Sept / Oct 2021

CCS Inc

<BITS & BYTES> Newsletter

www.ccsinfo.com
262-522-6500

Product Spotlight

INSIDE THIS ISSUE:

EMBEDDED ETHERNET DEVELOPMENT KIT

The Embedded Ethernet Development Kit supports development with Microchip's ENC28J60 ethernet transceiver. This kit includes the powerful PCWH Integrated Development Environment with compiler support for Microchip's PIC® PIC10, PIC12, PIC16 and PIC18 families and an ICD-U64 in-circuit programmer/debugger that supports C-aware real time debugging. The prototyping board features a PIC18F4620 attached to a potentiometer, RS-232 level converter, pushbutton and three LEDs.

support@ccsinfo.com sales@ccsinfo.com

Hex File Names

By: CCS Staff

Do you maybe have a lot of projects named “main”? How about one source base that through conditional compilation could produce any one of six hex files? Maybe you just need to know exactly what version the hex file for a project is from? This article describes the technique to get specifically named hex files, different from the project name.

The fundamental principle is to use #export to define the name of the hex file. The simple syntax is:

```
#export(file=”myproject.hex”, hex)
```

You need to give it the filename and the format (hex). This alone can help a lot for many situations.

The next example shows how to include a version in the hex file name:

```
#define FW_VER_MAJOR 1
#define FW_VER_MINOR 14

#define strzz(x) #x
#define strz(x) strzz(x)

#export(file=”myproject_”          \
         strz(FW_VER_MAJOR) “.” strz(FW_VER_MINOR) “.hex” , hex)
```

The strz/strzz macros are used to convert the text define to a quoted string. In C, consecutive strings are appended together as if there was a single long string. The resulting filename for this example will be:

```
myproject_1.14.hex
```

Finally consider adding in the product name, that may be different based on conditional compilation. We will also show how to identify a release with diagnostics.

```
#if PRODUCT==PROD_WIZBANG
     #define APP_NAME   “WIZBANG”
#elif PRODUCT==PROD_WIZBANG_PLUS
    #define APP_NAME   “WIZBANG_PLUS”
#elif PRODUCT==PROD_SUPERBANG
    #define APP_NAME   “SUPER-BANG”
#else
     #error Unknown Product
#endif

#ifdef DEBUG
     #define HEX_TAIL  “_diagnostic.hex”
#else
    #define HEX_TAIL  “.hex”
#endif

#export(file=APP_NAME “_”          \
           strz(FW_VER_MAJOR) “.” strz(FW_VER_MINOR) HEX_TAIL, hex)
```

If you use the CCS IDE for development then the IDE will know the correct hex file name from the last build. Debugging and programming will work as expected.

Sometimes the filename is not enough. The compiler also allows you to put the same kind of information inside the hex file. For example:

```
#hexcomment\ APP_NAME _ FW_VER_MAJOR . FW_VER_MINOR
```

This can be done in addition to or instead of the custom hex filename. The directive puts a comment in the hex file that is ignored by the device programmer. The \ after hexcomment tells it to put the comment at the end of the file.

Without the \ the comment is put at the top of the file. The CCS device programmers will pop up any comments at the top of the file before programming. For example:

```
#hexcomment NOTICE: This file is only to be used for APP_NAME
```

This simple technique can be a powerful tool to properly identify your hex files.

