HW #9

ECEN 621

Mrinmoy Sarkar

Date: 11/13/2019

LaunchPad Implementation:

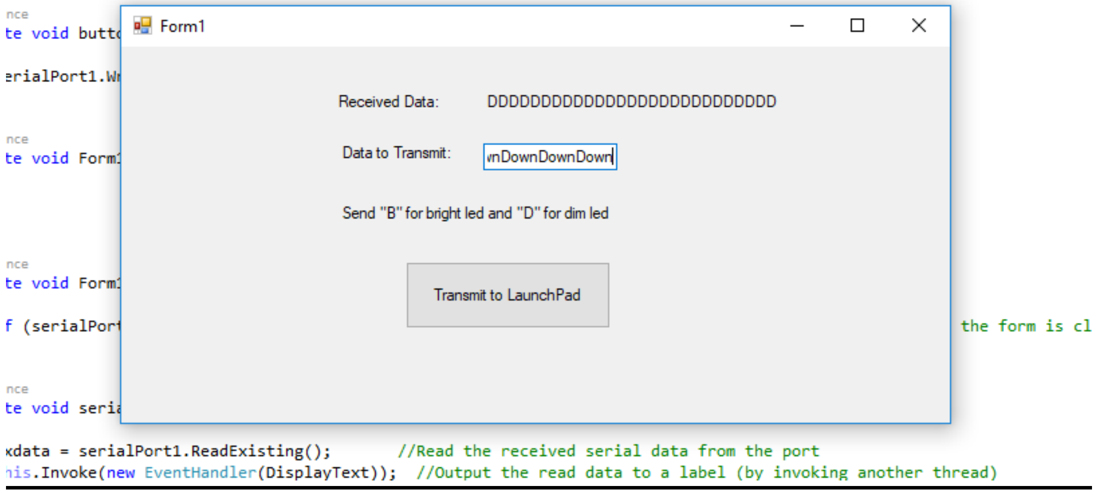
```
//******************************************************************************
#include "msp.h"


volatile uint16_t JOYresult; // store the ADC value of the joystick

// Configure system clock to 12Mz
// Alternatively you may simply define _SYSTEM_CLOCK as 12000000 in the
system_msp432p401r.c file
void clock_config_12MHz(){
    CS->KEY = CS_KEY_VAL;                    // Unlock CS module for register
access
    CS->CTL0 = 0;                           // Reset tuning parameters
    CS->CTL0 = CS_CTL0_DCORSEL_3;           // Set DCO to 12MHz (nominal, center of
8-16MHz range)
    CS->CTL1 = CS_CTL1_SELA_2 |             // Select ACLK = REFO
            CS_CTL1_SELS_3 |                // SMCLK = DCO
            CS_CTL1_SELM_3;                 // MCLK = DCO
    CS->KEY = 0;                            // Lock CS module from unintended
accesses
}

//Congigure neeed I/O ports
void port_config(){
    P1->SEL0 |= BIT2 | BIT3;    // select the eUSCI funtions of P1.2 and P1.3
    P1->DIR &= ~BIT2;           // P1.2 is the receive data (RXD) input pin
    P1->DIR |= BIT3;            // P1.3 is the transmit data (TXD) input pin

//    P1->DIR |= BIT0;            // The Red Led will be used for visual output
//    P1->DIR &= ~(BIT1|BIT4);    // Configure pins P1.1 and P1.4 as inputs for S1
and S2 switches
//    P1->REN |= (BIT1|BIT4);     // Activate pull resistors for S1 and S2
//    P1->OUT |= (BIT1|BIT4);     // Connect Pull resistors to Vcc to make them
pull-up resistors

    //configure pwm for MKII Red led
    P2->DIR |= BIT6;
    P2->OUT |= BIT6;
    P2->SEL0 |= BIT6;
    TIMER_A0->CCR[0]=999; //period
    TIMER_A0->CCR[3]=500; // duty cycle
    TIMER_A0->CCTL[3]=TIMER_A_CCTLN_OUTMOD_7;
    TIMER_A0->CTL=TIMER_A_CTL_TASSEL_2 | TIMER_A_CTL_MC_1 | TIMER_A_CTL_CLR;

    // configure ADC for JoyStick

    // Configure GPIO for ADC
    P4->SEL1 |= BIT4; // Enable A/D channel A9
    P4->SEL0 |= BIT4;

    // Turn on ADC14, extend sampling time to avoid overflow of results
    ADC14->CTL0 = ADC14_CTL0_ON | ADC14_CTL0_MSC | ADC14_CTL0_SHT0__192 |
ADC14_CTL0_SHP | ADC14_CTL0_CONSEQ_3;
    ADC14->MCTL[0] = ADC14_MCTLN_INCH_9; // ref+=AVcc, channel = A9
    ADC14->MCTL[1] = ADC14_MCTLN_EOS;

    ADC14->IER0 = ADC14_IER0_IE1; // ADC interrupt enable for ADC14->MEM[1]
    SCB->SCR |= SCB_SCR_SLEEPONEXIT_Msk; // sleep on exit
    // Start conversion-software trigger
