AN1292 Demonstration ReadMe for the dsPICDEM™ MCLV-2 Development Board with the dsPIC33EP256MC506 Internal Op Amp PIM (MPLAB 8)

1.1 INTRODUCTION

This document describes the setup requirements for running the Sensorless FOC algorithm with a PLL Estimator, which is referenced in AN1292 "*Sensorless Field Oriented Control (FOC) for a Permanent Magnet Synchronous Motor (PMSM) Using a PLL Estimator and Field Weakening (FW)"* using a dsPICDEM™ MCLV-2 Development Board in the Internal Op amp configuration.

1.2 SUGGESTED DEMONSTRATION REQUIREMENTS

MPLAB and C30 versions used:

- MPLAB version 8.84 (or later)
- C30 version 3.31 (or later)

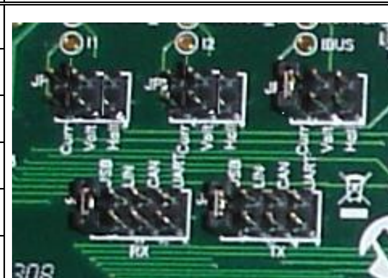
Hardware used with part numbers, available from www.microchipdirect.com:

- dsPICDEM MCLV-2 Development Board (DM330021-2)
- dsPIC33EP256MC506 Internal Op amp PIM (MA330031)
- 24V Power supply (AC002013)
- 24V Hurst motor (AC300020)

1.3 HARDWARE SETUP

The following hardware setup allows the Sensorless FOC algorithm to run on the dsPICDEM MCLV-2 Development Board using Op amps that are internal to the dsPIC33EP256MC506 device.

1. With the dsPICDEM MCLV-2 Development Board disconnected, and making sure there is no power, open the enclosure and set up the following jumpers:

Jumper	Pins to Short	Board Reference
JP1	Don't care	
JP2	Don't care	
JP3	Don't care	
JP4	USB position	
JP5	USB position	
J5	Don't care	
JP11	Don't care	

2. Connect the motor phases to the output header, J7. The winding color can be connected in any order to M1, M2, and M3 since it is a sensorless control algorithm. The Green wire does not have internal connection in the motor, so it can be left unconnected.

3. Connect the Internal Op amp Configuration Board into J14. Ensure that the matrix board is correctly oriented before proceeding.

4. Connect the 24V power supply to the dsPICDEM MCLV-2 Development Board, using the J2 connector.

5. Connect the programmer/debugger to the J11 connector.

6. For enhanced demonstration, the application requires the Real-Time Data Monitor (RTDM). Users can connect a mini-USB cable from their computer to the J8 connector of the dsPICDEM MCLV-2 Development Board.

Notice that when the development board is powered and connected to the USB host for the first time, the driver needs to be installed on the host for proper operation.

a) Extract the `PC_USB_driver_for_win2k_xp_vista32_64.zip` archive file to a local directory. This file is part of the ZIP file of the code.

b) When prompted to select the driver for new USB device found, select the driver from the ones provided corresponding to the operating system used: Windows 2000, XP, or Vista (32- or 64-bit). Wait for the indication that the new device was installed properly and is ready to be used. Once the USB driver is installed, it will emulate a Serial COM Port, visible in the Windows Device Manager. A message indicating that the driver has not passed Windows logo certification may appear. Click **Continue Anyway**.

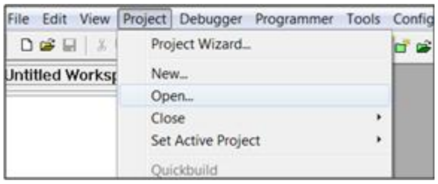
c) When the USB driver is installed, a new COM port should show up in Windows device hardware manager. This should be the COM port used for Enhanced Demonstration.

