The University of Arizona
Department of Aerospace and Mechanical Engineering
Mechatronics Laboratory
Instructor: Professor. Eniko T. Enikov, enikov@email.arizona.edu

Laboratory Task Sheet 15

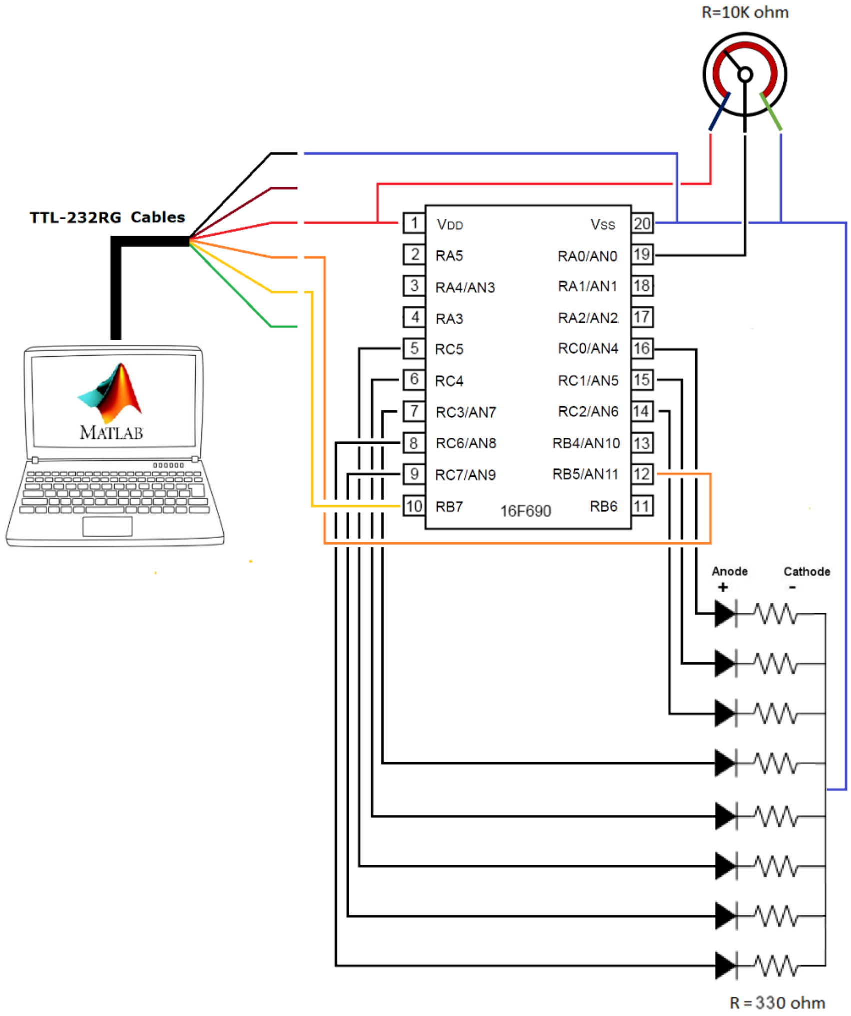
Title: RS232 Communication

Registers to be learned: TXSTA & BAUDCTL & SPBRG & TXREG & RCSTA & PIR1 & RCREG

Objective: Program the microcontroller to read the potentiometer's analog signal via Analog to Digital Convertor (ADC) and send the result of the conversion to MATLAB by using the TTL-232RG Cable and plot the data in real time. In MATLAB, divide the received data by four and send the result back to the Microcontroller and display it on the Linear Array of LEDs.

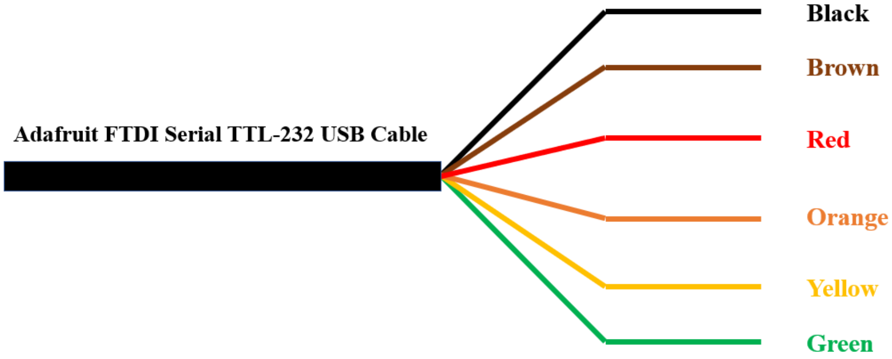
Tasks

1. Create the circuit below using a linear array of LEDs, a bank of resistors, a potentiometer, and a TTL-232RG Cable. **(In this task the circuit is powered through the cable)**



TTL to USB Serial Converter Generic Cable

The TTL-232RG generic cables are a family of USB to TTL serial UART converter cables incorporating FTDI's FT232RQ USB to Serial UART interface IC device which handles all the USB signaling and protocols. The cables provide a fast, simple way to connect devices with a logic level serial interface to USB. Each TTL-232RG generic cable contains a small internal electronic circuit board, utilizing the FT232R, which is encapsulated into the USB connector end of the cable. The other end of the cable is wire ended. The cables can be used for "TTL" or interface logic over a range to voltage levels.



Pin 1	**GND**	**Ground Supply**
Pin 2	**CTS # Input**	**Clear to Send Control Input Handshake Signal**
Pin 3	**VCC**	**Power Supply**
Pin 4	**TXD Output**	**Transmit Asynchronous Data**
Pin 5	**RXD Input**	**Receive Asynchronous Data**
Pin 6	**RTS # Output**	**Request To send Control Output Handshake Signal**

2. Make a copy of the P16f690_Template file and name it TASK15Group00. Open the file in MPLAB Software and use the table below to construct the code.

Suggested Code Structure
Define all the necessary Memory Bytes
Start
Call Initialization **Go to Main**
Main Call Delay Use ADCON0 register to initiate the conversion Wait until the conversion is done BANK1 Use ADRESL Register to move the low byte of the conversion to the Work Register BANK0 Call Send Use ADRESH Register to move the high byte of the conversion to the Work Register Call Send Call Receive Display received byte on PORTC Go to Main
Send Send the Work Register to TXREG Register Use RCSTA Register to enable Continuous Send BANK1 Check TXSTA to see the Transmit Shift Register Status If it is full, stay here If it is empty (data is sent), go ahead BANK0 Return
Receive Use RCSTA Register to enable Asynchronous Mode Receiver Use PIR1 Register to check if the buffer is empty (data is received) Move the received data from RCREG Register to work register. Return
Delay Make a delay for 20 µs Return
Initialization Bank2 Use ANSEL and ANSELH Registers to define all the ports as digital Use ANSEL and ANSELH Registers to define PORTA0 as analog Bank1

```
Use OSCCON Register to set the oscillator on 8 MHz
Use ADCON1 Register to set the ADC (Analog to Digital Convertor) clock on FOSC/16
Use TXSTA Register to empty Transmit Shift Register
Use TXSTA to set Baud Rate on High
Use TXSTA Register to set the Transmission on the Asynchronous Mode
Use TXSTA Register to enable Transmission
Use BAUDCTL Register to set Baud Rate Generator on 8 bits
Move decimal 25 to SPBRG Register
Use TRISA Register to define PORTA0 as input
Use TRISB Register to define PORTB5 as input
Use TRISB Register to define PORTB7 as output
Use TRISC Register to define PORTC as output
Bank0
Use ADCON0 Register to enable ADC
Use ADCON0 Register to set PORTA0 as input channel of the convertor
Use ADCON0 Register to make the convertor Right Justified
Return

end
```

3. Program the microcontroller and test it on the circuit.
4. Demonstrate the result to the instructor.
5. Upload the code on D2L and save it for yourself.