Quick Start Guide for EV-MCS-ISOINVEP-Z Isolated Inverter (Enhanced Protection) Platform

Rev. 0.2

Created:	8/7/2015 11:34 AM
Author:	Dara O’Sullivan
Last Modified:	8/17/2015 3:36 PM
Modified by:	Dara O’Sullivan

1 Contents

2 Revision History

Version	Modified By	Date	Comments
0.1	Dara O’Sullivan	8/7/2015	
0.2	Dara O’Sullivan	8/17/2015	Updated board photo

3 Overview

The isolated inverter (enhanced protection) platform is a power board that runs from a dc input in the range (24-800Vdc) and provides a three-phase variable frequency, variable voltage, and variable dead-time PWM output to a motor or load. The inverter is provided as an open loop platform with no feedback loops – however these can be added by the application developer. Isolated current and voltage feedback are provided to the control side of the board via sigma-delta modulators (AD7403), and these can be used for development of sensor-less control algorithms. The inverter is made up of a three-phase six-IGBT bridge, with the IGBTs rated at 1200V and driven by six isolated gate drivers (ADuM4135). A DC – rather than AC- input is provided to allow flexibility on the dc bus voltage level (rather than it being limited to the ac line peak).

This document will give a high level introduction to the EV-MCS-ISOINVEP isolated inverter platform and will provide a step-by-step approach that will get a motor/load up running with the platform and an ADSP-CM408 EZkit, although the platform can be controlled from any variable frequency PWM source. Instructions are provided on hardware setup, executable download, IAR project building and graphical user interface (GUI) operation.

3.1 System requirements

Before you start working on the inverter platform, please make sure you have the hardware and software listed below.

Required Hardware

- ADSP-CM408F EZ-KIT rev 0.2 (This is ordered separately from the EV-MCS-ISOINVEP-Z)
- EV-MCS-ISOINVEP-Z power board, including the provided control power supply and adapter board (for connection to EZkit)
- Power board supply (not provided). The board is optimized to work from a DC supply in the range 24Vdc-800Vdc. The power rating of the supply should be sufficient to match the load being driven. The power board is rated up to 2kVA. There is a series connected diode in the input, so if needed, the power board can be driven from an ac supply. However output power is limited in this case.
- If a Segger J-Link Lite debugger is not available (although this is automatically provided with an ADSP-CM408 EZ-KIT) a USB to serial converter (not provided) is required to download executable demo code to the EZ-KIT. A USB to serial converter is also required if the user wishes to visualize data variables using the Graphical User Interface (GUI).

Optional Hardware

- A three-phase load is strictly speaking, optional, however in order to evaluate the inverter components it is more useful to have current flowing to an output. This load will typically be a three-phase motor, such as an induction motor or permanent magnet synchronous motor, but a three-phase, three-wire resistive load can also be connected. This is not supplied with the board.

- Segger J-Link Lite debugger (This comes with the ADSP-CM408F Ezkit and can be used for executable download, and code development, but is not necessary for basic setup)
- If a full AC front-end is required, the inverter platform can be utilized in conjunction with the ADP1047 evaluation board up to 300W, or the ADP1048 evaluation board up to 600W (see http://www.analog.com/en/power-management/digital-power-management/adp1047/products/EVAL-ADP1047/eb.html and http://www.analog.com/en/power-management/digital-power-management/adp1048/products/EVAL-ADP1048/eb.html)

Required Software

- ADSP-CM40x SW Enablement Package version 2.0.0 (available here : https://ez.analog.com/docs/DOC-12607)
- Inverter demo program executable and linker map file (available here: https://ez.analog.com/docs/DOC-12608) for ADSP-CM408 Silicon Rev 0.0 only.

Optional Software

- ADIMonitor Graphical User Interface (also available here: https://ez.analog.com/docs/DOC-12608)
- IAR Embedded W orkbench (for code development, version 7.4 or higher)
- Segger J-Link Lite driver software (v4.97j or higher)
- IAR Embedded Workbench inverter demo program C project (also available here: https://ez.analog.com/docs/DOC-12608) for ADSP-CM408 Silicon Rev0.0 or Rev H.

4 Hardware Setup

This section will describe how to setup the hardware for a demo motor control application. This only has to be performed once, when bringing up a new platform. The application is designed to spin a three-phase motor in open loop Volts/Hertz control. The control settings are matched to a typical 2-pole induction motor; however other motors will also spin, but the speed reference may not match the actual speed of the motor if the pole count is different.

Connect the EZ-kit to the adapter board and the adapter board to the power board as shown in Figure 1.

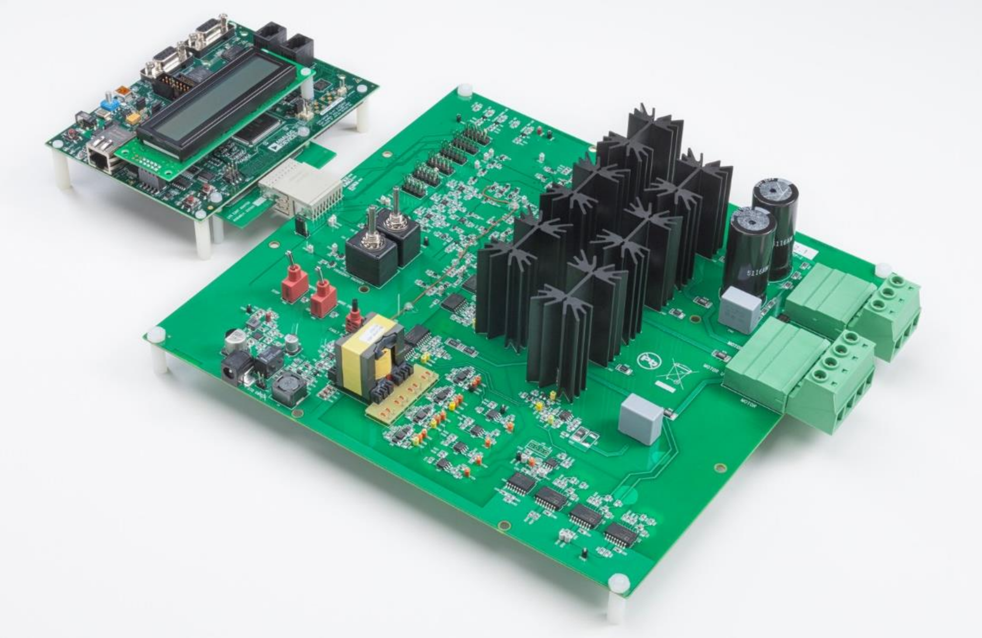


Figure 1 Inverter Board connected to CM408F EZ-kit.

Ensure that the Motor/Load and Power Connectors are wired as shown in *Figure 2*, *Table 1* and *Table 2*. As noted previously, the three-phase load can be a resistor bank or a three-phase motor. The power board is designed to operate from a DC bus supply but can also be run from AC with limited output power.

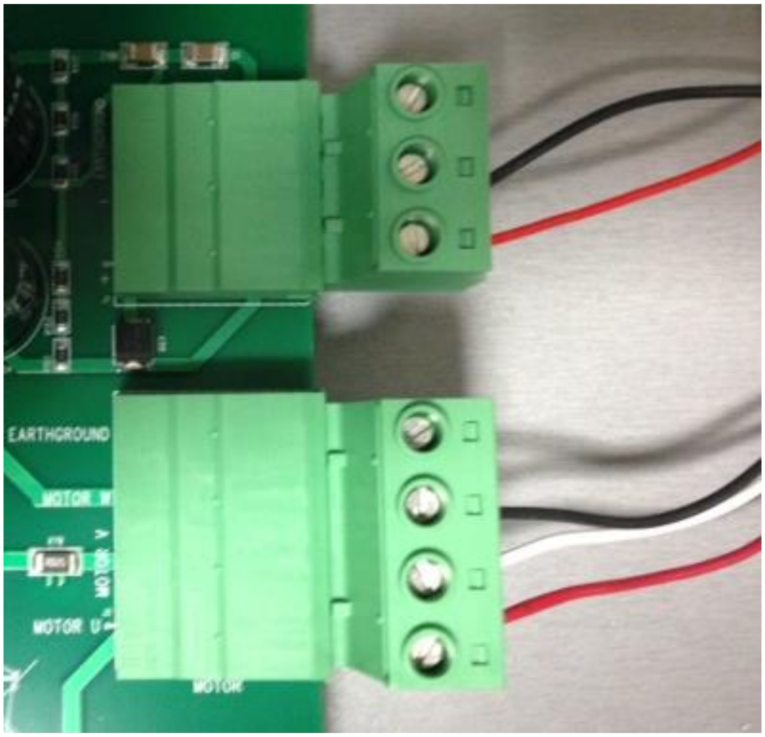


Figure 2 Wiring of Motor- and Power Connector.