    ADC14->CTL0 |= ADC14_CTL0_ENC | ADC14_CTL0_SC;
    NVIC->ISER[0] = 1 << ((ADC14_IRQn) & 31); // Enable ADC interrupt in NVIC
module

}

// Configure the eUSCI for 9600 Baud UART communication
void uart_config(){
    EUSCI_A0->CTLW0 |= EUSCI_A_CTLW0_SWRST; // Put eUSCI in reset
    EUSCI_A0->CTLW0 = EUSCI_A_CTLW0_SWRST | // Remain eUSCI in reset
            EUSCI_B_CTLW0_SSEL__SMCLK |      // Configure eUSCI clock source for
SMCLK
            EUSCI_A_CTLW0_PEN | // enable parity
            EUSCI_A_CTLW0_PAR; //even parity

    // Baud Rate 19200
    EUSCI_A0->BRW = 39;                    // 12000000/16/9600
    EUSCI_A0->MCTLW = (0x00 << EUSCI_A_MCTLW_BRS_OFS) | (1 <<
EUSCI_A_MCTLW_BRF_OFS) |
            EUSCI_A_MCTLW_OS16;

    EUSCI_A0->CTLW0 &= ~EUSCI_A_CTLW0_SWRST; // Initialize eUSCI
    EUSCI_A0->IFG &= ~EUSCI_A_IFG_RXIFG;    // Clear eUSCI RX interrupt flag
    EUSCI_A0->IE |= EUSCI_A_IE_RXIE;        // Enable USCI_A0 RX interrupt
}
// Main function
int main(void)
{
    clock_config_12MHz(); //Or simply define _SYSTEM_CLOCK as 12000000 in the
system_msp432p401r.c file
    port_config();
    uart_config();

    NVIC->ISER[0] = 1 << ((EUSCIA0_IRQn) & 31);// Enable eUSCIA0 interrupt in NVIC
module

    __enable_irq();// Enable global interrupt
    // Enter LPM0
    __sleep();


}

// UART interrupt service routine for Bright and dim of Red Led OF Booster Pack
void EUSCIA0_IRQHandler(void)
{
    if (EUSCI_A0->IFG & EUSCI_A_IFG_RXIFG) // Whenever a character is received
    {

        while(!(EUSCI_A0->IFG & EUSCI_A_IFG_TXIFG)); // Check if the TX buffer is
empty first
        //EUSCI_A0->TXBUF = EUSCI_A0->RXBUF;   // Echo received from the PC
character back to the PC
        if(EUSCI_A0->RXBUF =='B')      //If a "B" is received from the PC Keypboard
        {
            TIMER_A0->CCR[3]=900; // duty cycle
        }
        else if(EUSCI_A0->RXBUF =='D')      //If a "D" is received from the PC
Keypboard
        {
            TIMER_A0->CCR[3]=100; // duty cycle
        }
    }
}

// ADC14 interrupt service routine
void ADC14_IRQHandler(void)
{
    if (ADC14->IFGR0 & ADC14_IFGR0_IFG1)
    {
        JOYresult = ADC14->MEM[0]; // Move A9 results, IFG is cleared. vertical
joystick

        if(JOYresult < 3000)
        {
            EUSCI_A0->TXBUF = 'D'; // send 'D' to PC
        }
        else if(JOYresult > 10000)
        {
            EUSCI_A0->TXBUF = 'U'; // send 'U' to PC
        }
        __delay_cycles(2000); //small delay
    }
}
```

