Lab #2
USART

(w/o printf and UART Setup)

Ver 1.00

PART I

Using the code found in the ATmega328P datasheet (http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf) Chapter 20 Pages 185, 186 and 189. Combine those functions into one ATMEL Project (making sure NOT to turn on UART in the project setup).

NOTE – for registers in the sample code, replace the lower case n with a 0 (zero). Example: UCSRnB will be UCSR0B in your code.

Modify USART_Init so that the USART is set up for NO STOP BITS (see register UCSR0B).

In main() call the USART_Init (to set up the USART) then, using the USART_Transmit() function multiple times, send your initials to the USART port.

Note: you do need to include the atmel_start.h library, but DO NOT include the ateml_start_init(); line in your main().[1]

Demo this code to the instructor.

PART II

Using the functions that already exist, get a character from the USART using USART_Recieve. Add 1 to the value received then print the new character using USART_Transmit.

Demo this code to the instructor.

PART III

Create a new function called USART_SendStr that takes a STRING (aka a char array) and prints each character in the array, by calling the USART_Transmit function. You will need to use strlen to determine the length of the string and a FOR loop to accomplish this.

Print "hello world" to the usart using the new function.

Note: you will need to include the string.h library.

Demo this code to the instructor.

[1] Atmel_start.h allows us to use the register names for the ATmega328p (indirectly) so the library is needed, but the function call does something to disrupt control of the USART so it cannot be included.

PART IV

Using the function from Part II, print to the USART "Count = x" where x is the value of an integer variable that is incremented by one each loop. You will need to use the C function itoa() to change the integer value to a string.

PART V

For this part we will be using a 16x2 SerLCD. NOTE: This is a 3.3v device and the ATmega328p is a 5v device. To ensure that you do not damage the SerLCD, a 5v to 3.3v Logic Level Bi-Directional Converter **MUST** be used.

Here is the connections for the device:

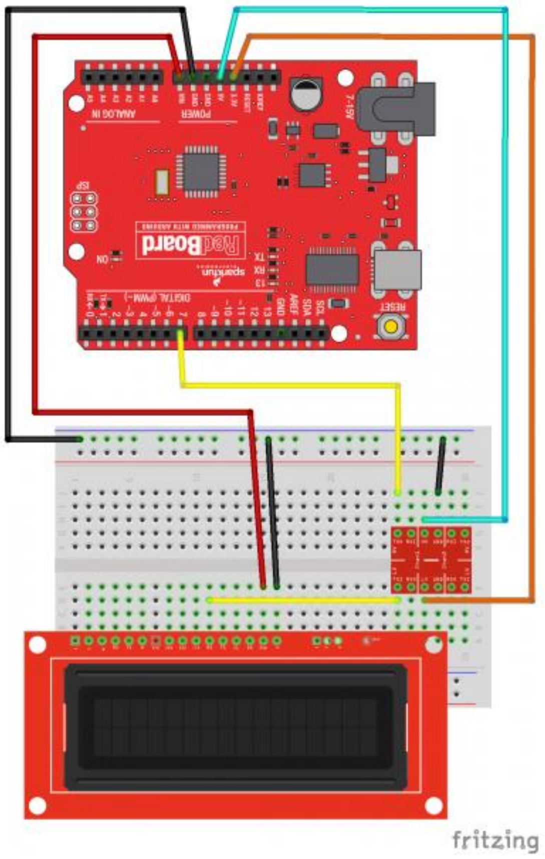


Figure 1- SerLCD, Level Converter and Arduino WIring[2]
NOTE -Rx and Tx will not match above (see next page)

[2] From https://learn.sparkfun.com/tutorials/avr-based-serial-enabled-lcds-hookup-guide/serial-uart-hardware-hookup

A closeup of the Level Converter follows:

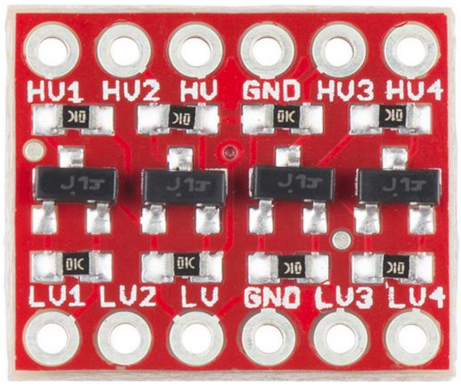


Figure 2- 3.3V / 5v Bi-directional level converter[3]