1.4 SOFTWARE SETUP AND RUN

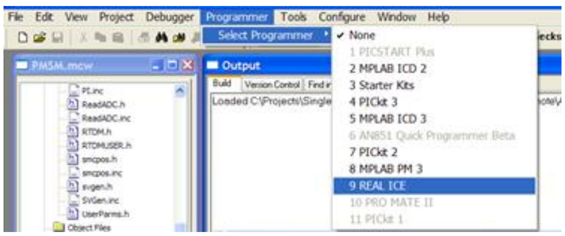
1.4.1 Basic Demonstration

This demonstration consists of running the motor using a push button and varying the speed with a potentiometer. The software, which is available for download from the Microchip website, is already configured for enabling the basic demonstration.

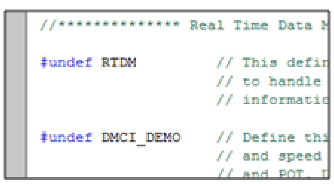
1. Start MPLAB IDE and open the `PMSM.mcp` workspace.



2. Select the desired programmer/debugger. In this example, REAL ICE™ is selected.



3. Make sure that `RTDM` and `DMCI_DEMO` are not defined in the `UserParms.h` file. This allows the push button and the potentiometer to have control over starting and stopping the motor and its speed. If this is defined, the motor will not start until the proper procedure is followed for the DMCI demonstration. Refer to Enhanced Demonstration Using Real-Time Data Monitor (R if the DMCI demonstration is required.

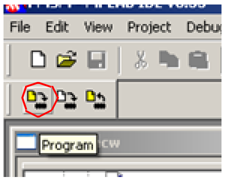


Also, in the `UserParms.h` file, ensure that `BIDIRECTIONAL_SPEED`, `TUNING`, `OPEN_LOOP_FUNCTIONING`, and `TORQUE_MODE` are not defined.

```
/* In this mode the speed doubling is no longer possible */
#undef BIDIRECTIONAL_SPEED

/* define the follwing TUNING for slow acceleration ramp ins
#undef TUNING

/* open loop continuous functioning */
/* closed loop transition disabled  */
#undef OPEN_LOOP_FUNCTIONING

/* definition for torque mode - for a separate tuning of the
 controllers, tuning mode will disable the speed PI controll
#undef TORQUE_MODE
```


4. Build the code by selecting the **Release** mode from the drop-down list and clicking the **Build All** icon.

5. Download the code to the target device on the dsPICDEM MCLV-2 Development Board.



6. Run or stop the motor by pressing S2. You can double the speed by pressing S3.

7. Vary the motor speed using the potentiometer (labeled POT).

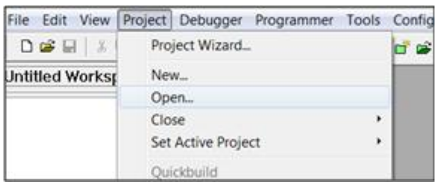
8. To double the speed of the motor, press S3. Pressing S3 again will reduce the speed of the motor by 50%.

9. Press S2 to stop the motor.

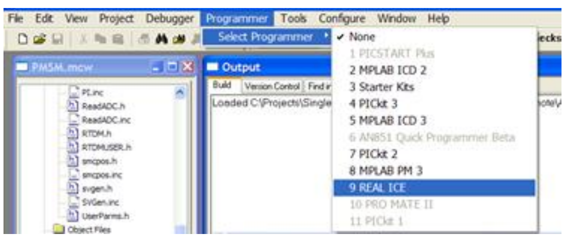
1.4.2 Enhanced Demonstration Using Real-Time Data Monitor (RTDM) and Dynamic Monitor and Control Interface (DMCI)

1. In order to utilize RTDM communication for this demonstration, a mini-USB c-onnection is required. Connect a mini-USB cable from your computer to the J8 connector on the dsPICDEM MCLV-2 Development Board, labeled USB.

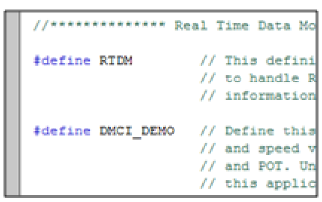
2. Start MPLAB IDE and open the `PMSM.mcp` workspace



3. Select the desired programmer/debugger. In this example, REAL ICE™ is selected.

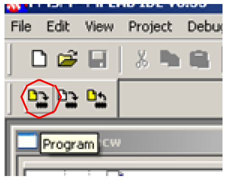


4. Make sure that `RTDM` and `DMCI_DEMO` are defined in the `UserParms.h` file. This allows DMCI to have control over starting and stopping the motor and its speed. If this is not defined, the motor will not start until the S2 push button is pressed.