Windows Form App:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        String rxdata;  // serial data received
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            serialPort1.PortName = "COM8";  // Choose the UART serial port associated
with the MSP432 LaunchPad (See "Device Manager")
            serialPort1.BaudRate = 19200;    // Use the same baudrate that the MSP432
is configured to
            serialPort1.Parity = System.IO.Ports.Parity.Even; // set even parity
            serialPort1.Open();             // Open the UART serial port for
communication witht he MSP432 LaunchPad
        }

        private void button1_Click(object sender, EventArgs e)
        {
            serialPort1.Write(textBox1.Text);      // Send the data from the textbox to
the serial port
        }

        private void Form1_FormClosed(object sender, FormClosedEventArgs e)
        {

        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (serialPort1.IsOpen) serialPort1.Close();      // Close the connection
to the MSP432 Launchpad when the form is closed (odApp exited)
        }

        private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
        {
            rxdata = serialPort1.ReadExisting();       //Read the received serial data
from the port
            this.Invoke(new EventHandler(DisplayText));  //Output the read data to a
label (by invoking another thread)
            if (rxdata.Contains("D"))               //If the recieved data is "D"
            {
                SendKeys.SendWait("Down");    //so send the letters "D", "o","w", and
"n" to the current Windows App in the forground (whatever the App may be)
            }
            else if (rxdata.Contains("U"))           //If the recieved data is "U"
            {
                SendKeys.SendWait("Up");    //so send the letters "U", and "p" to the
current Windows App in the forground
            }
        }
        private void DisplayText(object sender, EventArgs e)
        {
            label3.Text = rxdata;        //Output the received data to the label
        }
    }
}
```

Screenshot of Windows Form App:



```
nce
te void butt
erialPort1.Wr
nce
te void Form1
nce
te void Form1
f (serialPort                                          the form is cl
nce
te void seri
xdata = serialPort1.ReadExisting();       //Read the received serial data from the port
his.Invoke(new EventHandler(DisplayText));  //Output the read data to a label (by invoking another thread)
```

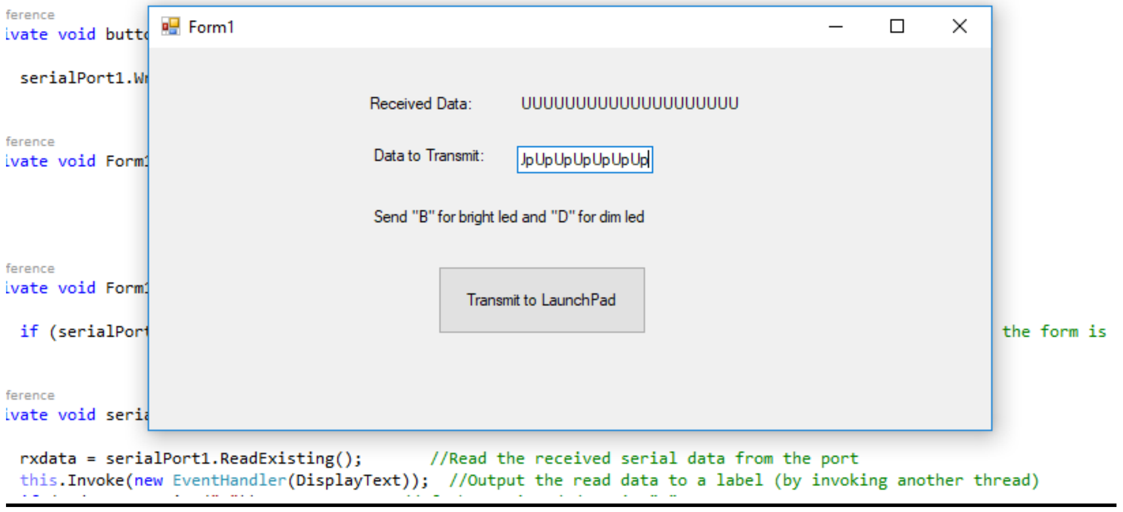
Form1

Received Data: DDDDDDDDDDDDDDDDDDDDDDDDDDD

Data to Transmit: wnDownDownDown

Send "B" for bright led and "D" for dim led

Transmit to LaunchPad



