1610 CLC assistant manual

Materials List:

Software:

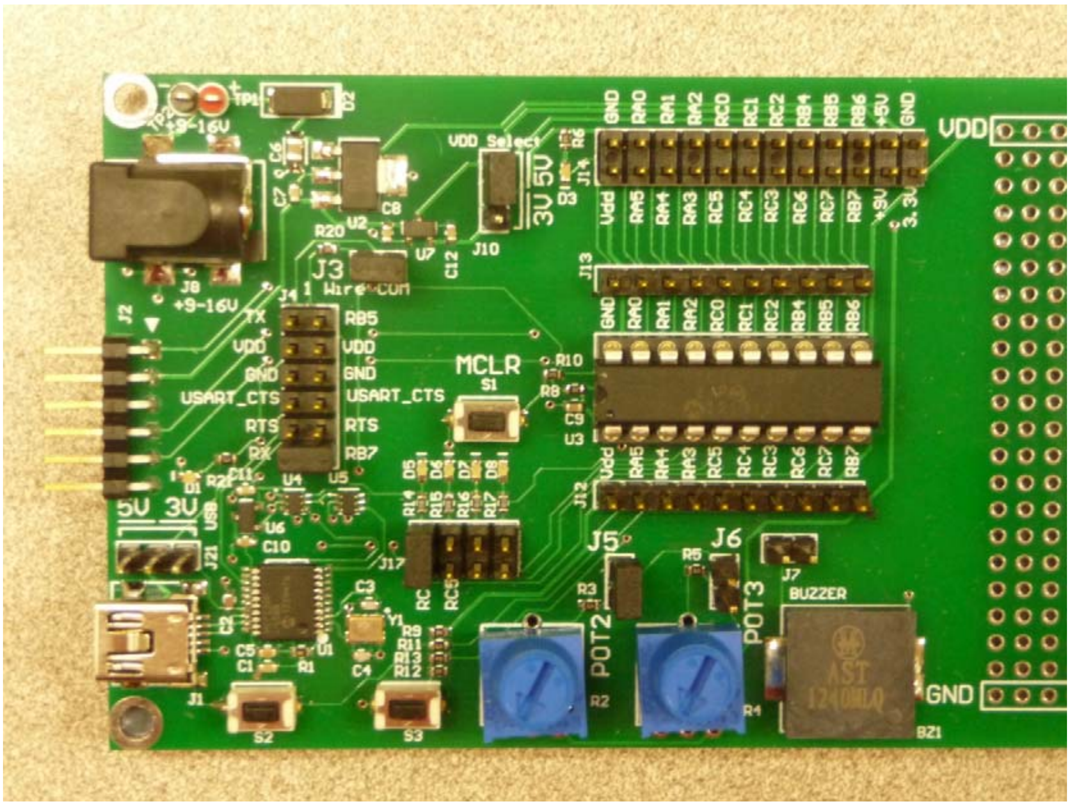
1. MPLAB X **v1.41** located at: http://www.microchip.com/pagehandler/en-us/family/mplabx/
2. C compiler XC8 **v1.00** located at: http://www.microchip.com/pagehandler/en_us/promo/mplabxc/
3. CLC Designer Tool v1.0.0.4 located at: http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en553474
4. PIC MCU Communicator IV v4.3.1.9

Programmer:

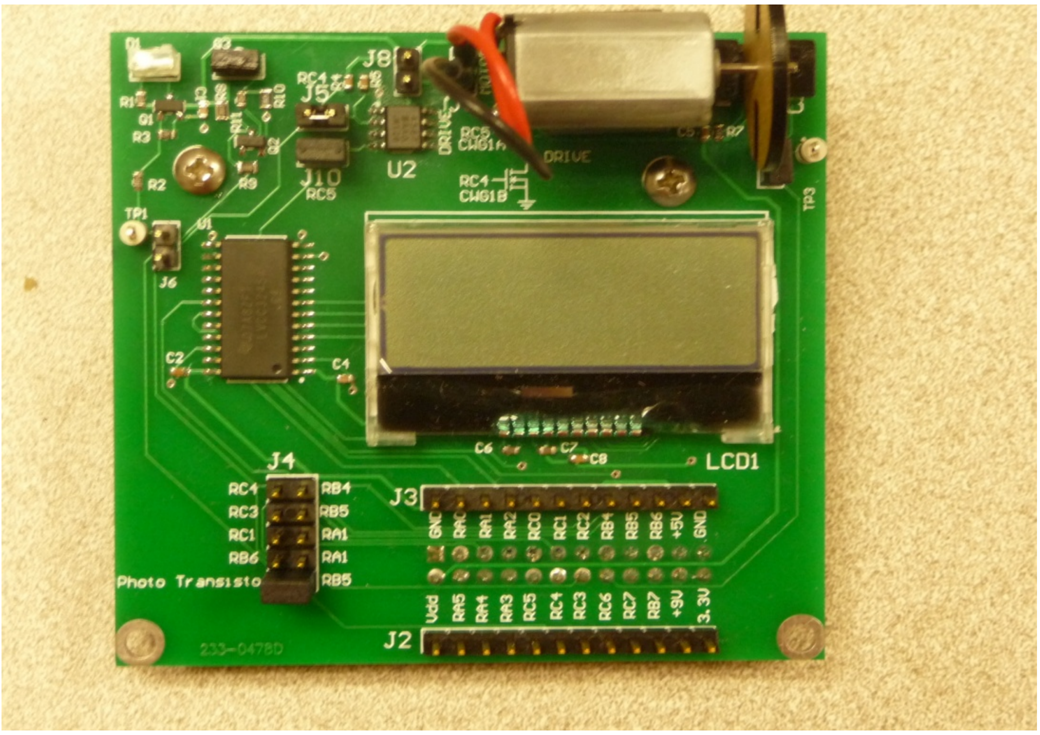
1. PicKit 3 with USB cable (mini USB connector).

Hardware:

1. 1509 Enhanced Midrange Development Board



2. New Peripheral CWG Daughter Board



<u>Test board:</u>

	Assist	Test board
LAB1 Manchester encoder by CLC	1. Jumpers 2. CLC Design 3. Manchester.c (*.inc)	LAB2 board
LAB2 Manchester decoder (CLC,NCO)	1. Jumpers 2. CLC Design 3. Manchester.c (CLC,NCO)	LAB1 board
LAB3 Motor control by PC GUI (PWM & CWG)	1. Jumpers 2. Hex file program 3. MCU Communicator (PWM control setting)	NA
LAB4 Motor control & speed measurement & Manchester communication	1. Jumpers 2. CLC Design 3. Main.c (CWG,PWM) 4. Manchester.c (CLC,NCO)	LAB2 board

<u>PS. Failed to get Device ID</u>

Take off J21jumper and the daughter board could solve the problem.

<u>PS. install MCP2200 deriver in LAB3</u>

<u>LAB 1 – Manchester Encoder</u>

<u>Jumper Settings:</u>

<u>1509 Enhanced Midrange Development Board</u>

J17: Place a shunt connector vertically on the far left of J17. This connects the indication LED, D5.

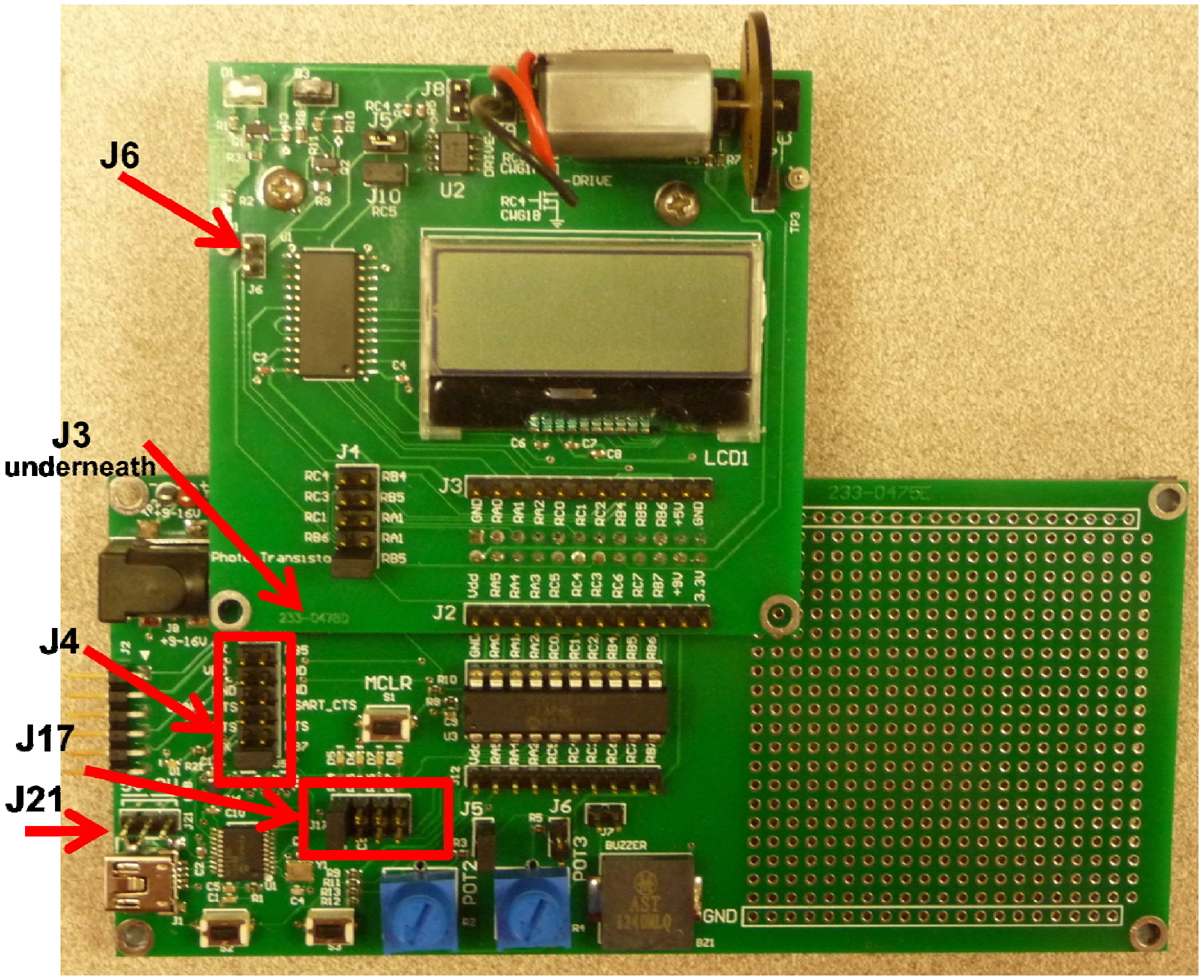
J4: Place a shunt connector horizontally on the bottom of J4.

J3: Place a shunt connector on J3 (located underneath the CWG daughter board).

J21: Place a shunt connector horizontally on the far left of J21 (5volts).

<u>New Peripheral CWG Daughter Board</u>

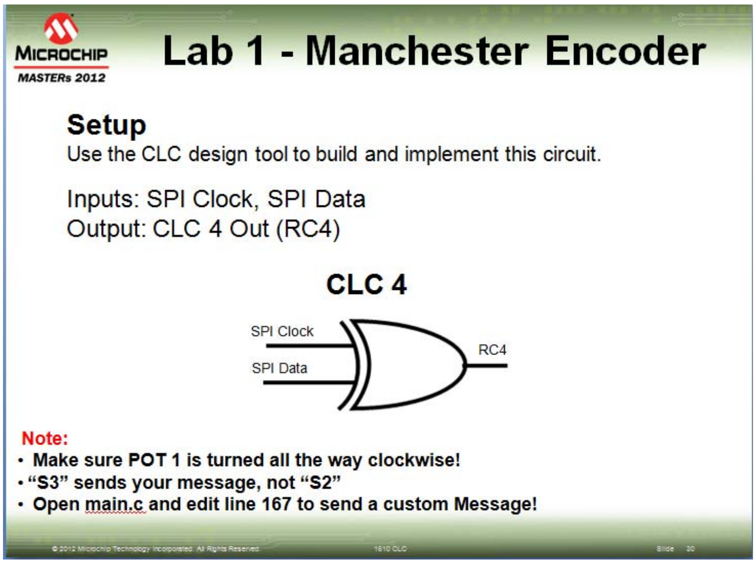
J6: Place a shunt connector vertically on J6. This connects the IR transmit LED.

