**Pre-Lab Requirements**

There are no pre-lab requirements for this Lab. 😊Just make sure you and your partner both bring your laptops, LaunchPads, mypartsKits and breadboards to lab this week and next week.

**What You Need to Bring to Lab** (This is 1 point of your lab grade.)

* 2 MSP430s
* Laptop (The Lab computers are slow and sometimes unreliable, so bring yours.)
* TI myPartsKit & Breadboard (You will need many jumper wires and male to female connectors, so both partners need to bring his/her myParts Kit.)

**Serial Communication**

Background

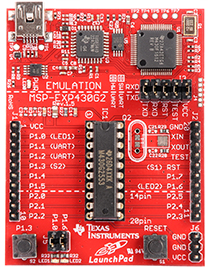
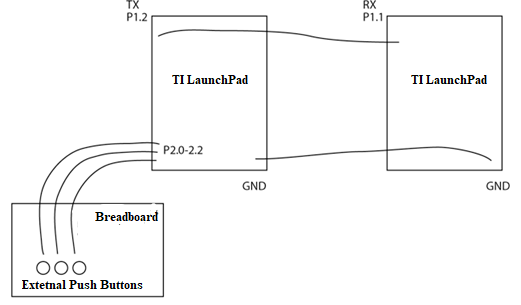
There are two categories of serial communications: synchronous and asynchronous. Synchronous serial communication uses a dedicated communication channel to send a clock signal, which provides the timing between the transmitting and receiving devices.  Asynchronous communication uses an agreed upon baud rate that is relatively slow compared to typical synchronous communication rates. There are several versions of both synchronous and asynchronous serial communication.  The MSP430 microcontroller offers three different types of communications using its USCI (Universal Serial Communication Interface) peripheral: [SPI](http://www.ece.utep.edu/courses/web3376/msp430_spi_demo.html) (synchronous), [IIC](http://www.ece.utep.edu/courses/web3376/msp430_iic_demo.html) (synchronous) and UART (asynchronous).   We will use UART (Universal Asynchronous Receive and Transmit) for this lab.

Although PC port types are being replaced with much faster interfaces like USB, the simplicity of the serial port ensures its continued use in embedded applications.  For **wired transmission and reception**, the LaunchPad uses Pin P1.2 for transmission and Pin P1.1 for reception, as shown in Fig 10.1.

Lab Objective

In this Lab we will develop a simple transmitter - receiver module in which the two microcontrollers will exchange 1-byte messages. Based on the user input from external push buttons the transmitter will generate three different signals. On the receiver side, the microcontroller will (1) trigger different patterns on the built-in LEDs and (2) will display the 1-byte message sent by the transmitter on an LCD.

**Two weeks will be dedicated to this lab. Part 1 (LED patterns) will be completed during the first week and part 2 (LCD display) will be completed during the second week.**

Receiver (P1.1)

Transmitter(P1.2)

Fig 10.1- Transmitting and Receiving Pins on the MSP430

**Submit Homework** (after the lecture)

Have your laptop open and the code for the homework assignment given in Lab 9 (Procedure Part 6) ready to run. Your instructor will install the code from your laptop to your LaunchPad and verify correct operation. You and your partner must each demonstrate homework as it is an individual rather than a group assignment.

Note: When you test your homework at home and get it working, make sure to take a video clearly showing your results and explaining how you got them. **Your face must be seen** in the video. This will help us grade you effectively, in case you are unable to bring your laptop to the Lab or you face any technical issues while running the code in Lab. No points will be deducted if your video is clear and original.

**Procedure** (Now you may work with your partner as a team.)

Part 1 – Connecting External Buttons to the Transmitter

You will need 3 external push buttons, which will be provided by your instructor. Connect these buttons to the Transmitting LaunchPad as shown in Figure 10.2. Make sure you place the legs of the push button such that they fall symmetrically across the middle of the board. Two legs should lie in the column marked ‘d’ and the other two should lie in the column marked ‘g’, as shown in Figure 10.3. Connect one push button to each of the LaunchPad pins P2.0, P2.1 and P2.2. **Use 330Ω (pull down) resistors between the LaunchPad pins and Ground - as shown in Figure 10.2 - to tie the LaunchPad pins to ground while the button is not being pressed.** Use of these pull-down resistors minimizes the chance of floating values, which may introduce errors. **When a button is pressed, the corresponding LaunchPad pin will be connected to Vcc.** Note: When writing your code, keep the “buttonstate” high. The power point slide set titled "External Buttons" (from Dr. Florence) has been uploaded to the Laboratory folder on e-Learning as an additional reference for you.

A circuit board

Description generated with high confidence

Figure 10.2 - Interfacing External Push Buttons with the TI LaunchPad

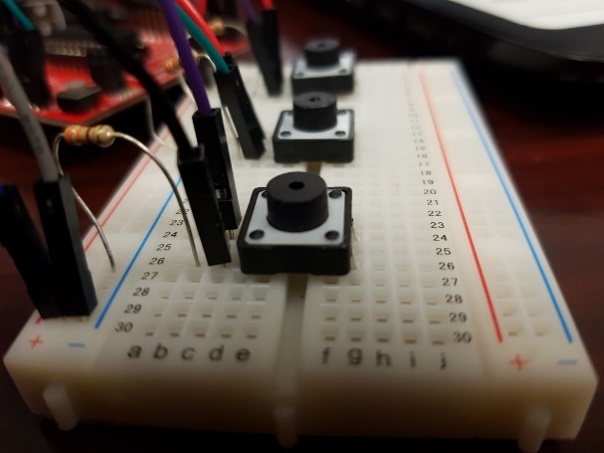


Figure 10.3 - Placing the Push Buttons in Columns ‘d’ and ‘g’

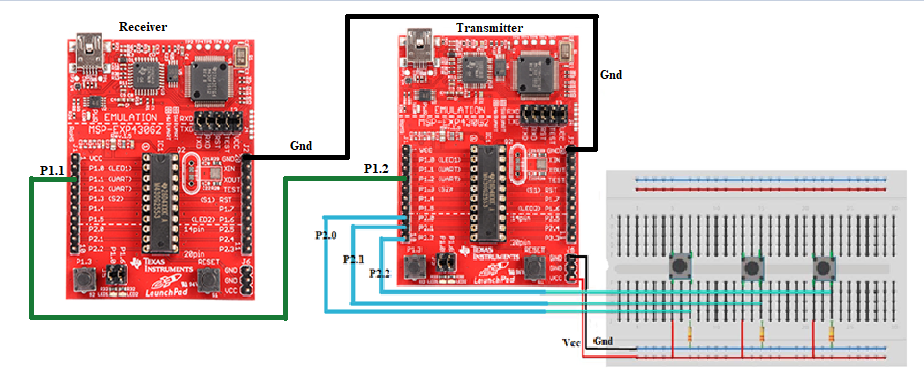


Figure 10.4 - Transmitter and Receiver Hardware Connections

Part 2 – Building the Transmission Channel

Create a wired transmission channel from one LaunchPad to another by connecting pin P1.2 of the transmitting LaunchPad to pin P1.1 of the receiving LaunchPad as shown in Figure 10.4.

Part 3 – Energia Code for the Transmitter and Receiver

1. Because you will be using external push buttons for the transmitter and an LCD to view the received transmission, the codes for the LaunchPads serving as transmitter and receiver will be different. One member of each team should write the code for the transmitter while the other will write the code for the receiver. Both codes are easy enough and use the concepts for push-buttons, LEDs and LCDs used in previous labs.
2. The built-in function needed to Code for the Transmitter is- **Serial.write().** This function writes binary data to Serial Ports. Here in our case the serial port is the Transmitting port P1.2. So, Serial.write(value) will send a value to Pin P1.2 of the LaunchPad which will then get transported via the Female-Female jumper wire to the Receiving port P1.1 of the Receiver. Additional information on the syntax of this function is given at- <http://energia.nu/Serial_Write.html> .
3. Since you need to transmit a different 1-byte value for each of your three buttons, choose three different integer values to write using the Serial.write(value) call. Each call should be inside an if-else statement, which will be called when the ButtonState of one of the push buttons is HIGH (i.e. the button is pressed).
4. The built-in functions needed to Code for the Receiver are **Serial.available() and Serial.read().** Serial.available() tell the processor how many bytes have been stored in the serial receive buffer. Serial.read() reads the data from the buffer. The syntax for Serial.available() is given at- <http://energia.nu/Serial_Available.html>, and the syntax reference for Serial.read() is given at- <http://energia.nu/Serial_Read.html> .
5. Store the Serial.read() value in a variable. This variable value tells which button has been pressed. Implement an if-else statement or switch-case logic to control the receiver's LEDs as follows:

* When the **first transmitter button** is pressed, the **receiver's red LED** should blink.
* When the **second transmitter button** is pressed, the **receiver's green LED** should blink.
* When the **third transmitter button** is pressed, **both of the receiver's LEDs** should blink.

Table 10.1 summarizes the logic for the communication link.

Table 10.1 - Communication Between the Transmitter and Receiver

|  |  |
| --- | --- |
| **Transmitter** | **Receiver** |
| Button 1 pressed | Red LED turns ON |
| Button 2 pressed | Green LED turns ON |
| Button 3 pressed | Both LEDs turn ON |
| No button pressed | Both LEDs OFF |

1. **Demonstrate your transmitter-receiver function to your instructor. This week's lab grade will depend on how well it works. If you have time, you may continue with Part 4 or you may begin working on the homework assignment, which is given in Part 6 of the Procedure.**

Part 4 – Adding an LCD to the Receiver (Week 2)

1. Check your trans-receiver to make sure it is working properly.
2. Recreate the LCD interface from Lab 9 on the receiver side of your trans-receiver.
3. Modify the receiver code such that when the buttons are pressed on the transmitter, the LCD display output corresponds to the activity of the receiver's LEDs. Table 10.2 summarizes the expected behavior of your trans-receiver.

Table 10.2 - Final Trans-Receiver Results (to be presented Week 2)

|  |  |  |
| --- | --- | --- |
| **Transmitter** | **Receiver Board** | **Receiver LCD Display Message** |
| Button 1 pressed | Red LED turns ON | “Red LED On” |
| Button 2 pressed | Green LED turns ON | “Green LED On” |
| Button 3 pressed | Both LEDs turns ON | “Both LEDs On” |
| No button pressed | Both LEDs OFF | “Hello EE/CE 1202” |

1. Include appropriate delays in your code so that the LCD is easy to read. Keeping the button pressed should give the required Display while releasing it should display – “**Hello EE/CE 1202”** or any text of your choice.
2. **Demonstrate your results to your instructor.**

Part 5 – Wrap-up

1. As you cannot take the LCD home, you will be graded on whatever could be completed during your lab session. If you complete the lab work early you may start on the final homework assignment (See Part 6.)
2. At the end of your lab session, unplug the LaunchPad from your computer and disconnect your circuit.
3. Return all lab supplies to their proper locations. Pack up your belongings and take them with you. Leave your workstation at least as clean as you found it.

Part 6 – Homework

1. This last homework assignment is a small project which you and your partner will complete together. You will present your results to the class during your lab session the week of April 24-27.
2. You have three options for this final project, each of which is described in the following pages. All projects will be graded on the same scale. You may go beyond the requirements of the project you select to earn bonus points. In addition, we have included an extra credit (optional) challenge.
3. You will be allowed 3 minutes for your oral presentation, which should be organized as follows:

* Each partner will have one minute to explain his/her contribution to the project. Both partners MUST speak.
* You will present a video no more than one minute in length demonstrating your project results. Both team members must be seen in the video.