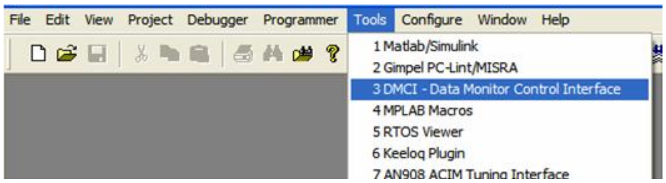


5. Build the code by selecting the **Release** mode from the drop-down list and clicking the **Build All** icon.

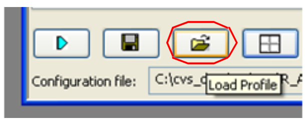
6. Download the code to the target device on the dsPICDEM MCLV-2 Development Board.



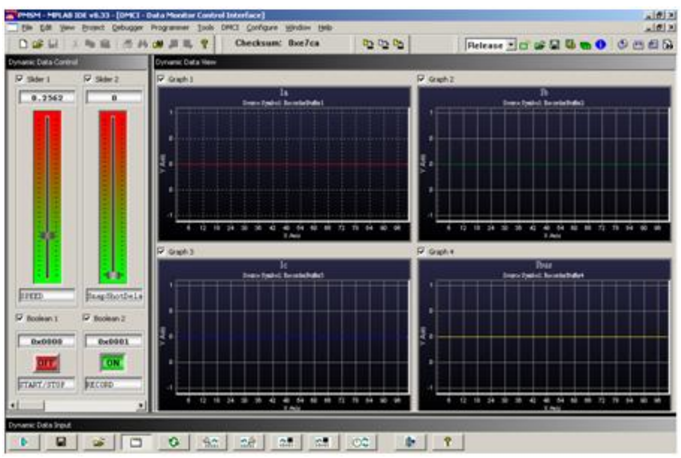
7. Open the DMCI window by selecting *Tools>DMCI – Data Monitor Control Interface*.



8. Click **Load Profile**, and from the same folder where your project resides, load the `DEMO.dmci` file, which contains a previously configured profile.

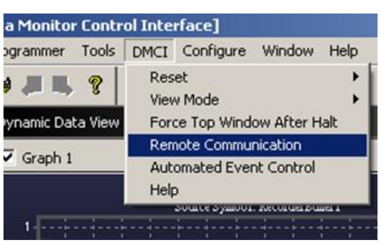


9. The DMCI window appears as follows:

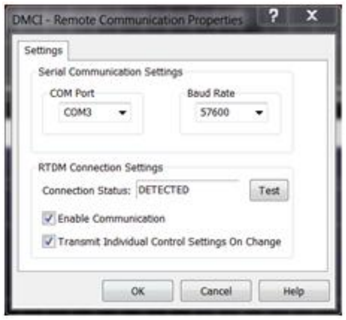


Please consult the *"Real-Time Data Monitor User's Guide"* (DS70567) for additional settings needed for a RDTM connection. This document explains the steps needed for the proper communication settings between the Host and Embedded side.

10. Select *DMCI>Remote Communication* to connect RTDM with your computer.

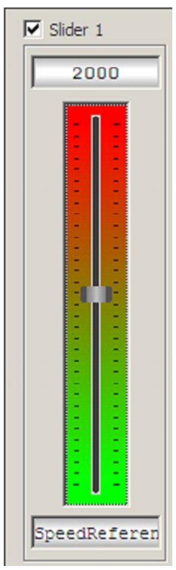


11. Remote Communication needs to be established, as indicated in the following figure (the communication baud rate should be set to 57600, while the COM port used depends on your particular settings).



12. Once communication is detected, make sure the **Enable Communication** box is checked and click **OK**.

13. Using the SpeedReference slider, adjust the value to 2000. Please note that positive and negative references are possible; therefore, bidirectional functioning is selected by default with `RTDM`.