Pin	***Color in Figure 3***	***Signal***
1	*Red*	*Motor phase U*
2	*White*	*Motor phase V*
3	*Black*	*Motor phase W*
4		*Shield/Earth*

Table 1 Motor Connector (J2)

Pin	***Color***	***Signal***
1	*Red*	*DC bus +ve*
2	*Black*	*DC bus -ve*
3		*Shield/Earth*

Table 2 Power Connector (J1)

Plug in the power adapter to connector J3 to provide the control power supply to the boards. Do not plug in the ADSP-CM408 EZkit. It derives its power supply from the inverter board. However, JP2 jumper must be set on the EZkit to enable “5V_INPUT” rather than “WALL_POWER”; see *Figure 3*.

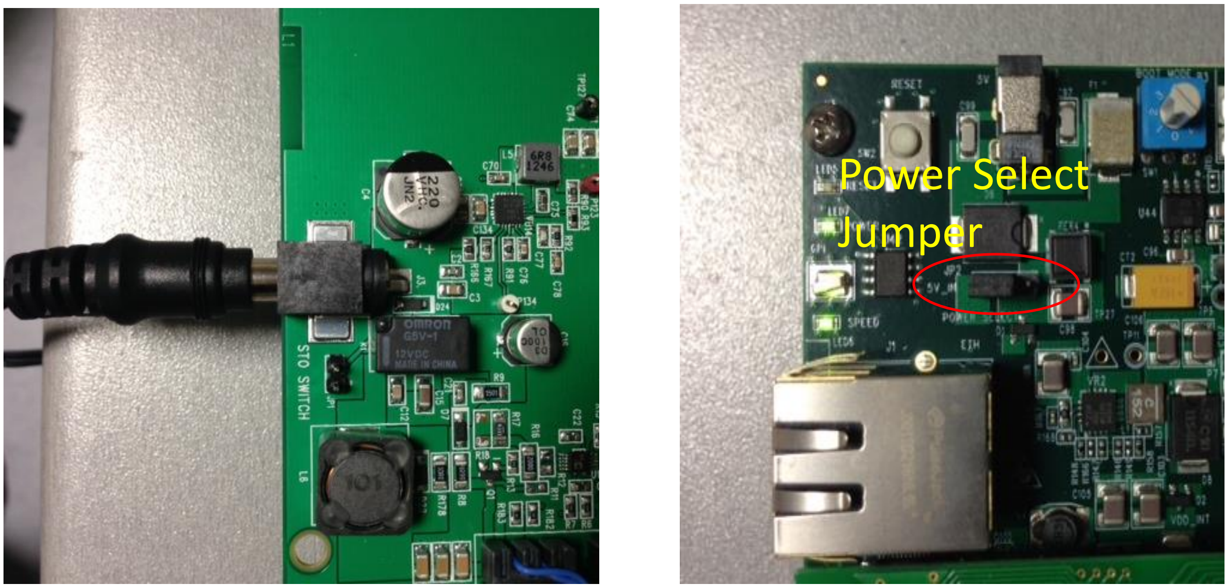


Figure 3 Control Power Supply Input and Power Select Jumper

There are four sigma-delta modulators which provide isolated current and voltage measurement or feedback. The precision and group delays of these signals are controlled by the setup of the digital filter parameters used in the software. The default configuration on the board is for the following signals to be measured:

- U phase motor current
- V phase motor current
- W phase motor current
- DC bus voltage

There is additional optional feedback of the UV and VW line to line motor voltages. These can be exchanged for the DC bus voltage measurement and W phase current measurement by removing R69, R66, R240 and R241 zero ohm resistors, and populating R176, R177, R67 and R81 with zero ohm resistors.

5 Software Setup

The software setup steps are as follows:

1. Download and install the ADSP-CM40x SW Enablement Package from the link shown previously in “SW Requirements”. This includes the serial boot-loader which is needed for download of the executable to the processor.

2. Download and install the GUI from the Engineer Zone link provided in "SW Requirements". This requires the .NET framework to be on the PC and it will prompt the user to download this if it is not detected.
3. For ADSP-CM408 silicon version 0.0 (see processor face for silicon versions), method (a) or method (b) can be used. For silicon version H, currently **only method (b) can be used**.

   (a) Download the inverter demo executable program from the Engineer Zone link provided in "SW Requirements" and program this to the processor board (ADSP-CM408 EZkit)

   OR

   (b) Compile and download the C code demo project from the Engineer Zone link provided in "Optional Software".

The first two steps are fairly self-explanatory. Step 3(a) – programming of the executable to the processor – can be carried out in two alternate ways. These are detailed in Section 5.1 and Section 5.2. Step 3(b) – compile and download from IAR Embedded Workbench – is outlined in Section 5.3.

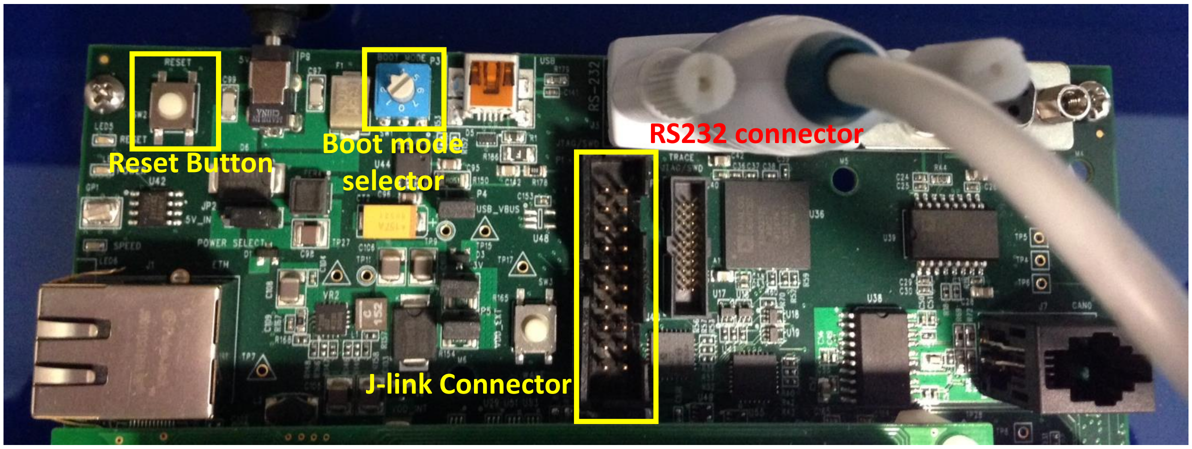
5.1 *Programming with Serial Downloader*

The serial downloader ("wsd.exe") is provided as part of the ADSP-CM408 SW Enablement Package and once this has been installed, assuming default installation directory structures, the downloader is found in C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\tool\UARTFlashProgrammer. This method uses the **MC_Demo.hex** executable provided at the Engineer Zone link.

