    // register, determines the baud rate of the UARTn.
        *U2LCR &= ~(1<<7); // Disable access to Divisor Latches.
        *PINSEL0 |= (5<<20); //Configure P0.10 as Tx and P0.11 as Rx
        *U2IER = 1;       //Enables the Receive Data Available interrupt for UARTn
        *ISER0 |=(1<<7);  //UART2 Interrupt Enable
}

void UART2_IRQHandler(void){
        char k;
        int i;
        k = *U2RBR;
        char c[] = "hola mundo  \r\n";
        c[11]=k;
        for(i=0;c[i];i++)  //transmit a predefined string
                enviar(c[i]);
}

void enviar (char c)
{
        while((*U2LSR&(1<<5))==0); //check if UnTHR contains valid data or is empty.
        *U2THR = c;
}


void clockConfig (void)
{
        *SCS = 32;  /*Main Oscillator is enabled  */
        while ((*SCS & (1<<6)) == 0);/* Wait for Oscillator to be ready    */
        //-------------------------------------------------------------------------------------------
        *CCLKCFG = 0x3;     /* Setup Clock Divider: pllclk is divided by 4 to produce the CPU clock. */
        //-------------------------------------------------------------------------------------------
        *PCLKSEL0  = 0x0;    /* Peripheral Clock Selection        */
        *PCLKSEL1  = 0x0;
        //-------------------------------------------------------------------------------------------
        *CLKSRCSEL = 0x1;   /* Select Clock Source for PLL0       */
        //-------------------------------------------------------------------------------------------
        *PLL0CFG   = 0x50063;     /* configure PLL0:: M = 100 N = 6     */
        *PLL0FEED  = 0xAA;
        *PLL0FEED  = 0x55;
        //-------------------------------------------------------------------------------------------
        *PLL0CON   = 0x01;          /* PLL0 Enable           */
        *PLL0FEED  = 0xAA;
        *PLL0FEED  = 0x55;
        while (!(*PLL0STAT & (1<<26)));/* Wait for PLOCK0        */
        //-------------------------------------------------------------------------------------------
        *PLL0CON   = 0x03;          /* PLL0 Enable & Connect           */
        *PLL0FEED  = 0xAA;
        *PLL0FEED  = 0x55;
        while (!(*PLL0STAT & ((1<<25) | (1<<24))));/* Wait for PLLC0_STAT & PLLE0_STAT */
}
```