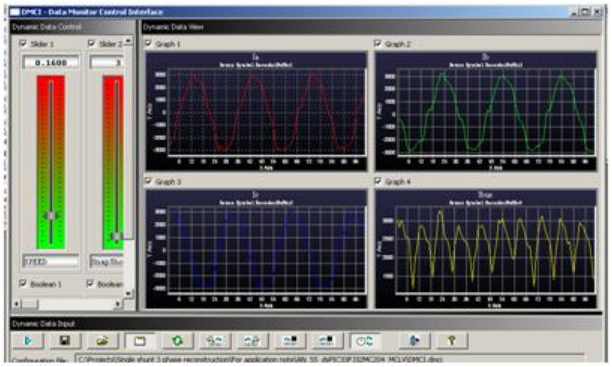


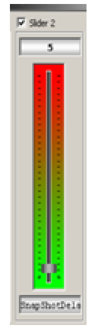
14. Press START/STOP in the DMCI window to start the motor at initial speed.

15. To plot variables in real-time, enable Automated Event Control by clicking the DMCI icon.

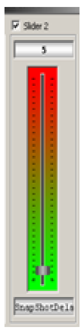
16. Use the SpeedReference control to increase or decrease the speed.You may want to click on the slider frame while using the up/down or Page Up/Page Down keys on your keyboard to increase or decrease the reference speed. Please note that negative references are available, which reverse the motor's direction of rotation.

17. The DMCI window shows variables plotted in real time and are updated automatically.





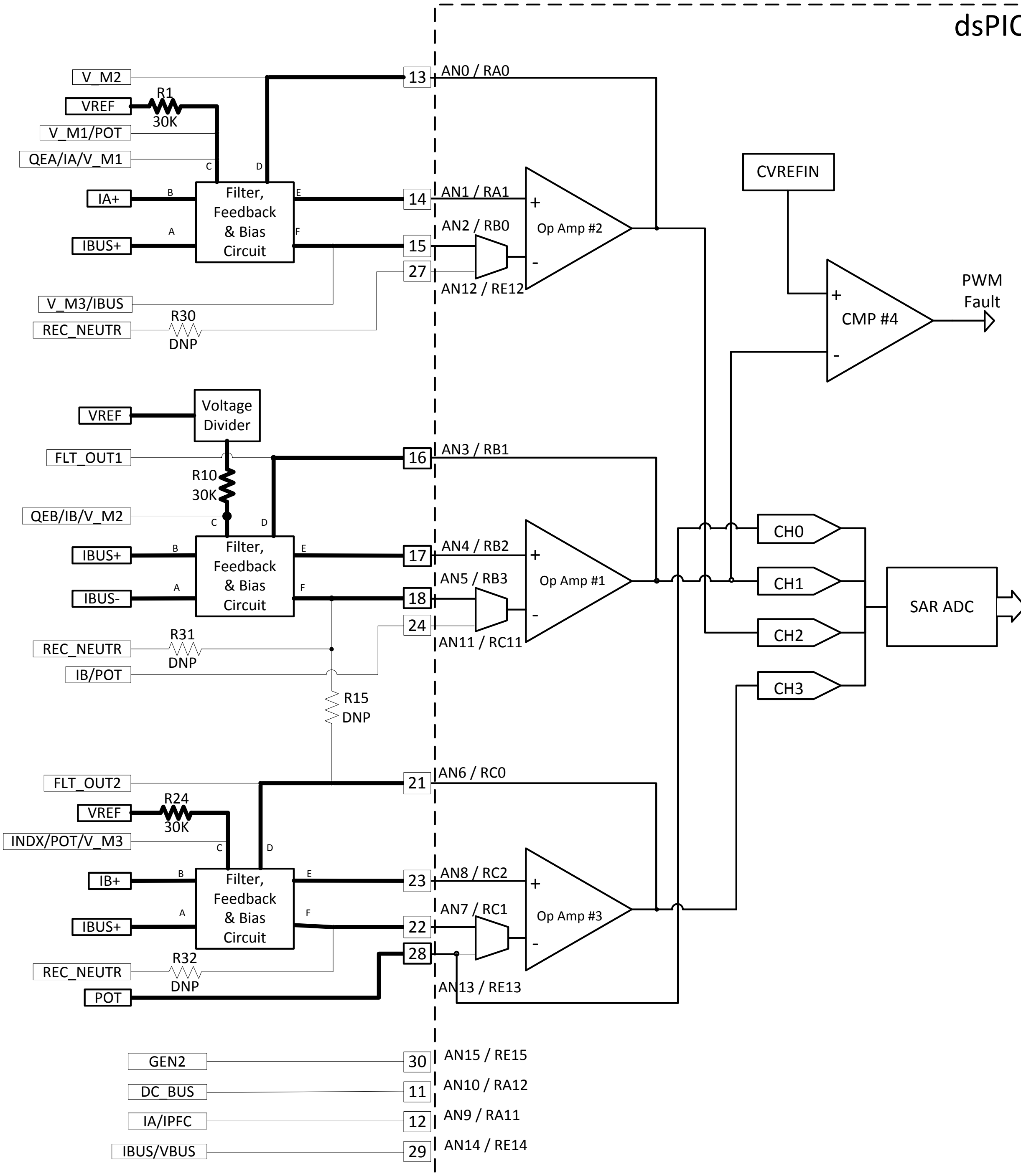
18. To change the time window to see more time on each plot, change the value of the SnapShotDelay, which controls how the buffers are being filled in the code.