Steps:

1. Connect the EZ-kit to the PC using the USB-UART cable: UART (RS-232) connector on the EZkit, USB on the PC

2. Select Boot Mode 3 (UART Boot) on the EZkit using the selector switch P3 (adjacent to the power connector)



3. Power up the EZkit

4. Open the Windows Serial Downloader (wsd.exe) and select the appropriate options:

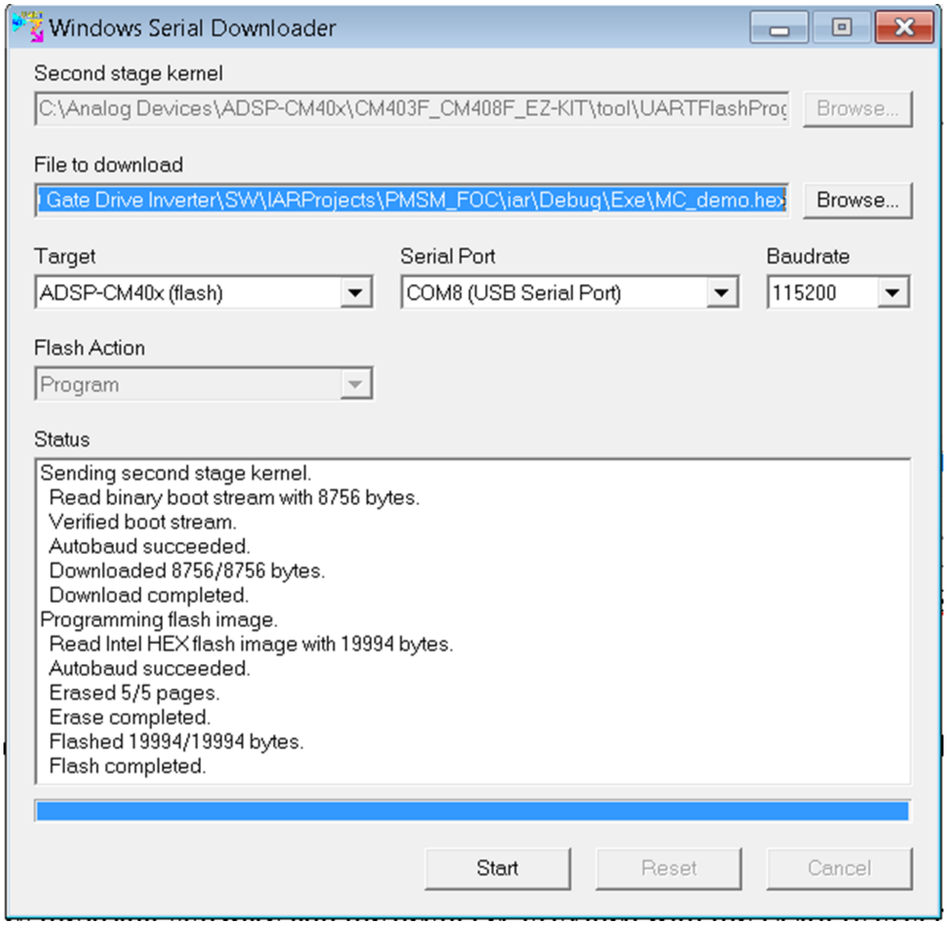
Target: ADSP-CM40x (flash)

Serial Port: Select the COM port to which the USB-serial cable is connected (if in doubt, check under ‘Device Manager->Ports’

Baudrate: 115200

5. Click "Browse" under "File to download" and navigate to the provided executable “MC_demo.hex”

6. Select "Start"



Second stage kernel: C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\tool\UARTFlashProg

File to download: Gate Drive Inverter\SW\IARProjects\PMSM_FOC\iar\Debug\Exe\MC_demo.hex

Target: ADSP-CM40x (flash)

Serial Port: COM8 (USB Serial Port)

Baudrate: 115200

Flash Action: Program

Status

```
Sending second stage kernel.
 Read binary boot stream with 8756 bytes.
 Verified boot stream.
 Autobaud succeeded.
 Downloaded 8756/8756 bytes.
 Download completed.
Programming flash image.
 Read Intel HEX flash image with 19994 bytes.
 Autobaud succeeded.
 Erased 5/5 pages.
 Erase completed.
 Flashed 19994/19994 bytes.
 Flash completed.
```

7. Once the operation completes, select Boot Mode 1 (SPI Flash Boot) on the EZkit, using the selector switch P3 (adjacent to the power connector)

8. Reset the EZkit (SW2 adjacent to power connector)

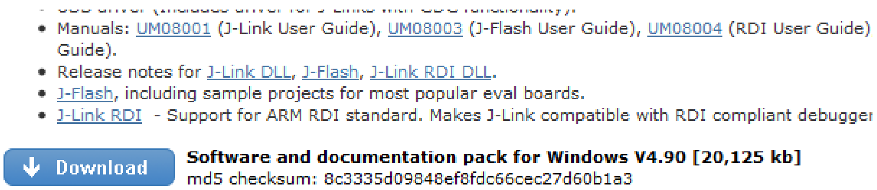
At this point the application should now be executing or loaded from flash, and the GUI can be connected to run the motor.

5.2 *Programming with Segger J-Link*

This method uses the JLink software and the debugger provided with the EZkit to program the processor flash memory.

1. Download and install Segger JLink Software (http://www.segger.com/jlink-software.html)

2. Click the blue download button under Windows. The version number may be different from the one shown below.



- Manuals: UM08001 (J-Link User Guide), UM08003 (J-Flash User Guide), UM08004 (RDI User Guide) Guide).
- Release notes for J-Link DLL, J-Flash, J-Link RDI DLL.
- J-Flash, including sample projects for most popular eval boards.
- J-Link RDI - Support for ARM RDI standard. Makes J-Link compatible with RDI compliant debugger

Software and documentation pack for Windows V4.90 [20,125 kb]
md5 checksum: 8c3335d09848ef8fdc66cec27d60b1a3

3. Enter your Segger JLink Lite serial number in the next box. This number is printed on the microcontroller on the debugger board.

4. Click the confirmation box on the next screen, download the software and install (default location is C:\Program Files(x86)\Segger)

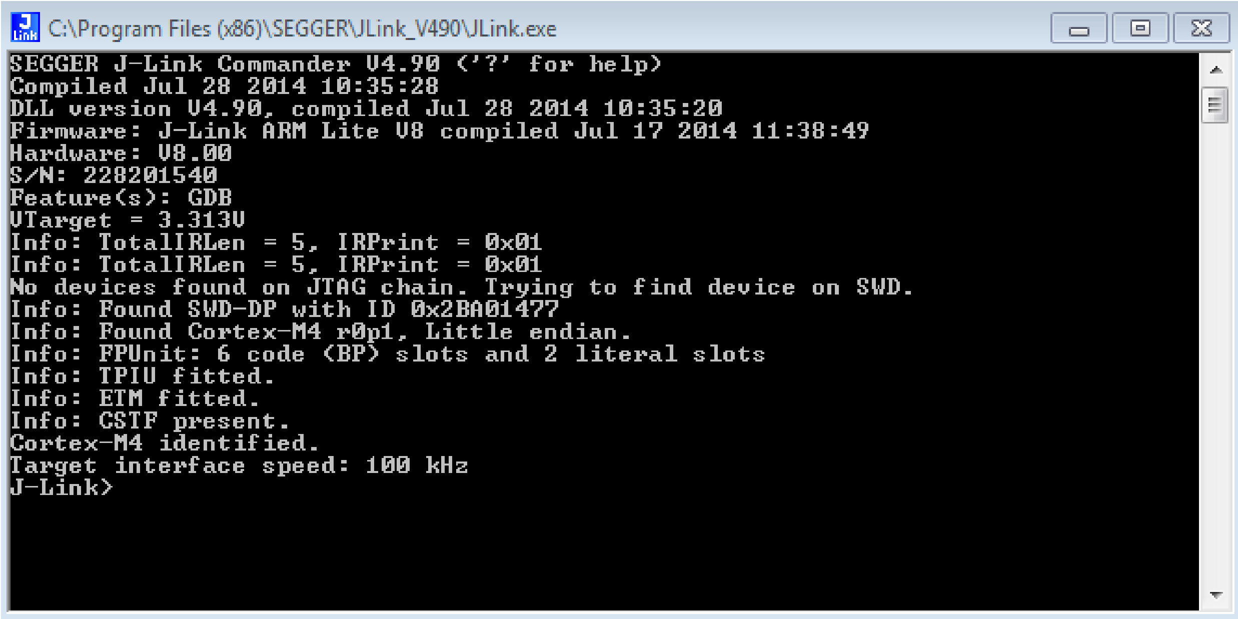
5. Connect the JLink to the PC USB port and the other end to the 20 pin connector on the CM40x board.

6. The PC will start automatically start installing drivers for your Jlink . Let it complete and turn on power to the EZkit.

7. Turn on power to the CM408x board.

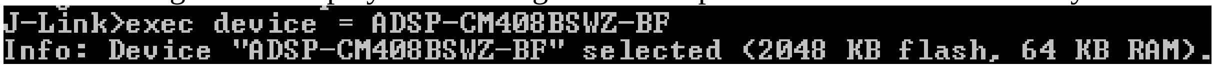
8. Go to the location where Segger was installed and click on JLink .exe. In this case it is in: C:\Program Files (x86)\SEGGER\JLink_V490 (The version number and folder name may be different to the one in this example.)