Project Time Tracking

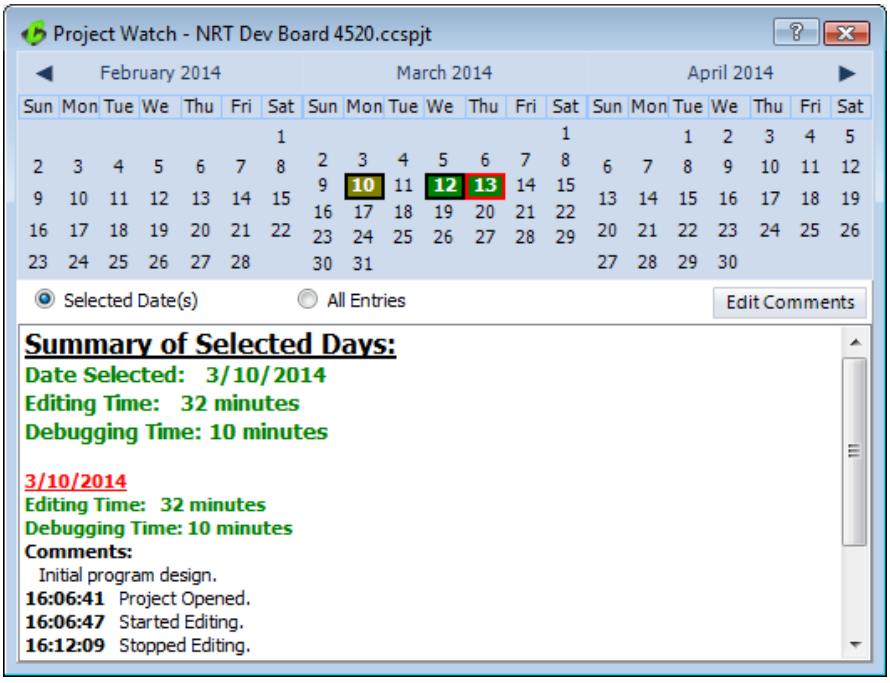
By: CCS Staff

Ever wonder how much time it took to code and debug a project? The Project Watch utility within the CCS C-Aware IDE makes keeping track of project development easier than ever!

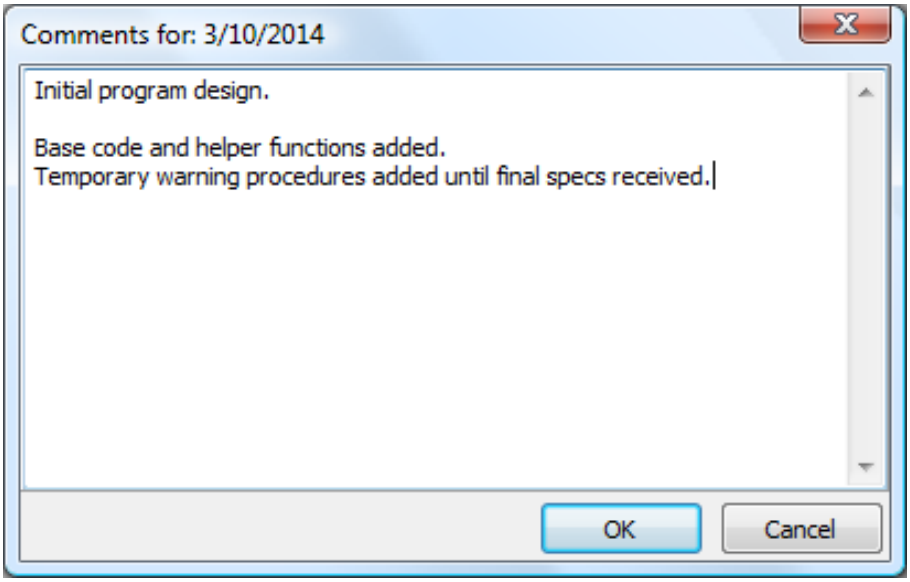
The Project Watch tool provides the user detailed log information on the current working project. Time-management can be viewed through either a 'Daily Activity' log, including total editing and debugging time, or 'Specific Activity' log containing time the specific activity occurred.

Long periods of time where the keyboard is not used are excluded from the times.

View logs, contained in the Project Watch tool by selecting between two view modes: 'Selected Date(s)' or 'All Entries'. The 'Selected Date(s)' view shows the log information for the currently selected day(s) on the calendar. The 'All Entries' view shows the entire log that has been collected for the project, including a summary of the log and detailed information separated by each day.



Additionally, comments can be added or modified to a Project Watch log from a pop-up menu by right-clicking on any date that contains log information. Any changes to the comments will be saved to the project log and displayed in the log information window.



The Project Watch feature is available under the 'View' ribbon in the IDE. 

Signature Headaches

By: Mark Siegesmund

Back in 1995 the most popular web browser company in the world, Netscape, decided to use asymmetric encryption to allow people to safely enter credit card numbers in the web browser. Mathematicians started talking about asymmetric encryption in the late 70's. There are two keys, one to encrypt and a different one to decrypt. It is very difficult to figure out one key unless one has the other. This way a website can send an encrypt key to the browser and the browser can encrypt the data to be sent. Even if someone is able to see the encrypted data and the encrypt key, the data can not be decrypted without the decrypt key. Part of the magic involves very large prime numbers and complex equations.

That is not all the browser did for SSL (Secure Socket Layer, AKA https). This also created the idea where each SSL website would have a certificate issued by a trusted authority. That certificate indicates the web site domain is encrypted by the authority. The browser can decrypt the certificate to learn the verified name of the company that is running the web site. In this case, it is the encrypt key that remains secret. The idea was to prevent a user from being spoofed into entering their credit card on a web site that appears to be a well known site, but is not.