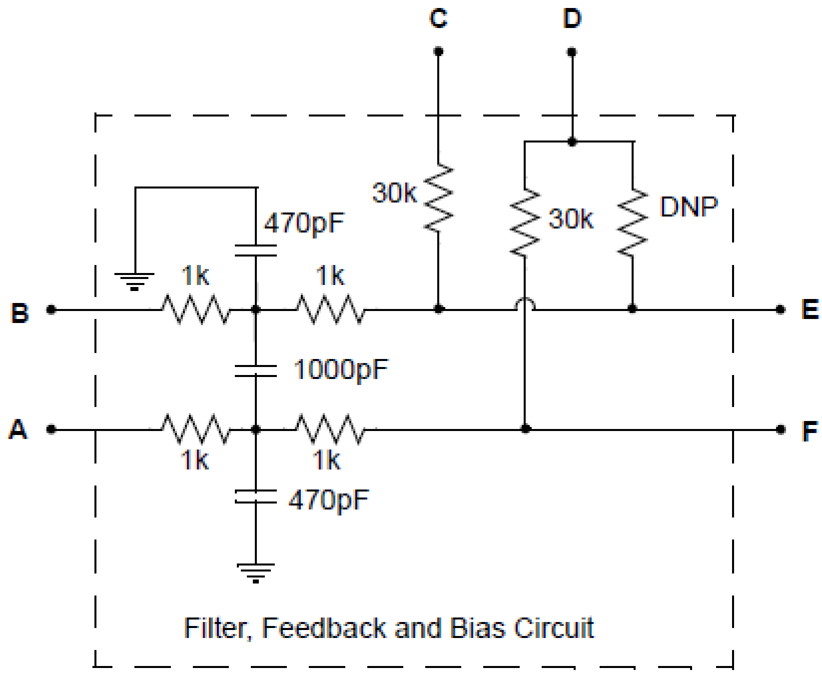
1.5 I/O CONFIGURATION

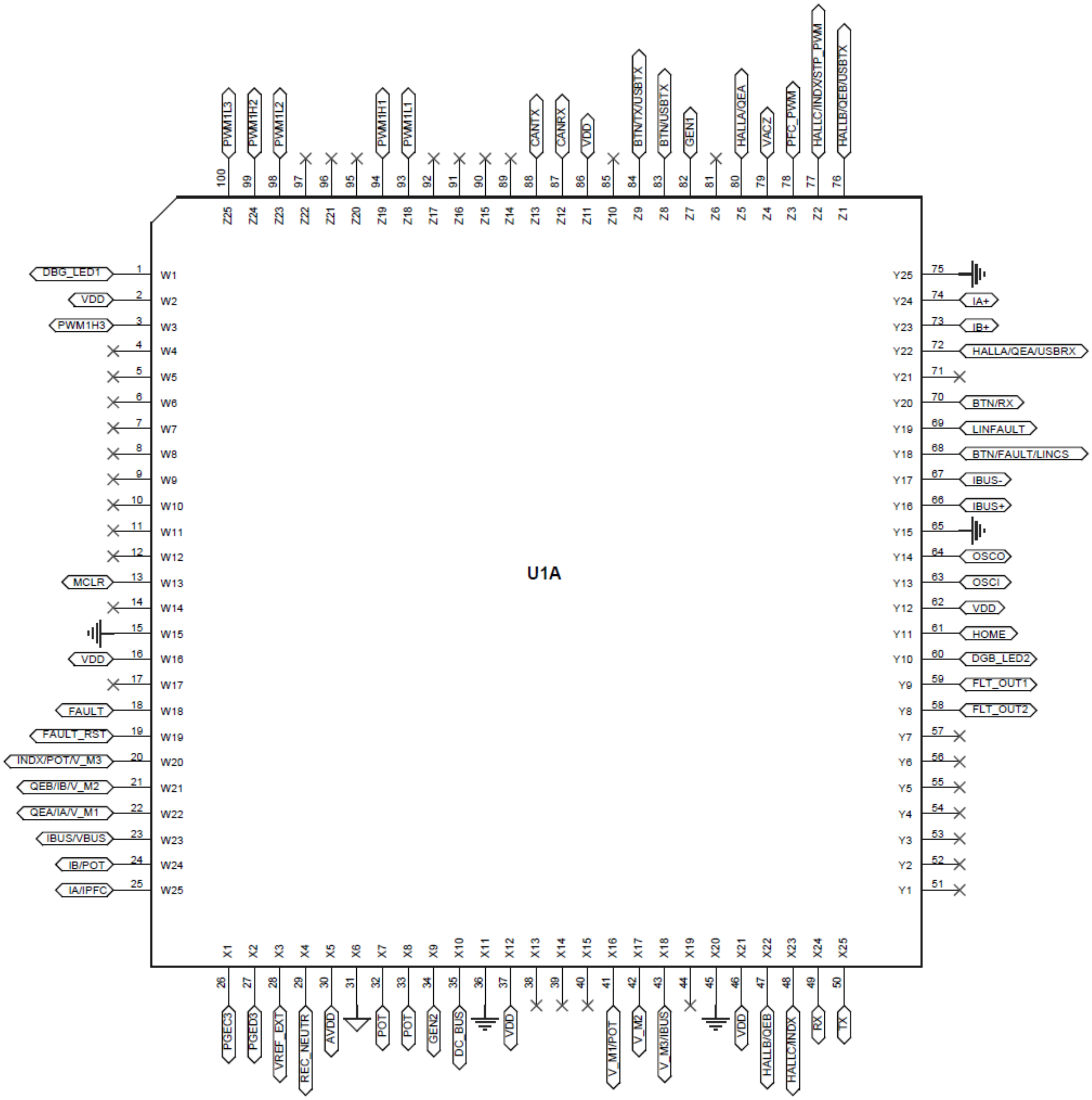
1.5.1 Analog I/O Configuration

The following figure shows a block diagram of the analog signal paths on the PIM (MA330031) and a description of their connections inside the dsPIC DSC device (dsPIC33EP256MC506). The analog signal paths used in this demonstration are highlighted. For details regarding the PIM schematics, refer to the PIM information sheet document, available at www.microchip.com/pims.



Note: Connections depicted inside the dsPIC block depend on the configuration settings selected in the software.


C
D
30k
470pF
30k
DNP
1k
1k
B
E
1000pF
A
F
1k
1k
470pF
Filter, Feedback and Bias Circuit



1.5.2 Digital I/O Configuration

Functional Description	Device Pin Function	Input/Output
PWM	RB10 through RB15	Output
Switch S2	RG7	Input
Switch S3	RG6	Input
UART RX	RC5	Input
Debug LED1	RD6	Output
Debug LED2	RD5	Output
Test Point	RD8	Output
UART TX	RF1	Output