9. The following should be displayed in the Jlink window. It should identify that a Cortex-M4 is connected to the Jlink .



```
SEGGER J-Link Commander V4.90 ('?' for help)
Compiled Jul 28 2014 10:35:28
DLL version V4.90, compiled Jul 28 2014 10:35:20
Firmware: J-Link ARM Lite V8 compiled Jul 17 2014 11:38:49
Hardware: V8.00
S/N: 228201540
Feature(s): GDB
VTarget = 3.313V
Info: TotalIRLen = 5, IRPrint = 0x01
Info: TotalIRLen = 5, IRPrint = 0x01
No devices found on JTAG chain. Trying to find device on SWD.
Info: Found SWD-DP with ID 0x2BA01477
Info: Found Cortex-M4 r0p1, Little endian.
Info: FPUnit: 6 code (BP) slots and 2 literal slots
Info: TPIU fitted.
Info: ETM fitted.
Info: CSTF present.
Cortex-M4 identified.
Target interface speed: 100 kHz
J-Link>
```

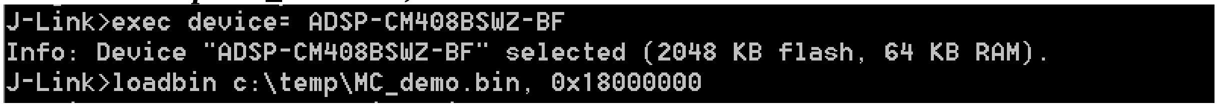
1. Enter the following at the prompt:
   exec device = ADSP-CM408BSWZ-BF
   The following will be displayed indicating that the chip has been identified correctly:



```
J-Link>exec device = ADSP-CM408BSWZ-BF
Info: Device "ADSP-CM408BSWZ-BF" selected (2048 KB flash, 64 KB RAM).
```

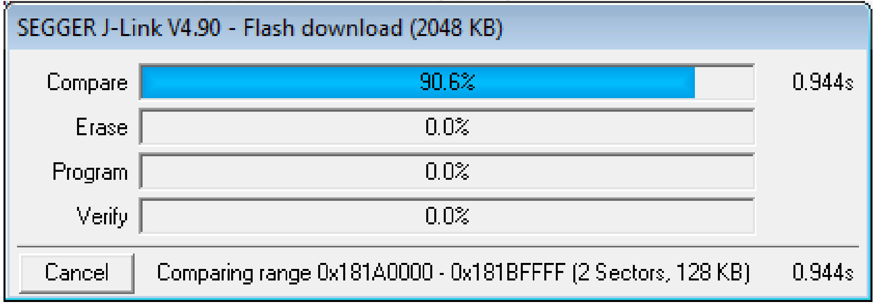
2. Copy the binary file you need to load into the CM40x in a known location. For this example, we will be placing the file **MC_demo.bin** in C:\temp

3. In the JLink window, type the following to load the binary code at address 0x1800 0000, which is the start of flash.
   loadbin c:\temp\MC_demo.bin, 0x18000000



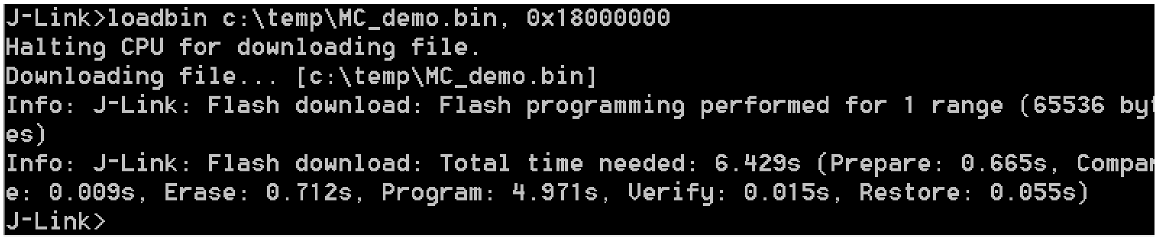
```
J-Link>exec device= ADSP-CM408BSWZ-BF
Info: Device "ADSP-CM408BSWZ-BF" selected (2048 KB flash, 64 KB RAM).
J-Link>loadbin c:\temp\MC_demo.bin, 0x18000000
```

A new window pops up, indicating progress:



Compare 90.6% 0.944s
Erase 0.0%
Program 0.0%
Verify 0.0%
Comparing range 0x181A0000 - 0x181BFFFF (2 Sectors, 128 KB) 0.944s

The JLink window will also show information about the flash procedure.



```
J-Link>loadbin c:\temp\MC_demo.bin, 0x18000000
Halting CPU for downloading file.
Downloading file... [c:\temp\MC_demo.bin]
Info: J-Link: Flash download: Flash programming performed for 1 range (65536 byt
es)
Info: J-Link: Flash download: Total time needed: 6.429s (Prepare: 0.665s, Compar
e: 0.009s, Erase: 0.712s, Program: 4.971s, Verify: 0.015s, Restore: 0.055s)
J-Link>
```

4. Reset the EZkit (SW2 adjacent to power connector)

At this point the application should now be executing or loaded from flash.

5.3 *Compile and Download from IAR Embedded Workbench*

First, get the C code demo project from the relevant Engineer Zone link and copy it to a local disk drive. The path of the MC project will be referenced as $MC_DIR$ in the following. For example, MC_DIR could be “C:\CM408Demo\”

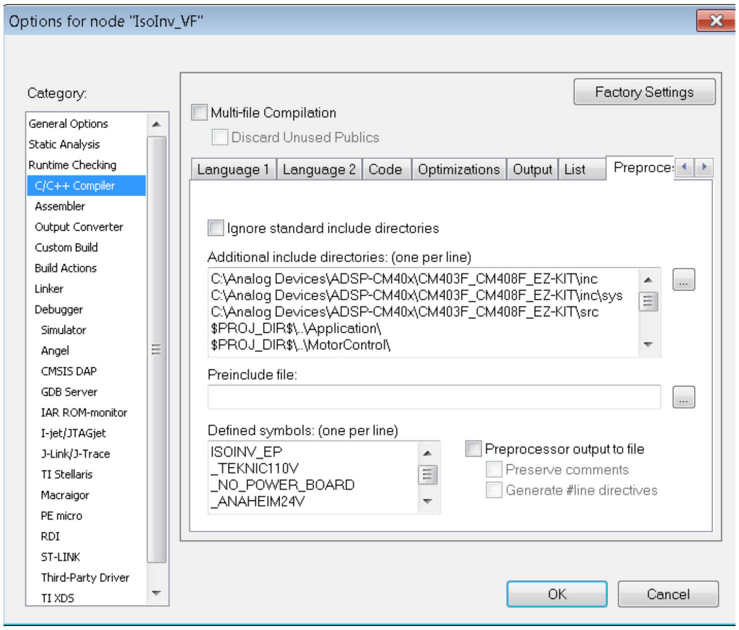
Before proceeding, ensure that the ADSP-CM40x SW Enablement package has been downloaded and installed as mentioned previously. Follow the instruction given during the installation.

To prepare the IAR Embedded Workbench environment, follow the steps below.

1. Open *$MC_DIR$\SW\IARProjects\IsoInv_VF\iar\IsoInv_VF.eww* in IAR EWB (Open Workspace)
2. Go to “*Project→Options→General Options→Target”* and ensure that a floating point unit (FPU) is selected. In the “*Library Configuration”* tab, ensure that “Use CMSIS” is selected.
3. Go to “*Project→Options→C/C++ Compiler→Preprocessor→Define Symbols”,* see Figure 4 and check that the correct additional include libraries and pre-compiler options are selected for the demo platform used. Possible configurations for the preprocessor defines are shown in Table 3. The additional include libraries are:

   C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\inc
   C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\inc\sys
   C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\src
   $PROJ_DIR$\..\Application
   $PROJ_DIR$\..\MotorControl
   $PROJ_DIR$\..\System
   $PROJ_DIR$\..\ADIMonitor
   $PROJ_DIR$\..\Measure
   $PROJ_DIR$\..\Communication
   $PROJ_DIR$\..\Protection
   $PROJ_DIR$\..\ScalarControlAlgorithm
   $PROJ_DIR$\..\ADIMonitorLibrary\inc\

4. Ensure that the linker library files are listed as shown in Figure 5 and that the Linker configuration file in the *Linker->Config* tab is set to $PROJ_DIR$\CM40z_384_2048_copy_to_SRAM.icf.