```
ference
ivate void butt
  serialPort1.Wr
ference
ivate void Form1
ference
ivate void Form1
  if (serialPort                                        the form is
ference
ivate void seri
  rxdata = serialPort1.ReadExisting();       //Read the received serial data from the port
  this.Invoke(new EventHandler(DisplayText));  //Output the read data to a label (by invoking another thread)
```

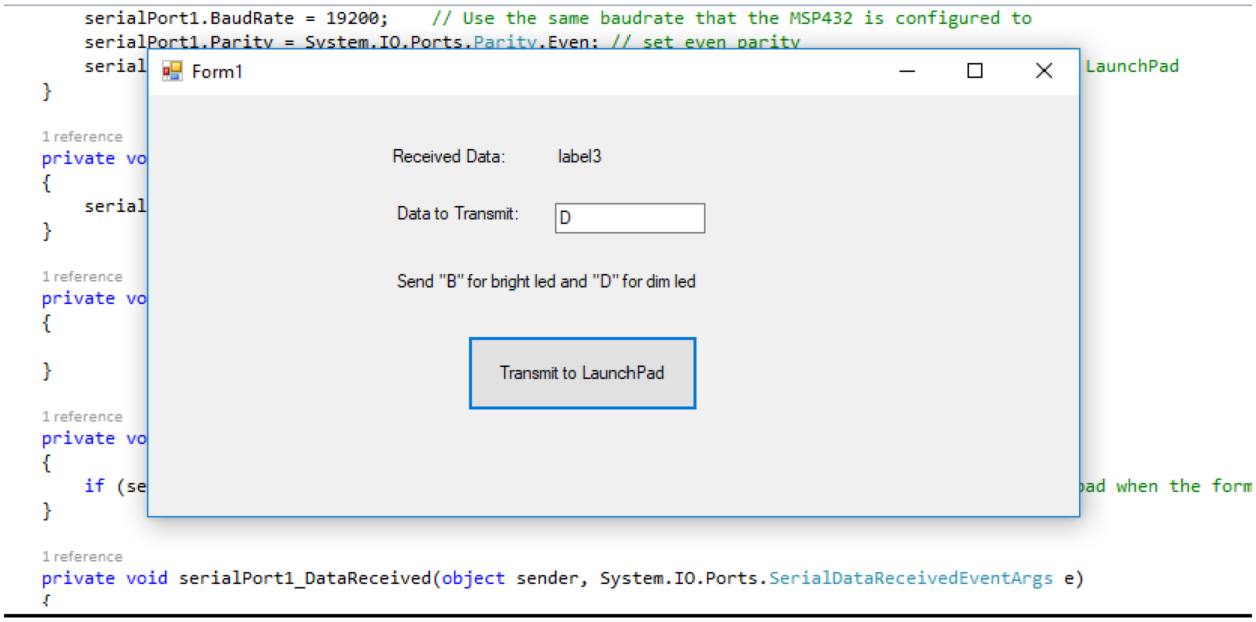
Form1

Received Data: UUUUUUUUUUUUUUUUUUUU

Data to Transmit: JpUpUpUpUpUpUp

Send "B" for bright led and "D" for dim led

Transmit to LaunchPad



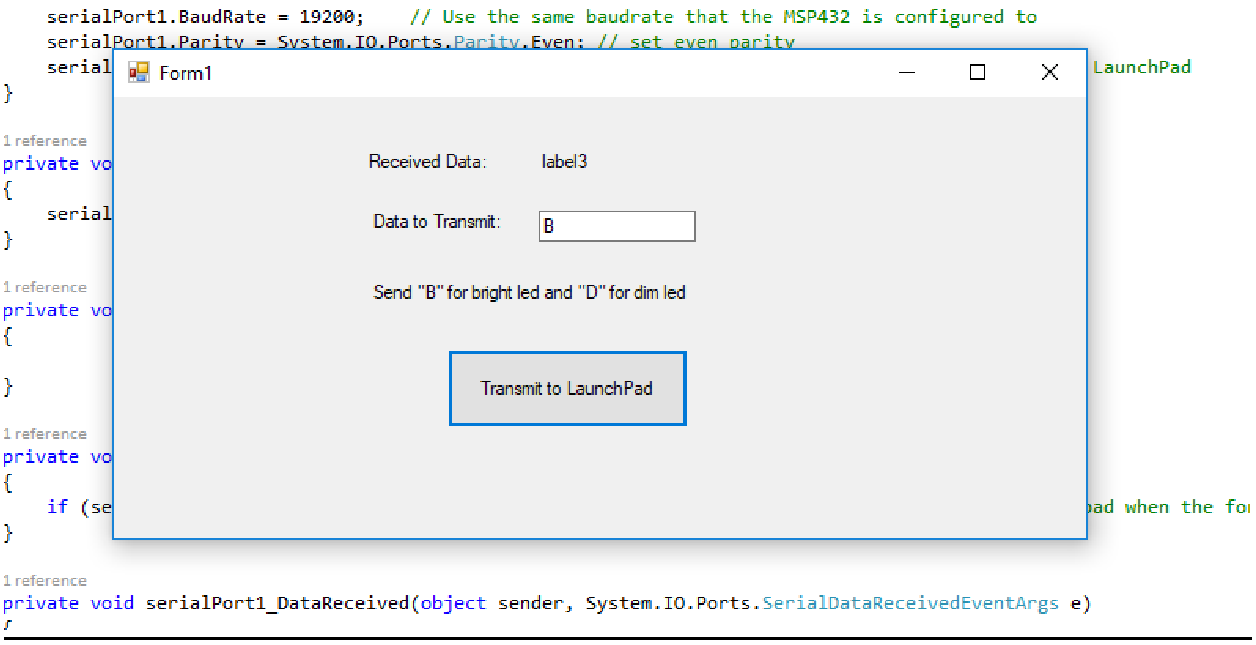
```
        serialPort1.BaudRate = 19200;    // Use the same baudrate that the MSP432 is configured to
        serialPort1.Parity = System.IO.Ports.Parity.Even; // set even parity
        serial                                                                              LaunchPad
    }

    1 reference
    private vo
    {
        serial
    }

    1 reference
    private vo
    {

    }

    1 reference
    private vo
    {
        if (se                                                                              ad when the form
    }

    1 reference
    private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
```

Form1

Received Data: label3

Data to Transmit: D

Send "B" for bright led and "D" for dim led



```
    serialPort1.BaudRate = 19200;    // Use the same baudrate that the MSP432 is configured to
    serialPort1.Parity = System.IO.Ports.Parity.Even; // set even parity
    serial                                                                                  LaunchPad
}

1 reference
private vo
{
    serial
}

1 reference
private vo
{

}

1 reference
private vo
{
    if (se                                                                                  ad when the fo
}

1 reference
private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
```

Form1

Received Data: label3

Data to Transmit: B

Send "B" for bright led and "D" for dim led