

# Neue Tools von Microchip!

## PICkit 5 und MPLAB ICD 5 jetzt erhältlich

Von Tam Hanna, für Microchip

Im Gegensatz zu früher benutzt kaum ein Elektroniker mehr Eigenbau-Brenner zur Programmierung von PIC-Controllern. Mit dem PICkit steht ein kostengünstiges Tool vom Hersteller zur Verfügung, das überdies mit vielen weiteren Controllerfamilien kompatibel ist. Für die professionelle Fehlersuche steht überdies der mächtige ICD-Debugger zur Verfügung. Vor wenigen Wochen hat Microchip beide Geräte nutzwertgesteigert - in diesem Artikel wird ein wenig mit den neuen Funktionen experimentiert und gezeigt, wie sich die neuen Versionen in schon vorhandene Infrastruktur einbinden.

### Jetzt mit USB-C!

Microchip ist seit jeher bemüht, den „Kampf um das Kabel“ in Häusern von MPLAB-Entwicklern so weit wie möglich zu entschärfen. Das ICD 3 nutzte einen Stecker vom Typ USB-A, während das PICkit 4 einen Micro-USB-Stecker mitbrachte.

Die Standardisierung im Bereich der USB-Stecker führt nun dazu, dass sowohl PICkit 5 [1] als auch ICD 5 [2] einen der in so gut wie allen modernen Mobiltelefonen verbreiteten USB-C-Stecker mitbringen. Für den Besitzer der Geräte ergeben sich daraus mehrere Vorteile. Erstens ist es fortan problemlos möglich, die von Telefon und Co. sowieso mitgeführten Kabel „zu recyceln“ - dies spart auf Geschäftsreisen Platz und mitunter Geld, weil man USB-C-Kabel ob der weiten Verbreitung im Smartphone-Bereich so gut wie überall ausborgen kann. Neuerung Nummer zwei betrifft insbesondere Besitzer der „größeren“ Programmiergeräte-Variante ICD 5. Da mehr Energie verfügbar ist, ist es dem ICD 5 möglich, der Applikationsschaltung bis zu 1 A zur Verfügung zu stellen. Auf diese Art und Weise erspart man sich in vielen Fällen die Mitnahme eines zweiten Netzgeräts, das die Energie-Versorgung des zu programmierenden DUTs sicherstellt. Das kleinere Gerät ist dabei nach wie vor auf rund 150 mA Strombudget beschränkt - wohl ein Entgegenkommen an die wesentlich kleinere Hardware.

Im Bereich der eigentlichen Programmiergeschwindigkeit gibt es durch USB-C übrigens keine nennenswerten Vorteile - die beiden Programmierer arbeiten nach wie vor „nur“ mit USB 2.0 High Speed, was in der Praxis aber wegen der kleinen Dateigröße der meisten

Mikrocontroller-Images nicht ins Gewicht fallen dürfte. Optimierungen im Bereich der verbauten Hardware erhöhen allerdings die „gefühlte Geschwindigkeit“.

### Mit der Macht des Power over Ethernet!

Insbesondere in Hochspannungs-Systemen ist „galvanische Trennung“ zwischen dem Programmier-PC und dem Device unter Test wünschenswert - fragen Sie den Autor nicht, wie man flucht, wenn eine damals teure AMD-K6-Workstation vom DUT „gegrillt“ wird. Während USB und Co. grundlegenden Schutz versprechen, ist richtige galvanische Isolation (und räumliche Trennung) über ein Ethernet-Netzwerk der wahrscheinlich beste Weg.

Besitzer des ICD5 finden auf der Rückseite den in **Bild 1** gezeigten Port. Neu an ihm ist nun, dass er auf Wunsch auch über Power over Ethernet die „Gesamtapplikation“ versorgt - je nach Konfiguration Ihres Power over Ethernet-Netzwerks kann die Applikationsschaltung dabei mit durchaus erheblichen Strömen versorgt werden.

Freundlicherweise ist MPLAB X auch zur direkten Ansprache des ICD 5 per Ethernet bereit; es ist also nicht notwendig, zusätzlich ein USB-C-Kabel zwischen PC und Programmiergerät zu verdrahten.

Dass diese Konfiguration auch zu Experimenten mit Continuous Integration beziehungsweise Continuous Delivery einlädt, sei angemerkt - allerdings bieten „ältere“ Varianten des Programmiergeräts im Zusammenspiel mit einem Ethernet-Anschluss und einem 9-V-Steckernetzteil ähnlichen Arbeitskomfort.

Das Steckernetzteil ist in Version 5 übrigens nicht mehr mit von der Partie. Dies weist darauf hin, dass die Stromversorgung (logischerweise) alternativ auch über das erhöhte Strombudget von USB C beziehungsweise USB 3.0-Ports erfolgen kann.

### PICkit 5: Im Dienste der Bequemlichkeit...

Die eigentliche Entwicklung eines elektronischen Systems ist in nur allzu vielen Fällen nur der erste Teil einer umfangreichen Wertschöpfungskette. Wer nicht von Anfang an auch an „Design for Manufactu-



Bild 1. Dieser Port ermöglicht das Anschließen per Power over Ethernet.

Figure 2-1. System Power Supply Control Using the PIC16F15244 Family of Microcontrollers

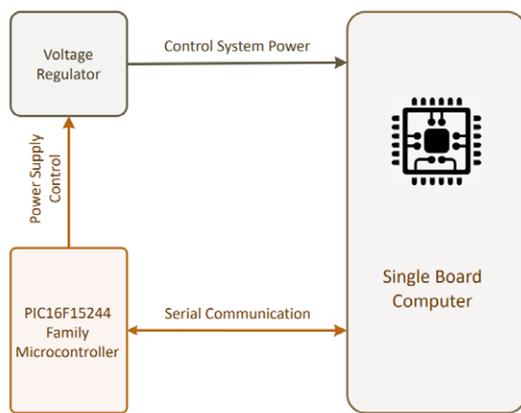


Bild 2. Auch der Management-Mikrocontroller benötigt Firmware!  
(Bildquelle: [3])

ring“ denkt, hat im Rahmen der Fertigung oft unerwartet hohe Kosten und / oder Ärger.

Ein gutes Beispiel hierfür wären Systeme, die nach dem vom Autor in der Application Note AN4121 [3] beschriebenen und in **Bild 2** zusammengefassten Schema einen „Management-Mikrocontroller“ mit einem Prozessor kombinieren. Wer hier nicht mitdenkt und beispielsweise den Programmierport für den Controller schlecht platziert, erhöht die Fertigungskosten immens.

Während Entwickler nicht damit überfordert sind, die Firmware auf jede einzelne Platine unter Nutzung von MPLAB auszuliefern, ist diese Vorgehensweise spätestens ab einer gewissen Menge von produzierten Geräten unproduktiv - die traurige praktische Erfahrung des Autors lehrt, dass „Hilfswillige“ oder die berühmt-berüchtigte Hilfskraft ohne nennenswerte Kenntnisse mit der Bedienung einer IDE wie MPLAB und Co überfordert sind und im besten Fall Fehler macht (und im schlimmsten Fall die Arbeit aufgibt).

Mit dem PICKit 5 und seiner Funktion *Programmer to Go* schafft Microchip hier Abhilfe. Im Prinzip ist diese Funktion dabei schon seit einigen Versionen mit von der Partie - Geheimnis ist der in der **Bild 3** gezeigte MicroSD-Slot, der eine FAT32-formatierte Speicherkarte aufnimmt. Unter Nutzung von MPLAB kann der Entwickler dann ein automatisiert einzusetzendes Firmware-Image festlegen, das der Programmier-Adapter dann ohne große menschliche Intervention auf angeschlossene Zielgeräte ausliefert.

Neu ist beim PICKit 5 die Möglichkeit, mehr als ein Firmware-Image in den *Programmer To Go*-Speicherbereich zu kopieren. Ein Programmiergerät lässt sich so zur „Parametrisierung beziehungsweise Endfertigung“ mehrerer Assemblies einspannen. Da die eigentliche Auslösung des Programmiervorgangs nach wie vor über den bekannten und unter dem Logo versteckten Taster erfolgt, muss die Auswahl des Images über ein anderes Kommunikations-Interface erfolgen.

### ...und der Freunde des Bluetooth-Funkstandards

Zur Inbetriebnahme der neuen Programmier-Geräte ist MPLAB in einer Version besser als 6.10 erforderlich - die eigentliche Installation erfolgt so, wie es von MPLAB erwartet wird. Neben der bei Microchip traditionell separat zu erfolgenden Installation des XC8-Compilers ist darauf zu achten, dass Sie MPLAB im Rahmen des ersten Starts in der Windows-Firewall sowohl den Zugriff auf lokale als auch auf entfernte Netzwerke erlauben müssen. Dies ist insbesondere dann wichtig, wenn Sie die größere Version des Programmiergerät unter Nutzung seiner Ethernet-Schnittstelle in Betrieb nehmen wollen.

Im nächsten Schritt ist die Provisionierung einer SD-Karte erforder-



Bild 3. Speicherkarten-Slot.

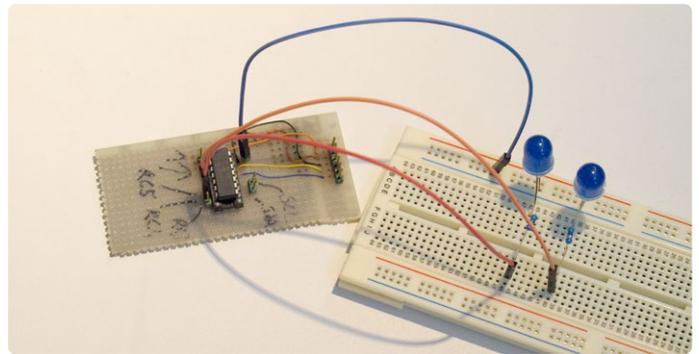


Bild 4. Dieses Board dient als Testsystem.

lich. Der Autor erbeutete ein 16 GB großes Medium aus einem herumliegenden Raspberry Pi und formatierte es unter Ubuntu mit einer FAT32-Partition. Achten Sie dabei auf eine Eigenheit des Disks-Snapins - von Haus aus erzeugt es keine Partition, weshalb eine Extra-Einladung erforderlich ist.

Als Opfer soll der jungfräulichen MPLAB-Installation ein auf dem PIC16F1503 basierendes Board dienen, das der Autor aus einem vergangenen Automotive-Consultingprojekt übrig hatte (siehe **Bild 4**). Entscheiden Sie sich im ersten Schritt wie bekannt für die Option *File* → *New Project*, um danach ein Projekt für den XC8-Compiler zu generieren. Im nächsten Schritt folgt ein Start von MCC, um ein grundlegendes Projektskelett zu generieren. Der Autor geht in den folgenden Schritten von soliden Grundkenntnissen im Bereich der PIC-Programmierung aus, weshalb das allgemeine Handling nicht weiter besprochen wird.

### Windows 7 ist unerwünscht!

Zu beachten ist, dass Microchip explizit davon abrät, die fünfte Version der Programmiergeräte mit Windows 7 zu nutzen - der Autor verwendete deshalb ausschließlich Windows 10 und 11. Im Internet finden sich allerdings einige Berichte, die im Allgemeinen davon berichten, dass es „funktioniert“ - zumindest dann, wenn das Programmiergerät regelmäßig Verbindung mit einem Rechner mit Windows 10 oder Windows 11 aufnehmen kann, wo MPLAB im Rahmen einer testweisen Auslieferung eines Kompilats verschiedene Firmware-Upgrades und Housekeeping-Tasks ausführt.

Wichtig ist lediglich, dass zwei der GPIO-Pins als Ausgang deklariert sind. Der Autor wird in den folgenden Schritten die Pins RC4 und RC5 als Ausgänge exponieren, weil sie in der Vergangenheit in einem Kundenprojekt zum Einsatz kamen. Achten Sie danach jedenfalls darauf, den Code zu generieren.

Zwecks einfacherer Unterscheidbarkeit wird der Autor dem hier als *RunMeQuick1* bezeichneten Beispiel-Projektskelett ein nach folgendem Schema aufgebautes Programm einschreiben, dass die beiden

Leuchtdioden langsam, aber gleichzeitig zum Aufleuchten bringen wird:

```
void main(void)
{
    // initialize the device
    SYSTEM_Initialize();

    // When using interrupts, you need to
    // set the Global and Peripheral Interrupt Enable bits
    // Use the following macros to:

    // Enable the Global Interrupts
    //INTERRUPT_GlobalInterruptEnable();

    . . .
    IO_RC4_SetHigh() ;
    IO_RC5_SetHigh() ;
    while (1)
    {
        IO_RC4_Toggle() ;
        IO_RC5_Toggle() ;
        __delay_ms(1000);
    }
}
```

Im nächsten Schritt öffnen wir die *Project Properties*-Ansicht, wo Sie - wie in **Bild 5** gezeigt - die Einstellungen für die Einrichtung des *Programmer To Go*-Betriebs vorfinden.

Am Wichtigsten ist dabei die Rubrik *Image Name* - dabei handelt es sich um den Namen, den MPLAB der herausgeschriebenen Datei zuweist. Die Option *Send image to tool* weist die IDE dann dazu an, die MicroSD-Speicherkarte des angeschlossenen Programmiergeräts mit dem generierten Image zu beladen.

Das Markieren der Checkbox *Program Device* ist derzeit optional - es entscheidet, ob neben der eigentlichen *Programmer to Go*-Funktion auch ein angeschlossener PIC-Mikrocontroller mit dem bereitgestellten Image zu beschreiben ist.

Die Nutzung dieser Option kann beispielsweise dann sinnvoll sein, wenn sie in einem Aufwasch sowohl das *Programmer to Go*-Image aktualisieren möchten als auch ein (eventuelles) Evaluationsboard für einen letzten Rauch-Test mit aktueller Firmware beladen wollen. Die eigentliche Auslieferung muss dann unbedingt über das in **Bild 6** gezeigte Menü erfolgen, das eine Aktualisierung des *Program-*

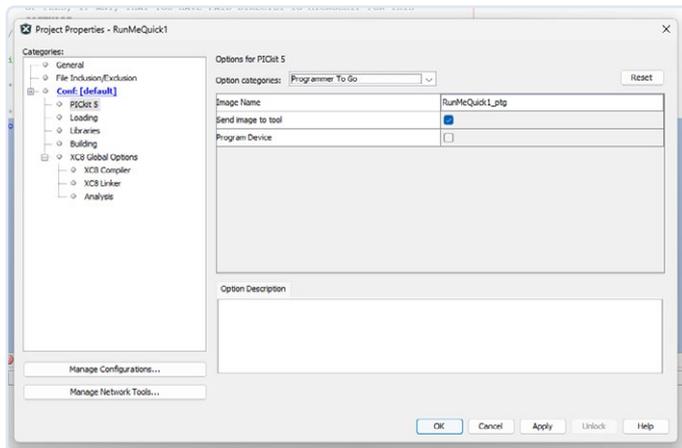


Bild 5. MPLAB unterstützt Entwickler bei der Konfiguration des *Programmer to Go*-Modus.

*mer to Go*-Images befiehlt.

Während der Auslieferung kommt es in manchen Fällen zu einem nach Schema *Transmission on endpoint 2 failed (err = -109)* aufgebauten Fehler. In Tests des Autors ließ sich dieser so gut wie immer durch einen Neustart des PCs beheben. Wer sein MPLAB frisch installiert, dem wird an dieser Stelle übrigens ebenfalls zu einem Reboot geraten. Das Deployment ist abgeschlossen, wenn in der Statuskonsole die Meldung *The debug tool is in programmer to go mode* erscheint.

Zur Illustration der Wertsteigerung der zweiten Generation des PICKit 5 kehren wir im nächsten Schritt in die MPLAB-Startpage zurück, wo wir ein weiteres, im Allgemeinen identisches Projekt mit dem Namen *RunMeQuick2* erzeugen. Auch hier gilt, dass ein Start von MCC erforderlich ist. Der für die beiden Signal-LEDs vorgesehene Code sieht nun allerdings folgendermaßen aus:

```
void main(void)
{
    . . .

    IO_RC4_SetLow() ;
    IO_RC5_SetHigh() ;
    while (1)
    {
        IO_RC4_Toggle() ;
        IO_RC5_Toggle() ;
        __delay_ms(1000);
    }
}
```

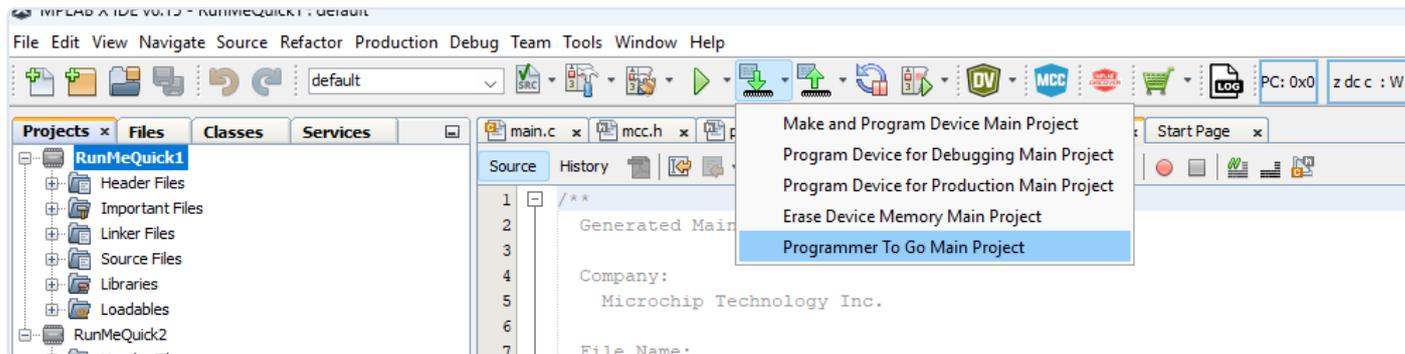


Bild 6. Mit diesem Eintrag wandert das Image auf die SD-Karte.

Zur Parametrisierung des angeschlossenen PICKit ist dann – logischerweise – eine abermalige Ausführung erforderlich, die ebenfalls mit der Ausgabe der Statusmeldung *The debug tool is in programmer to go mode* abgeschlossen werden muss. Im nächsten Schritt können Sie den Programmer auch schon von der Workstation trennen. Um das Ganze zu testen, wird der Autor sein PICKit danach mit einem gewöhnlichen Handy-Ladegerät verbinden, das von Experimenten mit einem Chromebook übrig geblieben ist. Die fröhlich hellgrün blinkende Leuchtdiode weist darauf hin, dass das Gerät im *Programmer to Go*-Modus auf Handlungsanweisungen wartet.

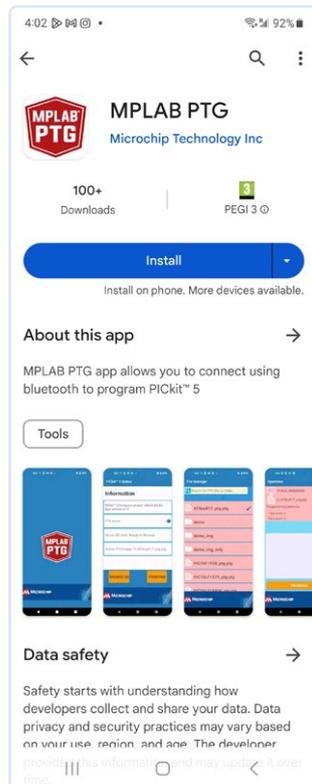


Bild 7. Diese Applikation erleichtert die Remote-Programmierung.

Der nächste Akt ist das Öffnen des Play- oder Apple-App-Stores, um die neue und nur mit dem Bluetooth-Modul des PICKit 5 kompatible Steuerungsapplikation herunterzuladen. Der Autor wird in den folgenden Schritten mit seinem Samsung Z Flip arbeiten; die Applikation präsentiert sich im Play Store wie in **Bild 7** gezeigt.

Wegen der Nutzung von Bluetooth LE gilt dabei, dass das Programm im Rahmen des ersten Starts verschiedene Permissions einfordern wird, was aufgrund von Googles Parametern erforderlich und in keiner Weise kritisch oder umgehbar ist.

Im nächsten Schritt erscheint auch schon der Scanning-Dialog, der alle in der Umgebung gefundenen Geräte auflistet. In Tests des Autors war es manches Mal erforderlich, auf den *Cancel*-Knopf zu klicken, bevor die Liste mit den gefundenen Programmiergeräten anklickbar war.

Nach getaner Arbeit präsentiert das System jedenfalls die in **Bild 8** und **Bild 9** gezeigten Bildschirme, die den direkten Einsatz der jeweils gewünschten Firmware auf angeschlossene Zielgeräte erlauben.

Die Handhabung dieser Telefon-Applikation erweist sich nach Ansicht des Autors für Hilfskräfte leichter erlernbar als die Arbeit mit dem oft haarigen *Project Explorer* von MPLAB. Außerdem gilt, dass ein Telefon fast überall zu haben ist und die sonst erforderliche (vergleichsweise leistungsstarke) Workstation entfällt – insbesondere in großen Deployments reduziert dies die Kosten erheblich.

## Blick zurück im Dank

Als nächstes Experiment folgt die Rückkehr auf die Windows-10-Workstation, mit der der Autor unter anderem das im Kasten befindliche Lehrbuch realisiert hat. Von Haus aus findet sich dort MPLAB in Version 5.45 – der Autor hat diese Version im kommerziellen Einsatz. Nun bietet sich deshalb der abermalige Besuch der MPLAB-Webseite an, wo wir nun jedoch Version 6.15 herunterladen. Das Installieren der neuen Variante von MPLAB erfolgt dann direkt über die schon vorhandene.

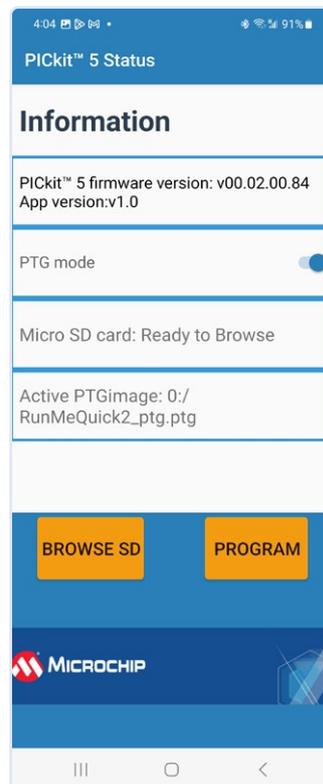


Bild 8. Das Startmenü...

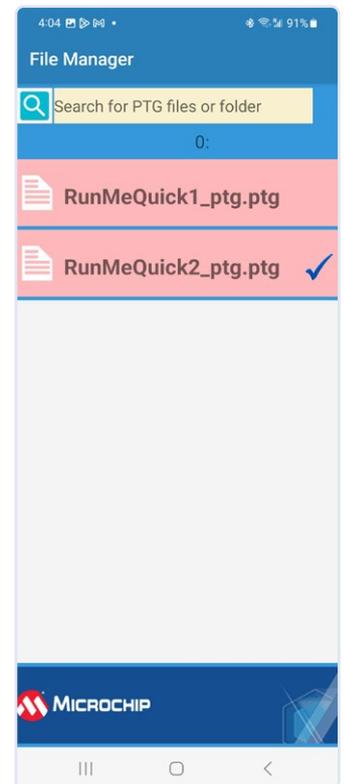


Bild 9. ...und die Auswahl des zu deployenden Images.

Nach dem erfolgreichen Durchlaufen des Installationsprozesses bot MPLAB 6.15 an, die Einstellungen der Vorgängerversion als Basis für die Parametrisierung der frischen Installation heranzuziehen. Danach war wie immer das Abnicken der diversen Einstellungen der Windows-Firewall erforderlich, die eigentliche IDE startete dann nach dem Herunterladen der *Microchip Offline Help*-Komponente.

Im Rahmen des erstmaligen Starts der IDE erfolgte ein umfangreicher Parsing-Prozess, der unter anderem alle Projektdateien auf den aktuellsten Stand zu bringen suchte. Darüber hinaus wurden die diversen in MPLAB X angelegten Caches aktualisiert, will sagen, mit den in den Projekten befindlichen Informationen bevölkert.

Im nächsten Schritt entschied sich der Autor für das Demo-Beispiel *CH9-Demo1* und verband das aus seinem Lehrbuch bekannte Board mit dem Programmiergerät und dem PC. Die Kompilation verlief erfolgreich; die eigentliche Firmware-Auslieferung scheiterte unter Windows 10 manchmal mit dem von weiter oben bekannten Fehler *Transmission on endpoint 2 failed (err = -109)*. Unterm Strich gilt allerdings, dass die neue Variante des PICKit 5 sich nahtlos in einen schon vorhandenen Entwicklungs-Workflow integriert – dank der Optimierungen erfolgt das Deployment des Codes in vielen Fällen schneller.

## Strom messen mit dem ICD5

Microchip erweitert die hauseigene MPLAB-IDE seit einiger Zeit um verschiedene Komfortfunktionen, die Entwicklern unter anderem die Visualisierung von in der Embedded-Applikation anfallenden Mess- und Tracking-Daten erleichtern soll.

Eines der interessanten Features des ICD 5 ist, dass es diese Daten-Visualisierungsengine von MPLAB bei Bedarf mit Informationen über den Stromverbrauch der angeschlossenen Applikationsschaltung ausstatten kann. Im Datenblatt verspricht man dabei eine Genauigkeit von 0,29 µA als Step – weitere Daten zur Genauigkeit des Powermonitors präsentieren sich wie in **Tabelle 1** gezeigt.

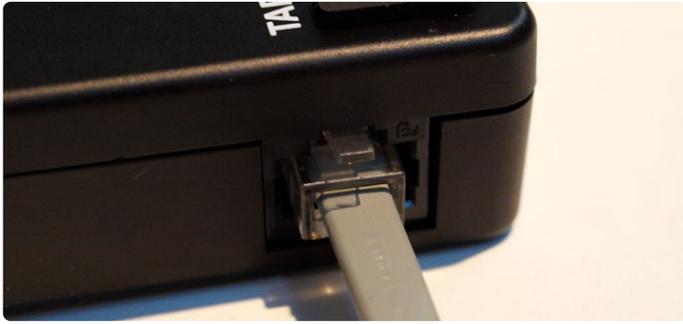


Bild 10. Ein kleines Ärgernis bei Nutzung des ICD 5.

**Tabelle 1. Genauigkeit des Powermonitors laut Hersteller.**

Current and Voltage	Resolution	Full Scale
Current	29 µA / step	1,0 A
Voltage	0,2087 mV / step	6,8 V

Klicken Sie im nächsten Schritt auf *Tools* → *Plug-in Downloads*, und entscheiden Sie sich im daraufhin erscheinenden Fenster für die Option *Microchip Plug-in Manager*. Prüfen Sie sodann, ob der *MPLAB Data Visualizer* schon in der Rubrik *Installed* aufscheint. Normalerweise ist dies bei frischen Installationen der Fall.

Logischerweise müssen Sie die Stromversorgung der Applikationsschaltungen unter Nutzung der im ICD 5 integrierten Energiequelle durchführen. Haarig erweist sich dabei übrigens vor allem der verwendete Stecker: Der Port des ICD 5 ist ein RJ45, während ältere Adapter wie der ICD 3 ein RJ11-Kabel aufweisen. **Bild 10** zeigt, dass der Verbinder zwar im Prinzip passt, aber eine komfortable und sichere Verbindung sieht anders aus.

Die weiter oben für die Arbeit mit dem PICkit realisierten LED-Beispiele taugen naturgemäß auch als Test für den Strom-Messer. Stecken Sie das PICkit ab und verbinden sie das ICD 5, und führen Sie testweise eine abermalige Auslieferung des Programms durch. Der Autor wird in den folgenden Schritten im Interesse der Bequemlichkeit auf ein USB-C-Kabel setzen und die TCP/IP-Features brach liegen lassen. Im Rahmen der erstmaligen Inbetriebnahme eines ICD muss MPLAB dabei übrigens ein einmaliges Update der Firmware des FPGAs durchführen, was etwas Zeit in Anspruch nimmt.

Interessant ist, dass dieser Prozess oft bei 93% stehen bleibt und danach „wie gewohnt“ weiterläuft (**Bild 11**). Warum MPLAB hier nicht das Erreichen eines 100%igen Aktualisierungsstands ausgibt, ist dem Autor nicht ganz klar.

Klicken Sie im nächsten Schritt auf die Option *Window* → *Debugging* → *Data Visualizer*, um den *Data Visualize*-Begrüßungsbildschirm zu starten. Klicken Sie danach in der Rubrik *Power* auf das *Play*-Symbol, um die Datenerfassung loszutreten.

MPLAB reagiert darauf mit der Einblendung des in **Bild 12** gezeigten Fensters, das auf das Fehlen einer festgelegten Visualisierung-Form hinweist.

Das Anklicken der *Plot Raw*-Funktion führt dann dazu, dass ein Diagramm aufscheint. Wer das mit beiden Leuchtdioden alternierend blinkende Programm zur Ausführung bringt, sieht dann, wie in **Bild 13** gezeigt, Informationen über den Stromverbrauch.

Interessantere Ergebnisse lassen sich unter Nutzung von *RunMeQuick1* ertrotzen; dieses Beispiel schaltet beide Leuchtdioden ein und aus (**Bild 14**).

## Fazit

Mit PICkit 5 und ICD 5 führt Microchip verschiedene Detail-Verbesserungen durch, die Entwicklern sowohl beim Debugging als auch bei der Serien-Fertigung unter die Arme greifen. Nach Ansicht des Autors sind die Geräte - die OEMSecrets-Bestpreise betragen zum Zeitpunkt der Drucklegung 86 € beziehungsweise 360 € - für das Gebotene in keiner Weise zu teuer; insbesondere Besitzer des PICkit 4, die sich über die alte Schnittstelle ärgern und ein Kabel weniger mitführen wollen, sollten unbedingt umsteigen. Die Stromanalyse ist ebenfalls sehr wertvoll, und erspart mitunter erhebliche Kosten für eine SMU oder GPIB-Karte. ◀

230571-02

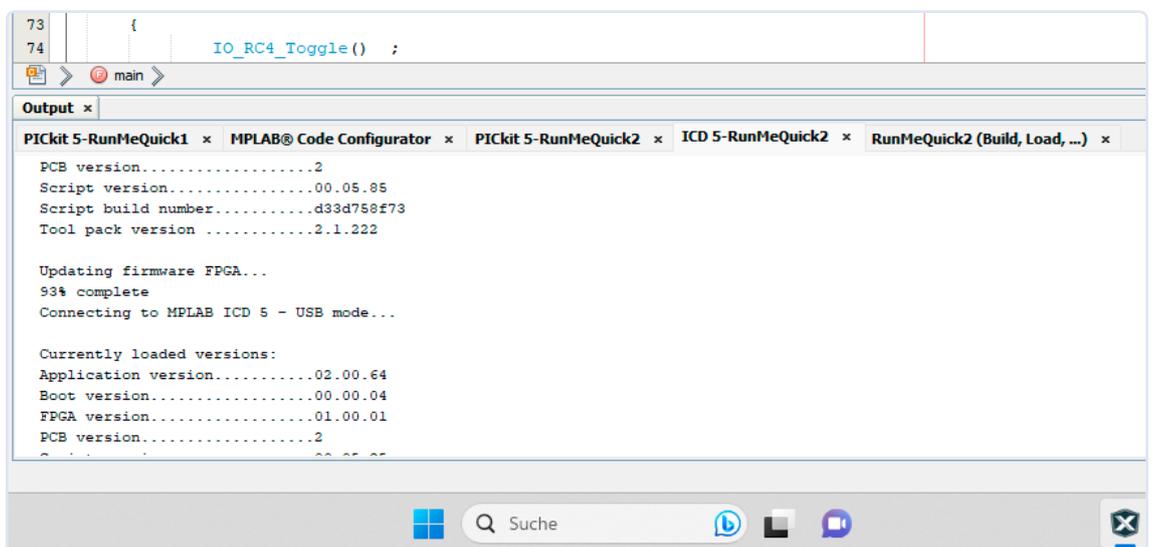


Bild 11. 93 % bedeutet hier den Abschluss der Aufgabe!

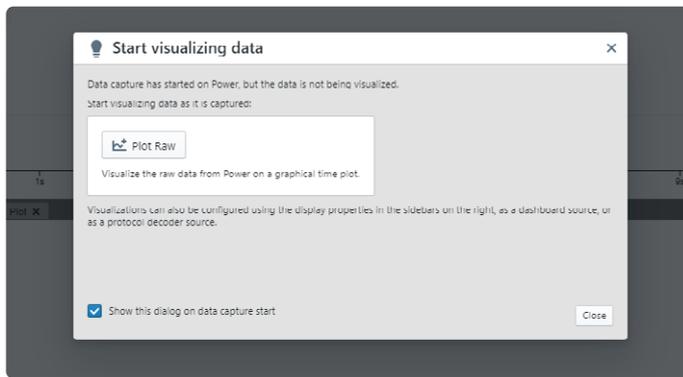


Bild 12. MPLAB bittet um die Festlegung des Visualisierung-Schemas.

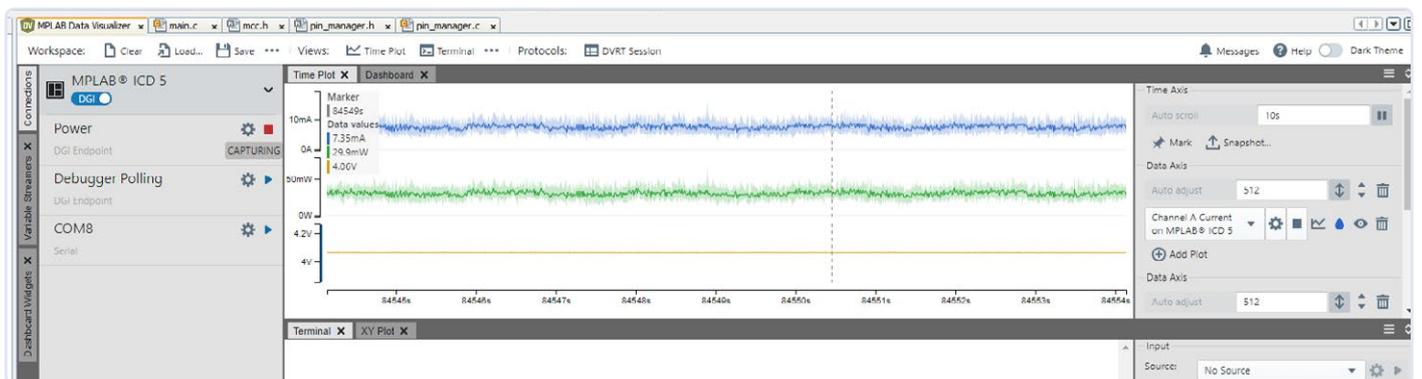


Bild 13. Es ist immer eine Leuchtdiode eingeschaltet.

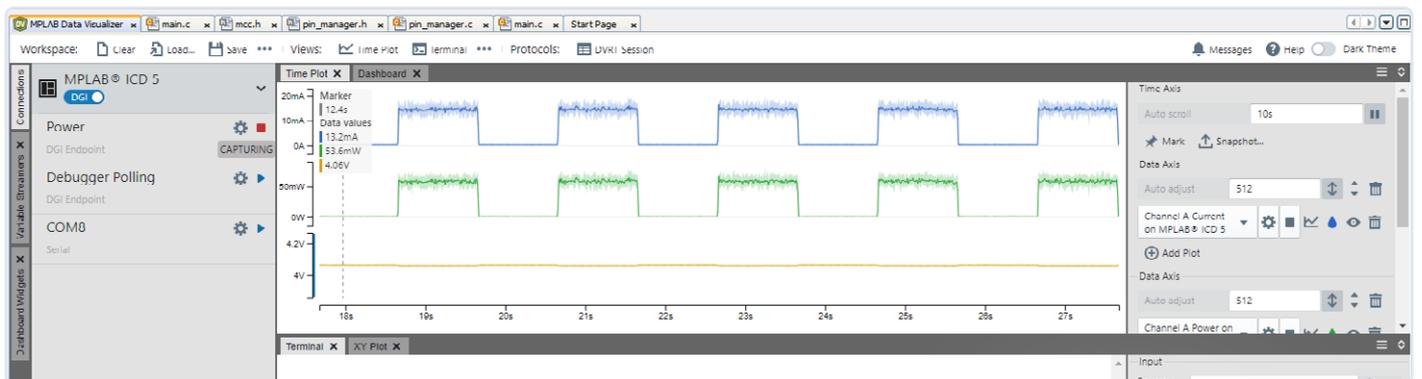


Bild 14. Der wechselnde Stromverbrauch ist klar ersichtlich.

### Fordern Sie noch heute Ihren 15%-Rabattcode für diese Tools an!

Geben Sie Ihre Kontaktdaten ein, wählen Sie das Tool aus, für das Sie einen 15%-Rabattgutschein erhalten möchten, und Sie erhalten Ihren persönlichen Gutscheincode, den Sie bei MicrochipDirect einlösen können.

[https://page.microchip.com/pic5\\_icd5](https://page.microchip.com/pic5_icd5)



### WEBLINKS

[1] MPLAB PICKit 5 von Microchip: <https://www.microchip.com/en-us/development-tool/pg164150>

[2] MPLAB ICD5 von Microchip: <https://www.microchip.com/en-us/development-tool/dv164055>

[3] Application Note AN4121: <https://www.microchip.com/en-us/application-notes/an4121>