□ Multi-file Compilation

□ Discard Unused Publics

□ Ignore standard include directories

Additional include directories: (one per line)

```
C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\inc
C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\inc\sys
C:\Analog Devices\ADSP-CM40x\CM403F_CM408F_EZ-KIT\src
$PROJ_DIR$\..\Application\
$PROJ_DIR$\..\MotorControl\
```

Preinclude file:

Defined symbols: (one per line)

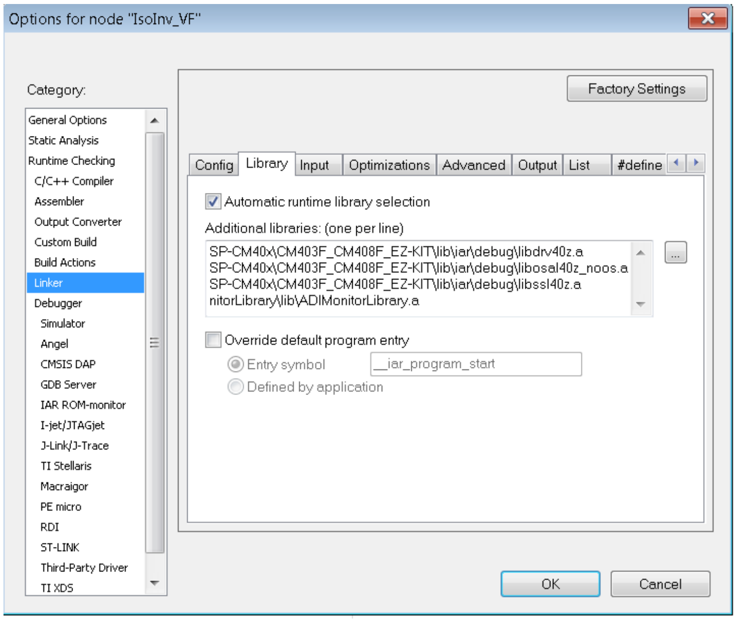
```
ISOINV_EP
_TEKNIC110V
_NO_POWER_BOARD
_ANAHEIM24V
```

□ Preprocessor output to file

□ Preserve comments

□ Generate #line directives

Figure 4: Include Directories and Preprocessor Defines



☑ Automatic runtime library selection

Additional libraries: (one per line)

```
SP-CM40x\CM403F_CM408F_EZ-KIT\lib\iar\debug\libdrv40z.a
SP-CM40x\CM403F_CM408F_EZ-KIT\lib\iar\debug\libosal40z_noos.a
SP-CM40x\CM403F_CM408F_EZ-KIT\lib\iar\debug\libssl40z.a
nitorLibrary\lib\ADIMonitorLibrary.a
```

□ Override default program entry

● Entry symbol __iar_program_start

○ Defined by application

Figure 5: Linker libraries

A number of different system configurations are supported by the demo project. The source code is always the same but configuration variables and pre-compiler defines vary. All supported configurations are listed in Table 3. Note that a preprocessor define with an _ e.g. _ISOINV_EP means that ISOINV_EP is NOT defined, and to define it the underscore can simply be deleted.

EWB Preprocessor Define	Effect
`ISOINV_EP`	Needs to be defined to use the EV-MCS-ISOINVEP-Z platform (ADuM4135 gate drivers). If using the EV-MCS-ISOINV platform (ADuM4223 gate drivers), remove this define (for Rev 2 or 3 ISOINV boards, add the define ISOINV_R23)

`WEGIM`
`ABBIM`
`ANAHEIM24V`
`TEKNIC110V`
	2 pole induction motor
	4 pole induction motor
	24V permanent magnet motor
	110V permanent magnet motor
	(Define one of these to match the inverter load. It is not critical that these match the load exactly – they mainly influence the V/f gain, frequency to speed ratio etc.)

`NO_POWER_BOARD`	Only this to test the project with no power board attached to the EZkit. It turns off all trips and power board feedback to enable the code to run with no inverter board attached.
`ARM_MATH_CM4`	Define to optimize use of CMSIS math function if they are being used

Table 3: Supported configurations.

The IAR project can now be built.

1. Open *$MC_DIR$\SW\IARProjects\PMSM_FOC\iar\pmsm_foc.eww* in IAR EWB (Open Workspace)
2. Click *Project→Rebuild All*
3. Build/link status can be obtained from the message window.

In its default configuration the Motor Control Demo rev 0.3.0 is set up to work with silicon rev. 0.0. However, with minor adjustments, silicon rev H is also supported. (Note: The silicon revision is indicated on the processor face). Silicon rev. H adds a Boot-ROM-based secure access protocol that imposes specific software requirements which must be met for successful booting. IAR-based application project files targeting silicon rev 0.0 should ***DEFINE*** the compiler preprocessor symbol "**ADI_NO_SECURITY_PROTOCOL**" in startup.c to circumvent security protocol code generation. Project files targeting silicon rev H should ***REMOVE*** the compiler preprocessor symbol "**ADI_NO_SECURITY_PROTOCOL**".
With rev H silicon, the code in startup.c should look like this:

```
// #define ADI_NO_SECURITY_PROTOCOL
```

For further information, please refer to"ADSP-CM40X ENABLEMENT SOFTWARE USER'S GUIDE".

With the linker file (.icf file) supplied with the demo project, CM40x boots from flash and then loads all code into SRAM and executes code from SRAM. In this mode it is only necessary to program memory when the program is changed.

To program the device, follow the steps below

1. Connect Segger J-Link debugger between a USB port on the PC and the Debug connector on the EZ-kit
2. Turn on the 5V power supply to the EZ-kit.
3. Go to IAR EWB and download program to target by clicking *Project→Download and Debug*
4. In IAR EWB, start the program by clicking *Debug→Go*

Once Flash has been programmed, Flash boot runs stand-alone and does not require IAR EWB+debugger.

6 Manual Mode

The board can also be run in manual mode with control of output frequency and dead-time being implemented from the dials on the board, and with no need of a PC connection to the EZkit. The system will run with default parameters, and the GUI can still be used for data visualization. However it should not be used for control in manual mode, as the control inputs may conflict with the manual inputs.

The steps to be followed for running the platform in manual mode are as follows:

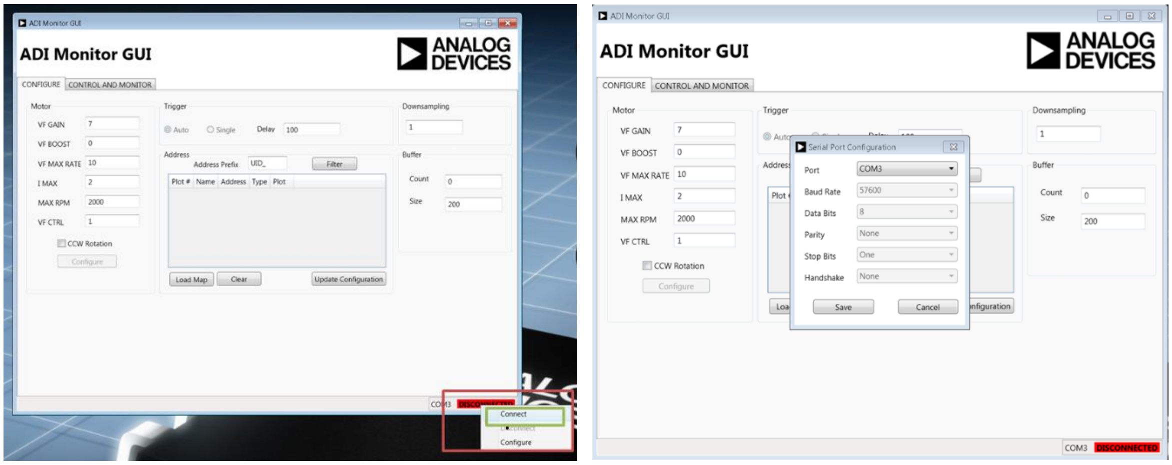
1. Plug in the 12V control power supply
2. Set the mode selector switch to *Manual*
3. Set the Start/Stop switch to *Stop*
4. Reset the EZkit
5. Ensure no fault LEDs are showing
6. Turn on the DC bus power
7. Set the Start/Stop switch to *Start*
8. Control motor speed/load voltage and frequency with the Frequency dial, and control PWM dead time with the Dead Time dial

A Safe Torque Off (STO) fault or shutdown can be emulated by shorting the jumper JP1. This will remove power to the gate drivers, and trigger a PWM fault in the processor. Removing the short and resetting the fault status by means of the momentary switch SW4 will cause the power board to restore power to the motor and ramp up the output PWM to its previous operating point (unless the Start/Stop button has been set to *Stop* in the meantime).

