Shallow Water FOCE Sensor Node Board
PIC Test Code Document, v0.3

Chad Kecy
Monterey Bay Aquarium Research Institute

June 9, 2014

Table of Contents

Introduction

The PIC test code for the Shallow Water FOCE Sensor Node (SN) Board is a modified version of code written for an older development board. The root command set remains the same; however, additional functions have been added. This test code allows the end user to manually perform basic tasks on the SN board. It is very useful for initial board testing and could be easily modified into operational code. The PIC code was written in C and compiled and loaded using MPLAB tools from Microchip.

This document will cover:

1) How to program the microcontroller on the SN board
2) How to establish serial communication with the SN board
3) The built in commands in the PIC Test Code
4) A simple load test using the commands

Getting Started

This document assumes that the user is beginning with a brand new, un-programmed SN board. The Hardware Interface Document covers how to initially physically configure the SN board. Do not continue with the programming until this has been done.

If you have an SN board which has already been programmed, you can skip to the serial communication and/or testing sections.

To begin programming and testing of the SN board, the following items will be required:

Computer with serial port (or USB)
Sensor Node board (properly configured)
DC Power Supply capable of 48Vdc at 2A
Power In cable for SN input
Serial cable for SN
MPLAB ICD3 Programmer (or equivalent)
PIC Programming cable + USB cable
MPLAB X software (free from Microchip)
PIC test code project

Optional items:
USB-to-Serial Adaptor
Sensor Node Board Hardware Interface Document v0.6
Multi-meter
Programmable DC Load
Load cable(s) for SN outputs
swFOCE Sensor Node Board schematic

Programming the SN Board

1) Set the DC power supply to 48Vdc and adjust the current limit to 2A.
    a. Turn off the power supply
2) Connect the power in wires from the power supply to the SN board
    a. The Phoenix connector is plugged into J1
3) On your computer, open up MPLAB X
4) In MPLAB X, load the PIC Test Code project

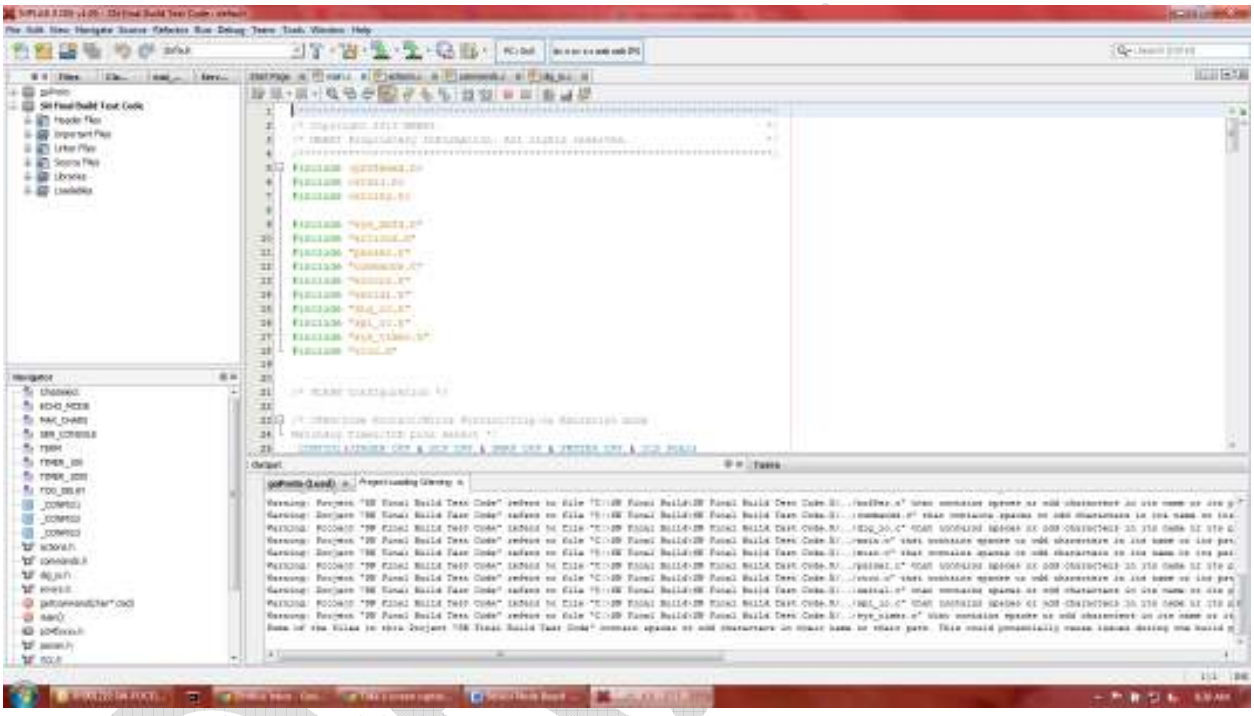


Figure 1: MPLAB X with PIC Test Code loaded

5) Attach the ICD 3 programmer to the computer with a USB cable
6) Attach the ICD 3 programmer to J6 on the SN board with the programming cable
    a. On the ICD3 , the power LED should be lit green and the active LED should be lit blue
7) Attach the DB9 side of the serial cable to the computer (or USB-to-Serial Adaptor)
8) Attach the serial cable to J10 of the SN board
    a. The ground wire from the serial cable should be attached to the ground on the DC Power Supply

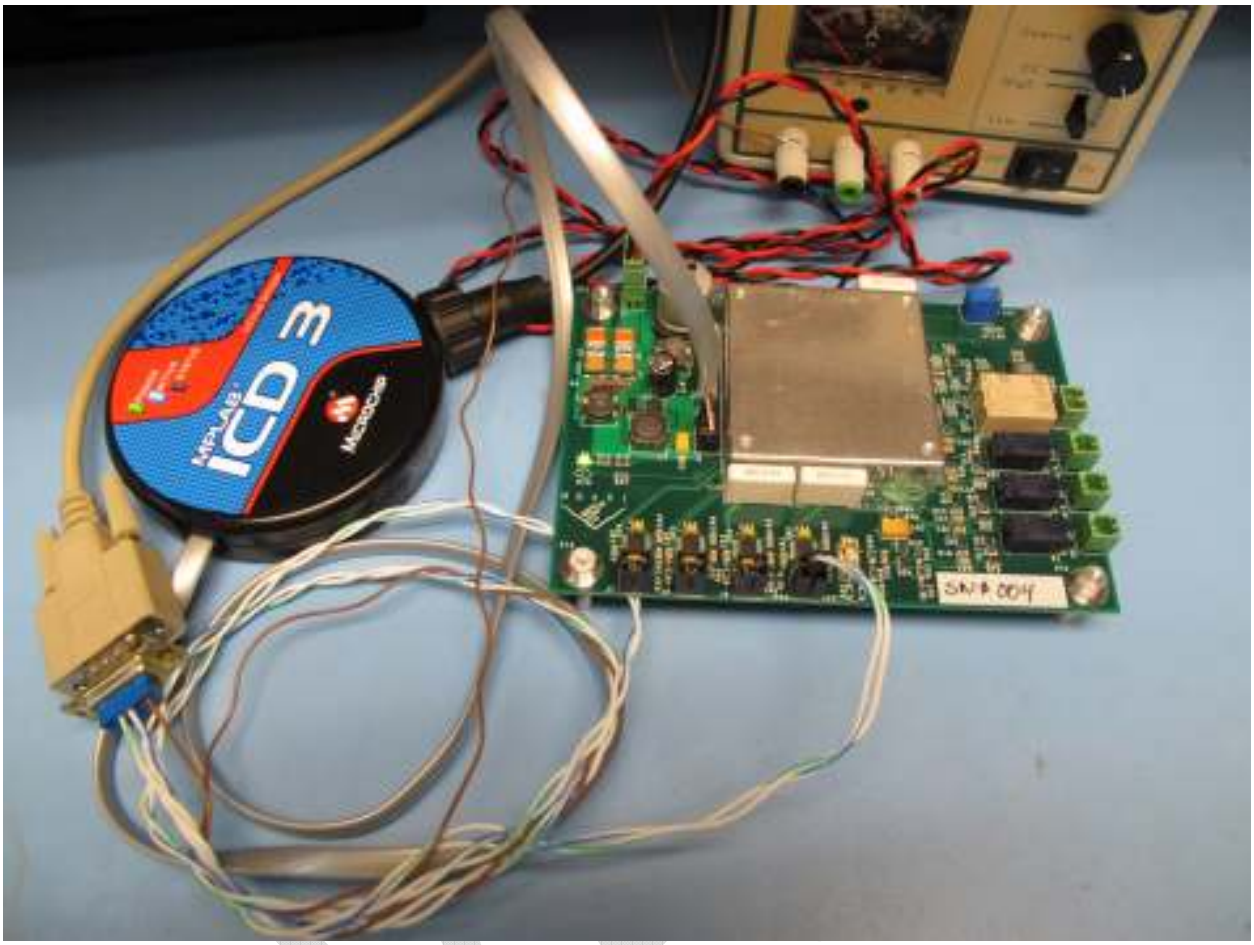


Figure 2: SN Board ready for programming

9) Turn on the power supply
    a. D2 on the SN board should be illuminated (provided R12 is installed)
    b. The power supply should be reading approximately 30ma.
10) In MPLAB X, hit the Make and Program Device button (icon on top row with downward arrow).
    a. A sub window at the bottom of the screen will display progress
    b. When complete, a Programming/Verify complete message will appear
11) Open a Hyperterminal window on the computer
12) Make a new connection with 9600,n,8,1 and no hardware control
13) Hit the Enter key
    a. a RDY message should appear on the screen
14) Short header JP1 with a jumper
    a. an acknowledgement message should appear on the screen

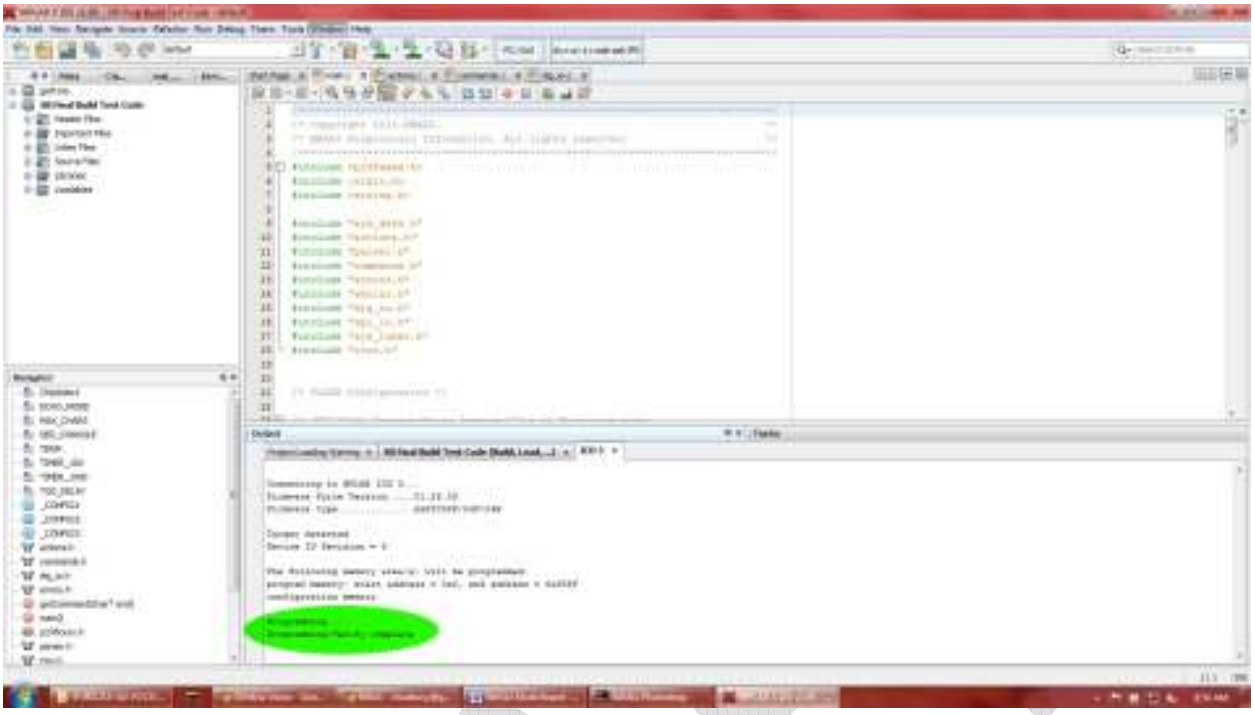


Figure 3: Programming successful

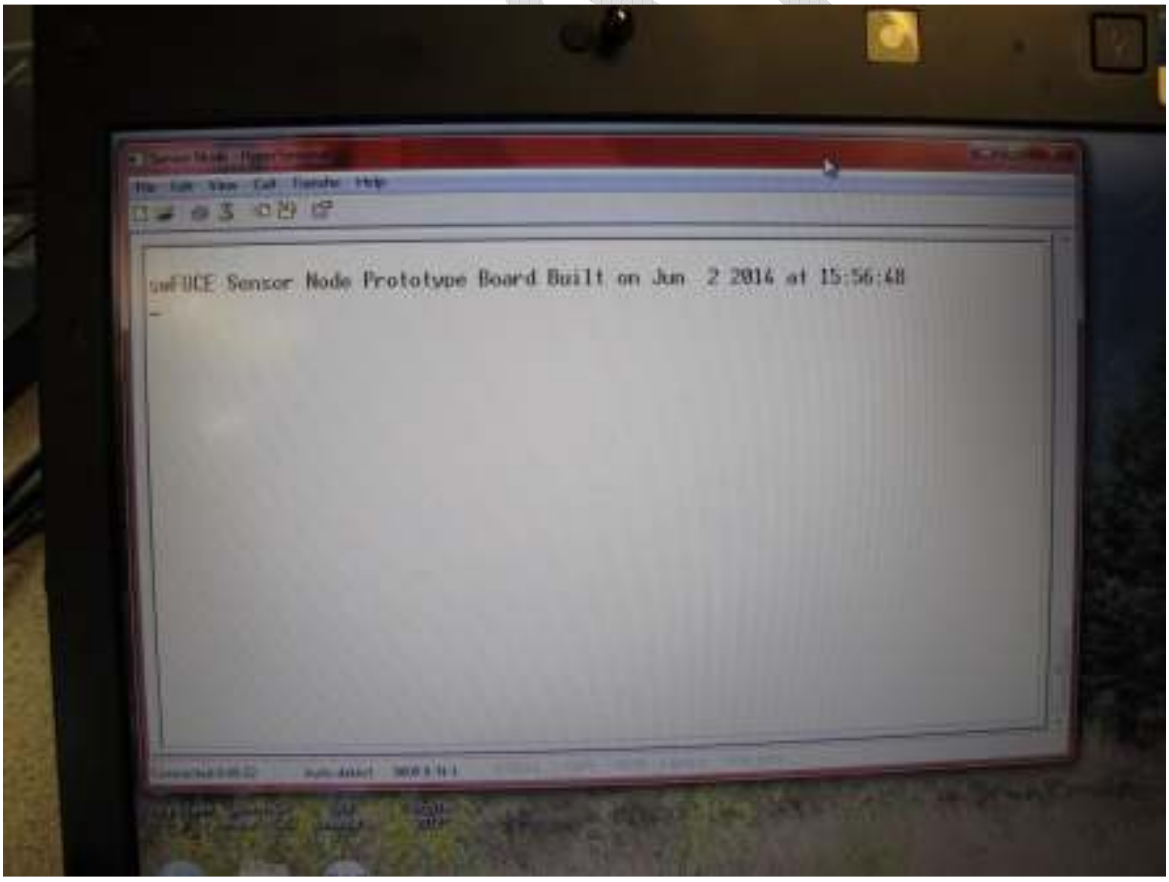


Figure 4: Hyperterminal window with acknowledgement message

Establishing Serial Communication with the SN Board

1) Set the DC power supply to 48Vdc and adjust the current limit to 2A.
    a. Turn off the power supply
2) Connect the power in wires from the power supply to the SN board
    a. The Phoenix connector is plugged into J1
3) Plug in the serial cable to the computer (or to USB-to-Serial adaptor)
    a. The Molex connector is plugged into J10 (serial port #4)
4) Open a Hyperterminal window on the computer
5) Make a new connection with 9600,n,8,1 and no hardware control
6) Turn on the power supply
    a. D2 on the SN board should be illuminated (provided R12 is installed)
    b. The power supply should be reading approximately 30ma.
7) In the Hyperterm window, an acknowledgement message should appear
8) The SN board is now ready to accept user commands

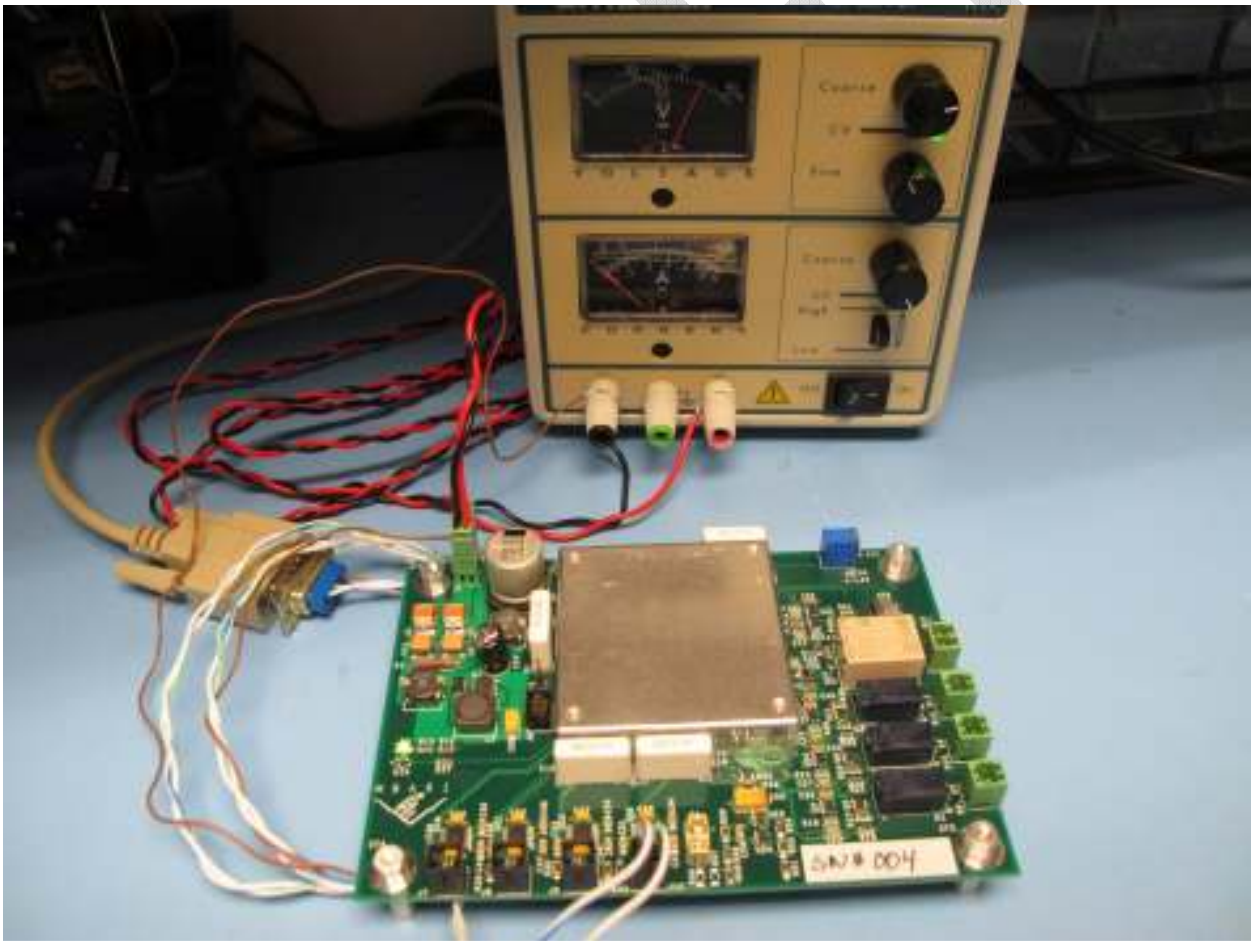


Figure 5: Initial Sensor Node board setup

Functional Commands

The root commands are top level commands which call lower level functions, as described below. The root commands are defined in commands.c; the lower level functions are defined in actions.c.

SET

These commands are used in conjunction with the SET command to change the indicated variable to a logical high.

DOUT

Function: Changes the selected digital out pin to a logical 1, or high.
Usage: SET DOUT x (where x is the software assigned BIT_OUT digital out line)

RELAYON

Function: Turns on the requested relay by sending out a 50msec pulse
Usage: SET RELAYON x (where x is the SN channel number, 1-4)

RELAYOFF

Function: Turns off the requested relay by sending out a 50msec pulse.
Usage: SET RELAYOFF x (where x is the SN channel number, 1-4)

GET

These commands are used in conjunction with the GET command to read back the indicated variable

DIN

Function: Reads the selected digital pins which are configured as inputs.
Parameters: A returned 0 is a logical low (0V) and a returned 1 is a logical high (5V)
Usage: GET DIN x (where x is the software assigned IN_BIT digital in line)

DOUT

Function: Reads the selected digital pins which are configured as outputs.
Parameters: A returned 0 is a logical low (0V) and a returned 1 is a logical high (5V)
Usage: GET DOUT x (where x is the software assigned BIT_OUT digital out line)

RELAY

Function: Reads the state of the requested relay.
Parameters: A returned 0 indicates relay is open, a 1 indicates relay is closed
Usage: GET RELAY x (where x is the SN channel number, 1-4)

ADC (Not yet implemeted)

Function: Reads the value of the requested channel of the ADS8344 (U20).
Parameters: Counts in 0 to 65536. Will need to convert, based on channel number.
Usage: GET ADC x (where x is the ADC channel number, 1-8)

CLR

These commands are used in conjunction with the CLR command to change the indicated variable to a logical low.

DOUT

Function: Changes the selected digital out pin to a logical 0, or low.
Usage: CLR DOUT x (where x is the software assigned BIT_OUT digital out line)

Command Examples

SET DOUT 11	This turns on the "Test LED"
GET DOUT 11	This returns the state of "Test LED", now a 1
CLR DOUT 11	This turns off the "Test LED"
GET DOUT 11	This returns the state of "Test LED", now a 0
GET DIN 4	This returns the state of ADC_BUSY line, either 0 or 1
SET RELAYON 1	This turns on relay #1
GET RELAY 1	This returns the state of relay #1, now a 1
SET RELAYOFF 1	This turns off relay #1
GET RELAY 1	This returns the state of relay #1, now a 0

Specific Pin Notes

BIT_OUT_1 through BIT_OUT_8 drive signals INSTR_x_ON and INSTR_x_OFF, which in turn control current through the coils of the four power switching relays on the SN board. These relays are latching style relays, meaning that only a temporary pulse is required to switch the state of the relays. Driving the coils constantly ON will degrade the coil and greatly reduced its lifespan. Use the RELAYON and RELAYOFF commands to change the states of the relays only.

Table 1: Digital Out & In Assignments

DIGITAL OUT LINES		
BIT_OUT	PIC Port	Name
0	D6	UART_1_*SHDN
1	E0	INSTR_1_ON
2	E2	INSTR_2_ON
3	E4	INSTR_3_ON
4	E6	INSTR_4_OFF
5	E1	INSTR_1_OFF
6	E3	INSTR_2_OFF
7	E5	INSTR_3_OFF
8	E7	INSTR_4_ON
9	D7	UART_2_*SHDN
10	D8	UART_3_*SHDN
11	F5	"Test LED", D19
12	F6	DC-DC_CNTL
13	G2	ADC_*SHDN
14	G6	SPI_*CS
15	B0	spare
16	B1	spare
17	B2	spare
18	B3	spare
19	B4	spare
20	B5	spare
21	C13	spare
22	C14	spare
23	D11	spare
24	F4	spare
25	dummyBit	N/A
26	dummyBit	N/A
27	dummyBit	N/A
28	dummyBit	N/A
29	dummyBit	N/A
30	dummyBit	N/A
31	dummyBit	N/A

DIGITAL IN LINES		
IN_BIT	PIC Port	Name
0	F0	INSTR_1_STATE
1	F1	INSTR_2_STATE
2	F2	INSTR_3_STATE
3	F3	INSTR_4_STATE
4	G3	ADC_BUSY
5	B8	CONFIG_BIT_0
6	B9	CONFIG_BIT_1
7	B10	CONFIG_BIT_2
8	B11	CONFIG_BIT_3
9	B12	CONFIG_BIT_4
10	B13	CONFIG_BIT_5
11	B14	CONFIG_BIT_6
12	B15	CONFIG_BIT_7
13	dummyBit	N/A
14	dummyBit	N/A
15	dummyBit	N/A

Load Testing of the SN Board

This section covers a basic load test of the SN board. This test will verify the proper operation of the relay switching circuits, DC-DC converter, and the current and voltage monitoring circuits. The proper operation of the PIC Test Code is also exercised.