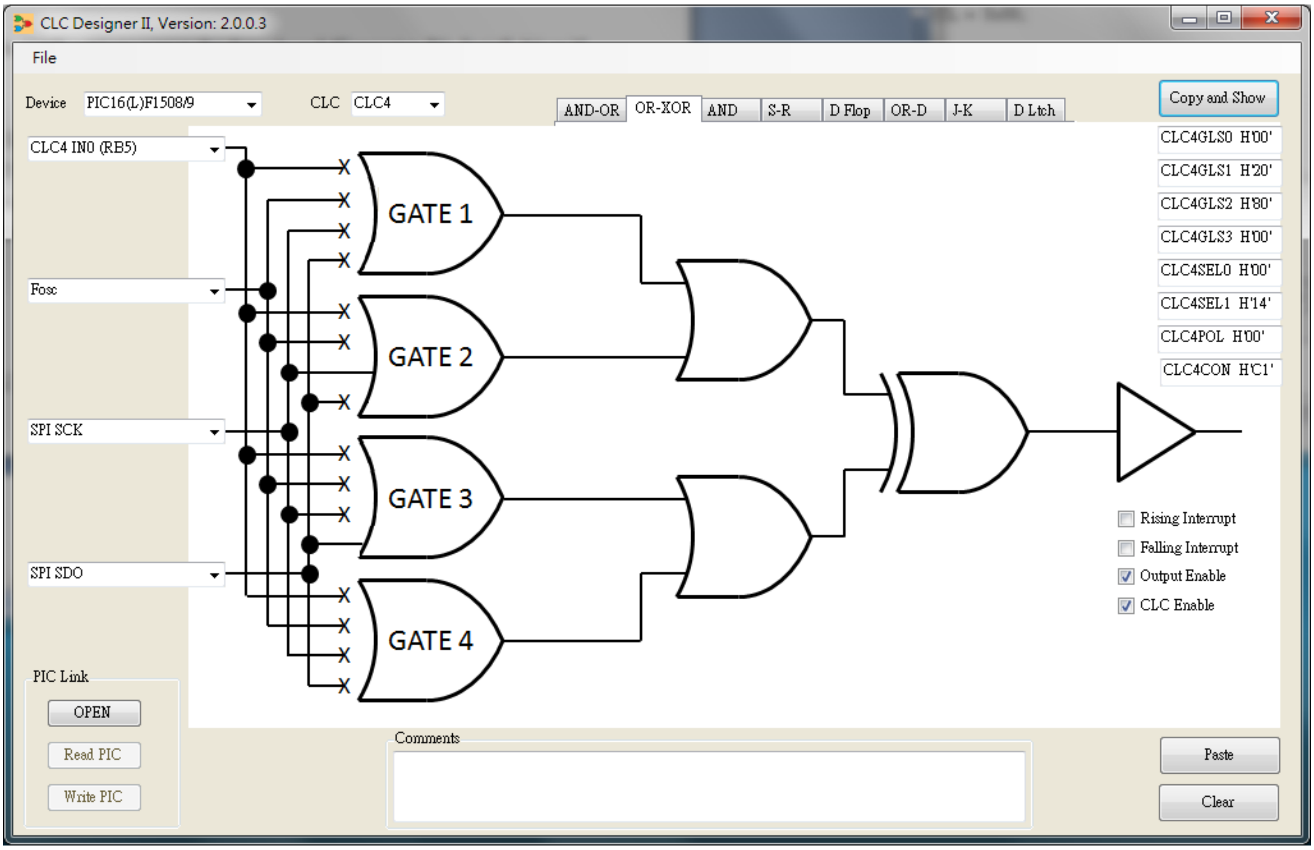


Procedure :

1. Open MPLAB X v1.41
2. In the tool bar at the top of the window, under the File drop-down menu, click on "Open project" and select the Manchester Encoder project at: "**C:\1610\Manchester Encoder.X**".
3. After expanding the project, expand the Source files directory. Now open manchester.c by double clicking on the file in the project window. Then in the code window, scroll down to function " encoder_int ". This will be where the CLC will be configured.
4. Next open the CLC Designer Tool GUI (Icon on Desktop)
5. Use the Pic16(L)F1509 datasheet, located in the 1610 director of this project or at Microchip.com, in conjunction with the CLC Designer tool to setup and implement the Manchester encoder.

   a. CLC4 Inputs: SPI Clock, SPI Data
   b. Use the OR-XOR logic function
   c. Outputs: CLC4 out (RC4)
   d. After making the CLC settings, click "Copy and Show", and under "File" save as C code with the file name: " lab1.inc" at location: " C:\1610\Manchester Encoder\src".
   e. Next go to the Manchester.c code window and under the function encoder_init, add #include "lab1.inc".

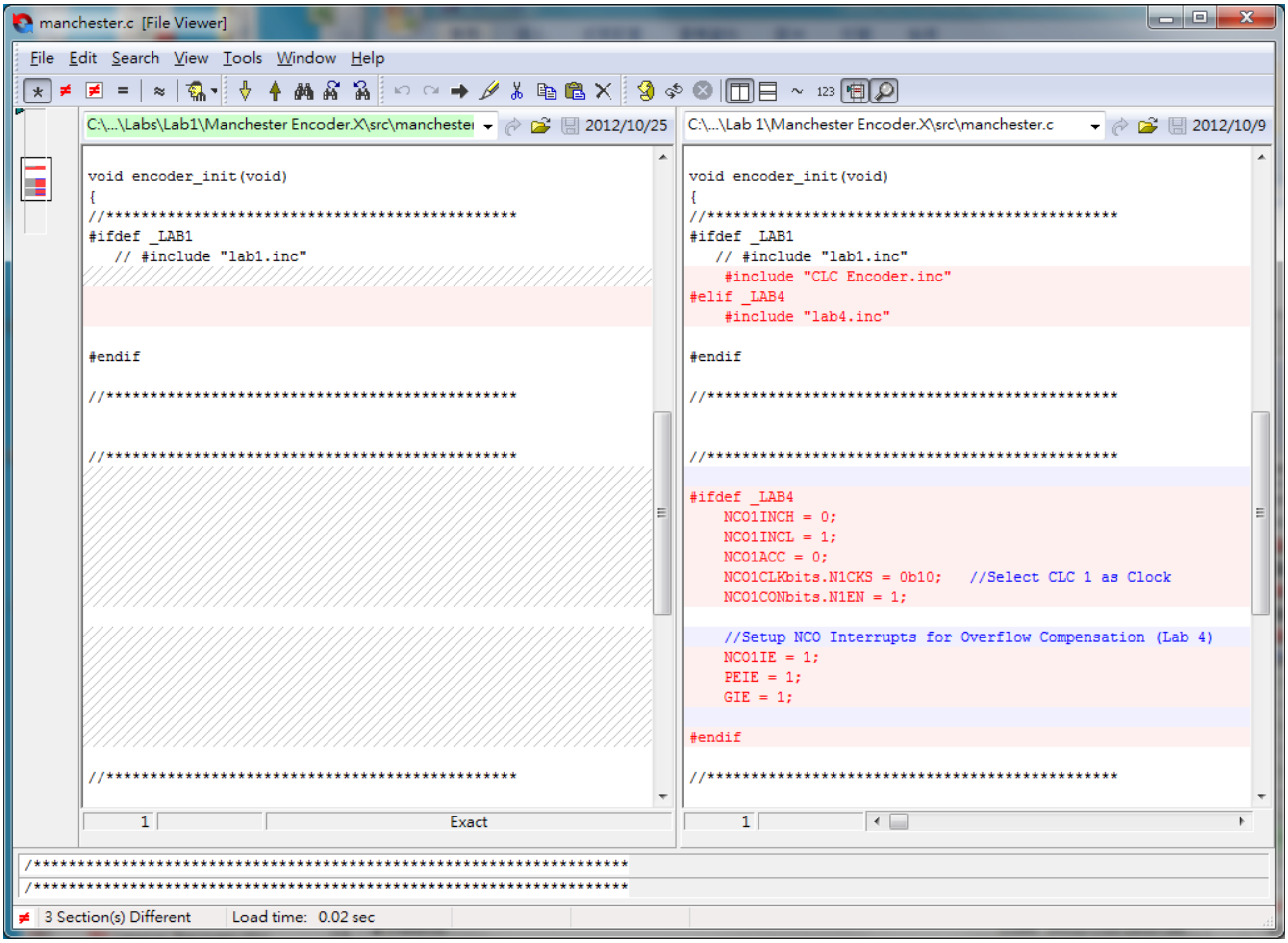




6. Plug in PICkit 3 to the computer and the demo board (J2), and program the part by clicking the "Make and Program Device" button.

7. Unplug the PICkit 3 and use the USB cable to power the demo board (Connector J1). Install driver if prompted.

8. The “S3” button on the 1509 Enhanced Midrange Development Board, sends your message. At the same time the indication LED (D5) should flash when S3 is pressed to show that data is being transmitted. Also, the words “Encoder: Testing # ” should appear on the LCD. The incrementing number is a looping counter to show a value that will be used in Lab 2.



LAB2 – Manchester Decoder

Jumper Settings:

1509 Enhanced Midrange Development Board

J4: Place a shunt connector horizontally on the bottom of J4.

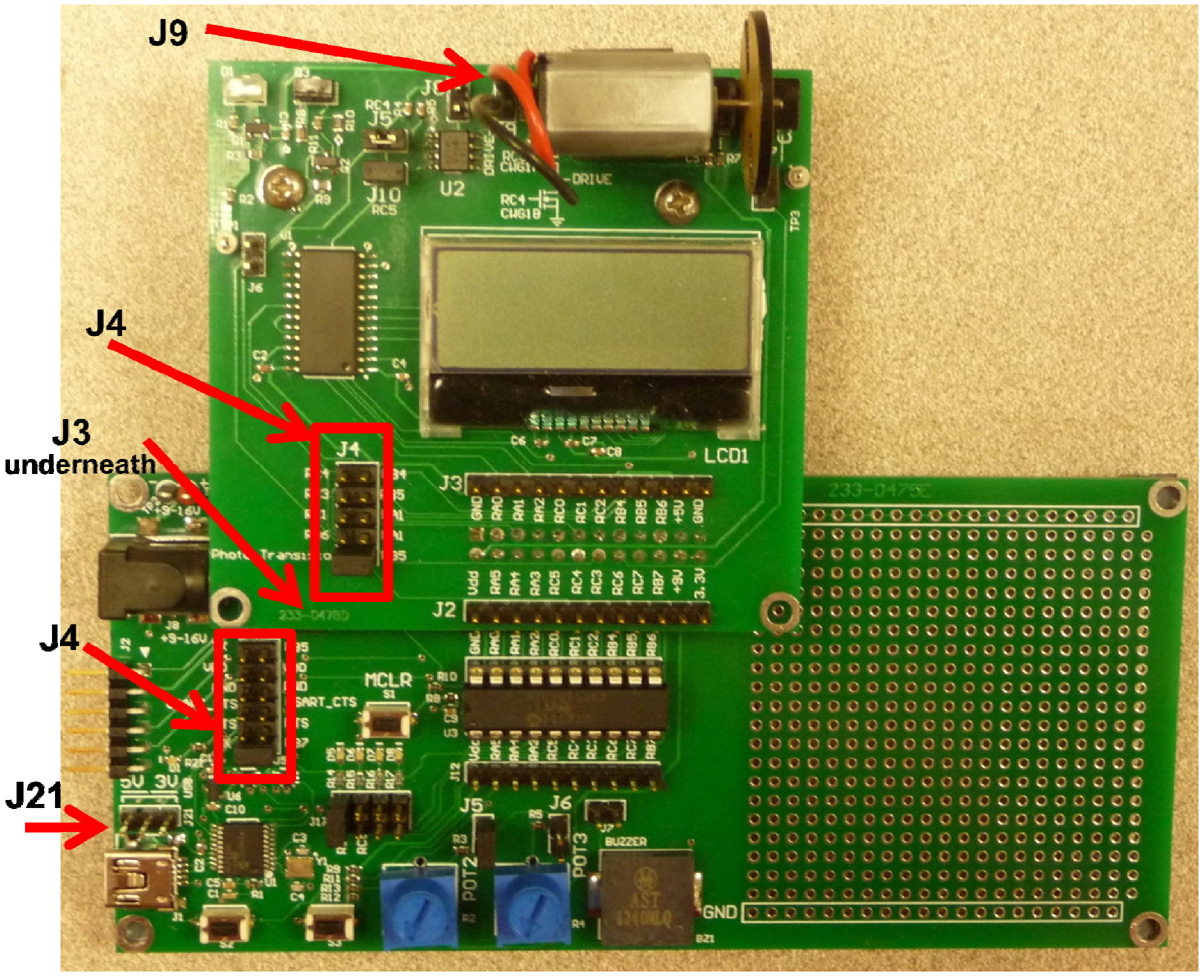
J3: Place a shunt connector on J3 (located underneath the CWG daughter board).

J21: Place a shunt connector horizontally on the far left of J21 (5volts).

New Peripheral CWG Daughter Board

J4: Place shunt connectors vertically on all locations (total of 5). This jumper connects all the components need to the PicMCU.

J9: Unplug the motor.



Procedure:

1. Open MPLAB X v1.41

2. In the tool bar at the top of the window, under the File drop-down menu, click on "Open project" and select the Manchester Decoder project at: "**C:\1610\Manchester Decoder.X**"

3. After expanding the project, expand the Source files directory. Now open manchester.c by double clicking on the file in the project window. Then in the code window, scroll down to function " decoder_int ". This will be where the CLC and NCO peripherals will be configured.