The LEDs can provide some level of diagnostics as to power supply busses and fault conditions. It should be noted that if the EZkit is reset, the frequency and dead-time settings revert to their default start-up values (zero PWM output, and 1μs dead-time), i.e. the dials are not absolute position related. The dead-time range allowed is 500ns to 20μs.

7 GUI Mode

Once the installation zip file is extracted, the GUI application is loaded by running the installer executable “ADIMonitorGUI.exe”. This GUI requires the .NET framework and this will be downloaded if it is not already on the PC. The first step is to set up the serial port (having connected the USB-serial converter to the EZkit RS232 port). This is accomplished by right-clicking the bottom right-hand corner of the GUI and selecting *Connect.* The program should automatically select the correct COM port, but if several options are available, right-click *Configure* and select the correct one. Once this is selected press, *Save* and then *Connect.*



VF GAIN 7
VF BOOST 0
VF MAX RATE 10
I MAX 2
MAX RPM 2000
VF CTRL 1
□ CCW Rotation
Delay 100
Address Prefix UID_
Downsampling 1
Count 0
Size 200

Port COM3
Baud Rate 57600
Data Bits 8
Parity None
Stop Bits One
Handshake None

Figure 6: Serial Port Setup

The next step is to configure the motor control settings. These are listed in *Table 4*, along with an explanation of their relevance. These settings are designed for open loop Volts/Hz control of a 2-pole motor. If a different number of poles is utilized, the system will still work, but the rpm to Hertz correlation will be incorrect.

Parameter	Open Loop V/f control
VF_GAIN	Volts per Hz representation
VF_BOOST	Low speed boost voltage representation
VF_MAX_RATE	RPM change per 0.1 ms sample
I_MAX	Not used
MAX_RPM	Max rpm
VF_CTRL	Set to 1 for open loop V/f
CCW rotation	Check for CCW rotation. Uncheck for CW rotation. When looking into shaft end of motor.
Speed reference	Speed reference

Table 4: Motor Control Settings

Once the configuration settings have been entered, press *Configure.* Typical settings for open loop control are depicted in *Figure 7.* Please note that the VF_GAIN and VF_BOOST settings do not directly refer to volts/Hz or volts quantities. Care should be taken when setting the VF_GAIN quantity. If it is set too high, the motor will draw large currents and an overcurrent trip may occur, especially if a permanent magnet synchronous motor is being used. Also if the VF_MAX_RATE is set too high, the applied motor voltage and frequency will ramp too quickly – especially if a large step change in speed reference occurs – potentially causing an overcurrent trip. If an overcurrent or PWM trip occurs, or an STO trip is created, press the GUI *Reset Trip* button (*Figure 8*) and then *Stop*. Reduce VF_GAIN or VF_MAX_RATE to limit the steady state and transient current amplitudes if necessary and re-start the motor. The fault LEDs should clear when the *Reset Trip* command is pressed.

Figure 7: Motor Control Configuration – Typical open loop settings.

.

8 Running the Motor

In order to control the motor, navigate to the *Control and Monitor* tab. From here, the motor can be started by pressing the *Start* button, and the speed varied by moving the slider or entering a speed in the numeric box. Dead time can also be varied by moving the slider or entering a ns value in the numeric box. This is illustrated in Figure 8.