HV pin goes to +5v

LV pin goes to +3.3V on Arduino and + on SerLCD

Ground (HV side) goes to Arduino GND pin

Ground (LV side) goes to SerLCD - Pin

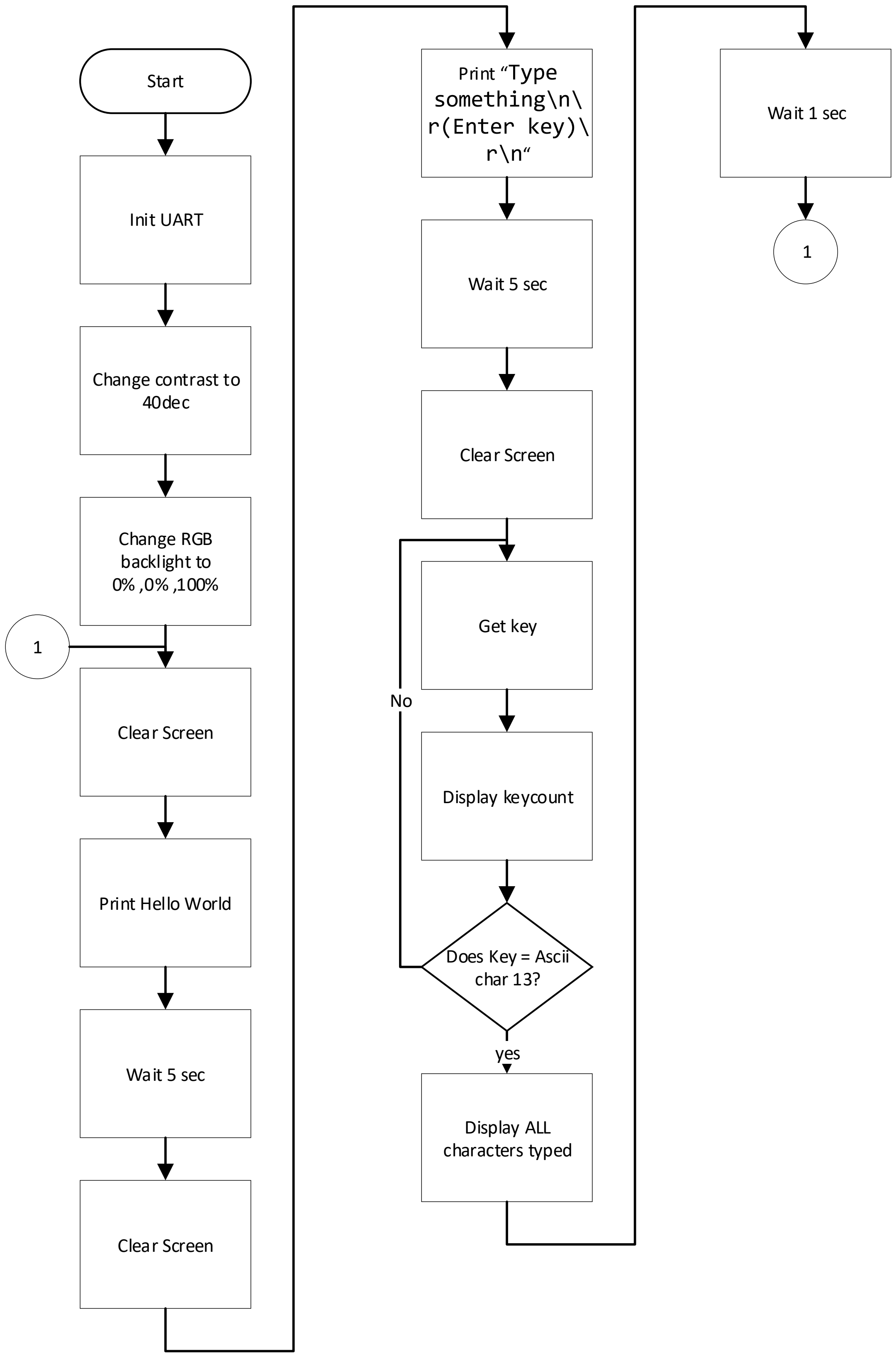
HV1 goes to TX on Arduino

LV1 goes to RX pin on SerLCD

NOTE: To send code to the Arduino from the PC, disconnect the TX line from the Arduino or the SerLCD might interfere with the code transmission.

Now that the LCD is wired, write a program that will do the following (using the routines already written in previous parts):

[3] From https://www.sparkfun.com/products/12009


Start
Init UART
Change contrast to 40dec
Change RGB backlight to 0% ,0% ,100%
1
Clear Screen
Print Hello World
Wait 5 sec
Clear Screen
Print "Type something\n\r(Enter key)\r\n"
Wait 5 sec
Clear Screen
Get key
Display keycount
No
Does Key = Ascii char 13?
yes
Display ALL characters typed
Wait 1 sec
1

For special commands, see table below[4]. So, to clear the screen you would have to send "|-" or "\x7c\x2d" (\x means send hex code that follows).

ASCII	DEC	HEX	Description
'|'	124	0x7C	Enter Settings Mode
ctrl+m	13	0x0D	Change baud to 9600bps
ctrl+x	24	0x18	Change the contrast. Follow Ctrl+x with number 0 to 255. 40 is default.
'+'	43	0x2B	Set RGB Backlight. Follow this command with three bytes representing Red, Green, Blue, backlight values. Supported on v1.1+ of SerLCD.
'-'	45	0x2D	Clear display. Move cursor to home position.
n/a	128-157	0x80-0x9D	Set the primary backlight brightness. 128 = Off, 157 = 100%.
n/a	158-187	0x9E-0xBB	Set the green backlight brightness. 158 = Off, 187 = 100%.
n/a	188-217	0xBC-0xD9	Set the blue backlight brightness. 188 = Off, 217 = 100%.

Demo this part to the instructor, comment your code and submit the full project via online submission.

[4] From https://learn.sparkfun.com/tutorials/avr-based-serial-enabled-lcds-hookup-guide/firmware-overview