4. Next open the CLC Designer Tool GUI (Icon on Desktop)

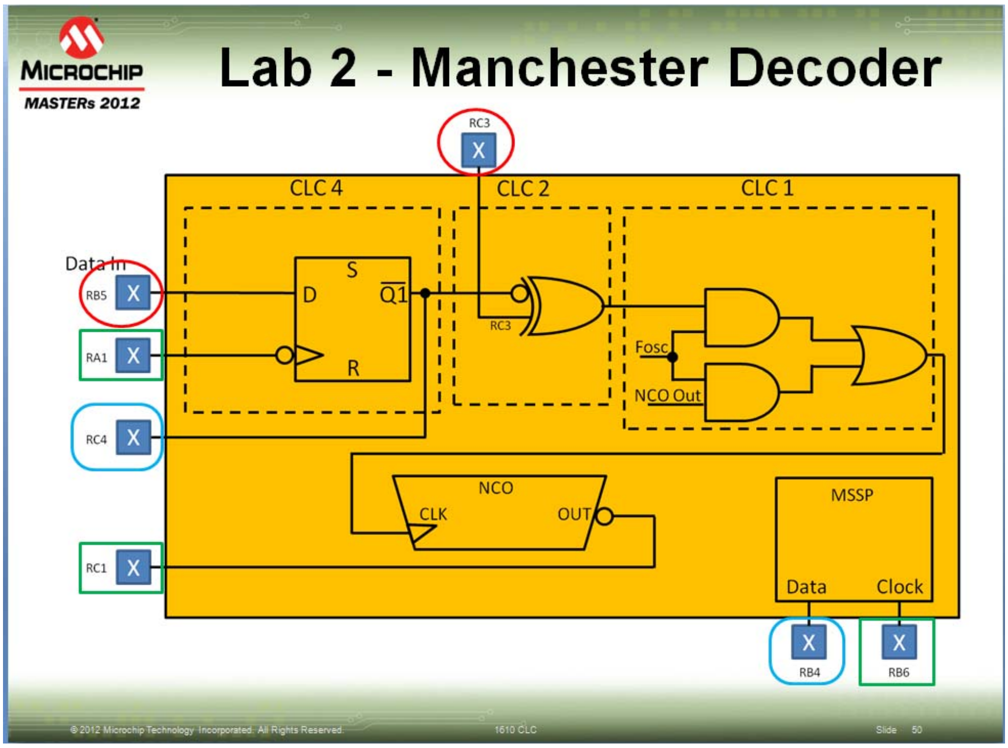
5. Use the Pic16(L)F1509 datasheet, located in the 1610 director of this project or at Microchip.com, in conjunction with the CLC Designer tool to setup and implement the Manchester decoder.

a. CLC4: Inputs: RB5 and RA1
Use the D-Flip Flop logic function
Outputs: CLC4 out (RC4)

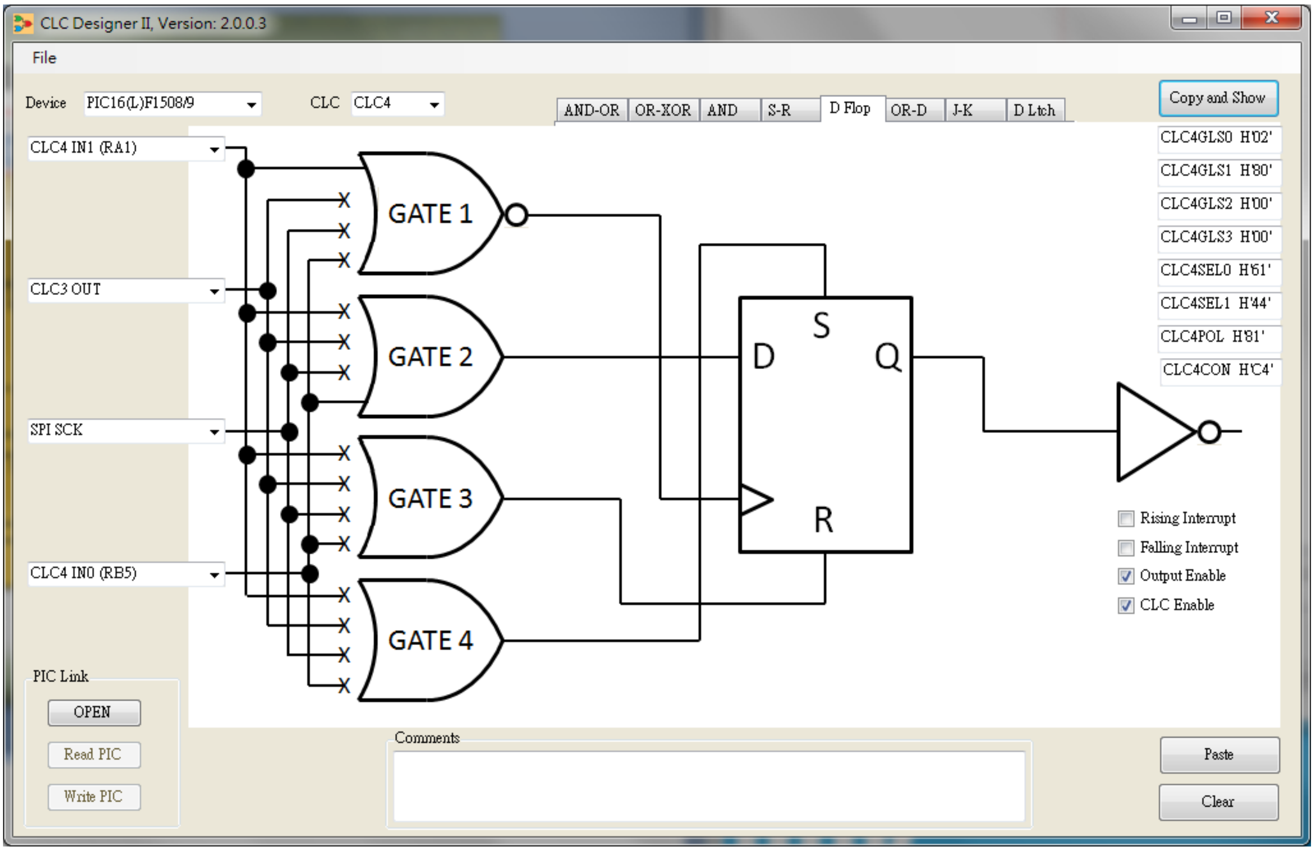
CLC2: Inputs: CLC4 and RC3
Use the OR-XOR logic function
Outputs: CLC2 out (RC0)

CLC1: Inputs: CLC2, Fosc and NCO out
Use the AND-OR logic function
Outputs: CLC1 out (RA2)

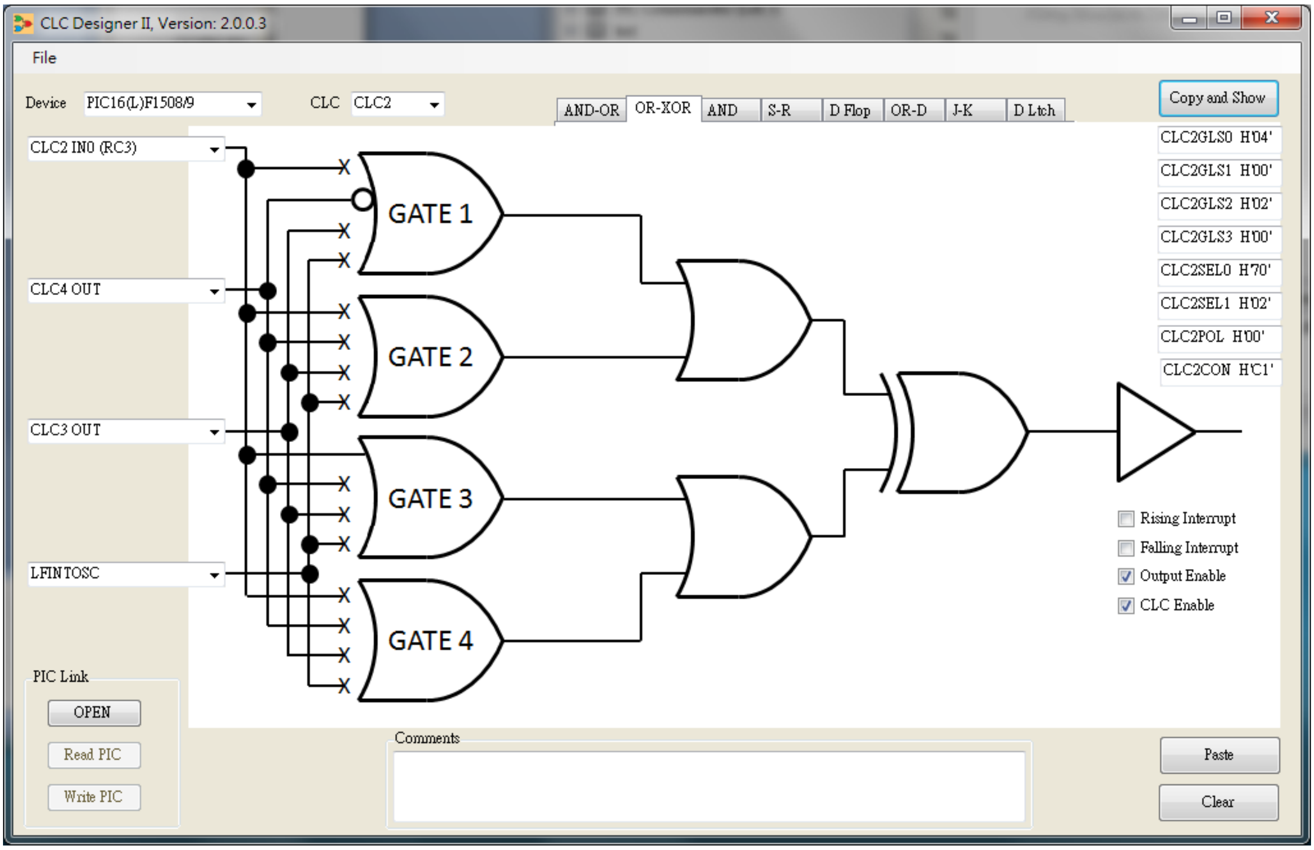
d. After making the CLC settings, click "Copy and Show", and under "File" save as C code with the file name: " lab2.inc" at location: " C:\1610\Manchester Encoder\src".
e. Next, go to the Manchester.c code window and under the function decoder_init, add #include "lab2.inc".

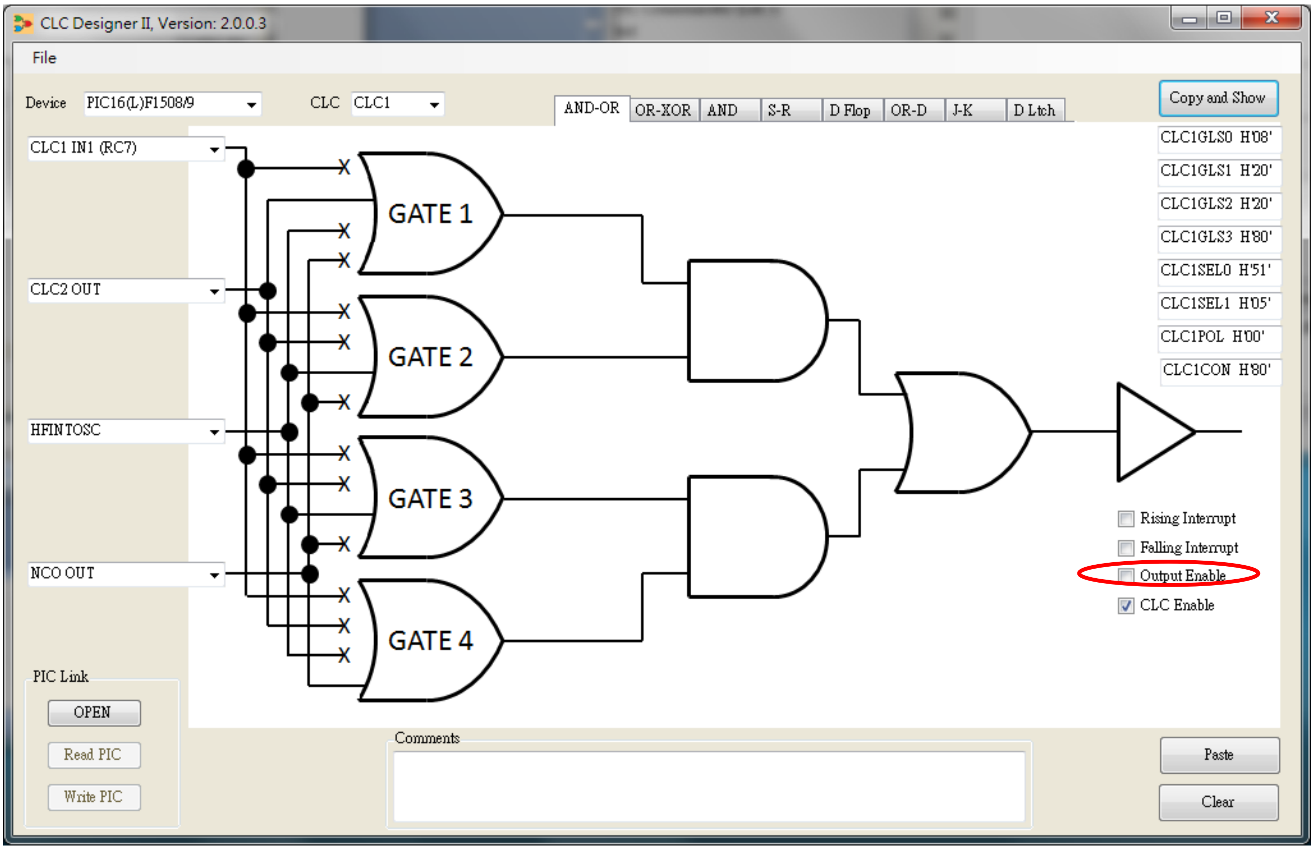


MICROCHIP
MASTERs 2012
Lab 2 - Manchester Decoder
RC3
CLC 4
CLC 2
CLC 1
Data In
RB5
D
S
Q1
R
RC3
Fosc
NCO Out
RA1
RC4
NCO
CLK
OUT
MSSP
Data
Clock
RC1
RB4
RB6
© 2012 Microchip Technology Incorporated. All Rights Reserved.
1610 CLC
Slide 50



CLC Designer II, Version: 2.0.0.3
File
Device PIC16(L)F1508/9
CLC CLC4
AND-OR
OR-XOR
AND
S-R
D Flop
OR-D
J-K
D Ltch
Copy and Show
CLC4GLS0 H'02'
CLC4GLS1 H'80'
CLC4GLS2 H'00'
CLC4GLS3 H'00'
CLC4SEL0 H'61'
CLC4SEL1 H'44'
CLC4POL H'81'
CLC4CON H'C4'
CLC4 IN1 (RA1)
CLC3 OUT
SPI SCK
CLC4 IN0 (RB5)
GATE 1
GATE 2
GATE 3
GATE 4
S
D
Q
R
Rising Interrupt
Falling Interrupt
Output Enable
CLC Enable
PIC Link
OPEN
Read PIC
Write PIC
Comments
Paste
Clear


