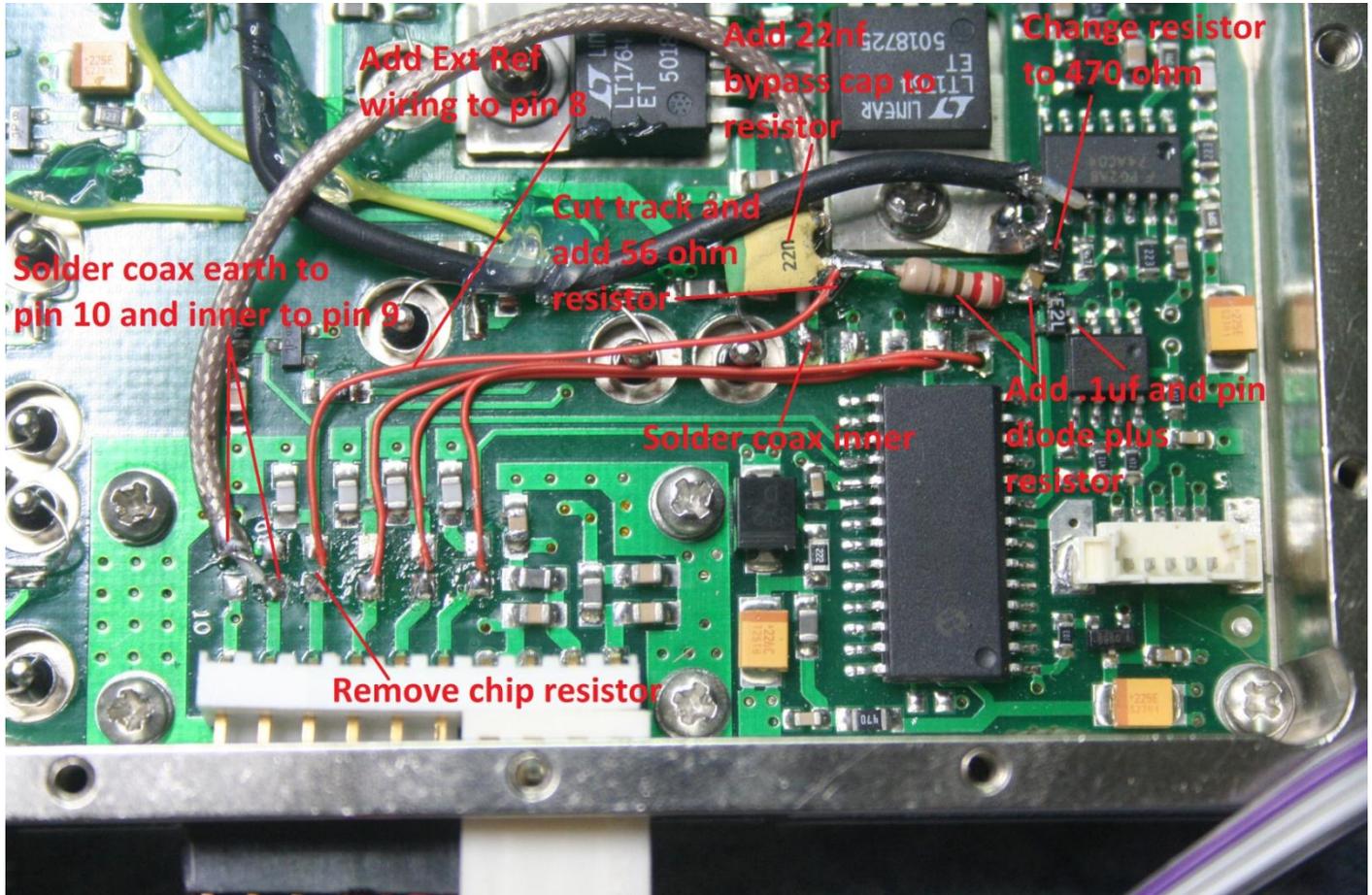


Reference Internal – External switch modification

To allow the selection of internal or external reference selection some modifications have to be made. The internal oscillator runs several functions with-in the unit. One signal is used for the 26 volt supply to drive the PLL, another derives the clock for the internal micro, which is no longer used, and thirdly it provides the negative bias for the GASFet amplifiers. I decided not to touch this oscillator as it could inadvertently cause loss of bias for the fet amplifiers and their possible destruction.

Pin 8 of the 74AC04 buffer feeds the PLL reference via a 47 ohm resistor. This resistor was removed and replaced by a 470 ohm resistor. A .1uf capacitor and a pin diode were added to the PLL side of the 470 ohm resistor.

In addition, the track was cut near where the signal goes into the PLL reference link and a 56 ohm resistor was added. Pin 9 of the interface connector has a piece of small coax (RG179) added to connect this pin to the PLL reference input. There is a 220 ohm resistor added to the junction of the .1uf and pin diode to provide bias for the pin diode so it can isolate the internal reference signal to the PLL. The DC bias control line is connected to pin 8 of the interface connector and a smd resistor is removed from the pin 8 position as per the attached photo.



When +5 volts is applied to pin 8 of the interface connector the pin diode is biased on and effectively shorts the internal reference signal going to the PLL (leaving all other oscillator functions running) and provides a termination to the external reference via the 56 ohm resistor.

Connecting an external reference will then operate the PLL. Any signal from -10dBm to +10dBm will function correctly. An external reference CANNOT be connected without power applied to pin 8 of the connector otherwise beating of the two signals will occur.

Out of Lock detection

The initial out of lock detect on the unit will no longer function with the modifications listed above. This is due to the initial way the system was designed, the O/L function from the PLL was multiplexed with the serial data line to the microprocessor.

The external microcontroller has been programmed to either load data on a write cycle or read the O/L function when not writing data to the PLL. The added resistor ensures the correct voltage levels are available for the external controller to detect the O/L signal.



18F2520 Controller PCB

A universal controller board was designed based on a Microchip 18F2520 microprocessor.

The board communicates with the Elcom synthesiser via the SPI bus interface to either load the frequency data or to read the O/L status from the synthesiser.

Port B.0 provides the out of lock signal. The pin is high on out of lock and low when locked.

This pin is 5 volt TTL from the microcontroller so caution should be exercised if connecting to external circuits.

The ICSP header is used to program the microprocessor.

The 12 volt supply header, SPI bus connector and PortB.0 are the only connections used. With modifications to the software the other pins can be used to select multiple frequencies from the unit.

Interface Connector pin out

Pin 1	+8 volts
Pin 2	+12 volts
Pin 3	Earth
Pin 4	Earth
Pin 5	Serial data (SDO) connection
Pin 6	Serial clock (SCK) connection
Pin 7	Latch enable (EN)
Pin 8	Internal/External Reference select
Pin 9	External 10Mhz reference input
Pin 10	Earth

Interconnection cable from controller to Elcom

Controller SPI connector

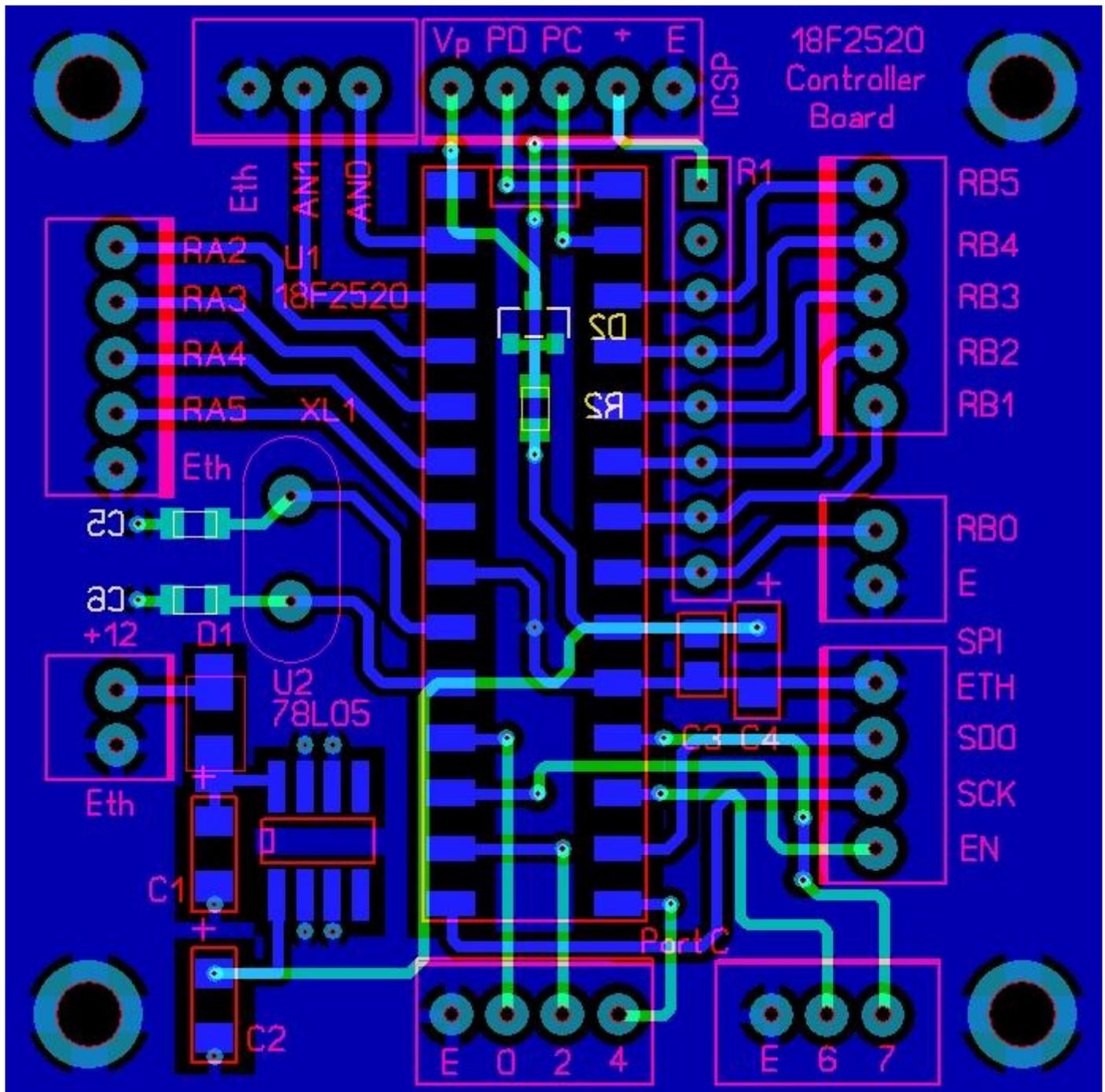
Elcom Connector

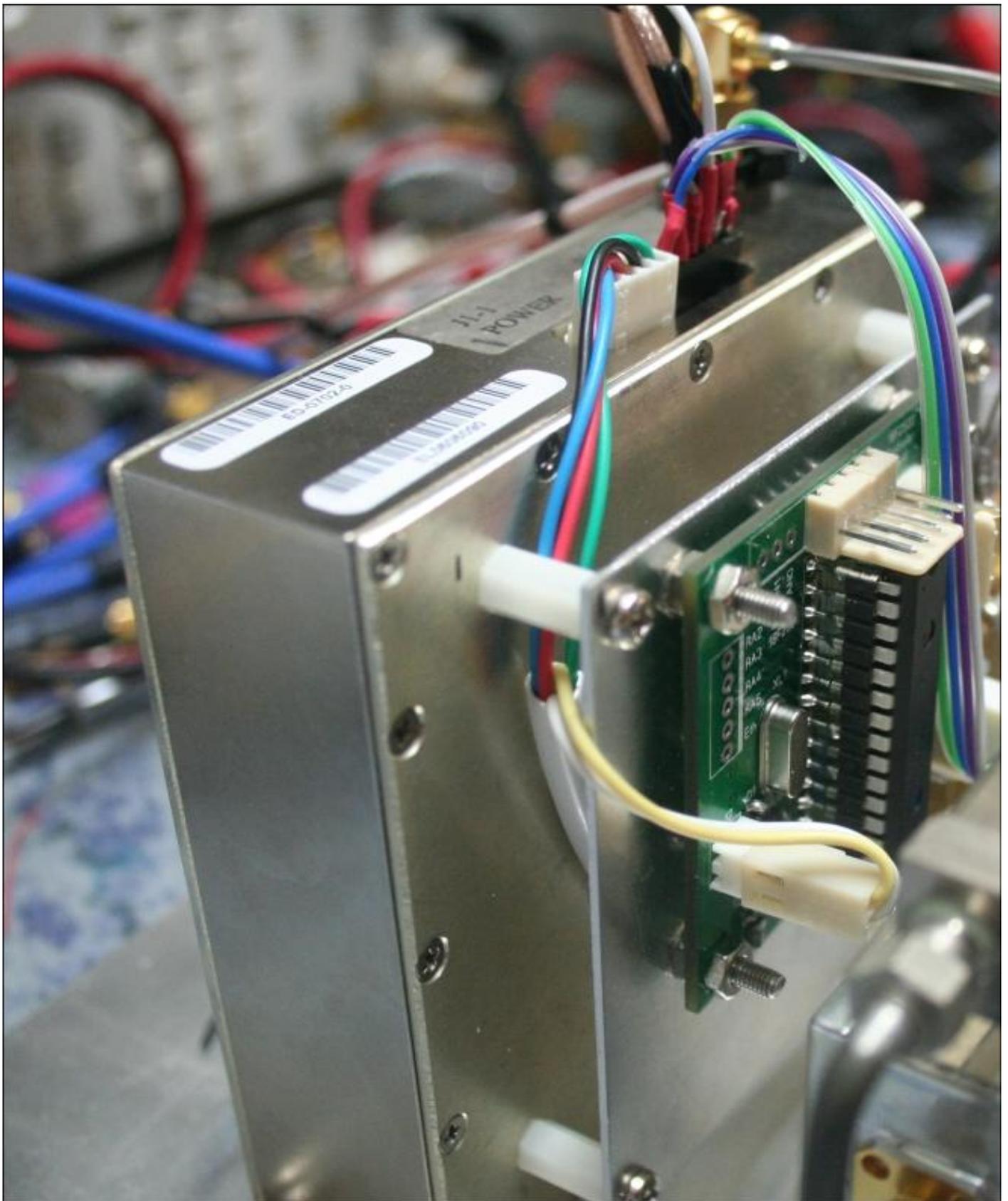
ETH -----Pin 10

SDO -----Pin 5

SCK -----Pin 6

EN -----Pin 7





The software is written in MikroBasic from MikroE.

The free version of the compiler is available from here : <http://www.mikroe.com/mikrobasic/pic/>

The calculation of the data for frequency selection is easily done from the Analogue Devices application available from here :

<http://www.analog.com/en/rfif-components/pll-synthesizersvcos/adf4252/products/EVAL-ADF4252/eb.html>

Download and install the application. When it loads you will have to select the ADF4252 synthesiser and ignore the warning about no USB found. This software is written for their development board

Screen shot from software showing hex data for the values used in the program.

The screenshot shows the 'Analog Devices Evaluation Software' window for the ADF4252. The interface is divided into several sections:

- (RF and IF Sections):** Contains control buttons for 'Counter Reset Disabled', 'CP 3State Off', and 'XO Enabled'. It also has radio buttons for 'Lowest Spur Mode', 'Low Noise and Spur Mode', and 'Lowest Noise Mode Muxout'. A dropdown menu for 'RF Digital Lock Detect' is set to 'Off'. Buttons for 'RF AND IF Powerdown Disabled', 'IF Powerdown Disabled', and 'RF Powerdown Disabled' are present.
- RF Section:** Displays 'RF VCO Output Frequency: 2362.80000MHz', 'PFD Frequency: 20000.00000kHz', 'REF IN Frequency: 10.00000MHz', 'Modulus: 100', 'Channel Step Size: 200.00000kHz', and 'RF Prescaler: 8/9'.
- IF Section:** Displays 'IF VCO Output Frequency: 540.00000MHz', 'PFD Frequency: 200.00000kHz', 'REF IN Frequency: 10.00000MHz', and 'IF Prescaler:'. Below these are buttons for 'CP Gain = 0', 'IF LDP = 3', and 'RF Phase Resync Disabled'.
- RF Section (Right):** Shows 'RF Charge Pump Current Setting: 4.3750 mA' and buttons for 'RF PD Polarity Negative', 'Doublor Enabled', 'RF Counter Reset Disabled', and 'CP 3 State Off'.
- IF Section (Right):** Shows 'IF Charge Pump Current Setting: 5.0000 mA' and buttons for 'IF PD Polarity Positive', 'Doublor Disabled', 'IF Counter Reset Disabled', and 'CP 3 State Off'.
- Warning:** A yellow box at the bottom left says 'USB Not Detected'.
- ADF4252 Register Section:** Features a table of 'Currently Loaded in Registers' and a set of update buttons.

MSB	Binary	LSB	Hex
000000101010001000100100			2A224
000000000110010101			195
0000011110000110			786

MSB	Binary	LSB	Hex
00111011000000001110000			380070
110001000001100100001			188321
000000001000111001000010			8E42
00000000000011000000011			603

Buttons for register updates: 'All IF Registers Updated', 'All RF Registers Updated', 'R4 Updated', 'R0 Updated', 'R5 Updated', 'R1 Updated', 'R6 Updated', 'R2 Updated', 'R3 Updated', and 'All Registers Updated'. Additional buttons for 'Frequency Sweep' and 'Frequency Hop' are located at the bottom right.

Screen shot showing PLL setup parameters

RF Output Frequency

Synthesizer Frequency

PLL

Enter the Reference Frequency (MHz):

Doubler Disabled: X 2

R:

Ref/2 Disabled:

PFD Frequency (kHz):

Enter the Channel Step Resolution (kHz):

N:

VCO

Enter the RF Output Frequency (MHz):

Legend:

- Changing Allowed
- Not Changeable

INT: MOD: FRAC: P: R:

PFD Freq = Reference Freq * Doubler / R INT = RF Freq / PFD Freq

Increment by one channel step (Normal Mode). Update R0.

Decrement by one channel step (Normal Mode). Update R0.

Update R0 and R1 (Normal Mode)

Exit Window

Update R0 and R1 (LCD Mode)

Increment by one channel step (LCD Mode). Update R0 and R1

Decrement by one channel step (LCD Mode). Update R0 and R1

LCD Mode: This will automatically calculate the lowest common denominator, and use this as the fraction numerator and denominator.
 Normal Mode: This will use the default modulus as the denominator.
 Example: If the fraction = 40/100. The LCD Mode will program the fraction as 2/5. The Normal Mode will program the fraction as 40/100


```

    Delay_ms(10)

    SPI1_Write ($06)    '06    'program the Master Register register
    SPI1_Write ($03)    '03
    Delay_ms(10)

    SetBit (PortC, 1)

    TRISC = $FF          'return PortC to inputs

    Delay_ms(2000)

end sub

main:

    TRISC = $FF          'Set Port C to inputs
    TRISB = $00          'Set PortB to outputs
    Delay_ms(200)

                                'At startup select channel not used in this version
' if PortD.4 = 1 then
Channel_1
' end if

' if PortD.5 = 1 then
' Channel_2
' end if

' if PortD.6 = 1 then
' Channel_3
' end if

' if PortD.7 = 1 then
' Channel_4
' end if

run:                                ' check PLL lock status

while true
    if Button(PORTC, 5, 10, 0) then
        PortB.0 = 1
    end if

    if Button(PORTC, 5, 10, 1) then    'PortB.0 = 0 when phase locked
        PORTB.0 = 0
    end if
wend
end.

```

