



SDR Cube Transceiver

SOFTWARE DEVELOPMENT KIT



Technical instructions and guidance for modifying the application software running on the SDR Cube Transceiver, or for developing new applications on this platform.

CONTENTS

1) Introduction.....	2
2) Minimum Requirements.....	2
3) Installing the Development Environment	3
4) Setting Up MPLAB Build Options.....	3
5) Compiling & Building with MPLAB.....	4
6) Programming with MPLAB + ICD2.....	5
7) Field Loading of Software Application.....	5
8) Hardware Architecture.....	7
9) Software Architecture	7
10) Memory Usage.....	8

1) Introduction

This Software Development Kit (“SDK”) is intended for software programmers wishing to modify the existing SDR Cube Transceiver application, and for those wishing to develop entirely new applications based on the SDR Cube hardware platform.

We shall not attempt to instruct on basics of software programming, Microchip development tools, or dsPIC hardware. Those attempting to use this SDK may refer to the plethora of good documentation and instructional material provided by Microchip at www.microchip.com.

A general understanding of the time critical embedded system programming is required. Good skills of C-language and preferably Microchip dsPIC assembler are also beneficial. Microchip dsPIC architecture and particularly dsPIC33FJ256GP710 device data sheet learning is recommended ... <http://ww1.microchip.com/downloads/en/DeviceDoc/70286C.pdf>

2) Minimum Requirements

In order to effectively use this SDK as intended, one must have ...

- a) **MPLAB** – The latest version of this Integrated Development Environment from Microchip is v8.31, as of this writing. You can download MPLAB for free from www.microchip.com/MPLAB
- b) **C30** – The latest version of the C compiler from Microchip is v3.2 or higher. The Student Edition download is free, and although the Microchip web page states that the optimization engine expires after 30 days’ use, the compiler may continue to be used effectively without the optimization well after this time period. You can download the latest version of C30 from www.microchip.com/C30
- c) **ICD2** – The most capable (and simultaneously low cost) programming/debugging pod available for this platform is the In Circuit Debugger 2 (“ICD 2”), available from a variety of sources including Microchip and Digi-Key.

- d) **Windows PC** – The tools described thus far only operate on Windows and the PC becomes your development workstation. We know the tools operate on Windows XP with Service Pack 2. They may also work on the newer Windows Vista operating system, or on the older Windows operating systems, but we have not tried them. As is most often the case when using a PC as development workstation, the best experience will be achieved using a powerful PC -- fast processor, much RAM, lots of hard disk space.

3) Installing the Development Environment

- a) Install the latest version of MPLAB.
- b) Install the latest version of the C30 compiler.
- c) Run MPLAB. (It takes almost a full minute for MPLAB to fully load in and start up.)
- d) Plug in the ICD2 programming pod to the computer's USB port. Follow any instructions that might appear for loading the driver into the Windows OS.
- e) In the Configure menu ... Select "Select Device", and then use the pull-down menu to select the dsPIC MPU used for the SDR Cube project: "dsPIC33FJ256GP710".
- f) In the Configure menu, select Settings ...
 - 1) Select the tab for Program Loading and uncheck the box for "Clear program memory upon loading a program". All other boxes except the top one should be unchecked.
 - 2) In the Projects tab, uncheck "Use one-to-one project-workspace model"
- g) Plug in the USB connection to the ICD2 programming pod. The system (and MPLAB) will automatically recognize the device.
- h) Under the Programmer menu, select "Select Programmer" and specify "MPLAB ICD2"
- i) Plug the ICD2 cable into the target hardware: P6 "ICD2" on the SDR Cube's DSP Board).
- j) Apply power to the Cube.
- k) In the Programmer menu, select Connect. You should see text appear in the Output window indicating that the ICD2 was able to successfully connect to the target hardware.

4) Setting Up MPLAB Build Options

A cumbersome-yet-important configuration item must be done when starting a new project, or when starting to work with the SDR Cube application: setting the build options to point the MPLAB environment to the proper folders containing the project. For this explanation, we will use the SDR Cube project as the example.

- a) **File ... Close Workspace** (if some workspace was already open on the screen.)
- b) **File ... Open Workspace**, and then navigate to the **sdr-cube.mcw** file in the top level of your project folder and open it. This will bring up the workspace window on the left with various **mcp** folder(s) containing the application(s), and the corresponding subfolders for Source Files, Header Files, Object Files, Library Files, Linker Script, and Other Files.
- c) Set the **Project Build Options ...**
 - Large Data Model
 - use 64-bit Double
 - heap 500
 - modify linker script for DSP library search

Remember special linker scripts for the flasher and SDR Cube project. SDR Cube project linker script should have:

```
** Memory Regions
   program (xr) : ORIGIN = 0xC00,

and DSP lib search options ...
   SEARCH_DIR(..\..\..\lib)
   SEARCH_DIR(..\..\..\support\gld)
   INPUT(-ldsp)
   INPUT(-lpic30)
   INPUT(-lm)
   INPUT(-lc)
```

The project require no compiler optimization switches. All very time critical things are assembler coded. Student edition of the compiler without optimization is used to compile the SDR Cube project.

d) Set MPLAB environment pointers for the SDR-Cube Program

- 1) **Set the SDR-Cube Program as active** – Right-click the top-level folder called “SDR-Cube.mcp”, hereafter called the Cube Program”. Right-clicking will produce a pop-up menu, from which you will select the “Set Active” item.
- 2) Right-click again SDR-Cube.mcp and this time select “**Build Options ...**”
- 3) In the Directories tab, pull down the menu for “Show directories for:” and select the **Assembler Include Search Path**.

The highlighted pathname in the larger window below this is the current path to the Output folder, which may or may not be correct. Click on the right end of that highlighted line to see a new control representing the navigation icon “...”. Clicking on this control will bring you to a folder structure on your PC through which you will use to navigate to the proper folder. Locate and select the top-level folder where your project files reside (C:\SDR-Cube). Click OK.

- 4) In the Directories tab, pull down the menu for “Show directories for:” and select the **Include Search Path**.

In a manner similar to how you navigated to the previous folder, navigate to the C:\SDR-Cube folder. Select this one, see it appear in the “Folder:” field, and click OK.

- 5) In the Directories tab, pull down the menu for “Show directories for:” and select the **Library Search Path**.

For this one, you need to ensure that the path is pointing to the C30 “lib” folder on your system. If not already there, start at the top to navigate to C:\Program Files\Microchip\MPLAB C30\lib. Once pointed there, click OK.

- e. **Click APPLY** to end this part of the process.
- f. In the MPLAB menu bar along the top of the screen, select **Project** and move your mouse down to **Build Configuration** item. Click Release if a checkmark is not already on that item.
- g. All done!

5) Compiling & Building with MPLAB

If all the paths were set correctly in Build Options, you should experience a successful compilation and build process in this section. (That is, for a known-good program such as the distribution files for the SDR Cube Transceiver.)

NOTE: For those familiar with the terms “build” and “make”, we specifically will instruct you to “make” these files here. “Build all” is a term that forces a compile and linking of everything in the project folder, whereas we need to compile and link selectively (i.e., “make”) for the two programs being combined into one object file image.

Make the SDR-Cube Program -- Click the top-level folder **SDR-Cube.mcp** to ensure that it is highlighted (selected) and then right-click that line and select Build All. Numerous lines will be displayed in the Output window on the right, indicating the compile and linking processes happening. If successful, it will again terminate with SUCCEEDED.

6) Programming with MPLAB + ICD2

You are now ready to program the target hardware.

- 1) If not already in place, make sure the ICD2 module is plugged into the USB port of your computer, and its target cable is connected to P6 on the Cube’s DSP Board.
- 2) In the MPLAB menu bar, select Programmer and then select **Connect**. You should see some text display in the Output window on the right of your screen, ending with “...Passed. MPLAB ICD2 Ready”. You might also see the LCD on the Cube come alive with residual text from the last time it was programmed and used.
- 3) Pull down the Programming menu item from the MPLAB menu bar, and this time select **Program**. You should see the following sequence of events displayed in the Output window while the target hardware is programmed ...

```
Programming Target...
...Validating configuration fields
...Erasing Part
...Programming Programming Executive
...Verifying Programming Executive
...Programming Program Memory (0x0 - 0x154FF)
Verifying...
...Program Memory
...Verify Succeeded
...Programming User IDs
...User ID Memory
...Programming Configuration Bits
.. Config Memory
Verifying configuration memory...
...Programming succeeded
18-Feb-2008, 13:30:25
```

```
MPLAB ICD 2 Ready
```

During the programming sequence, which normally takes less than a minute, you will see a green progress bar moving in the lower portion of the overall MPLAB window.

- 4) Turn off the Cube and disconnect the cable connecting the ICD2 to the DSP Board. Turn the Cube’s power back on again and it will power-up with its normal boot sequence, displaying the splash screen followed by a display of the transceiver’s “front panel”.
- 5) All done!

7) Field Loading of Software Application

The SDR Cube has a Field Programming feature that allows a user to send the hex file of a new/updated software application to the Cube and have it permanently burned into the controller’s memory.

a) Determine what COM port number your USB or RS-232 serial port is -- You can find this out by clicking START and then right-clicking My Computer. Select Properties and then the Hardware

tab. Click Device Manager and find the Ports line item in the list. Expand that line item and see that the Communications Port will have a COM number shown. That is the serial COM port number that you'll use.

b) Get the FlashCube.bat file (and associated other files) onto your local computer -- Save the files to a known location, such as at the root of your hard drive (C:\)

c) Edit the two batch files (those with the “.bat” extension) to use the COM port number appropriate for your system, as noted in step ‘a’ above. Typically one can edit a batch file by right-clicking the file and select Edit. If necessary, change COM1 in the command line to be the active COM number of your system. Then close and save the batch files.

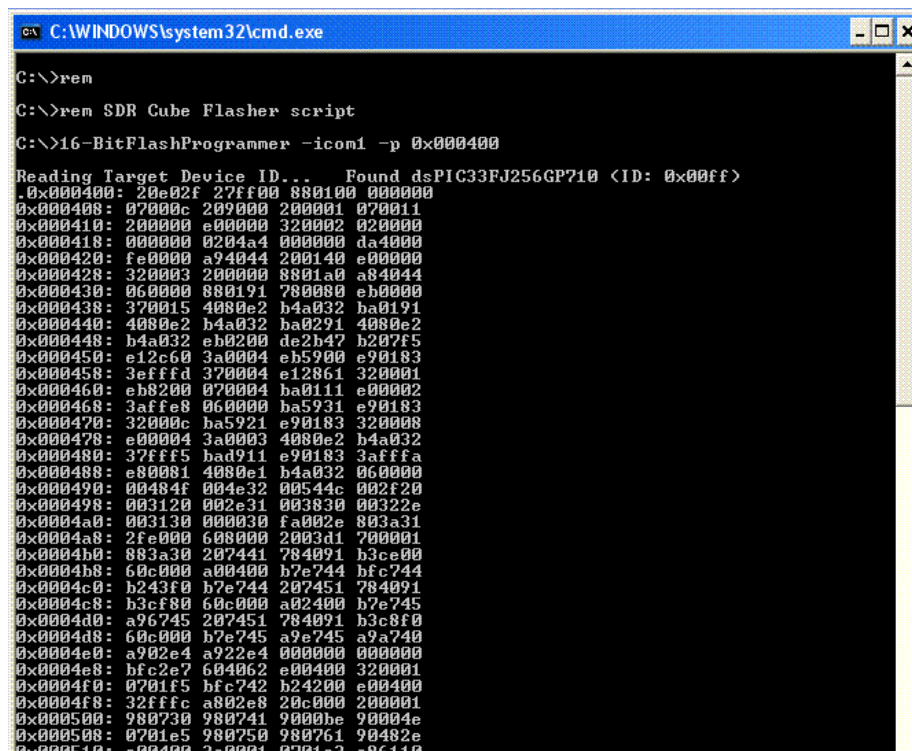
d) Connect the PC’s RS-232 Serial Port to the SDR Cube “Aux” port on the rear panel

A one-time preliminary step is to read a small memory segment from the SDR Cube. This will demonstrate to yourself that a successful connection has been made before you attempt to flash a new program into the Cube ...

e) Power up the Cube in a special way -- Hold down the "Menu" button on the Cube, while turning on the power switch. This commands the Cube to look to the serial port before it continues powering on normally. (If you have the cover open, notice that the red LED LD3 indicates that the running condition of the bootloader program.)

f) Run the ReadCube.bat file

g) The PC screen starts displaying the Cube memory contents in a black “Command Prompt” window, starting at location 0x400 -- Wait for the display on the PC to stop displaying the data, then compare it to the screen shot below. If it is generally like what we show, and if the memory display stops with a message: “Hit any key to continue”, then you indeed have a fully functioning serial interface! The numbers shown happen to be the contents of memory locations starting at address 0x400 in the dsPIC, which is the Cube software we programmed into the dsPIC before delivering it to you. (These numbers correspond to version 1.0 software - they will likely be somewhat different for follow-on versions.)



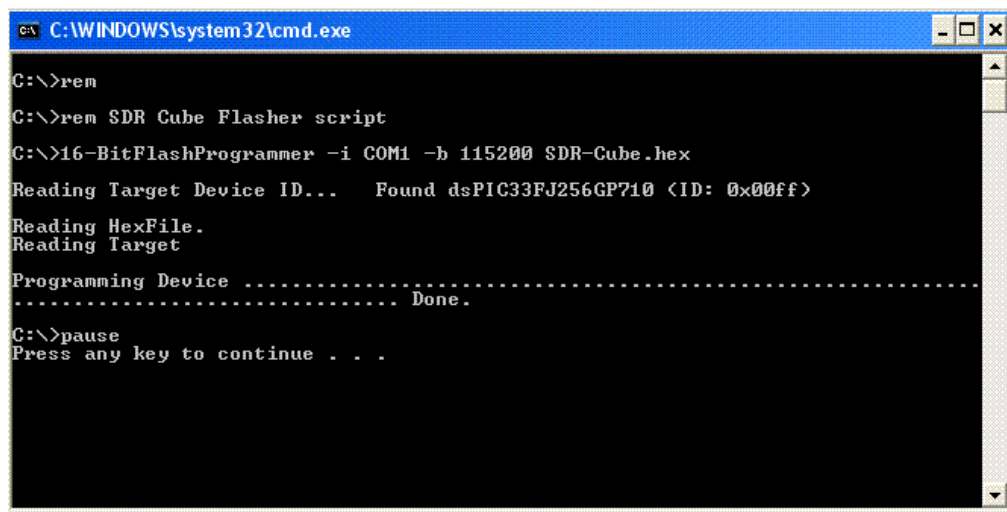
```
C:\WINDOWS\system32\cmd.exe
C:\>rem
C:\>rem SDR Cube Flasher script
C:\>16-BitFlashProgrammer -icom1 -p 0x000400
Reading Target Device ID... Found dsPIC33FJ256GP710 <ID: 0x00ff>
0x000400: 20e02f 27ff00 880100 000000
0x000408: 07000c 209000 200001 070011
0x000410: 200000 e00000 320002 020000
0x000418: 000000 0204a4 000000 da4000
0x000420: fe0000 a94044 200140 e00000
0x000428: 320003 200000 8801a0 a84044
0x000430: 060000 880191 780080 eb0000
0x000438: 370015 4080e2 b4a032 ba0191
0x000440: 4080e2 b4a032 ba0291 4080e2
0x000448: b4a032 eb0200 de2b47 b207f5
0x000450: e12c60 3a0004 eb5900 e90183
0x000458: 3efffd 370004 e12861 320001
0x000460: eb8200 070004 ba0111 e00002
0x000468: 3affe8 060000 ba5931 e90183
0x000470: 32000c ha5921 e90183 320008
0x000478: e00004 3a0003 4080e2 b4a032
0x000480: 37fff5 had911 e90183 3afffa
0x000488: e80081 4080e1 b4a032 060000
0x000490: 00484f 004e32 00544c 002f20
0x000498: 003120 002e31 003830 00322e
0x0004a0: 003130 000030 fa002e 803a31
0x0004a8: 2fe000 608000 2003d1 700001
0x0004b0: 883a30 207441 784091 b3ce00
0x0004b8: 60c000 a00400 b7e744 bfc744
0x0004c0: b243f0 b7e744 207451 784091
0x0004c8: b3cf80 60c000 a02400 b7e745
0x0004d0: a96745 207451 784091 b3c8f0
0x0004d8: 60c000 b7e745 a9e745 a9a740
0x0004e0: a902e4 a922e4 000000 000000
0x0004e8: bfc2e7 604062 e00400 320001
0x0004f0: 0701f5 bfc742 b24200 e00400
0x0004f8: 32ffffc a802e8 20c000 200001
0x000500: 980730 980741 9000be 90004e
0x000508: 0701e5 980750 980761 90482e
0x000510: e00400 3a0001 0201e3 a86110
```

Now that you know that you have a good connection to the Cube, you will next flash the new program into the Cube's controller ...

h) Once again, power up the Cube in a special way -- Hold down the "Menu" button on the Cube, while turning on the power switch. This commands the Cube to look to the serial port before it continues powering on normally.

i) Run the FlashCube.bat file

j) The PC starts sending the new program file to the Cube for flashing on a line-by-line basis. -- Again, if the enclosure cover is off, you will see the green LED on the DSP board change state each time a packet is received. Further, every time a line of data has been flashed, a decimal point ('.') is displayed. When about 100 lines of data have been, the PC displays "Hit any key to continue", and the Cube reboots and runs the new program. The FlashCube process is done!



```
C:\WINDOWS\system32\cmd.exe
C:\>rem
C:\>rem SDR Cube Flasher script
C:\>16-BitFlashProgrammer -i COM1 -b 115200 SDR-Cube.hex
Reading Target Device ID... Found dsPIC33FJ256GP710 <ID: 0x00ff>
Reading HexFile.
Reading Target
Programming Device .....
..... Done.
C:\>pause
Press any key to continue . . .
```

8) Hardware Architecture

Refer to the schematic diagrams. We also suggest to studying the dsPIC33FJ256GP710 dsPIC and TLV320AIC23B Codec data sheets and application notes.

9) Software Architecture

Flasher

SDR Cube flasher program is based on Microchip application note AN1097. The code is adapted to the SDR Cube hardware. Special Linker scripts are required for the Flasher and SDR Cube project to be located correctly in the memory space. Flasher code starts at 0x400, the Flash memory address. The SDR Cube application starts at 0xC00. See Memory usage and Microchip AN1097 for the details.

SDR Cube application (v1.00)

The SDR Cube application is run at 38.7072 MHz execution rate (FCY). The codec sample rate is set to 8 kHz (125us). All DSP calculations are accomplished within the 125 us period defined by the Codec interrupt structure. Time critical parts of the 125 us IRQ software are coded in Microchip dsPIC assembler language. Other time critical tasks like the keyer code are executed within a 1 ms real-time interrupt tick. General transceiver control is accomplished in the free running Main loop. The Main loop contains the user interface functions, frequency tuning RX/TX control etc. The band scope FFT

is also calculated in the Main loop when enough baseband data samples are available (every 16ms). If new functions are added, care must be taken not to slow Main loop too much. RX/TX control is also time critical task.

Additional information of the software architecture can be found from the SDR Cube source code and its comments.

10) Memory Usage

0x0157FE 0x00AED4	Free space v1.00 sw
0x000c02	SDR Cube code
0x000C00	Flasher delay parameter
0x00090A	Not used
0x000400	Flasher code
0x000200	IRQ vectors
0x000000	Reset vector

Transceiver configuration data

Transceiver configuration, setup and calibration data is stored in the external 256 kB 24LC256 serial EEPROM memory. Configuration data is reset to the “factory defaults” upon user request or if a new software version contains new configuration parameters.

SDR Cube v1.00 software uses about 27% of the dsPIC RAM and about 24% of the dsPIC Flash memory. There is plenty of room for future features and improvements.