Note that the exact load voltage at each load will depend on the trim up or trim down setting. This is a good time to adjust the trim to give the expected voltage at the expected load. There are load curves in the appendices for the output ports.

1) Setup the SN board as in the Establishing Serial Communication section of this document.
2) Attach a load cable to a dummy load (can be electronic or manual)
3) Attach the other end of the load cable to one of the output connectors (J2-J5)
    a. For this example, use J2 for channel #1
4) Set the dummy load to a resistance which will cause a low current to flow.
    a. i.e. for a 12V Converter, 120 ohms causes 100mA of load current
5) In the Hyperterm window, type SET RELAYON 1
    a. You should hear the relay click on and the expected load current displayed
    b. For this example, VLoad = 12.7V and ILoad = 0.105A
6) Type GET RELAY 1
    a. The SN board should reply Relay 1 is ON
7) With a voltmeter, measure the scaled load voltage at INSTR_VOLT (R7/C14)
    a. For this example (12V), VMeas = 0.771V
8) With a voltmeter, measure the scaled current for channel #1 at INSTR_1_CURR (R18/C25)
    a. For this example (100mA), IMeas = 0.135V (measuring a voltage which representative of the load current)
9) Change the load to a higher current
    a. Changing RLoad to 6 ohms causes 2A of load current
    b. For this example, VLoad = 11.98V and ILoad = 1.995A
10) With a voltmeter, measure the scaled current for channel #1 at INSTR_1_CURR (R18/C25)
    a. For this example (2A), IMeas = 2.48V (measuring a voltage which representative of the load current)
11) In the Hyperterm window, type SET RELAYOFF 1
    a. You should hear the relay click offf
    b. The displayed load current show drop to zero.

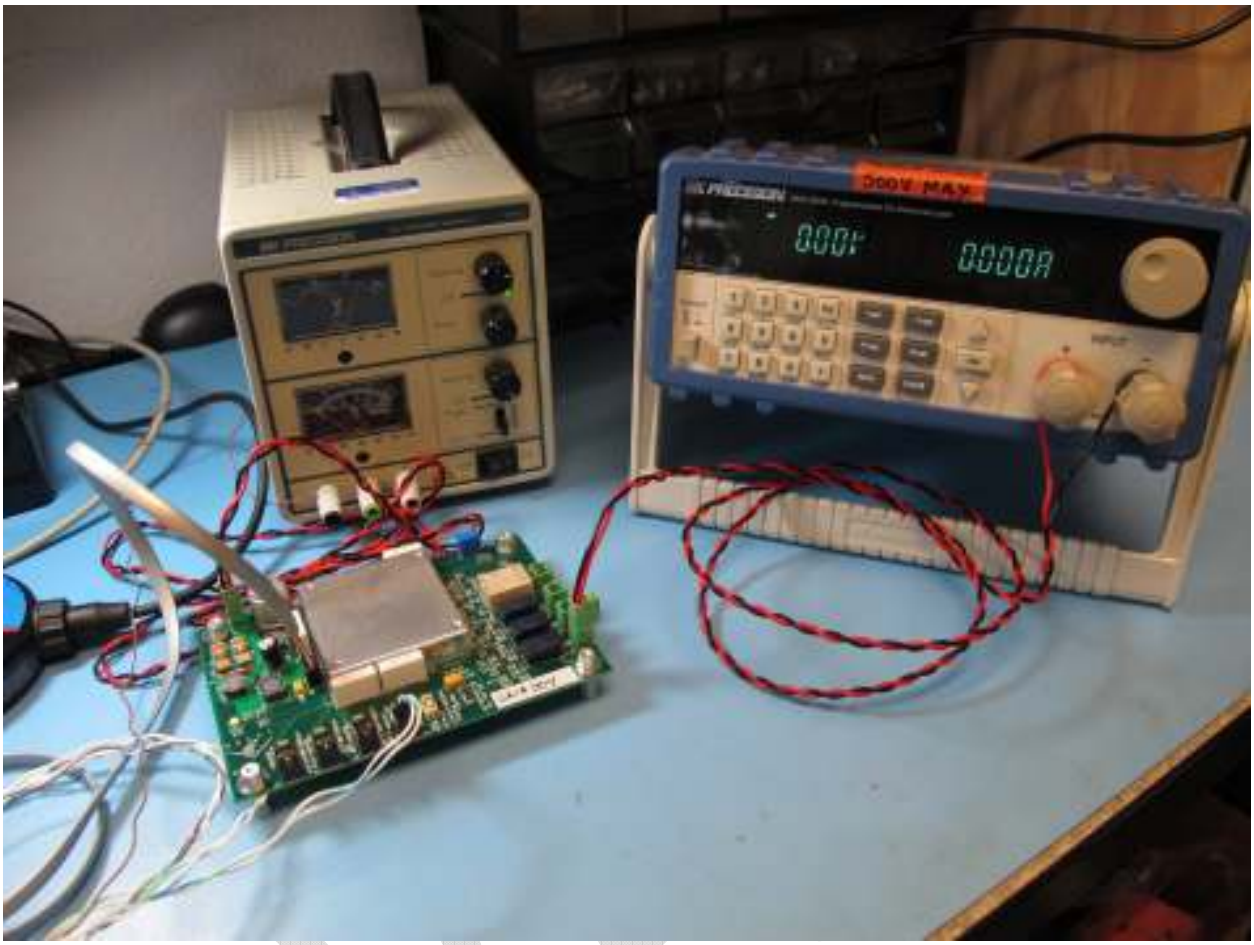


Figure 6: Load testing the SN board

PIC Code Documentation

Project Files

Header Files
actions.h
buffer.h
commands.h
dig_io.h
errors.h
misc.h
parser.h
rtcc.h
serial.h
spi_io.h
sys_def.h
sys_timer.h

Source Files
actions.c
buffer.c
commands.c
dig_io.c
main.c
misc.c
parser.c
rtcc.c
serial.c
spi_io.c
sys_timer.c

PIC Pins Configuration

Digital Inputs

```
#define TOTAL_IN_BITS  16

#define IN_BIT_0 PORTFbits.RF0 // INSTR_1_STATE
#define IN_BIT_1 PORTFbits.RF1 // INSTR_2_STATE
#define IN_BIT_2 PORTFbits.RF2 // INSTR_3_STATE
#define IN_BIT_3 PORTFbits.RF3 // INSTR_4_STATE
#define IN_BIT_4 PORTGbits.RG3 // ADC_BUSY
#define IN_BIT_5 PORTBbits.RB8 // CONFIG_BIT_0
#define IN_BIT_6 PORTBbits.RB9 // CONFIG_BIT_1
#define IN_BIT_7 PORTBbits.RB10 // CONFIG_BIT_2

#define IN_BIT_8 PORTBbits.RB11 // CONFIG_BIT_3
#define IN_BIT_9 PORTBbits.RB12 // CONFIG_BIT_4
#define IN_BIT_10 PORTBbits.RB13 // CONFIG_BIT_5
#define IN_BIT_11 PORTBbits.RB14 // CONFIG_BIT_6
#define IN_BIT_12 PORTBbits.RB15 // CONFIG_BIT_7
#define IN_BIT_13 dummyBit
#define IN_BIT_14 dummyBit
#define IN_BIT_15 dummyBit
```

Digital Outputs

```
#define TOTAL_OUT_BITS  32

#define OUT_BIT_0 LATDbits.LATD6 // UART_1_*SHDN
#define OUT_BIT_1 LATEbits.LATE0 // INSTR_1_ON
#define OUT_BIT_2 LATEbits.LATE2 // INSTR_2_ON
#define OUT_BIT_3 LATEbits.LATE4 // INSTR_3_ON
#define OUT_BIT_4 LATEbits.LATE6 // INSTR_4_OFF
#define OUT_BIT_5 LATEbits.LATE1 // INSTR_1_OFF
#define OUT_BIT_6 LATEbits.LATE3 // INSTR_2_OFF
#define OUT_BIT_7 LATEbits.LATE5 // INSTR_3_OFF

#define OUT_BIT_8 LATEbits.LATE7 // INSTR_4_ON
#define OUT_BIT_9 LATDbits.LATD7 // UART_2_*SHDN
#define OUT_BIT_10 LATDbits.LATD8 // UART_3_*SHDN
#define OUT_BIT_11 LATFbits.LATF5 // "Test LED"
#define OUT_BIT_12 LATFbits.LATF6 // DC-DC_CNTL
#define OUT_BIT_13 LATGbits.LATG2 // ADC_*SHDN
#define OUT_BIT_14 LATGbits.LATG6 // SPI_*CS
#define OUT_BIT_15 LATBbits.LATB0 // spare

#define OUT_BIT_16 LATBbits.LATB1 // spare
#define OUT_BIT_17 LATBbits.LATB2 // spare
#define OUT_BIT_18 LATBbits.LATB3 // spare
#define OUT_BIT_19 LATBbits.LATB4 // spare
#define OUT_BIT_20 LATBbits.LATB5 // spare
#define OUT_BIT_21 LATCbits.LATC13 // spare
#define OUT_BIT_22 LATCbits.LATC14 // spare
#define OUT_BIT_23 LATDbits.LATD11 // spare

#define OUT_BIT_24 LATFbits.LATF4 // spare
#define OUT_BIT_25 dummyBit
#define OUT_BIT_26 dummyBit
#define OUT_BIT_27 dummyBit
#define OUT_BIT_28 dummyBit
#define OUT_BIT_29 dummyBit
#define OUT_BIT_30 dummyBit
#define OUT_BIT_31 dummyBit
```

UART Pins

```
/* Map UARTs to pins */

/* Map UART #1 pins */
RPINR18bits.U1RXR = 24;  /* Make Pin RP24 U1RX */
RPOR5bits.RP11R = 3;     /* Make Pin RP11 U1TX */

/* Map UART #2 pins */
RPINR19bits.U2RXR = 22; // Make Pin RP22 U2RX
RPOR11bits.RP23R = 5;   // Make Pin RP23 U2TX

/* Map UART #3 pins */
RPINR17bits.U3RXR = 20; // Make Pin RP20 U3RX
RPOR12bits.RP25R = 28;   // Make Pin RP25 U3TX

 /* Map UART #4 pins */
RPINR27bits.U4RXR = 3; // Make Pin RP3 U4RX
RPOR2bits.RP4R = 30;   // Make Pin RP4 U4TX
```

SPI Pins

```
/* SPI Inputs (to PIC) */
RPINR20bits.SDI1R = 19; // SDI1 (MISO) RP19


/* SPI Outputs (from PIC) */
RPOR13bits.RP27R = 8; // SDO1 (MOSI) n=26 RP27
RPOR13bits.RP26R = 5; // SCK1OUT (SCK) n=26 RP26
RPOR10bits.RP21R = 4; // SS1OUT (*CS) n=20 RP21
```

Appendices

Power IN Cable

The power in cable delivers the source 48Vdc to J1 on the SN Board. It is recommended to use 18AWG wire. The mating inline plug for J1 is a Phoenix connector, PN# 1881325.

Appendix 1: Phoenix Connector 1881325

Appendix 2: Sample Power IN Cable

Power OUT Cable

The Power out cables delivers the output voltage to the four instruments. It is recommended to use 18AWG wire. The mating inline plug for channels #1-3 (J2, J3, J4) is a Phoenix connector, PN# 1881325. The mating inline plug for channel #4 (J5) is a Phoenix connector, PN# 1803578.

Appendix 3: Phoenix Connector 1803578

Appendix 4: Sample Power OUT Cable

Serial Cable

The serial cable connects the UARTs on the SN board to attached instruments, the Gateway Node, or a test computer. It is recommended to use 24AWG wire. The mating inline plug for the four serial connections is a Molex connector, PN# 50-57-9402. The other end of the cable is a simple DB9 connector. The pinout for the cable is:

DB9 Side	Molex Side
2	1
3	2
5	[to Power Supply Ground]

Appendix 5: Molex Connector 50-57-9402

Appendix 6: Sample Serial Cable

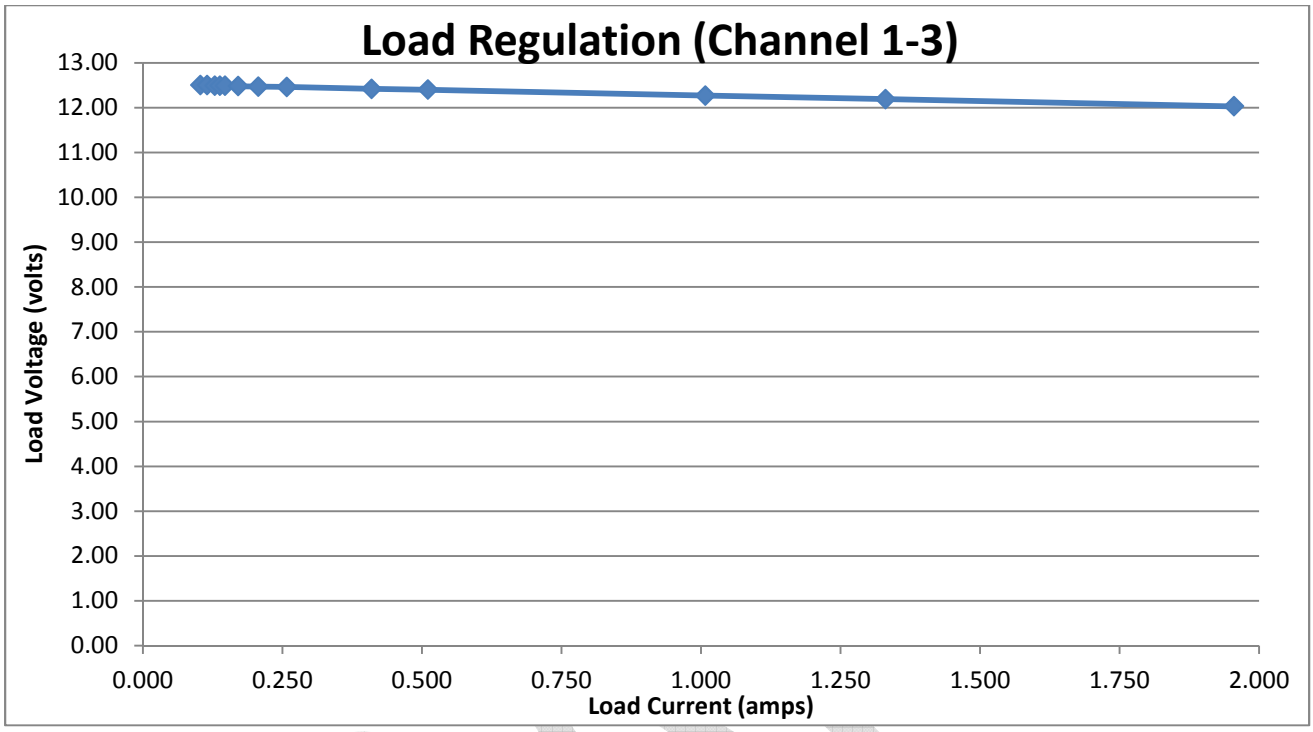
PIC Programming Cable

The PIC Programming cable connects the MPLAB ICD3 programmer to J6 of the SN Board. The ICD3 kit comes with a cable; one end needs to be modified to mate with the SN board. The mating connector is a Molex Connector, PN# 51110-0650.

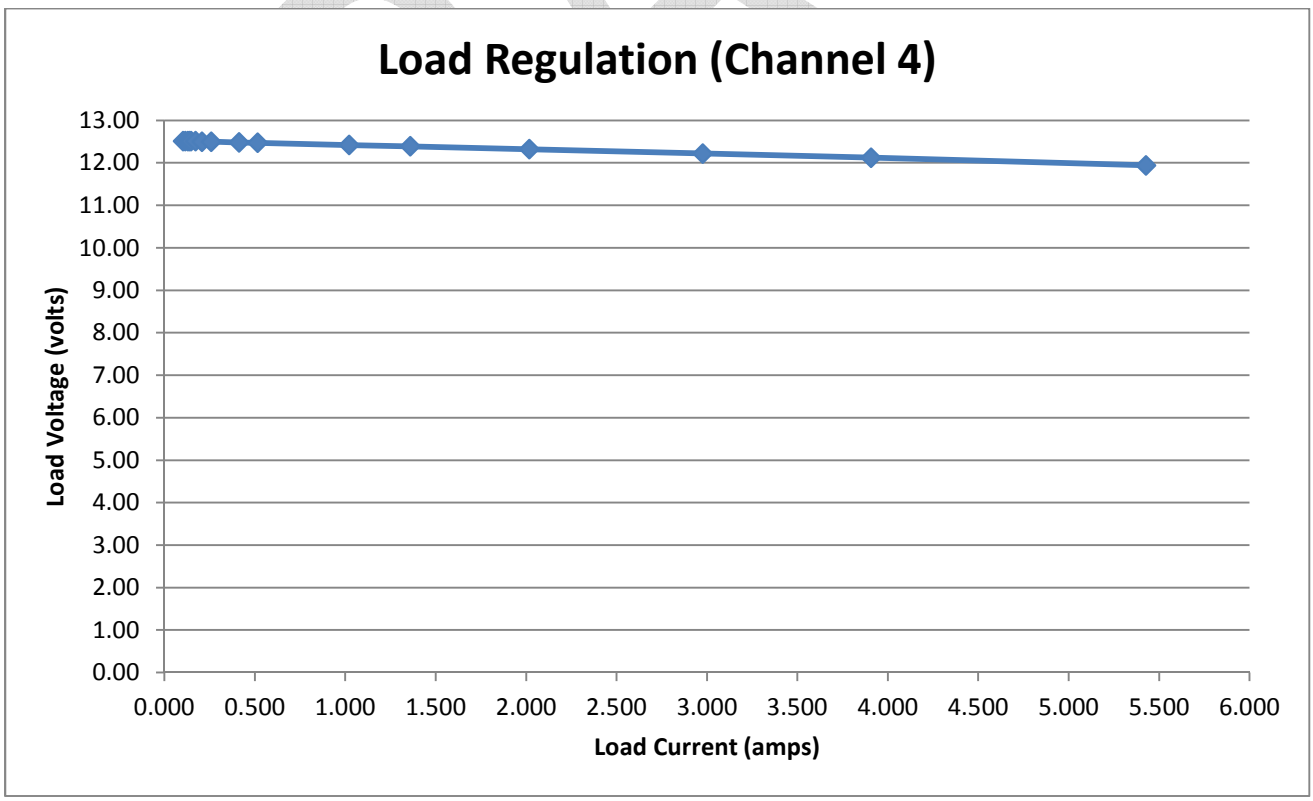
Appendix 7: Molex Connector 51110-0650

Appendix 8: Modified Programming Cable

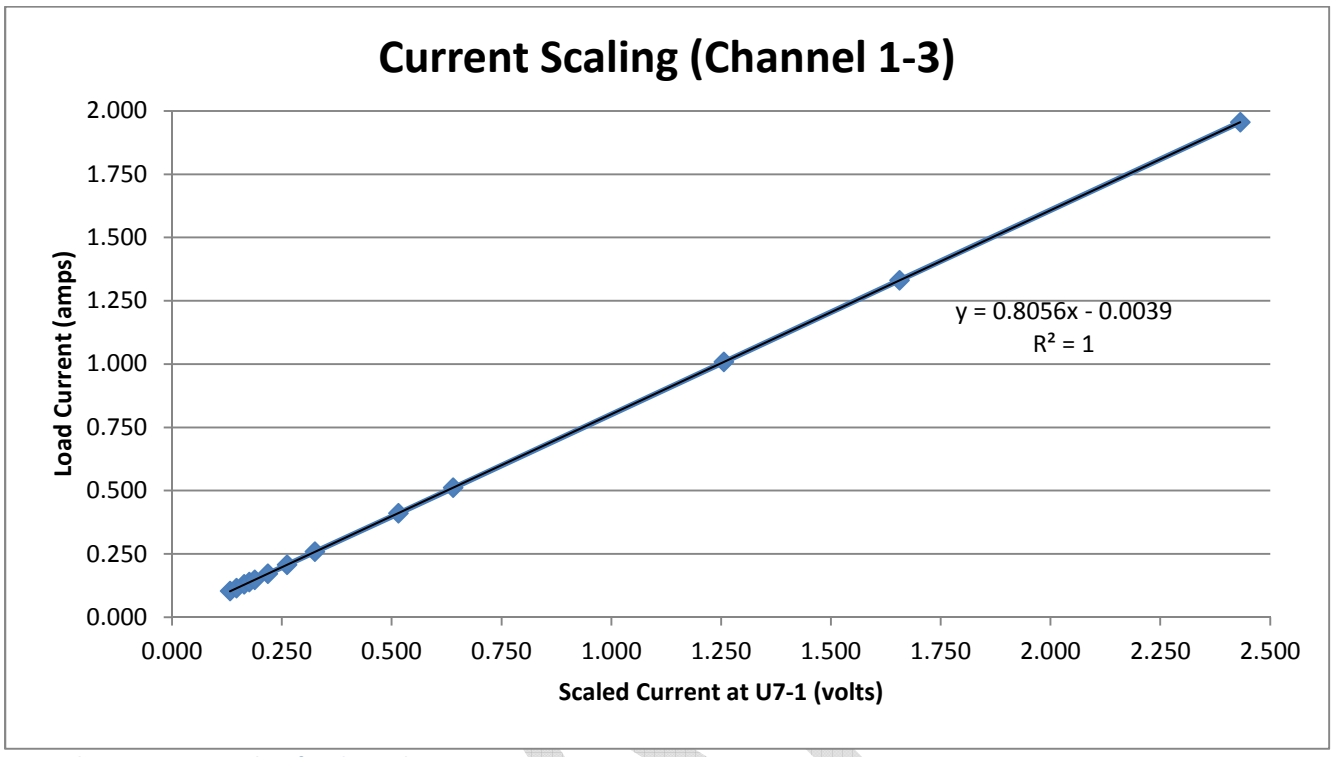
Load Curves



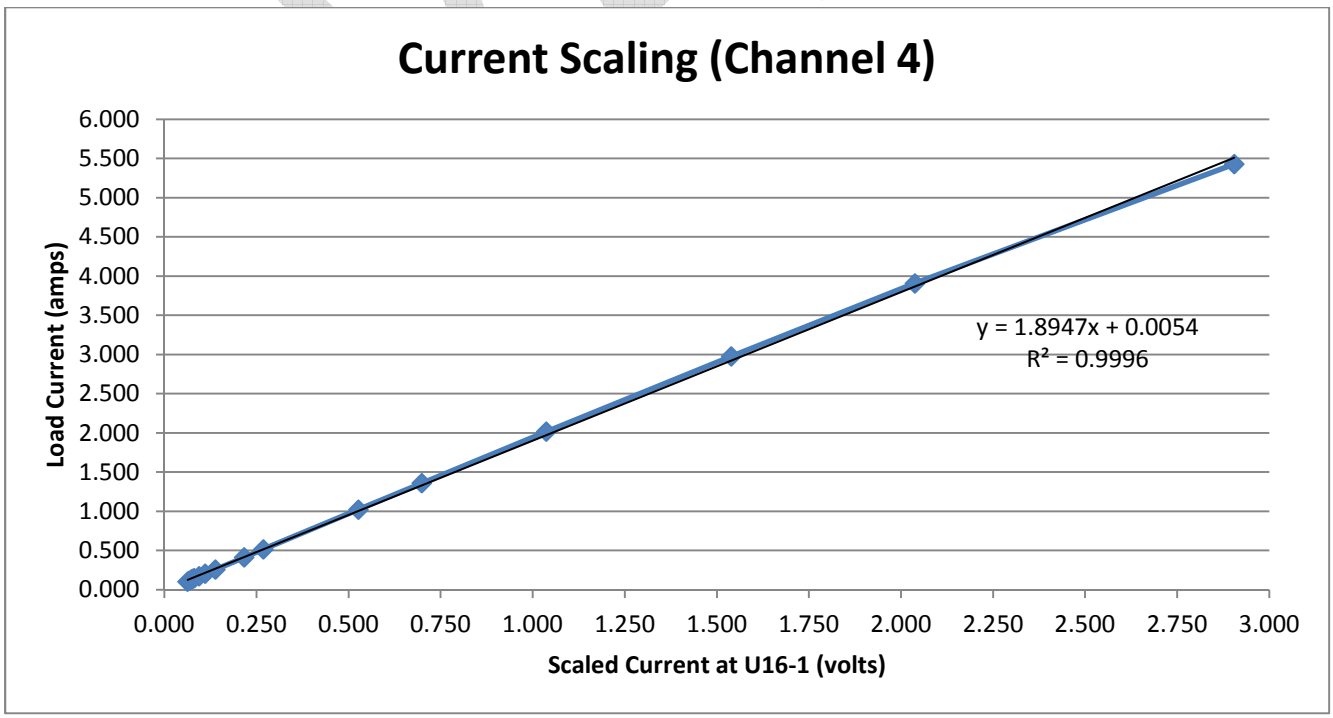
Appendix 9: Load Regulation for channels 1-3



Appendix 10: Load Regulation for channel 4



Appendix 11: Current scaling for channels 1-3



Appendix 12: Current scaling for channel 4