



[www.ccsinfo.com](http://www.ccsinfo.com)  
262-522-6500

# <Bits & Bytes> Newsletter

## INSIDE THIS ISSUE:

Pg 1  
LOAD-n-GO Handheld  
Programmer

Pg 2  
Debugging With Two  
Processors

Pg 3  
Debugging Support for Dual  
Core for dsPIC33CH Family

Pg 4  
Debug Features You May Not  
About

# Product Spotlight



## LOAD-N-GO HANDHELD PROGRAMMER

**LOAD-n-GO is a hand-held mobile programmer and goes where no PC or laptop has gone before. This device allows for quick field programming of targets with up to four firmware images. The programming process is as easy as 1-2-3 with the LOAD-n-GO. Simply load the program into the LOAD-n-GO via a USB cable, connect to the target and program! LOAD-n-GO debug support covers all targets that have debug mode when used in conjunction with the C-Aware IDE Compilers.**

[support@ccsinfo.com](mailto:support@ccsinfo.com)

[sales@ccsinfo.com](mailto:sales@ccsinfo.com)

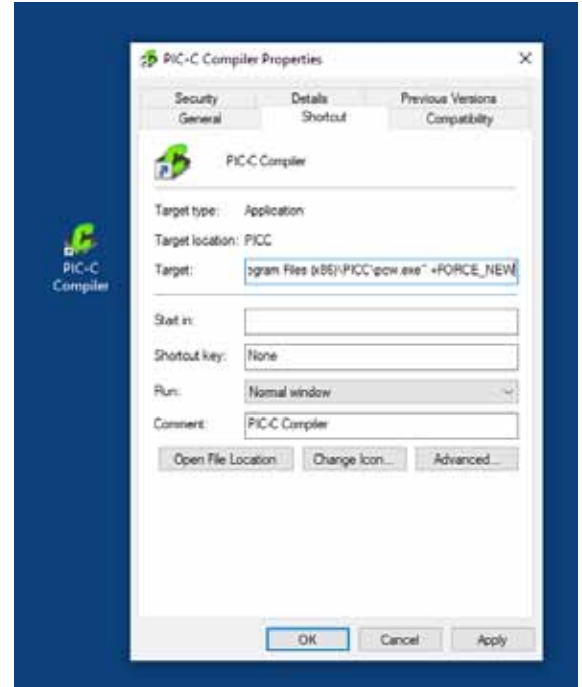
# Debugging With Two Processors

Sometimes there is a need to debug multiple processors in the same system. It is possible if you use the CCS ICD units to use multiple ICD units on the same PC. This allows you program or debug different target devices in separate processors concurrently.

By default the PCW IDE will only open one instance and additional files you attempt to edit appear as separate tabs. Although the IDE can only manage one project at a time it is possible to have multiple instances of the IDE open at the same time.

Procedure:

- Connect two ICD units and target devices.
- Right click on the desktop icon for the IDE and select PROPERTIES.
- At the end of the target line, after the " if there is one, add a space followed by:  
+FORCE\_NEW
- Start up the IDE by double clicking on the icon and load the first project and start debugging. With both ICD units connected you will be prompted (by serial number) for the ICD unit to use. You can use the CCSLOAD utility to give each unit a friendly name. You can also go the "Configure Port" button the control panel to switch ICD units.
- Double click again on the IDE icon and this time select the second project. You can start that debugger as well.



## CCS<sup>INC</sup> COMPILER FEATURE FOCUS



### #serialize

can be used to add serial number instructions into the hex file

```
const int32 SerialNum = 0;
#serialize(idserialNum, prompt="Enter the serial number")
...
printf("S/N: %LU \r\n",SerialNum);
```

The above code, when used with a CCS device programmer, will prompt the user for a serial number each time a chip is programmed. That serial number will be programmed into flash and can be accessed like any other constant in the program.

#serialize can instead used serial numbers saved in a file or simply increment the serial number with each chip programmed. It also can be set up to keep a log file of each serial number used.

The format can be binary, ASCII and it can be saved in Flash or EEPROM.

# Debugging Support for Dual Core dsPIC33CH Family

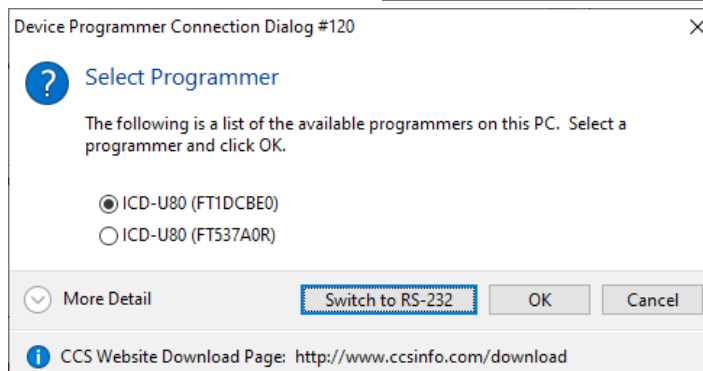
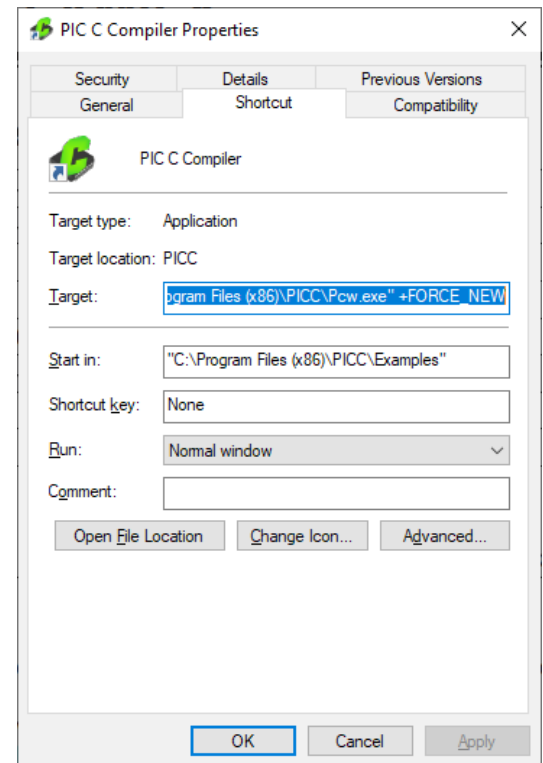
Introducing support added for debugging dsPIC33CH devices for both the master and slave cores to the CCS C Compiler and IDE using the CCS ICD-U80 or ICD-U64 device programmers. This support will be available starting with Compiler version 5.094 and ICD firmware version 3.42 and newer. In addition to debugging the slave core with the CCS tools the slave core can also be programmed starting with the previously mentioned version. Both of these features will aid in developing code for the dsPIC33CH dual core devices.

When debugging there are three setups that can be done. First debug only the master core, second debug only the slave core, or third debug both the master and slave core at the same time. When debugging the slave core the master core's configuration bit need to be set so that the slave core can be debugged. At minimum the S1\_ISOLAT, S1\_DEBUG and S1\_ICSPx, x being the debug pin to use for the slave core, should be set. The S1\_ISOLAT configuration fuse allows the slave core to operate even if the master core hasn't set the SLVEN bit in the MS1CON register, enabling the slave core to run, the S1\_DEBUG configuration fuse enables the slave core debugger and the S1\_ICSPx configuration fuses sets which S1MCLRx, S1PGCx and S1PGDx pins are being used to debug the slave core.

When debugging both the master core and slave core at the same time it requires that two instances of the CCS C Compiler and IDE be running, two device programmers, and the development board will have to have two ICD connectors on it connected to different MCLR, PGC and PGD pins, one for the master core and one for the slave core. To run two instances of the CCS C Compiler and IDE +FORCE\_NEW needs added to the target line of the icon used to launch the IDE, for example:

With the above change when using that icon to start the IDE will cause a new instance of the IDE to start allowing for more the one instance to be running. Once there are two instances of the IDE running the master core project should be opened in one and the slave core project should be opened in the other. Next starting with the master core project, build it with the appropriate slave core configuration fuses set and start the debugger. Assuming both ICDs are connected to the PC a selection box similar to the following should pop up:

Select the ICD that is connected to the master core's programming/debugging pins and select 'OK'. If you're not sure which ICD is connected to the master core you may want to disconnect the ICD connect to the slave core debugging pins from the PC and then start the master core's debugger, then when the master core's debugger is loaded the slave core ICD can be reconnected to the PC. Once the master code program debugger





is loaded, switch to the CCS IDE instance with the slave core project, build it and start its debugger. Once it's done loading both the master and slave core can be debugged at the same time.

One thing to be aware of the master code debugger using the MCLR pin of the PIC. If the master core's debugger pulls that pin low to reset the master core it will also reset the slave core. So if the master core debugger is reset or the program if reloaded when the slave core is also being debugged, the slave core's debugger needs to be reset to synchronize it with the slave core.



**CCS C Compiler Savings!**

**\$25 Off a Full Compiler or Compiler Maintenance**

**Use Code: Winter2022**

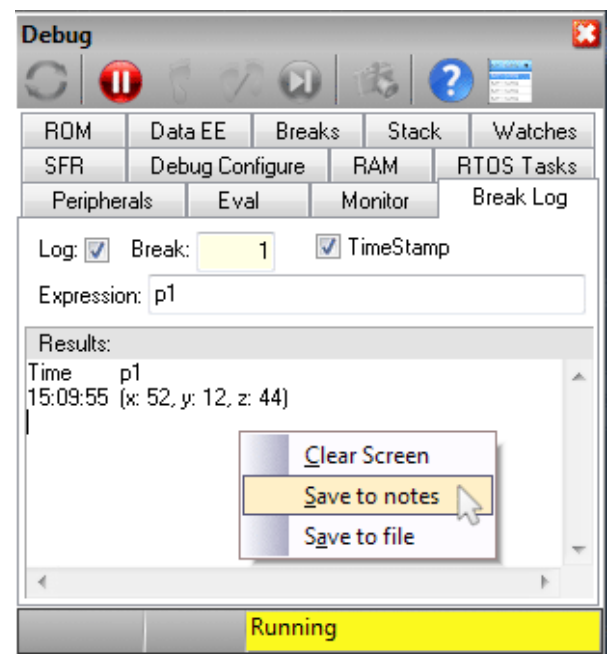
PIC<sup>®</sup> MCU  
**C COMPILER**

The image is a promotional banner for the CCS C Compiler. It features a background of a field of colorful tulips under a blue sky with clouds. On the left, a yellow circle contains the text "\$25 Off a Full Compiler or Compiler Maintenance". In the center, a purple rounded rectangle contains the text "Use Code: Winter2022". On the right, there is a CD-ROM with the text "PIC<sup>®</sup> MCU C COMPILER" and a small image of a computer monitor and keyboard.

## Debugger Features You Might Not Know About

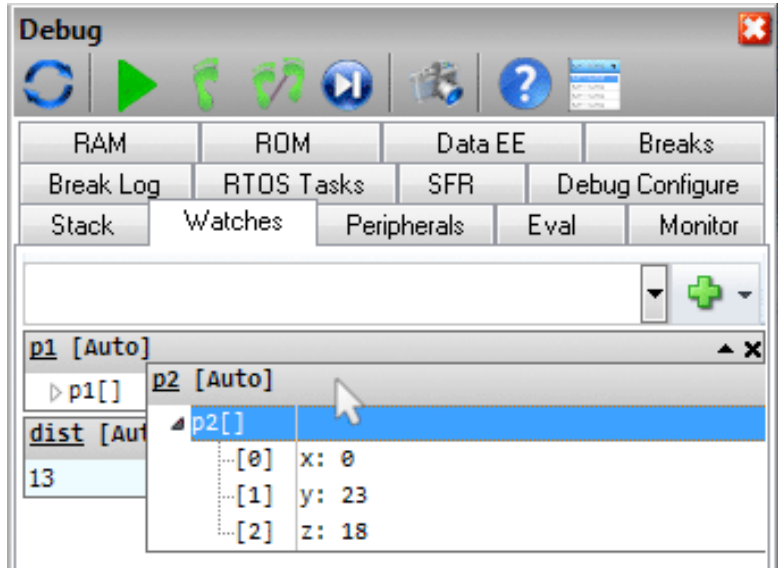
### Save to notes

The option to "Save to notes" is in the right-click menu of multiple debug windows which will append the selected data (such as break log or RAM data) to the end of the notes file associated to the project.



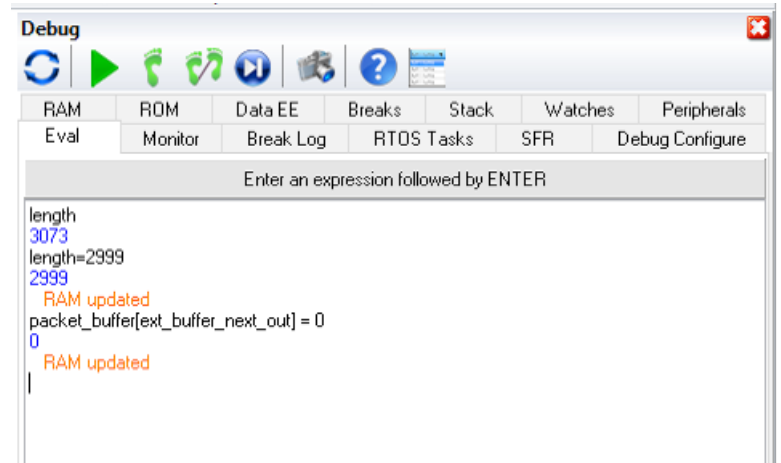
### Drag and drop “watch items”

Have an important watch variable that is in the middle or at the bottom of your watch list? You can now drag and drop watch items within the watch window to reorder them by your preference.



### Change RAM

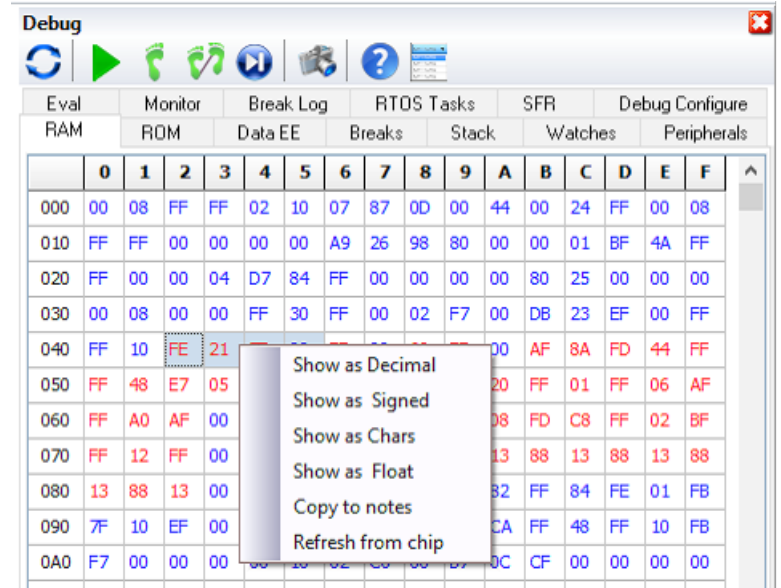
If during debugging you want to change the value of a variable or just see the result of a complex expression use the EVAL tab in the debugger. If you use the = operator then RAM is actually changed.



### Viewing RAM

Looking directly at the RAM data is easy with the RAM tab in the debugger. You might not know if you highlight some data you can use right click to interpret that data in various formats or just copy the data to your notes file.

There are a lot of helpful features in the debugger, many can be explored by reviewing the various tabs and doing some right clicking. If you have suggestions for more features please let us know.



# 8-Bit AVR<sup>®</sup> Support for Programmers



Programming support for all 8-bit AVR<sup>®</sup> microcontrollers. LOAD-n-GO, Prime8 and ICD-U80 supported. Programming adapter and cables available as separate purchase.

**8-bit AVR<sup>®</sup>  
Programming Adapter**  
53505-1867 | \$25.00

sales@ccsinfo.com  
262-522-6500 EXT 35  
[www.ccsinfo.com/NL0422](http://www.ccsinfo.com/NL0422)



More than 25 years experience in software, firmware and hardware design and over 500 custom embedded C design projects using a Microchip PIC<sup>®</sup> MCU device. We are a recognized Microchip Third-Party Partner.



[www.ccsinfo.com](http://www.ccsinfo.com)

**Follow Us!**



*AVR<sup>®</sup> is a registered trademark of Microchip Technology Inc.*