CLC Designer II, Version: 2.0.0.3
File
Device PIC16(L)F1508/9
CLC CLC2
AND-OR
OR-XOR
AND
S-R
D Flop
OR-D
J-K
D Ltch
Copy and Show
CLC2 IN0 (RC3)
CLC4 OUT
CLC3 OUT
LFINTOSC
GATE 1
GATE 2
GATE 3
GATE 4
CLC2GLS0 H'04'
CLC2GLS1 H'00'
CLC2GLS2 H'02'
CLC2GLS3 H'00'
CLC2SEL0 H'70'
CLC2SEL1 H'02'
CLC2POL H'00'
CLC2CON H'C1'
Rising Interrupt
Falling Interrupt
Output Enable
CLC Enable
PIC Link
OPEN
Read PIC
Write PIC
Comments
Paste
Clear


CLC Designer II, Version: 2.0.0.3
File
Device PIC16(L)F1508/9
CLC CLC1
AND-OR
OR-XOR
AND
S-R
D Flop
OR-D
J-K
D Ltch
Copy and Show
CLC1 IN1 (RC7)
CLC2 OUT
HFINTOSC
NCO OUT
GATE 1
GATE 2
GATE 3
GATE 4
CLC1GLS0 H'08'
CLC1GLS1 H'20'
CLC1GLS2 H'20'
CLC1GLS3 H'80'
CLC1SEL0 H'51'
CLC1SEL1 H'05'
CLC1POL H'00'
CLC1CON H'80'
Rising Interrupt
Falling Interrupt
Output Enable
CLC Enable
PIC Link
OPEN
Read PIC
Write PIC
Comments
Paste
Clear

6. Use the Pic16(L)F1509 datasheet, located in the 1610 directory of this project or at Microchip.com, to configure the NCO.

   a. Input: CLC1
      Outputs: NCO1 out (RC1)
   b. Fill increment Register, set the input clock to 4 Clock periods and use the Pulse Frequency mode. Remember, write to the HIGH register first, and then the LOW register. Values are latched after write to the LOW increment register.

   c. Setup the NCO to overflow every 3/4 of the input pulse.
      Remember: the Manchester is running @25KHz = 40us
      1 / (3/4 * 40us) = 33.33KHz
      NCOFoverflow = (NCOclk * Increment value) / (2^n) → where 'n' = 20 bits
      Increment value = (33.33KHz * (2^20) / 16MHz) = xxx **(2184=0x888)**

      NCO1INCH = ?? **(0x08)**
      NCO1INCL = ?? **(0x88)**

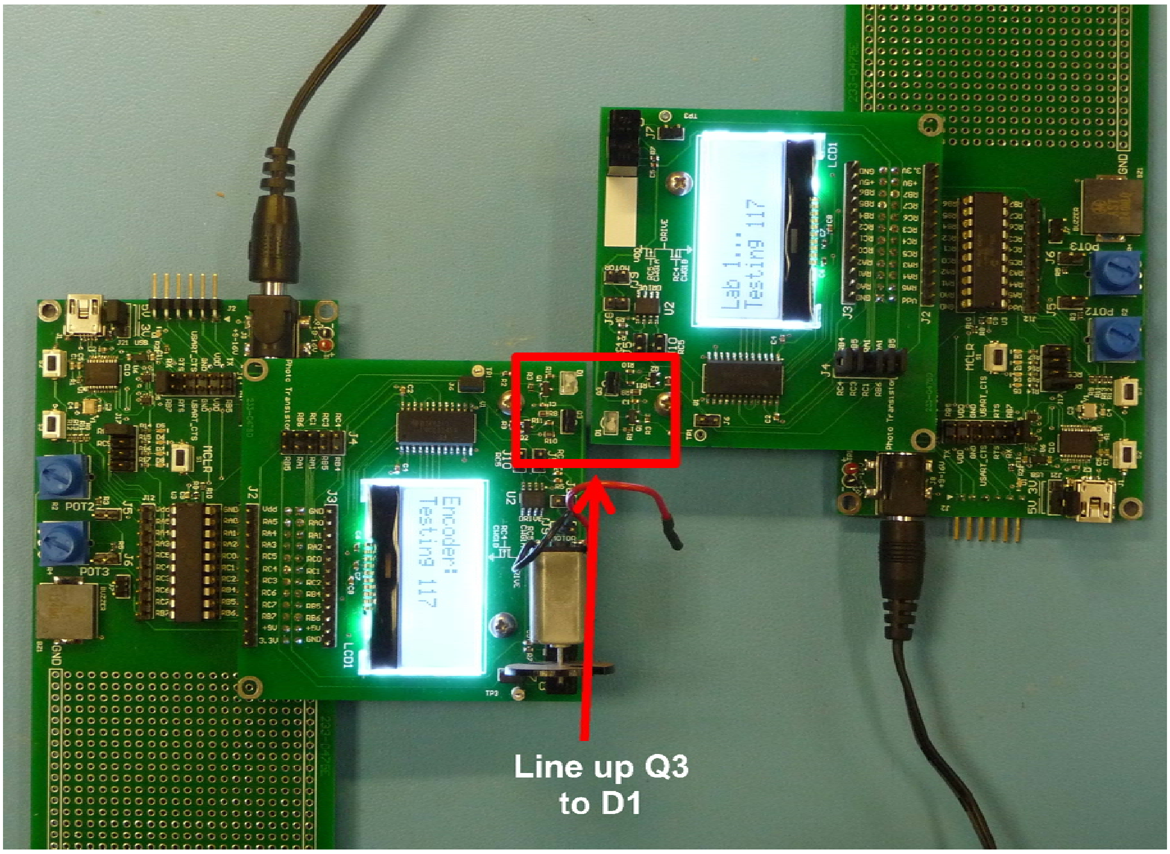
   d. Setup pulse width to be **4** clock:

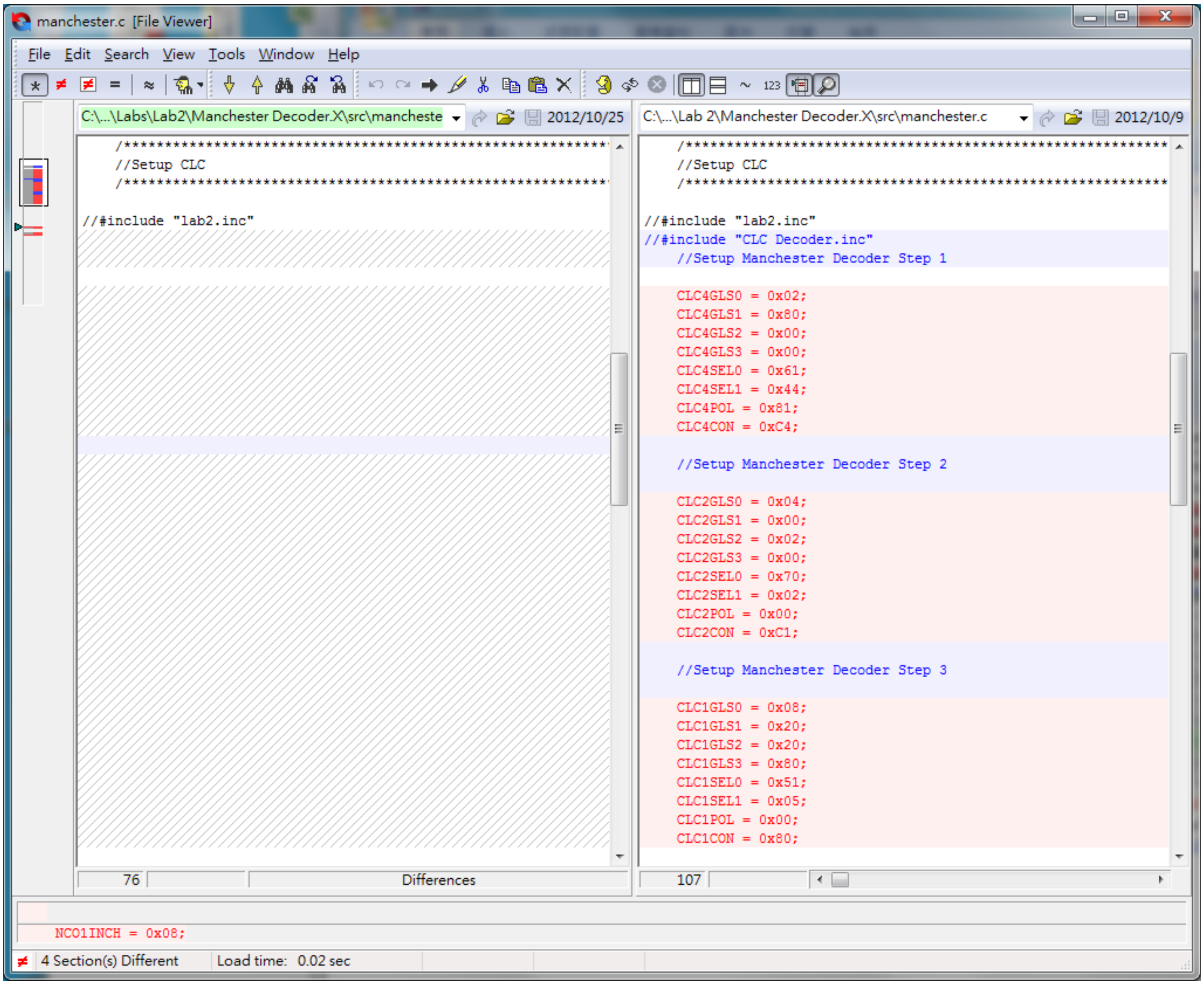
      NCO1CLKbits.N1PWS1 = ?? (Pulse width is 4 clocks) **(010)**
      NCO1CLKbits.N1CKS1 = ?? (Clock source is CLC1 out) **(10)**
      NCO1CON = ?? (NCO Enabled, Output Enabled, Signal is Active Low (datasheet is wrong)). **(0b11010001)**
   e. Next go to the Manchester.c code window and under the function decoder_init, and add the NCO register values.

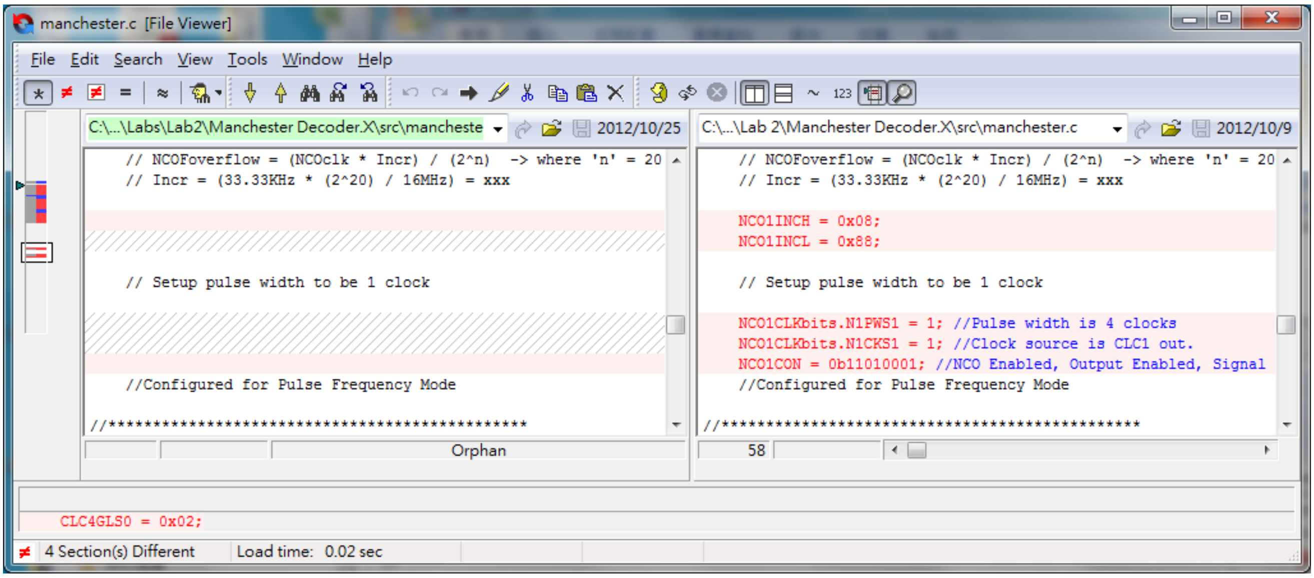
7. Plug in PICkit 3 to the computer and the demo board (J2), and program the part by clicking the "Make and Program Device" button.

8. Unplug the PICkit 3 and use the USB cable to power the demo board (Connector J1). Install driver if prompted.

9. **Test your project with a class helper.** After programming one set of board as a Manchester Decoder, program another set of boards as a Manchester Encoder (as in lab1). Then position the Encoder board so Q3 lines up with D1 on the Decoder board. Now, if you remember the "S3" button sends your message (on the encoder board setup). At the same time the indication LED (D5) should flash when S3 is pressed to show that data is being transmitted. Also, the words "Encoder: Testing # " should appear on the LCD. At the same time, the LCD on the Decoder board setup will display " Lab1…. Testing #". The looping counter value should be the same on both LCDs.


Line up Q3
to D1


manchester.c [File Viewer]
File Edit Search View Tools Window Help
C:\...\Labs\Lab2\Manchester Decoder.X\src\mancheste
2012/10/25
C:\...\Lab 2\Manchester Decoder.X\src\manchester.c
2012/10/9
/***
//Setup CLC
/***
//#include "lab2.inc"
//#include "CLC Decoder.inc"
//Setup Manchester Decoder Step 1
CLC4GLS0 = 0x02;
CLC4GLS1 = 0x80;
CLC4GLS2 = 0x00;
CLC4GLS3 = 0x00;
CLC4SEL0 = 0x61;
CLC4SEL1 = 0x44;
CLC4POL = 0x81;
CLC4CON = 0xC4;
//Setup Manchester Decoder Step 2
CLC2GLS0 = 0x04;
CLC2GLS1 = 0x00;
CLC2GLS2 = 0x02;
CLC2GLS3 = 0x00;
CLC2SEL0 = 0x70;
CLC2SEL1 = 0x02;
CLC2POL = 0x00;
CLC2CON = 0xC1;
//Setup Manchester Decoder Step 3
CLC1GLS0 = 0x08;
CLC1GLS1 = 0x20;
CLC1GLS2 = 0x20;
CLC1GLS3 = 0x80;
CLC1SEL0 = 0x51;
CLC1SEL1 = 0x05;
CLC1POL = 0x00;
CLC1CON = 0x80;
76
Differences
107
NCO1INCH = 0x08;
4 Section(s) Different
Load time: 0.02 sec


manchester.c [File Viewer]
File Edit Search View Tools Window Help
C:\...\Labs\Lab2\Manchester Decoder.X\src\mancheste
2012/10/25
C:\...\Lab 2\Manchester Decoder.X\src\manchester.c
2012/10/9
// NCOFoverflow = (NCOclk * Incr) / (2^n) -> where 'n' = 20
// Incr = (33.33KHz * (2^20) / 16MHz) = xxx
NCO1INCH = 0x08;
NCO1INCL = 0x88;
// Setup pulse width to be 1 clock
NCO1CLKbits.N1PWS1 = 1; //Pulse width is 4 clocks
NCO1CLKbits.N1CKS1 = 1; //Clock source is CLC1 out.
NCO1CON = 0b11010001; //NCO Enabled, Output Enabled, Signal
//Configured for Pulse Frequency Mode
//***
Orphan
58
CLC4GLS0 = 0x02;
4 Section(s) Different
Load time: 0.02 sec

LAB3 – Drive a Motor with the CWG

Jumper Settings:

1509 Enhanced Midrange Development Board

J4: Place a shunt connector horizontally on the bottom of J4.

J3: Place a shunt connector on J3 (located underneath the CWG daughter board).

J21: Place a shunt connector horizontally on the far left of J21 (5volts)

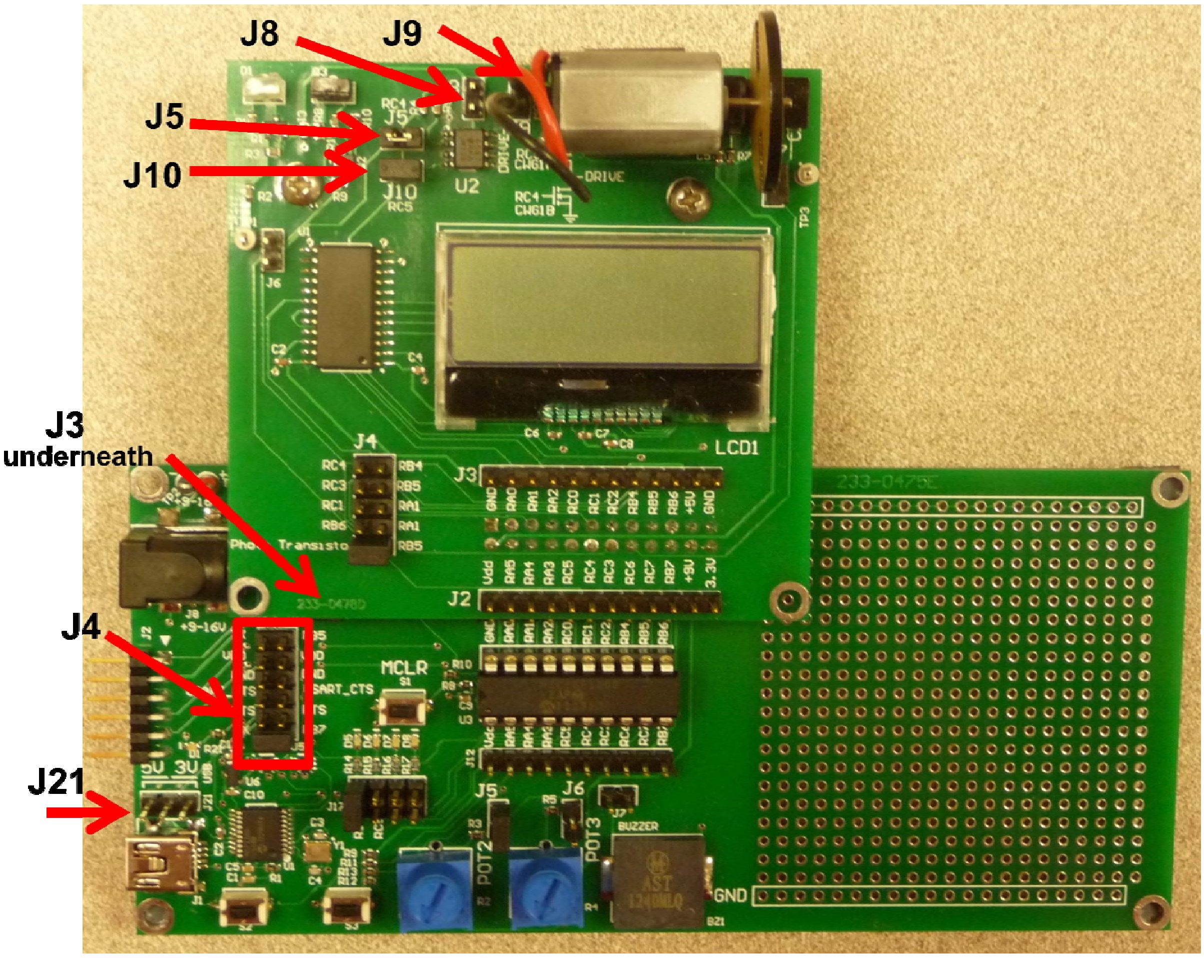
New Peripheral CWG Daughter Board

J5: Place shunt connectors horizontally on J5. This jumper connects the CWG output B (RC4) to the motor drive switch.

J10: Place shunt connectors horizontally on J10. This jumper connects the CWG output A (RC5) to the motor drive switch.

J8: Place a shunt connector vertically on J8. This jumper connects the motor drive switches to VDD.

J9: Unplug the motor until needed.

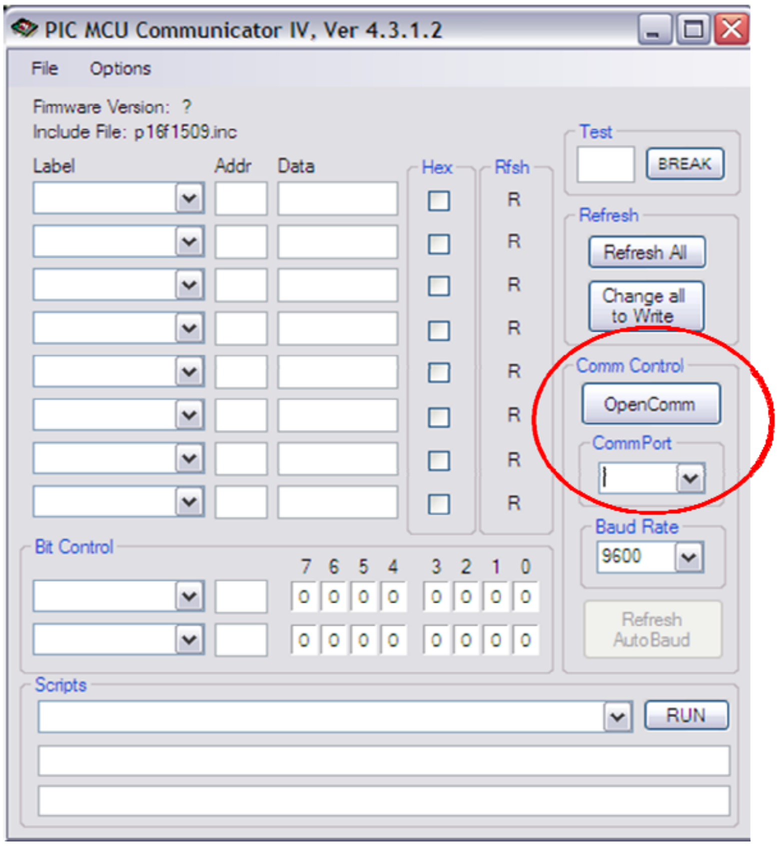


Procedure:

1. Open MPLAB X v1.41

2. In the tool bar at the top of the window, under the File drop-down menu, click on "Open project" and select the PIC Communicator project at: "**C:\1610\PIC Communicator (Lab3).X**".

3. Connect PICkit 3 and program device by clicking the "Make and Program Device" button.
Note: Make sure there are no jumpers on J5, J10 and J8, and the motor is not connected on J9 (see Jumper Settings section above).
4. Unplug the PICkit 3 and use the USB cable to power the demo board (Connector J1). Install driver if prompted.
5. Open the PIC Communicator GUI. Under "File" in the tool bar select " Open* .inc ". Here you will need to load the proper .inc file located at "C:\1610\CLCDesignerII\p16f1509.inc"
6. Select the proper Comm Port channel from the CommPort selector (usually the highest number), and press "OpenComm". If successful, the "Open Comm" button will change to "Close Comm".

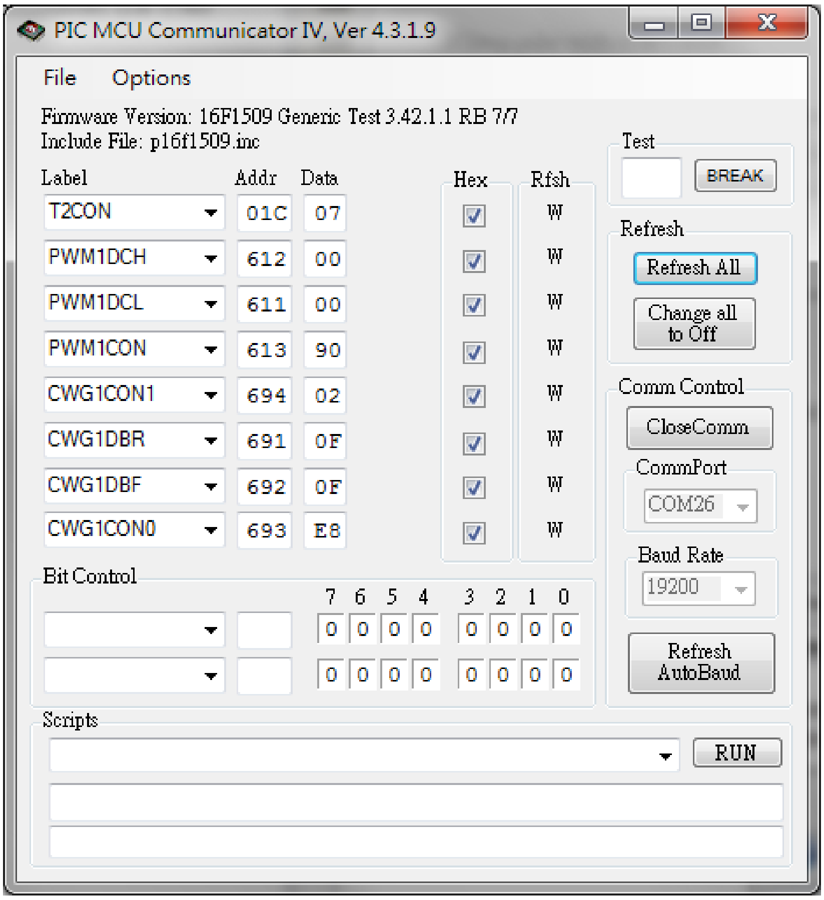


Remember to install MCP2200 driver in \Class Material 2\PicMCUCommIV

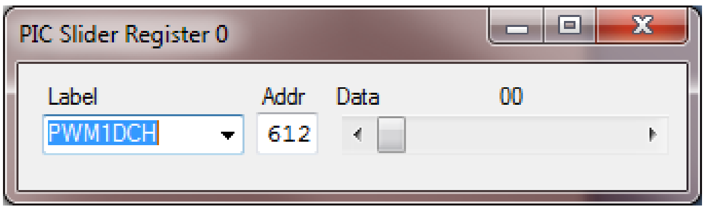
7. Connect Motor to J9 (red wire on top) and connect jumpers to J5, J10 and J8 on New Peripheral

CWG Daughter Board as stated in the Jumper Settings section above.

8. Use the PIC Communicator GUI to select and configure the proper register bit values for the PWM and CWG peripherals. Use the Pic16(L)F1509 datasheet, located in the 1610 directory of this project or at Microchip.com, to configure the PWM1 and CWG1 peripherals. They should be configured in this order:



8. Under "Options" in the tool bar, select "Add New Slider " or (Alt+S) and select register PWM1DCH. This will be your speed control. As you adjust the slider bar left or right, the speed of the motor will change accordingly.



LAB4 – Using the Manchester Encoder and Decoder to Control the motors speed.

Jumper Settings:

1509 Enhanced Midrange Development Board

J4: Place a shunt connector horizontally on the bottom of J4.

J3: Place a shunt connector on J3 (located underneath the CWG daughter board).