It is that second use that caught the eye of Microsoft in the XP days. They decided to use the certificate concept to add a signature to executable files loaded by Windows. It works like this: A software publisher gets a certificate from a trusted authority, the same people doing web site certificates. Using a tool from Microsoft, a signature using the certificate with the company name and a CRC of the executable file is appended to the file. When the OS loads the file, it can decrypt the signature, verify the CRC and decide how trusted the file is. For example if a message like "This program was published by CCS, Inc; if you trust that company press continue."

The only encryption algorithm used up to 2001 was called SHA1. In 2001 a new algorithm called SHA2 (AKA SHA256) came out that was considered harder to break with the newest fast computers. XP SP3 would accept either encryption but

the older XP versions would only take SHA1. One could sign a file with two signatures, SHA1 and SHA256 and it would all work, so this became the norm.

Starting with 64 bit Windows 7 Microsoft began requiring all drivers be signed with a certificate traceable to an authority they approved of. The way these certificates work is company A can issue a certificate to you and they have a certificate issued by company B and they have one issued by company C. This chain of certificates is all part of the signature. Microsoft maintains a list of the large handful of top level certificate issuers it trusts. Drivers were considered sensitive because they often operate in kernel mode where they can access any memory in the PC.

With all the websites in the world a lot of companies were issuing a lot of certificates and some of these authorities were not doing much to check for who they were issuing them to. The concept of an Extended Validation certificate (EV) was created where it requires a great deal of verification to ensure the certificate holder is who they say they are. Microsoft uses this in their SmartScreen web browser download checker. It flags any software without a EV certificate as suspicious.

This was all well and good until the release of Windows 10 that first was sent out October of 2020. Microsoft decided drivers loaded into Windows would only be accepted if they were signed by Microsoft. The way this worked is the software publisher would first sign the driver with an EV certificate, then send it to Microsoft. If approved, the publisher signature was stripped and a Microsoft signature applied. They would then send the package back to the publisher. So the whole world did not grind to a halt, they have exceptions as a part of a roll out plan. In general, right now anything signed in 2021 needs a Microsoft signature and most older files are accepted.

So here come the headaches for CCS. First I should point out even though we have a yearly developer subscription with Microsoft we did not get the memo about this new rule. In the old days Microsoft published a newspaper sent via snail mail to all developers. The last one we got was October of 2000. For a while we got news by CD and then that was replaced with a slew of online blogs.

Microsoft does not update all machines at the same time, so early this year we started getting some calls of trouble but not enough to be very concerned. There seemed to be work-arounds that got people going. Only new drivers had a problem and only on machines that did not have the driver previously installed. The error did not say “Sorry this driver must be signed by Microsoft”, it did say “Invalid signature hash.”

We put in a support ticket with Microsoft in May and were then told about the new rules. Our first step was getting an EV certificate. We had been using the ordinary certificates (OV) since the beginning but never bothered with the EV. It then took over two months and a pile of tickets in attempts to log into the Microsoft driver signing website. The problem turned out to be another account with the same company name was taken out 10 years ago and not used since. The error was not “Company name already used”, it was “Something went wrong, try again later.” The next problem was our files were not being accepted. Another dozen tickets and we found out the reason our certificate was not working was the authorities started issuing SHA3 certificates and the Microsoft website could not handle those yet. Windows is OK with SHA3 but not the website that accepts the driver submittals. After making a special request for an SHA2 certificate and another couple of weeks to figure out what the special website wanted, we started getting Microsoft signed drivers.

So far we have not been able to figure out how to append additional signatures on to the files so it seems we need to have new and old style drivers depending on the OS.

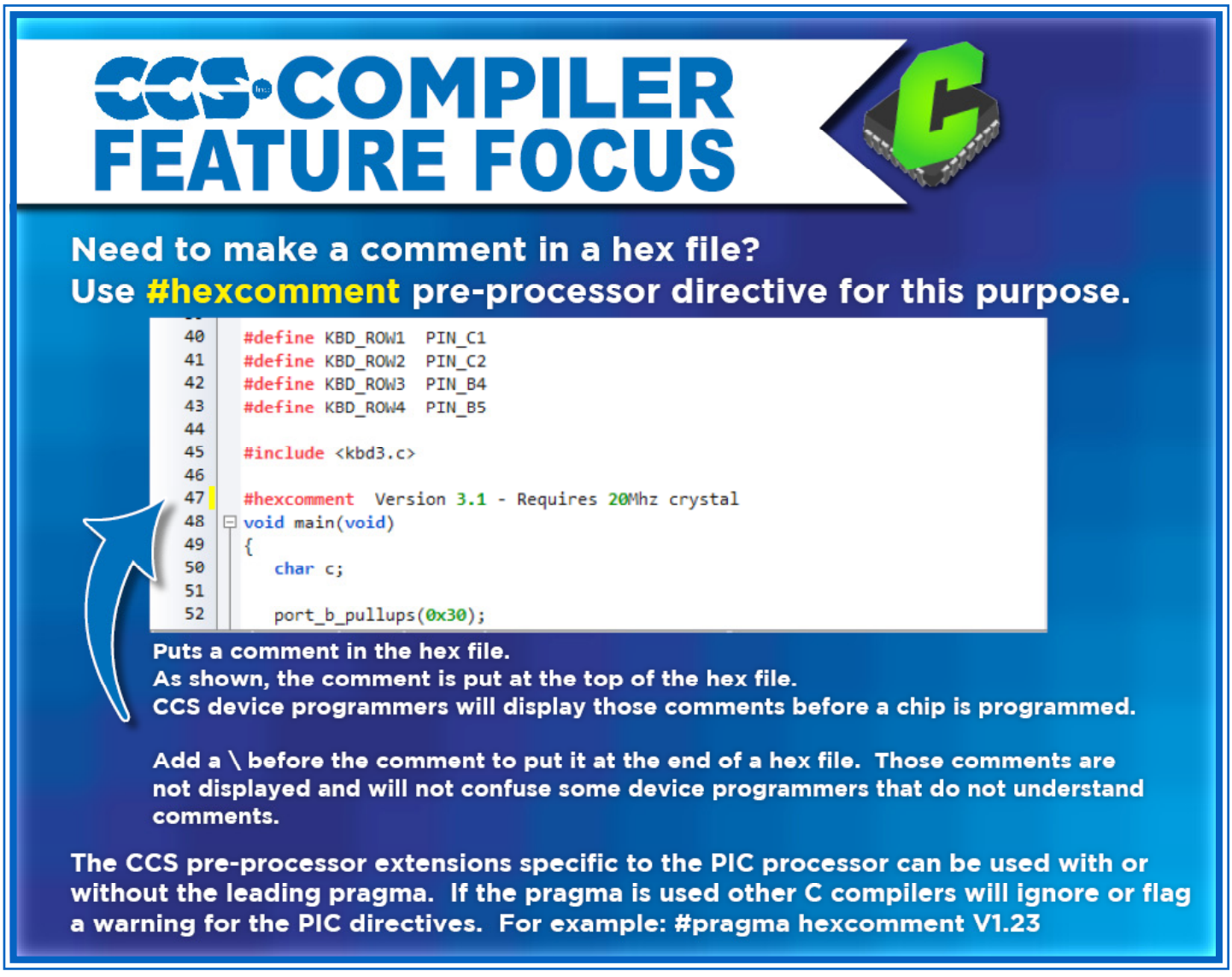
Many of our customers are using USB CDC drivers to talk to the PIC® MCU. This creates a COM port in Windows that can use the normal serial API functions. The driver for a CDC device is just a .inf file with VID, PID and company information, plus the .cat file with the signature. In Windows 10 no driver is required for CDC devices. You do need it for older versions of Windows. If you do want to use a .inf for a CDC device in Windows 10 then it does need the Microsoft signature even though the driver file is optional.

It is possible to load a driver into Windows that has a bad or wrong signature. This link details the procedure:
https://www.howtogeek.com/167723/how-to-disable-driver-signature-verification-on-64-bit-windows-8.1-so-that-you-can-install-unsigned-drivers/

The procedure will not work if you have secure boot enabled and on some PC's you need a password to disable secure boot. This also will not work on PC's that are in S-Mode. Only software products in the Microsoft store can be installed in S-Mode. This might be a hint that new rules for normal executable are coming.

Microsoft is evolving with encrypted certificates as are web browsers. If you ever try powering up a old PC you may find it can no longer browse the web because it uses an older encryption method and most websites are now the newest https. Asymmetric encryption technology can be used to send encrypted data to anyone you have not previously made secure contact with. It is finding use in a large variety of applications. One big use, that would not otherwise be possible, is cryptocurrency.

We are sorry for the inconvenience to customers who were trying to use our products during this adventure (nightmare). Many customers were able to use the above workaround and others returned product out of fear of damaging their system. We are now back to normal and everyone should be able to install our drivers.



CAN Bus Development Kit

Includes

- CAN Bus Development Board
- In-Circuit Debugger/ Programmer
- Exercise Tutorial
- 9V AC Adapters and Cables

sales@ccsinfo.com
262-522-6500 EXT 35
www.ccsinfo.com/NL521

CCS Inc

C Compiler Savings

$25 Off a Full Compiler or Compiler Maintenance

Use Code:Fall2021

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

www.ccsinfo.com

CCS Inc

Follow Us!