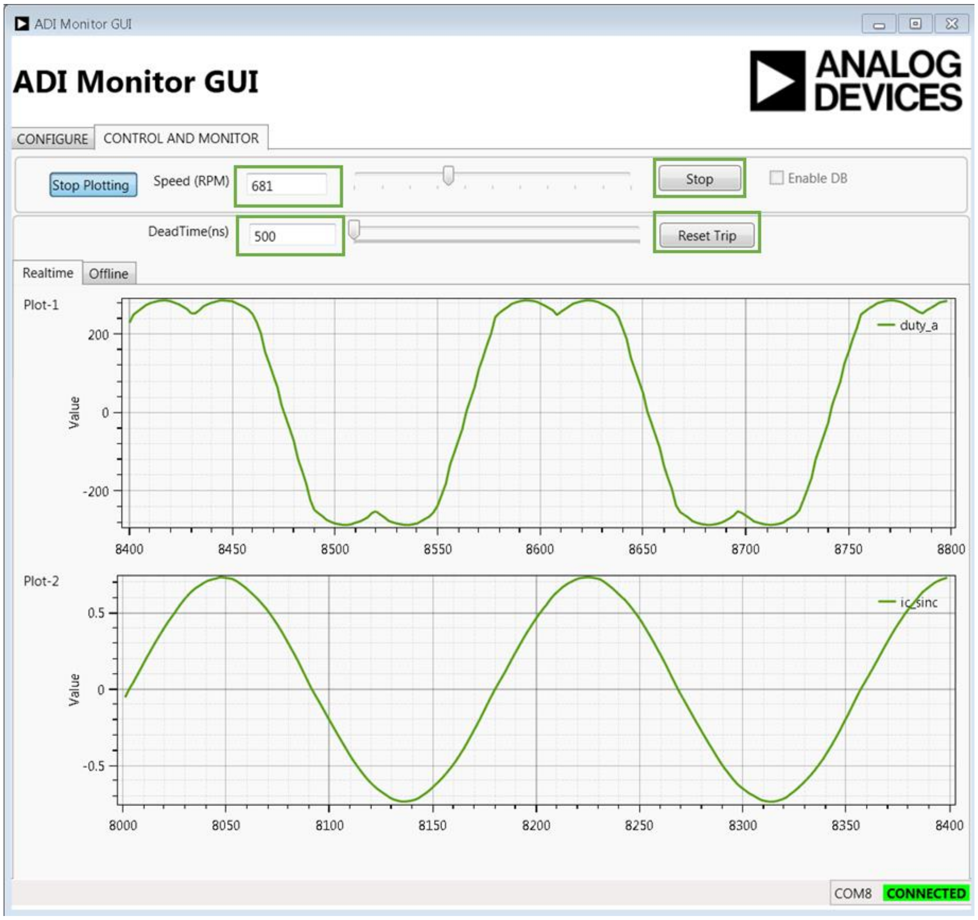


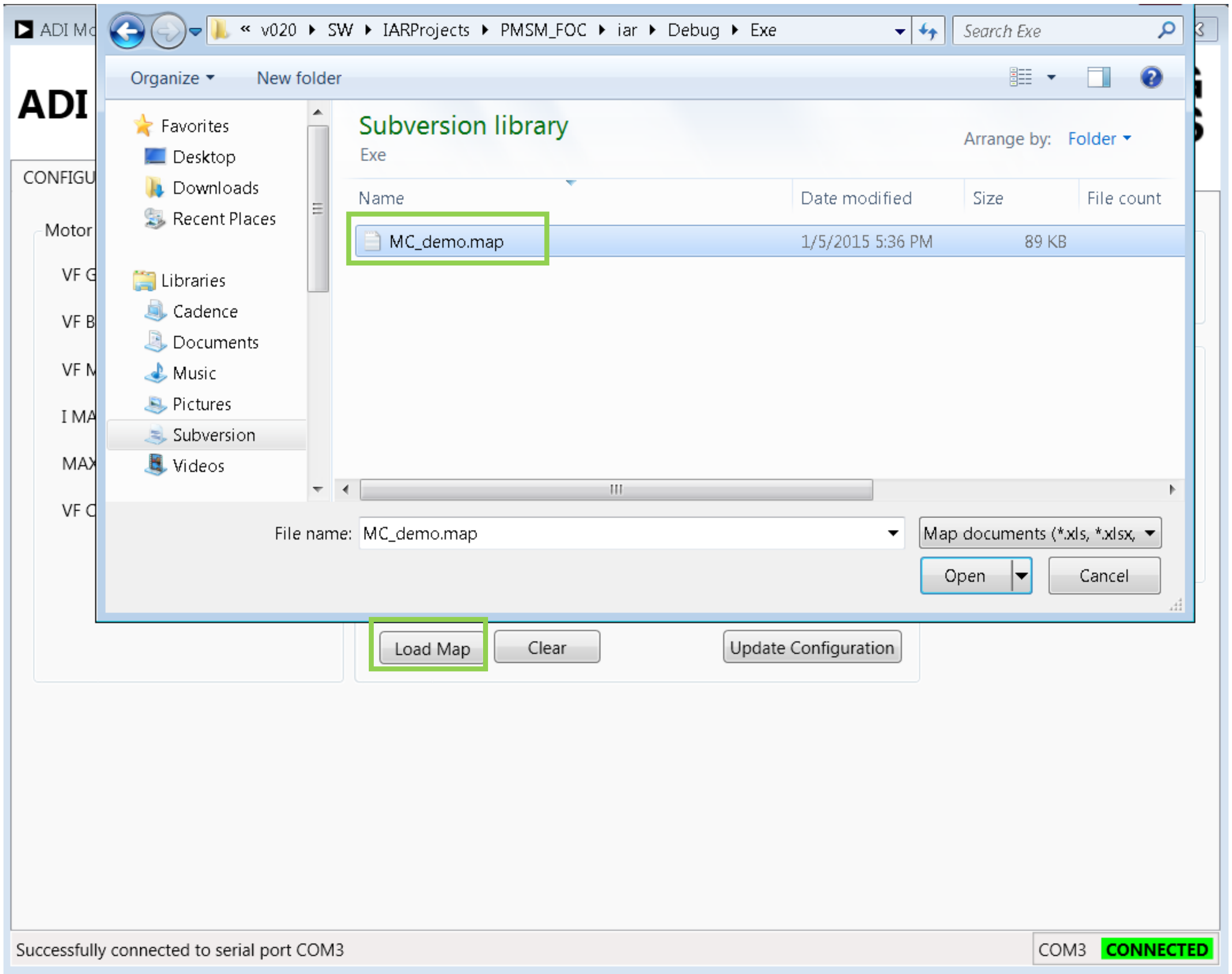
Figure 8: Motor Run Screen

9 Data Visualization

In order to import visualization data into the GUI, the linker map file (**MC_demo.map**) provided with the demo C code or executable must be loaded. This is selected from the *Configure* tab as illustrated in Figure 9 with the resultant variable list shown in Figure 10. The variables provided by default have a UID_ prefix in the map file and are listed in the Table below, along with their data types. It should be noted that the GUI cannot determine from the map file if a variable has signed or unsigned type, so if a variable is of signed type (e.g. duty_a) it is necessary to manually change the Type from UINT16 to SINT16 in the relevant column of Figure 10.

Variable Name	Variable	Data Type
duty_a	A phase duty cycle counter	SINT16

duty_b	B phase duty cycle counter	SINT16
duty_c	C phase duty cycle counter	SINT16
ia_sinc	U phase current – SINC Amps value	SINGLE
ib_sinc	V phase current – SINC Amps value	SINGLE
ib_sinc_raw	V phase current – SINC ADC value	SINT16
ic_sinc	W phase current – SINC Amps value	SINGLE
ic_sinc_raw	W phase current – SINC ADC value	SINT16
theta_vf	Open loop electrical angle	SINGLE
Vab	U phase-Vphase line to line voltage – SINC Volts	SINGLE
Vbc	U phase-Vphase line to line voltage – SINC Volts	SINGLE
Vca	U phase-Vphase line to line voltage – SINC Volts	SINGLE
Vdc	DC bus voltage	SINGLE
Vab_sinc_raw	U phase-Vphase line to line voltage – SINC ADC value	SINT16
Vdc_sinc_raw	DC bus voltage – SINC ADC value	SINT16
error_code	Fault code (0=No faults, 1= PWM trip, 2=SINC overcurrent trip, 3 =12V/3.3V supply missing, 4= Overtemp, 5=Gate driver fault, 6=DESAT fault)	UINT8



Name	Date modified	Size	File count
MC_demo.map	1/5/2015 5:36 PM	89 KB	

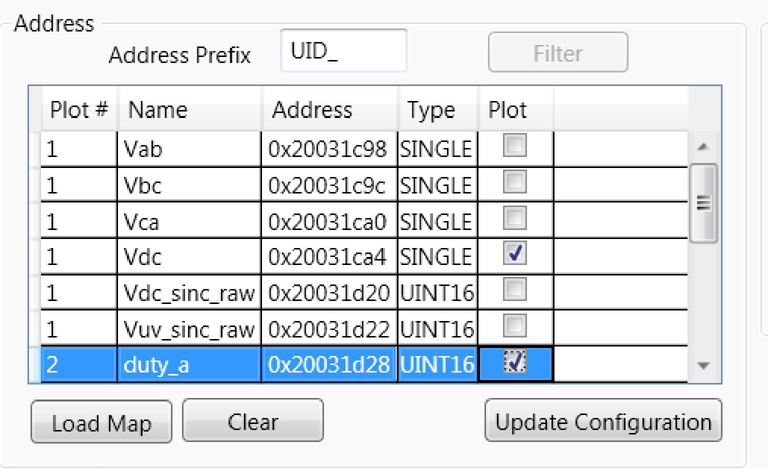
File name: MC_demo.map

Map documents (*.xls, *.xlsx,

Successfully connected to serial port COM3

COM3 CONNECTED

Figure 9: Load Map file for Data Visualization



Address Prefix UID_

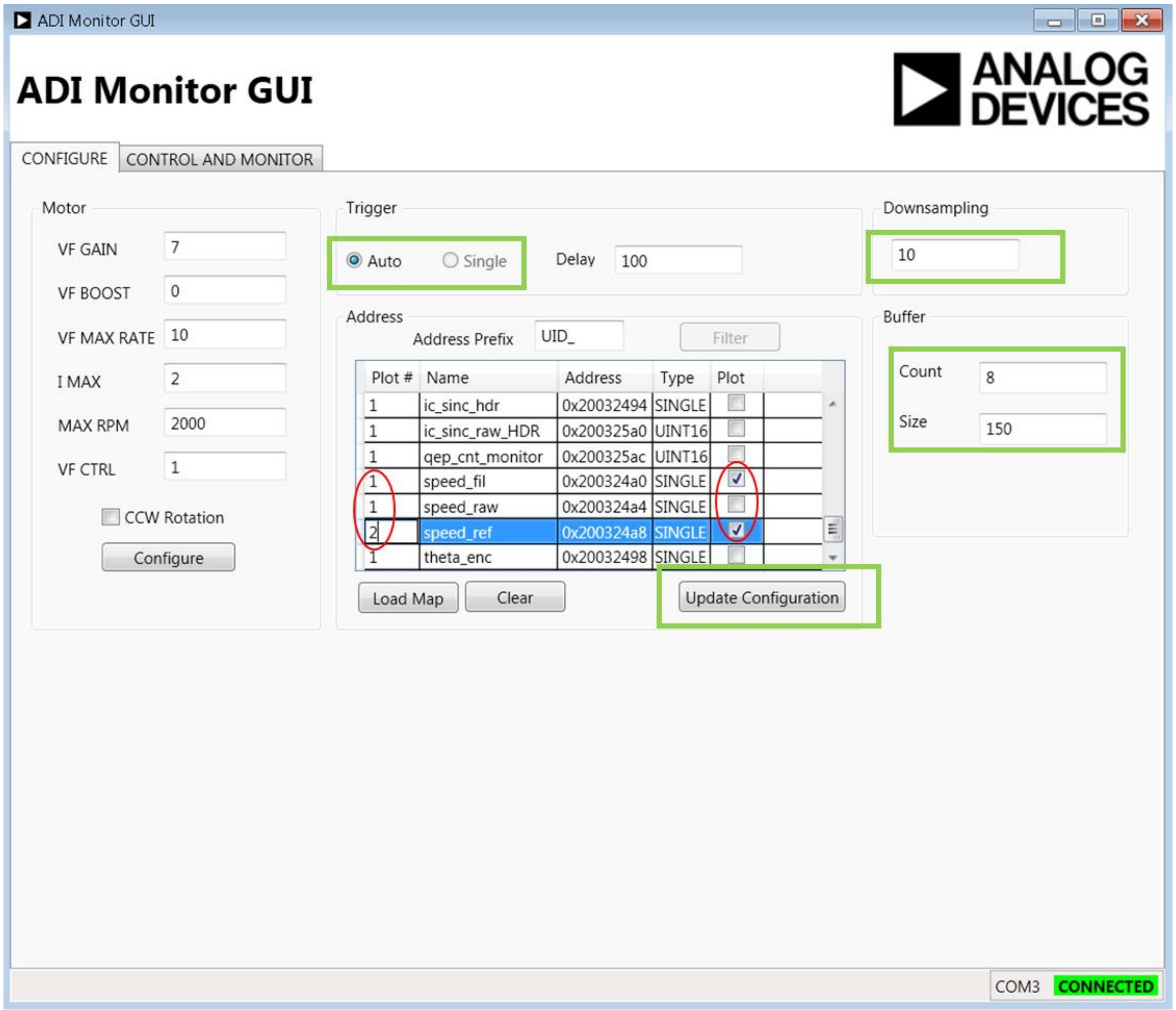
Plot #	Name	Address	Type	Plot
1	Vab	0x20031c98	SINGLE	□
1	Vbc	0x20031c9c	SINGLE	□
1	Vca	0x20031ca0	SINGLE	□
1	Vdc	0x20031ca4	SINGLE	☑
1	Vdc_sinc_raw	0x20031d20	UINT16	□
1	Vuv_sinc_raw	0x20031d22	UINT16	□
2	duty_a	0x20031d28	UINT16	☑