J21: Place a shunt connector horizontally on the far left of J21 (5volts).

J17: Place a shunt connector vertically on the far left of J17. This connects the indication LED, D5.

J5 : Place a shunt connector vertically on J5. This connects POT2 to the Pic MCU to vary the speed of the motor.

New Peripheral CWG Daughter Board

J7: Place a shunt connector vertically on connector J7. This jumper connects the Optical Encoder to the Pic MCU at RA4.

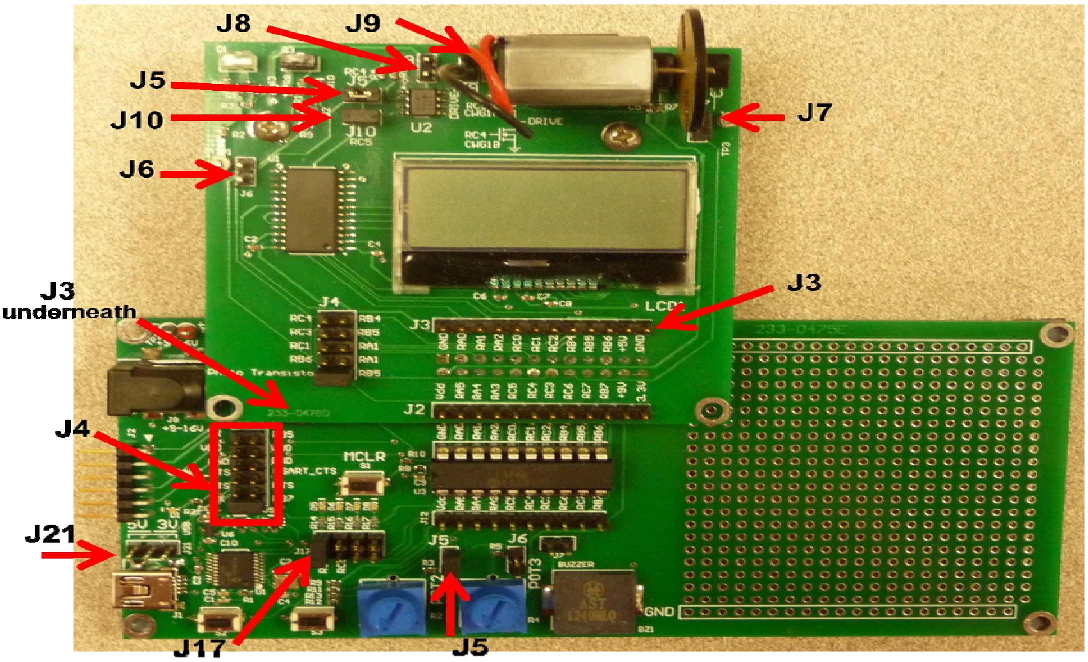
J6: Place a red jumper wire on the top pin of connector J6 and connect it to J3 pin 5 (RC0). This connects the transmitting IR LED to the Pic MCU.

J5: Place shunt connectors horizontally on J5. This jumper connects the CWG output B (RC4) to the motor drive switch.

J10: Place shunt connectors horizontally on J10. This jumper connects the CWG output A (RC5) to the motor drive switch.

J8: Place a shunt connector vertically on J8. This jumper connects the motor drive switches to VDD.

J9: Unplug the motor until needed.



Procedure:

1. Open MPLAB X v1.41

2. In the tool bar at the top of the window, under the File drop-down menu, click on "Open project" and select the Manchester Encoder project at: "**C:\1610\Lab4\Manchester Encoder.X**"

3. After expanding the project, expand the Source files directory. Now open manchester.c by double clicking on the file in the project window. Then in the code window, scroll down to the function " encoder_int ". This will be where the CLC will be configured.

4. Next open the CLC Designer Tool GUI (Icon on Desktop)

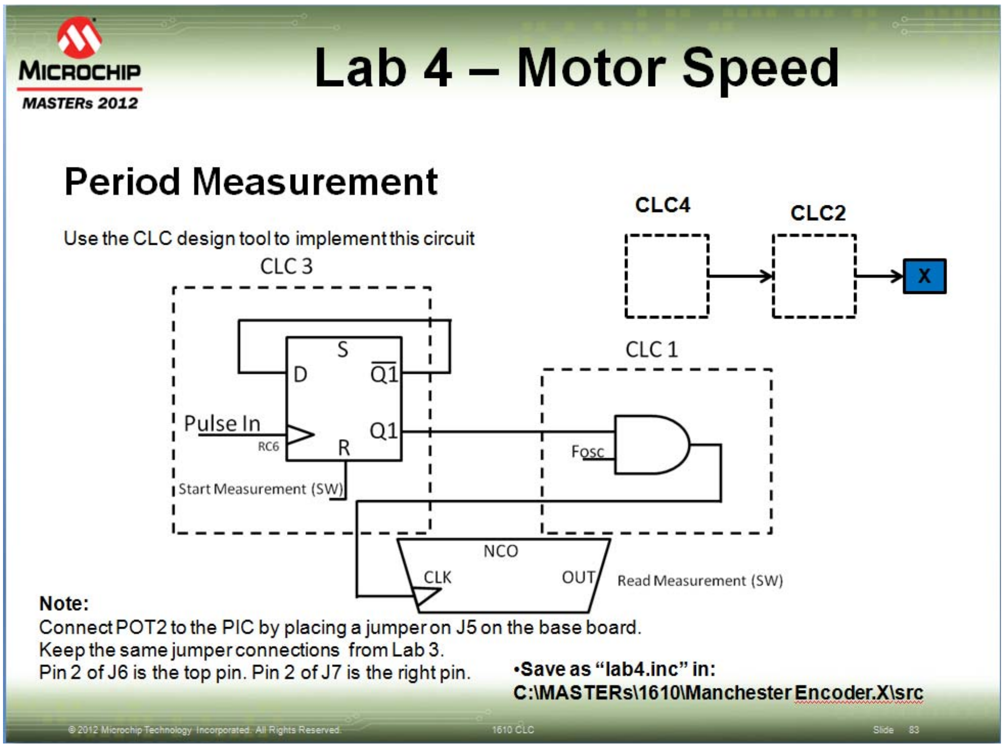
5. Use the Pic16(L)F1509 datasheet, located in the 1610 director of this project or at Microchip.com, in conjunction with the CLC Designer tool to setup and implement the Manchester encoder for Lab 4.

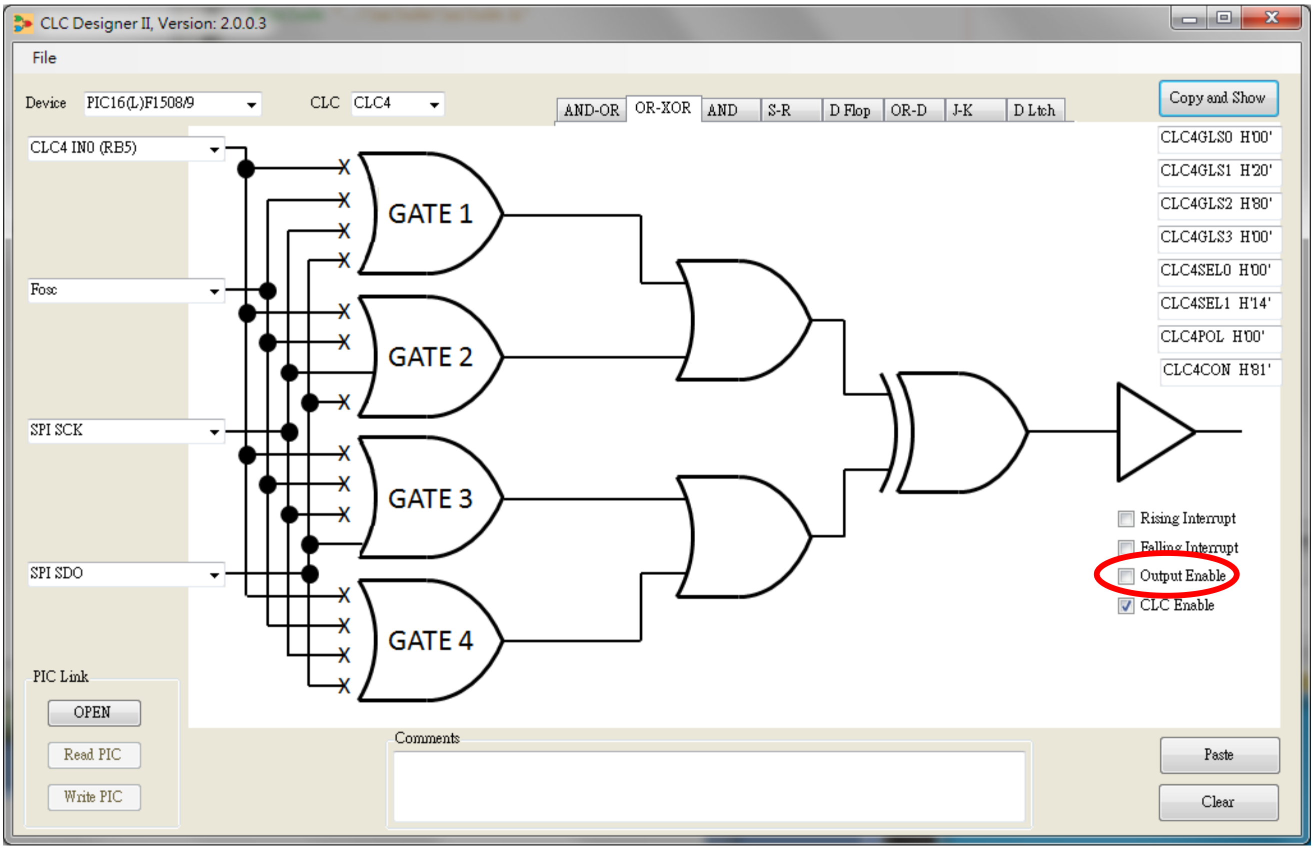
a. CLC1: Inputs: Fosc (Gate 2), CLC3 out (Gate 1)
Use the AND-OR logic function
Outputs: Enable CLC1

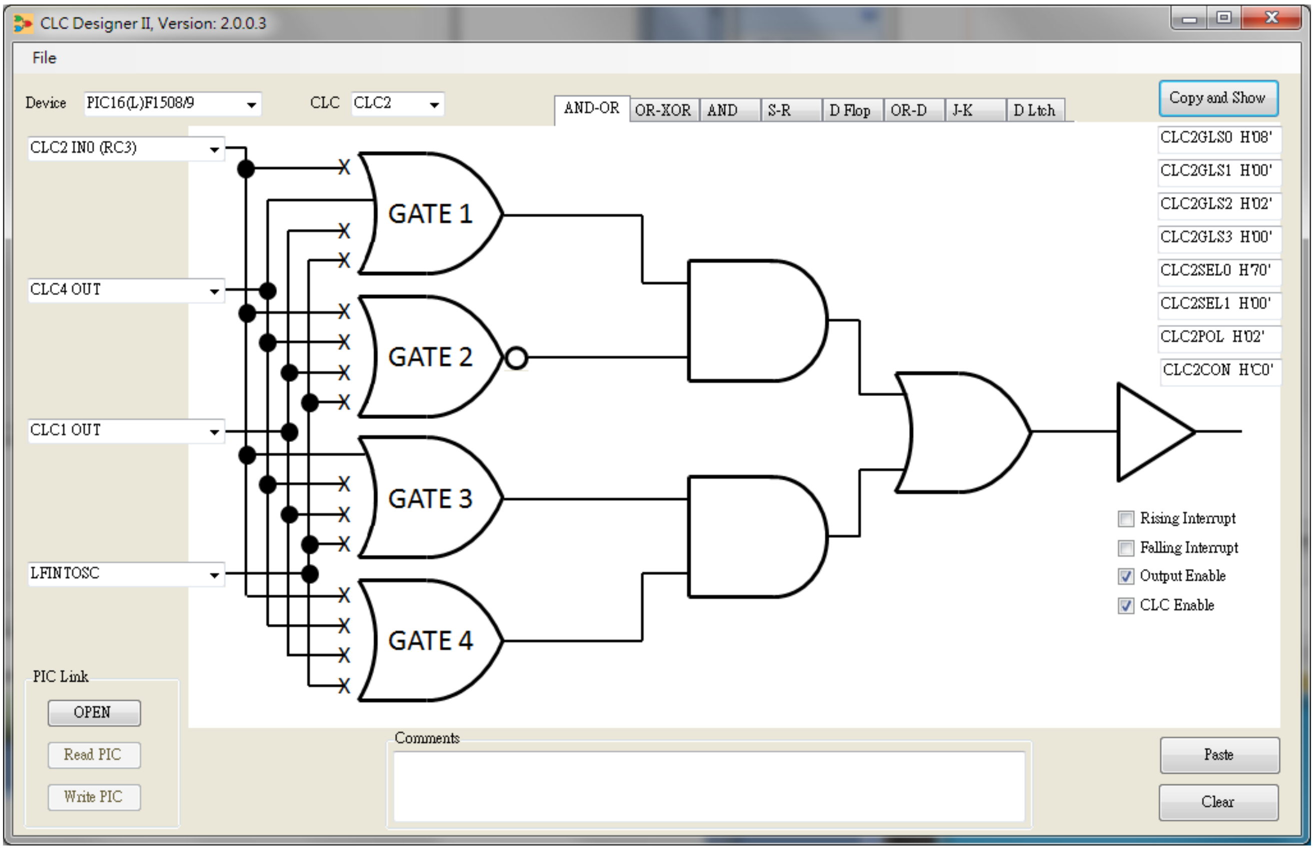
CLC2: Inputs: CLC2 input0 (Gate 3), CLC4 out (Gate 1), invert Gate 3 output
Use the AND-OR logic function
Outputs: Enable CLC2 and output

