Lab 4

Communication Between Two Microcontrollers

EEL 4746C / EEL 5756C: Microcomputers

Fall 2018

Student Name:	
Student ID:	
Section:	

Contents

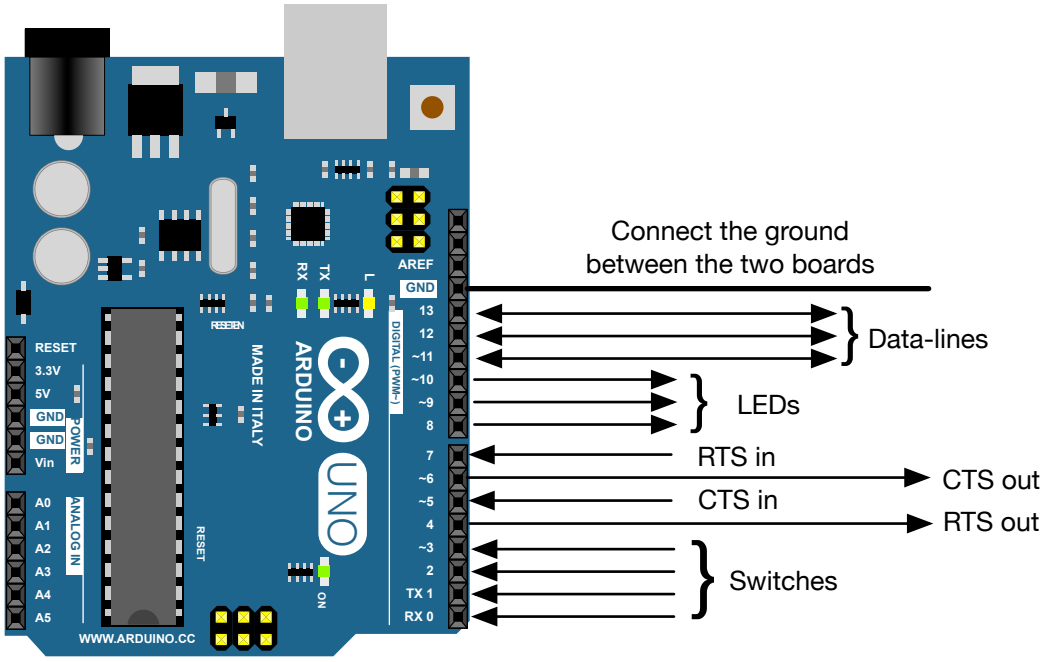
1 Objective

The objective of this lab is to be able to implement a simple asynchronous communication protocol between two microcontroller boards.

2 Introduction

This experiment will introduce you to the interface two microcontroller boards using a very simple communication. protocol that you are going to implement for this lab. The two microcontrollers will have symmetric connections and they will be running the same code. Each microcontroller will have four (4) switches connected to it and three LEDs. Three of the switches will be used to represent a three digits binary numbers the 4th switch is used to indicate the 3-bit number is ready to be sent. Whenever the microcontroller sense that it has a number ready to be sent, it will check if the second microcontroller is sending. If not, then it will begin the send procedure. Otherwise, you need to think of a way to differ from sending. When the microcontroller receives a number from the other microcontroller, it will output that on the LEDs.

2.1 The schematic



2.2 Notes

1) Make sure to connect the ground between the two boards.
2) For the data-lines connection between the two boards use a 1kΩ resistor between the two boards.
3) Make sure that you start your board with both boards having their data-lines set as inputs.
4) At any time, only one board can have the data-lines as outputs.
5) Before connecting any of the boards to one another, make sure to check your circuit and code with the instructor or the TA.

3 Guideline

1. Write a non-blocking subroutine that detects if bit 3 of port D data register bit went from 0→1→0. Hint: Use register two (R2) to store the state.
2. Write a subroutine that will read the 3-bit number from the least significant 3-bits of port D and store them in register 3 (R3).
3. Write a subroutine that will write a 3-bit number in register 4 (R4) to the LEDs on the least significant 3-bits of port B.
4. Write the pre-send subroutine that will read the RTS in, if zero then it will set RTS out, sleep for *n*-seconds and then read CTS in. If CTS in is 1, then proceed to send.
5. Write the send routine that will set the data-lines as outputs and put the data on the data-lines.
6. Write the post-send subroutine that will release the control lines and set the data-lines back to being inputs.
7. Write the subroutine that detects RTS in, sends the CTS out, and write the data to register 4 (R4).
8. The details of the communication protocol is left to you to determine.
9. Check each subroutine separately.