Figure 10: Visualization Variables

In order to select a variable for plotting, selection is by means of the check-box in the Plot column. The variable can be plotted in Plot 1 or Plot 2 (see Figure 8). There are certain constraints on the number of bytes and buffers that can be streamed using the GUI. The program will limit the total buffer size to 1.7k bytes. The number of variables plotted can be increased by reducing the buffer size, which is maximum 200, although a total maximum selected variable size of 8 bytes is allowed (e.g. 2xSINGLE or 4xINT16, or 1xSINGLE+2xINT16 etc).

The steps to prepare for plotting of variables are (refer to Figure 11):

1. Select all of the variables to be plotted and whether each one is to be on Plot 1 or Plot 2, bearing in mind the maximum buffer memory size (1.7k) and the maximum byte count per buffer (8).
2. Down Sampling factor – the buffer time slice length equals the PWM switching period x Down Sampling ratio, so for a down sampling ratio of 1, the sampling period will be 100µs, and with a buffer length of 200, the plot time slices will be of 20ms length. To look at longer time slices, increase the down sampling factor.
3. Press *Update Configuration*
4. On the *Control and Monitor* tab, press *Start Plotting.*
5. If the variables are to be changed, press *Stop Plotting* and return to the *Configure* tab to change the configured variables.



ADI Monitor GUI

Motor

VF GAIN	7
VF BOOST	0
VF MAX RATE	10
I MAX	2
MAX RPM	2000
VF CTRL	1

□ CCW Rotation

Trigger

● Auto ○ Single

Delay 100

Downsampling

10

Address

Address Prefix UID_

Plot #	Name	Address	Type	Plot
1	ic_sinc_hdr	0x20032494	SINGLE	□
1	ic_sinc_raw_HDR	0x200325a0	UINT16	□
1	qep_cnt_monitor	0x200325ac	UINT16	□
1	speed_fil	0x200324a0	SINGLE	☑
1	speed_raw	0x200324a4	SINGLE	□
2	speed_ref	0x200324a8	SINGLE	☑
1	theta_enc	0x20032498	SINGLE	□

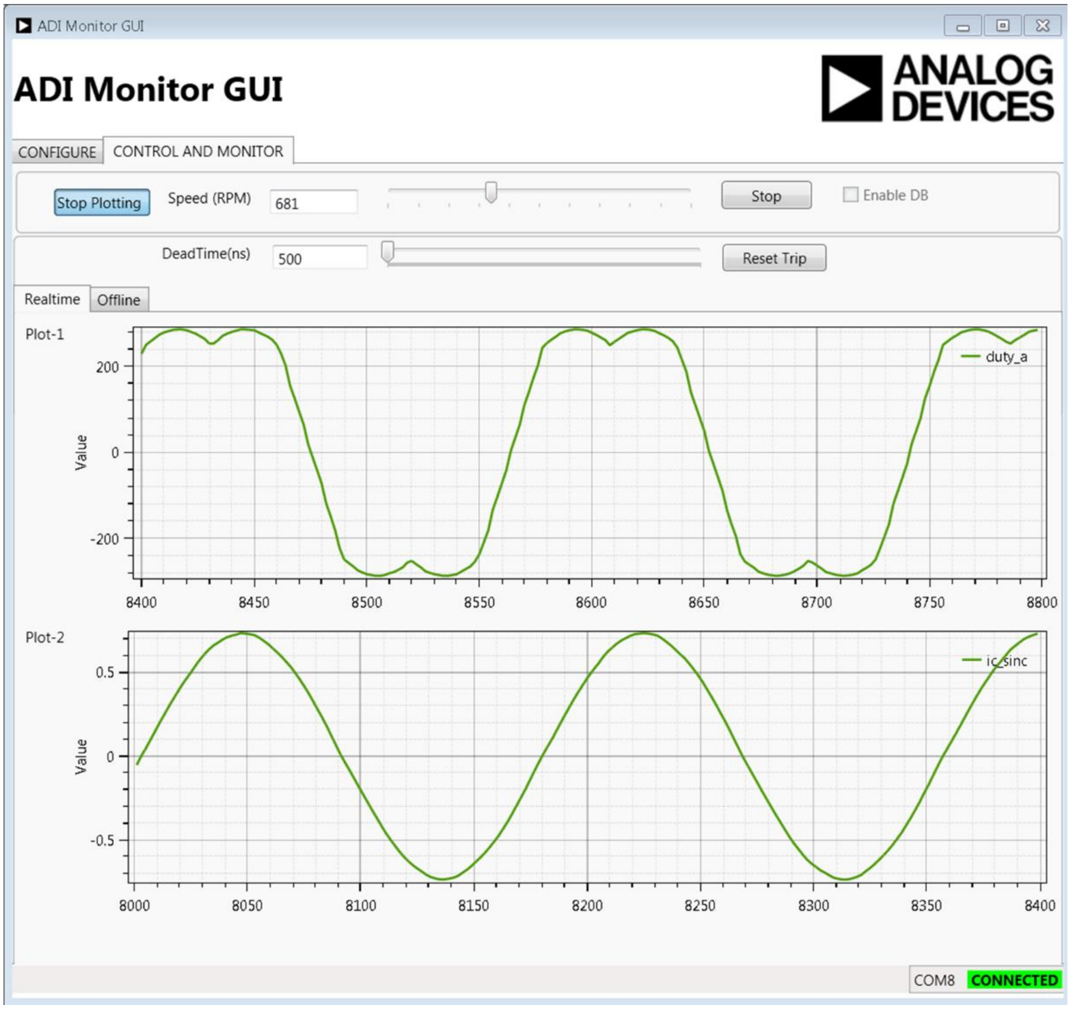
Buffer

Count	8
Size	150

COM3 CONNECTED

Figure 11: Setup of Variables for Plotting

The variables will start to plot in the *Control and Monitor* tab as illustrated in Figure 12.



Speed (RPM) 681

DeadTime(ns) 500

Figure 12: Variable plotting

10 Other Board Features

This quick start guide describes operation of the inverter platform as controlled from an ADSP-CM408 EZkit. However the platform is designed to be easily operable from any appropriate processor or FPGA board. The minimum requirement is for six PWM logic level signals (3.3V) to drive the IGBTs along with a gate driver enable signal. Feedback signals from fault logic, manual dials/switches and sigma-delta modulators can also be fed back to the processor or FPGA.

In addition, spare analog and digital pins are provided on the 0.1” header connectors that can be used to route other external signals such as motor encoder feedback to the ADSP-CM408 EZkit or to another processor or FPGA kit via the main P1 connector. Please see the schematics for more details on the external connectors J4-J9.

Many test points are also available to allow oscilloscope viewing of board signals. These include: