RS-232 Serial Communication Tutorial – Software
SERIN2 and SEROUT2 Tutorial
Cornerstone Electronics Technology and Robotics II

- **Administration:**
  - Prayer
- **Related Tutorial:**
  - RS-232 Serial Communication Tutorial - Hardware See: http://www.cornerstonerobotics.org/curriculum/lessons_year2/erii_rs232_1.pdf
- **General PicBasic Pro Program Listing:** http://www.cornerstonerobotics.org/picbasic.php
- **Tera Term Pro 3.1.3:**
  - Communication between a PC and a PIC microcontroller requires a terminal program (terminal emulator) installed on the PC. In 2002 Ayera Technologies released TeraTerm Pro 3.1.3
  - To download TeraTerm Pro 3.1.3, see: http://www.ayera.com/teraterm/download.cfm and download TeraTerm Pro Web 3.1.3.
  - To use TeraTerm Pro 3.1.3, open the ttpro313 folder.
  - Then click on the icon:

    Ttermpro.exe

    The icon may have ttermpro, Tera Term Pro, T. Teranishi beside it.

  - When the message box Tera Term: New Connection appears, select Serial and press the OK button.
- **HyperTerminal:**
  - Another terminal program that can be used is Microsoft's HyperTerminal.
  - Every Windows XP comes with HyperTerminal. HyperTerminal is found in your start menu via, Start Menu > Accessories > Communications > HyperTerminal.
  - Watch the HyperTerminal setup video at: http://www.box.net/shared/or7jm5xk8b.
  - At the Connection Description screen, type in a name such as 9600_link and choose any icon. Press the OK button.
  - At the Connect To screen, set the "Connect using:" to the proper com port – for example COM1. Press OK.
  - At the COM1 Properties screen, make the following settings:

Bits per second	**9600 not 19200** as shown in the video.
Data bits	8
Parity	None
Stop bits	1
Flow control	None

    Press the OK button

  - To use HyperTerminal again, it is found via, Start Menu > Accessories > Communications > HyperTerminal > Your HyperTerminal File Name.
  - Press OK.

- **New PicBasic Pro Commands for RS-232 Serial Communication:**
  - **SEROUT2:**
    - Format:

      SEROUT2 DataPin{\FlowPin},Mode,{Pace,}{Timeout,Label,}[Item...]

      Explanation:

      At this point, we will shorten the format to:

      SEROUT2 DataPin, Mode, [Item1]

      Send one or more Items to DataPin in standard asynchronous format. Mode is used to specify the baud rate and operating parameters of the serial transfer.

      For a more extensive explanation of SEROUT2, see:
      http://www.melabs.com/support/index.htm
      then look under the Compiler Documentation: and click on PICBASIC PRO Compiler Manual. Look for SEROUT2 command at about page 142 in the manual.

      Examples:

      SEROUT2 5, MODE, ["hello", 10, 13]

      Format: SEROUT2 Pin, Mode, [Item1]
      Pin = 5 (RB5 the transmit pin)
      Mode = 84 (9600 baud rate)
      [Item1] = ["hello", 10, 13]
      Transmits hello, then
      10 (the ASCII codes for line feed),
      and 13 (the ASCII code for carriage
      return to PC.

      SEROUT2 5, 84, ["POT = ", DEC x, 10, 13]

      Format: SEROUT2 Pin, Mode, [Item1]
      Pin = 5 (RB5 the transmit pin)
      Mode = 84 (9600 baud rate)
      [Item1] = ["POT = ", DEC x, 10, 13]
      Transmits POT = , the 10-bit
      value of x, 10 (the ASCII codes for
      line feed), and 13 (the ASCII code
      for carriage return) to the PC.

- **SERIN2:**
  - Format:

    SERIN2 DataPin{\FlowPin},Mode,{ParityLabel,}
    {Timeout,Label,}[Item...]

    Explanation:

    At this point, we will shorten the format to:
    SEROUT2 DataPin, Mode, {Timeout,Label,} [Item1]
    Receive one or more Items on DataPin in standard asynchronous format. Mode is used to specify the baud rate and operating parameters of the serial transfer. Timeout,Label may be included to allow the program to continue if a character is not received within a certain amount of time. Timeout is specified in 1 millisecond units. If the serial input pin stays in the idle state during the Timeout time, the program will exit the SERIN2 command and jump to Label.
    SERIN2 supports many different data modifiers which may be mixed and matched freely within a single SERIN2 statement to provide various input formatting. We will use the WAIT modifier.
    The list of data items to be received may be preceded by a qualifier between parentheses after WAIT. SERIN2 must
    receive this byte before receiving the data items. If any byte received does not match the byte in the qualifier, the qualification process starts over.

    For a more extensive explanation of look for SERIN2 command at about page 134 in the PBP manual.

    Examples:

    ```
    SERIN2 pinin, MODE, 1000, nodata, [DEC InputData]
                    SERIN2 uses a timeout and a label.
                    If we get no input data within 1000 ms,
                    we jump to nodata, and the program continues.
                    Format:SERIN2 Pin,Mode,TimeOut,Label,[Item1]
                    Pin = pinin,(RB2), Declared in variables
                    Mode = 84 (9600 baud rate)
                    TimeOut = 1000 ms
                    Label = nodata
                    [Item1] = [DEC InputData]

    SERIN2 pinin, MODE,[WAIT("a"),DEC inputData]
                    ' Format: SERIN2 Pin,Mode,[Item1]
                    ' Pin = pinin,(RB2), Declared in variables
                    ' Mode = 84 (9600 baud rate)
                    ' [Item1] = [Modifier WAIT("a"), which
                    ' makes the program wait
                    ' for the letter a to be typed in, then
                    ' DEC InputData, the input data.
    ```

- **SELECT CASE:**
  - Format:

```
SELECT CASE Var
        CASE Expr1 {, Expr...}
                Statement...
        CASE Expr2 {, Expr...}
                Statement...
        {CASE ELSE
                Statement...}
END SELECT
```

CASE statements are sometimes easier to use than multiple IF..THENs. These statements are used to compare a variable with different values or ranges of values, and take action based on the value.

The Variable to be used in all of the comparisons is specified in the SELECT CASE statement. Each CASE is followed by the Statements to be executed if the CASE is true. IS may be used to specify a comparison other than equal to. If none of the CASEs are true, the Statements under the optional CASE ELSE statement are executed. An END SELECT closes the SELECT CASE.

Examples:

```
        SELECT CASE  x

                CASE 1                ‘ If x = 1,  then y = 10
                        y = 10
                CASE 2, 3             ‘ If x = 2 or 3, y = 20
                        y = 20
                CASE IS > 5           ‘ (Comparison other than equal to)
                        y = 100        If x > 5, y = 100
                CASE ELSE             ‘ If none of the CASEs are true,
                        y = 0           then y = 0
        END SELECT          ‘Closes the SELECT CASE


SELECT CASE  Command

   CASE "a"             ' If "a" is entered, Command = a
      HIGH LED1         ' HIGH 0 turns on LED to RB0
      SEROUT2 pinout, Mode, ["LED1 ON",10,13]
                        ' Sends back "LED1 ON" to computer
   CASE "b"             ' If "b" is entered, Command = b
      LOW LED1          ' LOW 0 turns off LED to RB0
      SEROUT2 pinout, Mode, ["LED1 OFF",10,13]
                        ' Sends back "LED1 OFF" to computer
END SELECT
```

- Labs:
    - Perform RS-232 Serial Communication Tutorial – Software Lab 1 – Hello
    - Perform RS-232 Serial Communication Tutorial – Software Lab 2 – Potentiometer 8bit and 10-bit
    - Perform RS-232 Serial Communication Tutorial – Software Lab 3 – LED On/Off
    - Perform RS-232 Serial Communication Tutorial – Software Lab 4 – PWM Motor Control

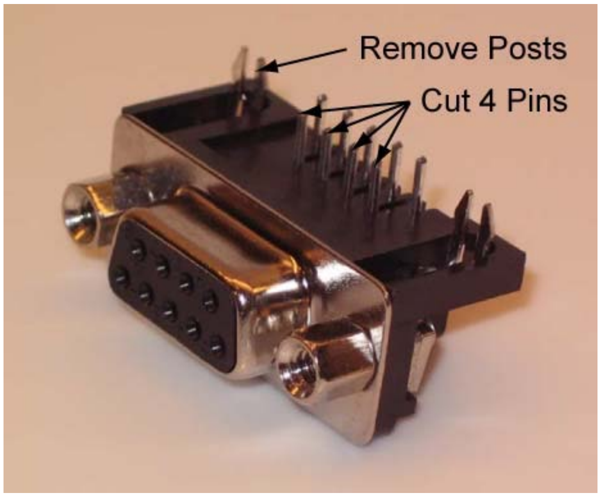
Cornerstone Electronics Technology and Robotics II
RS-232 Serial Communication Tutorial – Software Lab 1 – Hello

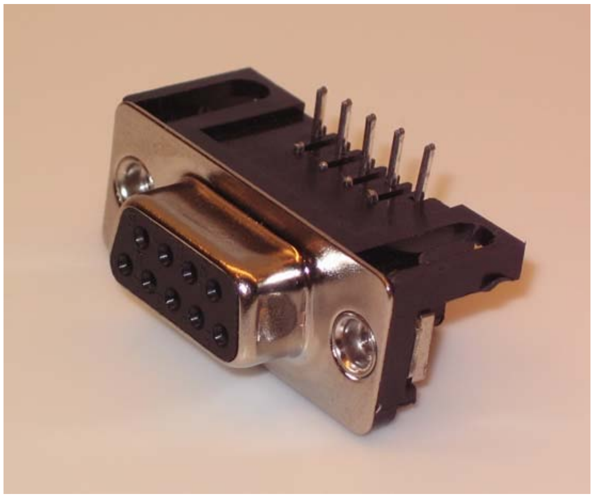
- **Purpose:** The purpose of this lab is to establish basic serial communications between the PIC microcontroller and a PC terminal program.

- **Apparatus and Materials:**

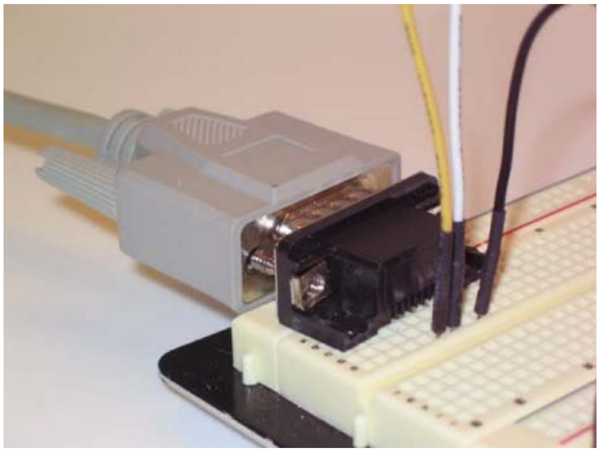
  - 1 – Breadboard with +5V Power Supply
  - 1 – MAX232 RS-232 Driver/Receiver
  - 1 – PIC16F88
  - 5 – 1 uF Capacitors
  - 1 – 4.7K Resistor
  - 1 – Modified DB9 Connector - Connector,D-SUB,.318"RT,9P-F, Short Metal Housing, Jameco Part # 104952

- **Procedure:**
  - Wire the serout2_hello circuit below. Read the schematic note.
  - To make connection between the DB9 connector and the breadboard, remove the mounting posts and 4 front pins on the connector. See the photos below. A header may be soldered onto the remaining 5 pins.

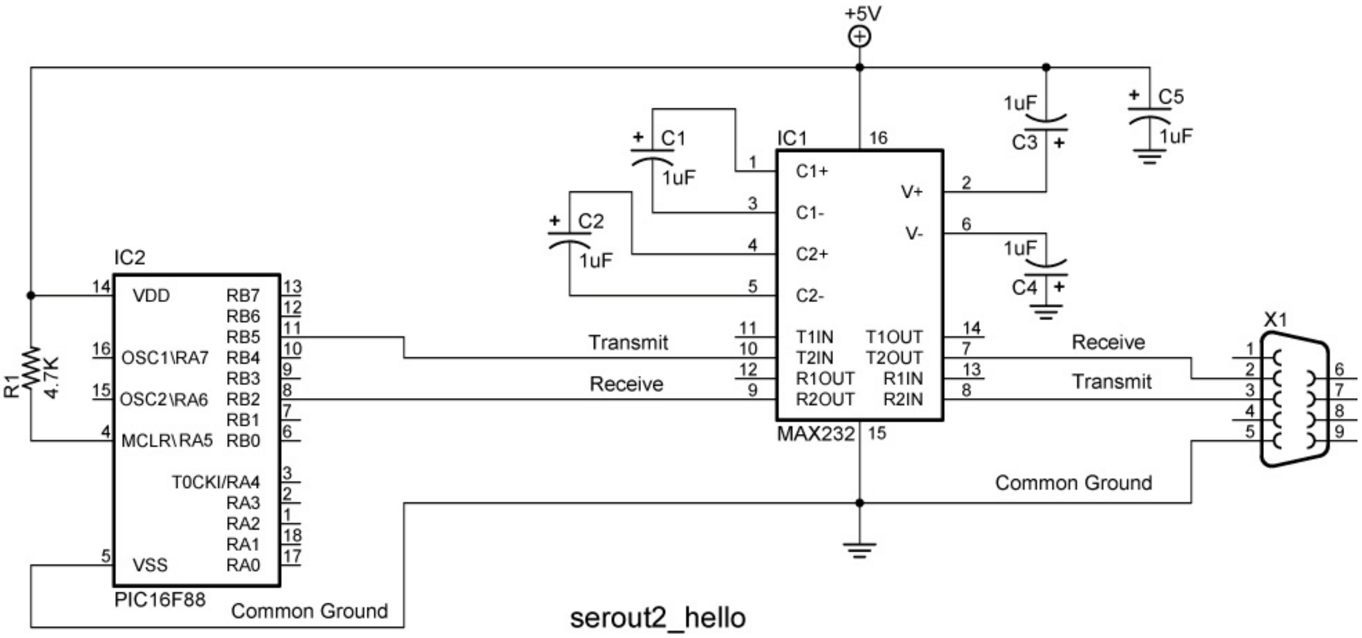




New DB9 Connector **Modified DB9 Connector**



Modified DB9 Connector Mounted on a Breadboard



NOTE: Make certain that the polarity of each capacitor is wired correctly.

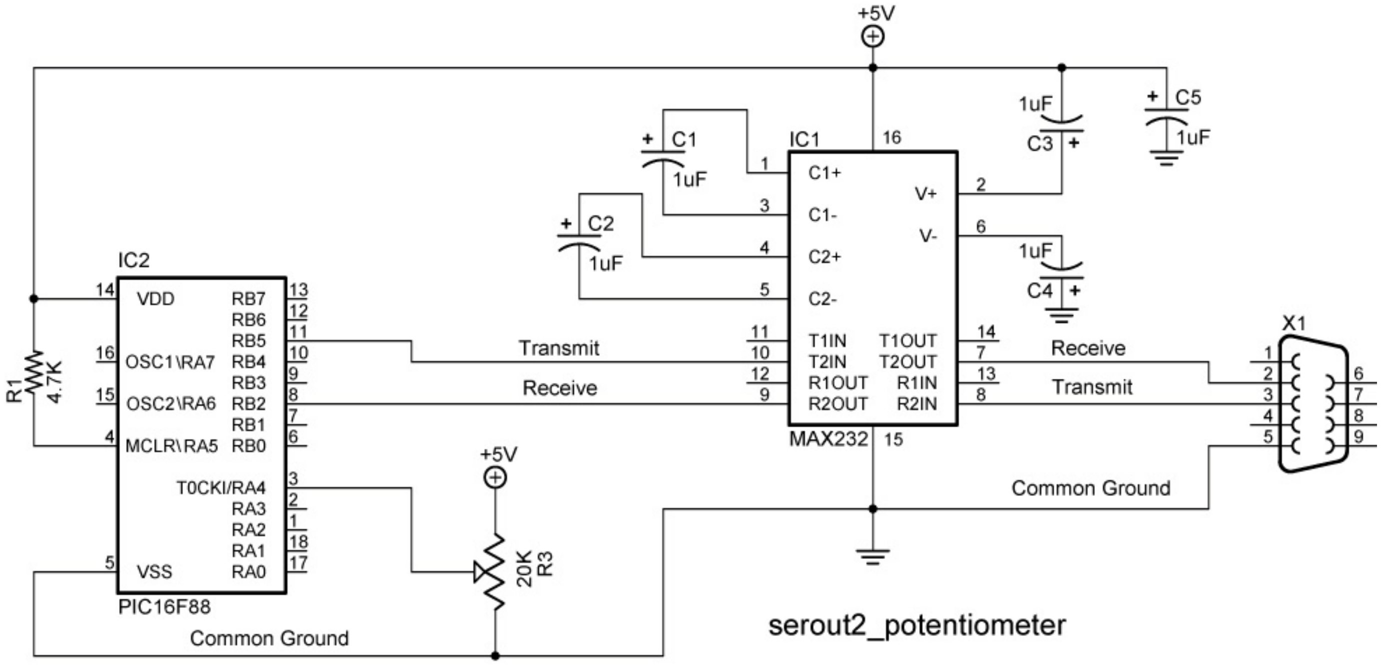
- o Program the PIC16F88 with **serout2 _hello**. Copy file from: http://www.cornerstonerobotics.org/code/serout2_hello.pbp
- o Use either TeraTerm Pro 3.1.3 or HyperTerminal as the terminal program.
- o For the PicBasic Pro program to work, the terminal program window must be the active window.
- o Reprogram the PIC16F88 to display additional hellos.
- o Save **serout2_hello** as **serout2_hello_modified.**
    - ▪ Remove “10” and the “13” from the SEROUT2 command and note the changes in the display. Load the program several times to demonstrate the changes.
    - ▪ Reinsert “10” back into the SEROUT2 command. Load the program several times to demonstrate the changes. Note the response.
    - ▪ Remove “10” and reinsert “13” back into the SEROUT2 command. . Load the program several times to demonstrate the changes. Note the response.

Cornerstone Electronics Technology and Robotics II
RS-232 Serial Communication Tutorial – Software Lab 2 – Potentiometer 8-bit and 10-bit

- **Purpose:** The purpose of this lab is to display an 8-bit potentiometer readings (0 – 255) and a 10-bit potentiometer readings (0 – 1023) on a PC terminal program.

- **Apparatus and Materials:**
  - 1 – Breadboard with +5V Power Supply
  - 1 – MAX232 RS-232 Driver/Receiver
  - 1 – PIC16F88
  - 5 – 1 uF Capacitors
  - 1 – 4.7K Resistor
  - 1 – 20K Potentiometer
  - 1 – Modified DB9 Connector - Connector,D-SUB,.318"RT,9P-F, Short Metal Housing, Jameco Part # 104952

- **Procedure:**
  - Wire the serout2_potentiometer circuit below.
  - Program the PIC16F88 with **serout2 _potentiometer_8bit.** Copy file from: http://www.cornerstonerobotics.org/code/serout2_potentiometer_8bit.pbp
  - Adjust the potentiometer and note the range of readings on the PC terminal program.
  - Program the PIC16F88 with **serout2 _potentiometer_10bit.** Copy file from: http://www.cornerstonerobotics.org/code/serout2_potentiometer_10bit.pbp
  - Again, adjust the potentiometer and note the range of readings on the PC terminal program.



NOTE: Make certain that the polarity of each capacitor is wired correctly.

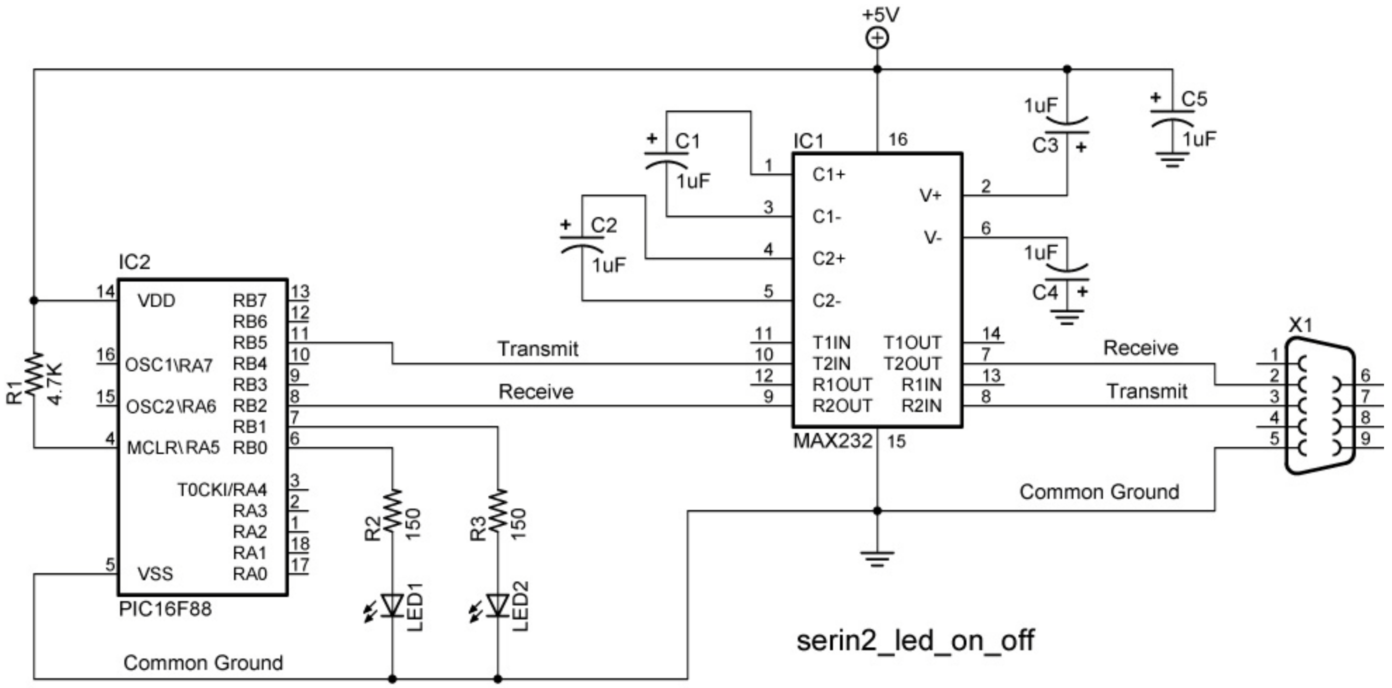
RS-232 Serial Communication Tutorial – Software Lab 3 – LED On/Off

- **Purpose:** The purpose of this lab is to control the output of a PIC from a PC terminal program.

- **Apparatus and Materials:**

  - 1 – Breadboard with +5V Power Supply
  - 1 – MAX232 RS232 Driver/Receiver
  - 1 – PIC16F88
  - 5 – 1 uF Capacitors
  - 1 – 4.7K Resistor
  - 3 – 150 Ohm Resistors
  - 3 – LEDs
  - 1 – Modified DB9 Connector - Connector,D-SUB,.318"RT,9P-F, Short Metal Housing, Jameco Part # 104952

- **Procedure:**
  - Wire the serin2_led_on_off circuit below.
  - Program the PIC16F88 with **serin2_led_on_off.** Copy file from: http://www.cornerstonerobotics.org/code/serin2_led_on_off.pbp
  - Type in one of the command options, a, b, c, or d and observe the LED response and the output to the PC terminal program.
- **Challenge:**
  - Save **serin2_led_on_off** as **serin2_led_on_off_modified.**
  - Add another LED to PORTB.4 and add two more options ("e" and "f") to the command list to turn the new LED on and off from the PC.



serin2_led_on_off

NOTE: Make certain that the polarity of each capacitor is wired correctly.

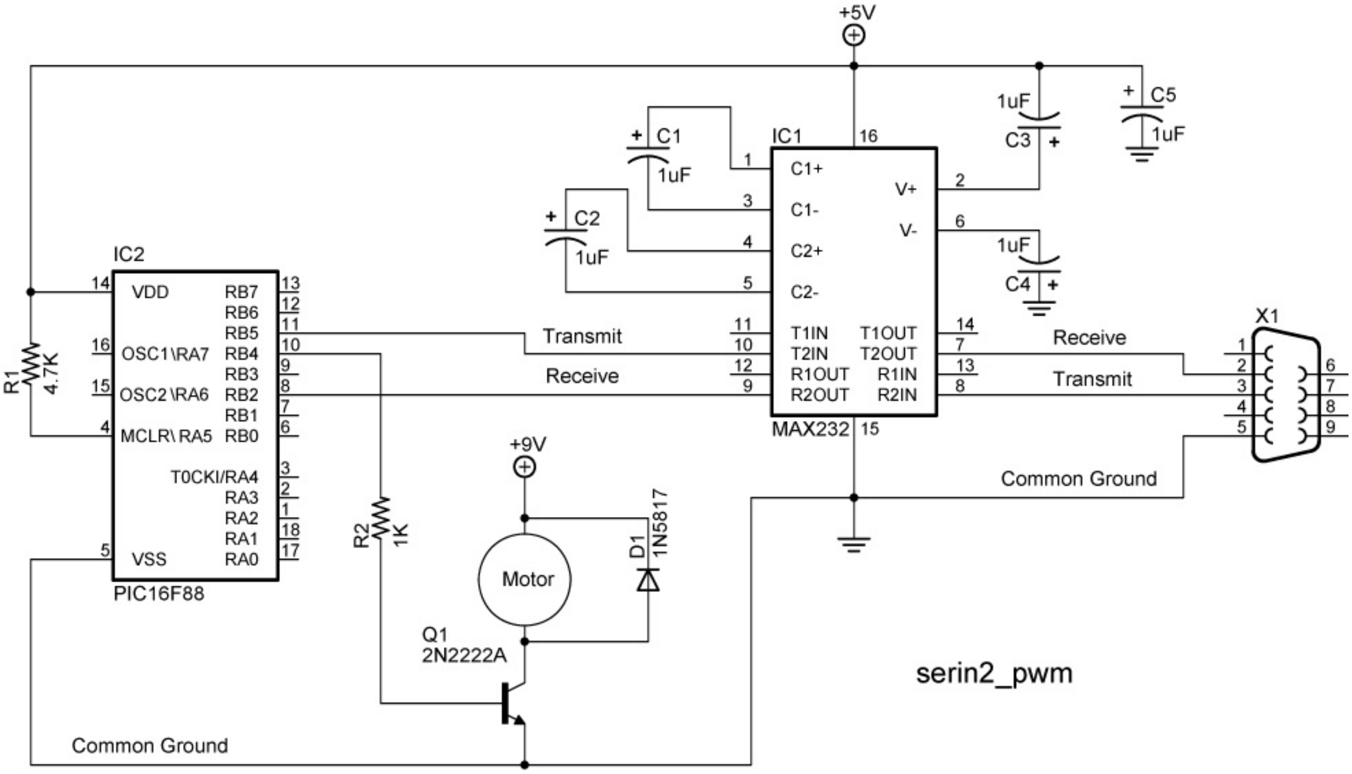
Cornerstone Electronics Technology and Robotics II
RS-232 Serial Communication Tutorial – Software Lab 4 – PWM Motor Control

- **Purpose:** The purpose of this lab is to change the Pulse Width Modulation (PWM) to control a motor from a PC terminal program.

- **Apparatus and Materials:**

    - 1 – Breadboard with +5V and +9V Power Supplies
    - 1 – MAX232 RS232 Driver/Receiver
    - 1 – PIC16F88
    - 5 – 1 uF Capacitors
    - 1 – 4.7K Resistor
    - 1 – 1K Resistor
    - 1 – Modified DB9 Connector - Connector,D-SUB,.318"RT,9P-F, Short Metal Housing, Jameco Part # 104952
    - 1 – 1N5817 Diode
    - 1 – 2N2222A NPN Transistor

- **Procedure:**
    - Wire the serin2_pwm circuit below.
    - Program the PIC16F88 with **serin2_pwm.** Copy file from: http://www.cornerstonerobotics.org/code/serin2_pwm.pbp
    - Type in one of the command options, a, b, c, or d and observe the LED response and the output to the PC terminal program.
    - Program the PIC16F88 with **serin2_pwm_wait.** Copy file from: http://www.cornerstonerobotics.org/code/serin2_pwm_wait.pbp
    - Follow the program instructions and change the motor speed from the PC.
- **Challenge:**
    - Save **serin2_pwm_wait** as **serin2_servo1.** Copy **servo1.pbp** from http://www.cornerstonerobotics.org/code/servo1.pbp. Use **serin2_servo1** and **servo1.pbp** as a references to control a servo from a PC.



serin2_pwm

NOTE: Make certain that polarity of each capacitor is wired correctly.