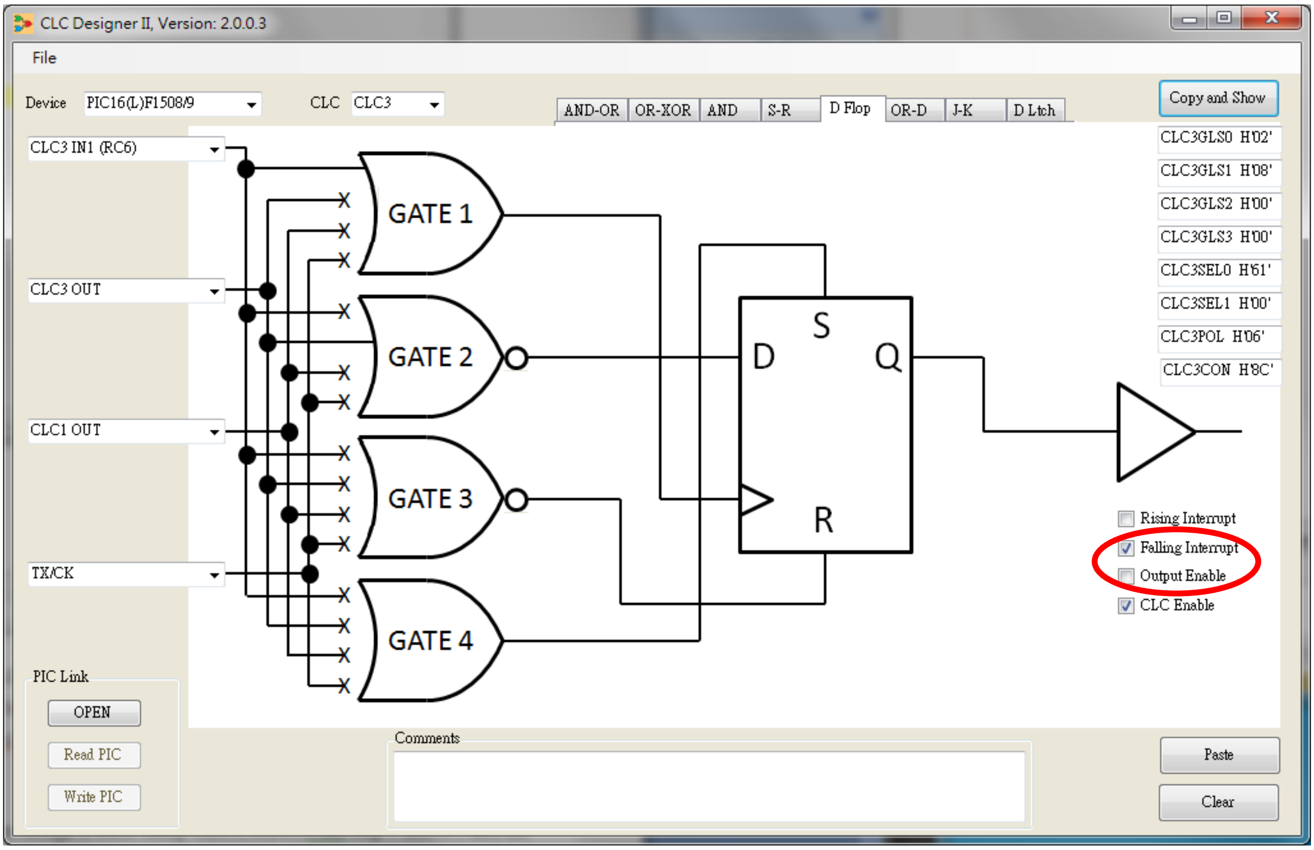
CLC3: Inputs: CLC3 output (Gate 2 and invert gate 2), CLC3 input1 (Gate 1), invert Gate 3 output
Use the D-Flop logic function
Outputs: Enable CLC3 and enable falling edge interrupt

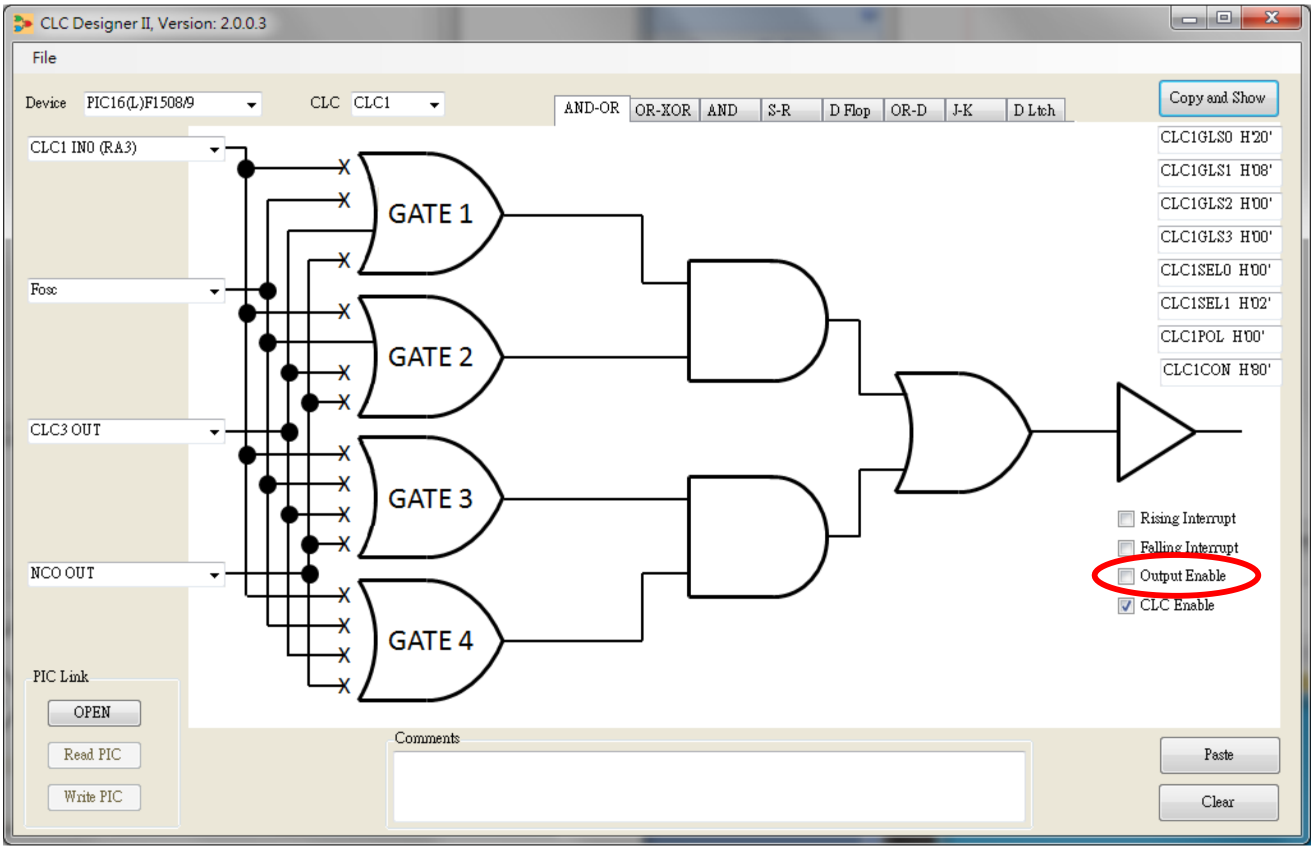
CLC4: Inputs: SPI SCK (Gate 2), SPI SDO (Gate 3)
Use the OR-XOR logic function
Outputs: Enable CLC4 enabled, do not enable CLC4 output, because it uses the same pin as the CWG (RC4). You will need to use a wire jumper from RC0 to pin 2 of J6 to make the encoder work.


MICROCHIP
MASTERs 2012
Lab 4 – Motor Speed
Period Measurement
Use the CLC design tool to implement this circuit
CLC 3
S
D
Q1
Pulse In
RC6
Q1
R
Start Measurement (SW)
CLC4
CLC2
X
CLC 1
Fosc
NCO
CLK
OUT
Read Measurement (SW)
Note:
Connect POT2 to the PIC by placing a jumper on J5 on the base board.
Keep the same jumper connections from Lab 3.
Pin 2 of J6 is the top pin. Pin 2 of J7 is the right pin.
•Save as "lab4.inc" in:
C:\MASTERs\1610\Manchester Encoder.X\src
© 2012 Microchip Technology Incorporated. All Rights Reserved.
1610 CLC
Slide 83


CLC Designer II, Version: 2.0.0.3
File
Device
PIC16(L)F1508/9
CLC
CLC4
AND-OR
OR-XOR
AND
S-R
D Flop
OR-D
J-K
D Ltch
Copy and Show
CLC4GLS0 H'00'
CLC4GLS1 H'20'
CLC4GLS2 H'80'
CLC4GLS3 H'00'
CLC4SEL0 H'00'
CLC4SEL1 H'14'
CLC4POL H'00'
CLC4CON H'81'
CLC4 IN0 (RB5)
Fosc
SPI SCK
SPI SDO
GATE 1
GATE 2
GATE 3
GATE 4
Rising Interrupt
Falling Interrupt
Output Enable
CLC Enable
PIC Link
OPEN
Read PIC
Write PIC
Comments
Paste
Clear


CLC Designer II, Version: 2.0.0.3
File
Device PIC16(L)F1508/9
CLC CLC2
AND-OR
OR-XOR
AND
S-R
D Flop
OR-D
J-K
D Ltch
Copy and Show
CLC2GLS0 H'08'
CLC2GLS1 H'00'
CLC2GLS2 H'02'
CLC2GLS3 H'00'
CLC2SEL0 H'70'
CLC2SEL1 H'00'
CLC2POL H'02'
CLC2CON H'C0'
CLC2 IN0 (RC3)
CLC4 OUT
CLC1 OUT
LFINTOSC
GATE 1
GATE 2
GATE 3
GATE 4
Rising Interrupt
Falling Interrupt
Output Enable
CLC Enable
PIC Link
OPEN
Read PIC
Write PIC
Comments
Paste
Clear


CLC Designer II, Version: 2.0.0.3
File
Device PIC16(L)F1508/9
CLC CLC3
AND-OR
OR-XOR
AND
S-R
D Flop
OR-D
J-K
D Ltch
Copy and Show
CLC3GLS0 H'02'
CLC3GLS1 H'08'
CLC3GLS2 H'00'
CLC3GLS3 H'00'
CLC3SEL0 H'61'
CLC3SEL1 H'00'
CLC3POL H'06'
CLC3CON H'8C'
CLC3 IN1 (RC6)
CLC3 OUT
CLC1 OUT
TX/CK
GATE 1
GATE 2
GATE 3
GATE 4
S
D
Q
R
Rising Interrupt
Falling Interrupt
Output Enable
CLC Enable
PIC Link
OPEN
Read PIC
Write PIC
Comments
Paste
Clear



b. After making all of the CLC settings, click "Copy and Show", and under "File" save as C code with the file name: " lab4.inc" at location: " C:\1610\Lab4\Manchester Encoder\src".

c. Next go to the Manchester.c code window and under the function encoder_init, add #include "lab4.inc".

6. Use the concepts outlined in Lab 2 to design a timer using CLC1, CLC2 (see step 5 above) and the NCO, that is capable of capturing the pulses coming from OPTO 2.

a. NCO setup: NCO1CON: Enable NCO
NCO1CLK: Select CLC1 as NCO clock
NCO1INC: Set NCO increment value for the maximum time and resolution
NCO1ACC: Set the NCO accumulator to zero
NCO1IE, PEIE, GIE: Enable the NCO, peripheral, and global interrupts

NCO1INCH = 0;
NCO1INCL = 1;
NCO1ACC = 0;
NCO1CLKbits.N1CKS = 0b10; //Select CLC 1 as Clock
NCO1CONbits.N1EN = 1;

NCO1IE = 1;

PEIE = 1;

GIE = 1;

b. Make all of these register settings to the NCO in the "encoder.int" function at location: " C:\1610\Lab4\Manchester Encoder\src".

7. Open main.c at location: " C:\1610\Lab4\Manchester Encoder\src", and add the CWG configuration settings used in Lab 3 and add it under the "CWG_Setup" function. Uncomment the "CWG_Setup" function line in the main function loop.

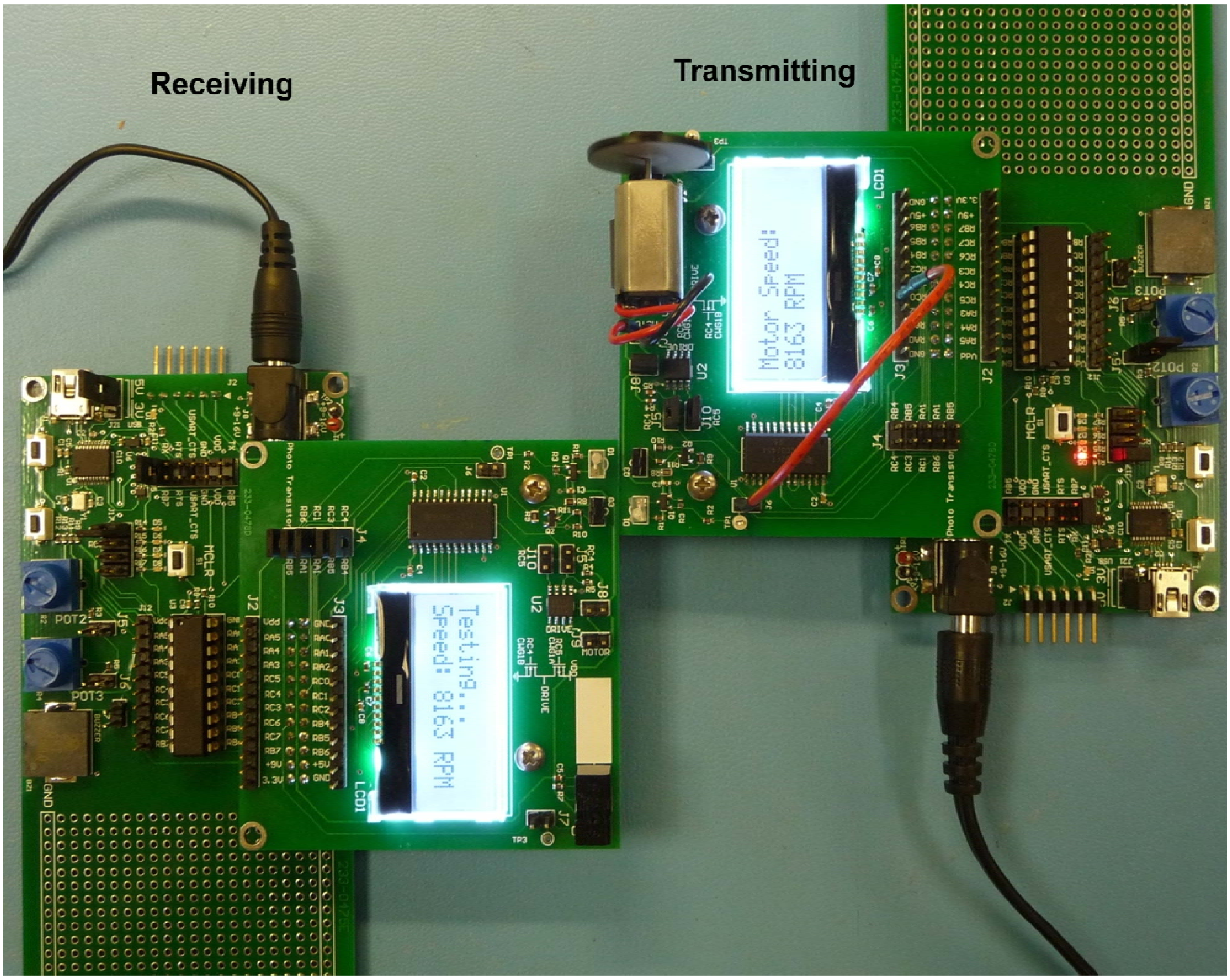
8. In the main while loop, the ADC is already setup to read from POT2 and transfer it to the duty cycle of PWM1.

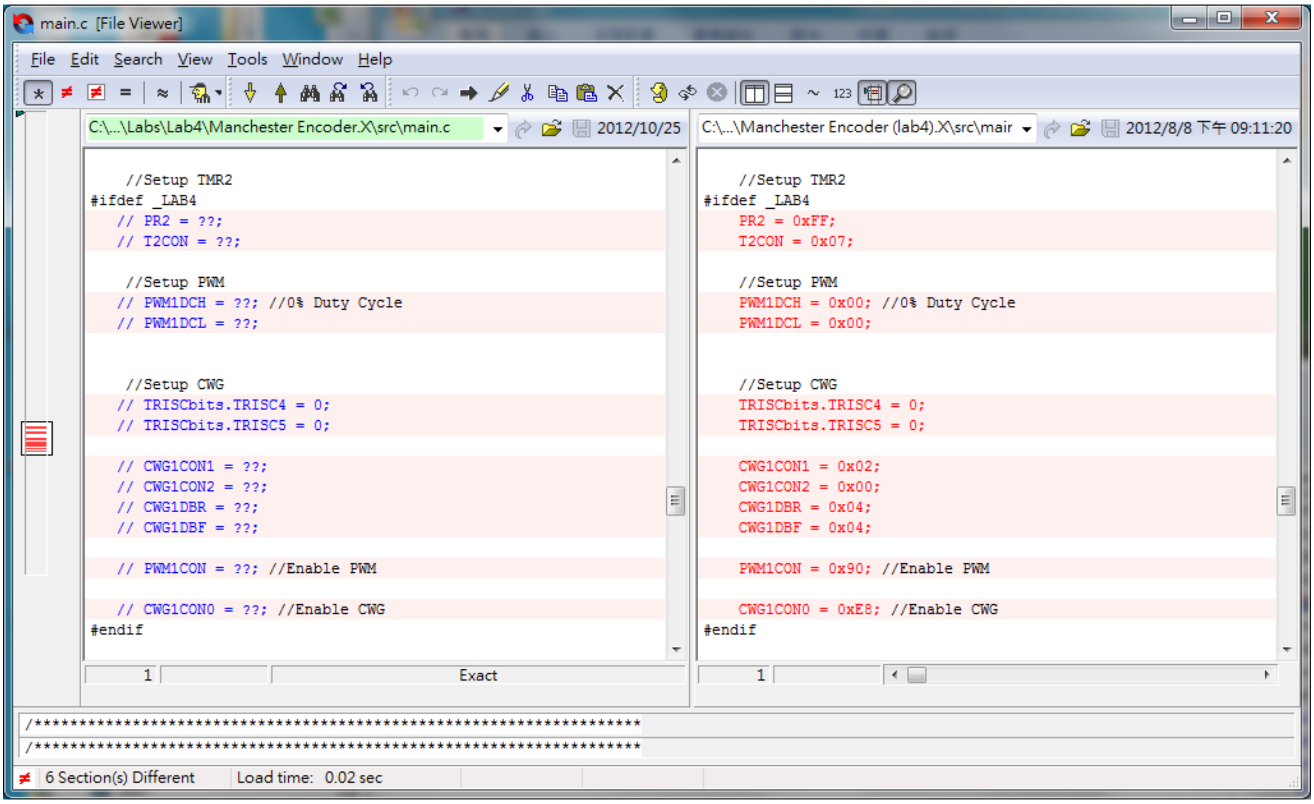
9. Plug in PICkit 3 to the computer and the demo board (J2), and program the part by clicking the "Make and Program Device" button.

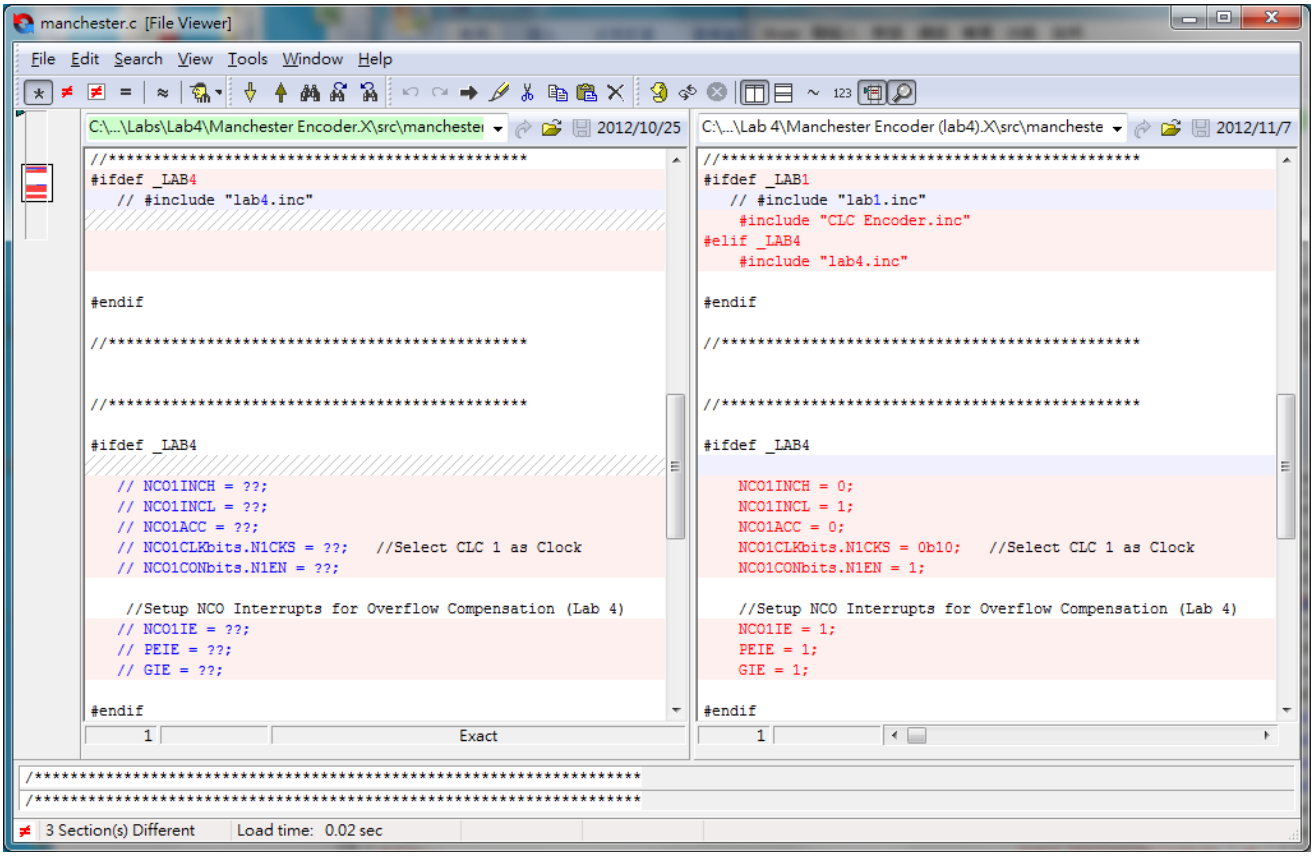
10. Unplug the PICkit 3 and use the USB cable to power the demo board (Connector J1). Install driver if prompted.

11. Now after adjusting POT2, press the "S3" button to send (transmit) your motors speed. At the same time the indication LED (D5) should flash when S3 is pressed to show that data is being transmitted. Also, the words "Motor Speed: # RPM " should appear on the LCD. The number displayed is the revolutions per minute of the motor as read by the optical encoder.

12. **To test your project with a class helper.** After programming one set of boards as directed in the previous steps and a second set of boards as a Manchester Decoder (as shown in lab2). Then position the transmit boards so Q3 lines up with D1 on the Decoder board. Now, if you remember the "S3" button sends your message (on the transmit board setup). At the same time the indication LED (D5) should flash on both sets of boards when S3 is pressed to show that data is being transmitted and received. Also, the words "Testing… Speed: # RPM " should appear on the receiving LCD. Thus, as you adjust the motor speed on the transmit board and press "S3", you will see the same motor speed displayed on the receiving board.


Receiving
Transmitting
Motor Speed:
8163 RPM
Testing...
Speed: 8163 RPM


main.c [File Viewer]
File Edit Search View Tools Window Help
C:\...\Labs\Lab4\Manchester Encoder.X\src\main.c
2012/10/25
//Setup TMR2
#ifdef _LAB4
// PR2 = ??;
// T2CON = ??;
//Setup PWM
// PWM1DCH = ??; //0% Duty Cycle
// PWM1DCL = ??;
//Setup CWG
// TRISCbits.TRISC4 = 0;
// TRISCbits.TRISC5 = 0;
// CWG1CON1 = ??;
// CWG1CON2 = ??;
// CWG1DBR = ??;
// CWG1DBF = ??;
// PWM1CON = ??; //Enable PWM
// CWG1CON0 = ??; //Enable CWG
#endif
Exact
C:\...\Manchester Encoder (lab4).X\src\mair
2012/8/8 下午 09:11:20
//Setup TMR2
#ifdef _LAB4
PR2 = 0xFF;
T2CON = 0x07;
//Setup PWM
PWM1DCH = 0x00; //0% Duty Cycle
PWM1DCL = 0x00;
//Setup CWG
TRISCbits.TRISC4 = 0;
TRISCbits.TRISC5 = 0;
CWG1CON1 = 0x02;
CWG1CON2 = 0x00;
CWG1DBR = 0x04;
CWG1DBF = 0x04;
PWM1CON = 0x90; //Enable PWM
CWG1CON0 = 0xE8; //Enable CWG
#endif
6 Section(s) Different
Load time: 0.02 sec


manchester.c [File Viewer]
File Edit Search View Tools Window Help
C:\...\Labs\Lab4\Manchester Encoder.X\src\mancheste
2012/10/25
#ifdef _LAB4
// #include "lab4.inc"
#endif
#ifdef _LAB4
// NCO1INCH = ??;
// NCO1INCL = ??;
// NCO1ACC = ??;
// NCO1CLKbits.N1CKS = ??; //Select CLC 1 as Clock
// NCO1CONbits.N1EN = ??;
//Setup NCO Interrupts for Overflow Compensation (Lab 4)
// NCO1IE = ??;
// PEIE = ??;
// GIE = ??;
#endif
Exact
C:\...\Lab 4\Manchester Encoder (lab4).X\src\mancheste
2012/11/7
#ifdef _LAB1
// #include "lab1.inc"
#include "CLC Encoder.inc"
#elif _LAB4
#include "lab4.inc"
#endif
#ifdef _LAB4
NCO1INCH = 0;
NCO1INCL = 1;
NCO1ACC = 0;
NCO1CLKbits.N1CKS = 0b10; //Select CLC 1 as Clock
NCO1CONbits.N1EN = 1;
//Setup NCO Interrupts for Overflow Compensation (Lab 4)
NCO1IE = 1;
PEIE = 1;
GIE = 1;
#endif
3 Section(s) Different
Load time: 0.02 sec