

## Ergänzung zum Beitrag in FA 12/19, S. 1158 f. „SDR-Experimente auf LW und MW mit dem STM32F429 Discovery“

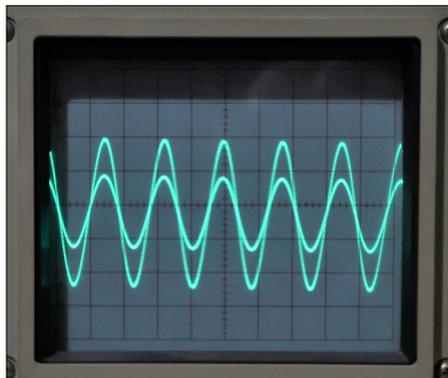


Bild 1: Signale am Eingang und Ausgang des Ausgangsfilter bei 6 kHz

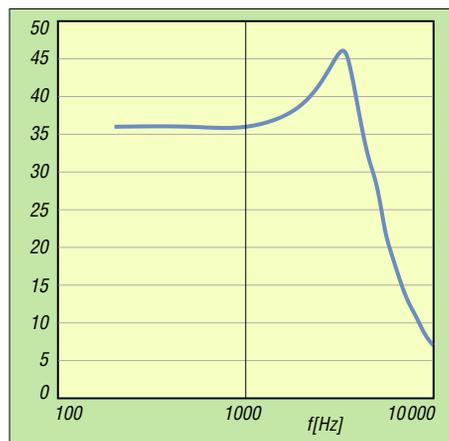


Bild 2: Charakteristik des Ausgangsfilters  
Fotos, Screenshots: PE3ES

Ergänzend zum Beitrag präsentieren wir hier noch einige Bilder, die in der gedruckten Ausgabe leider keinen Platz mehr fanden, sowie eine Anleitung zum Erstellen eigener Bin-Dateien.

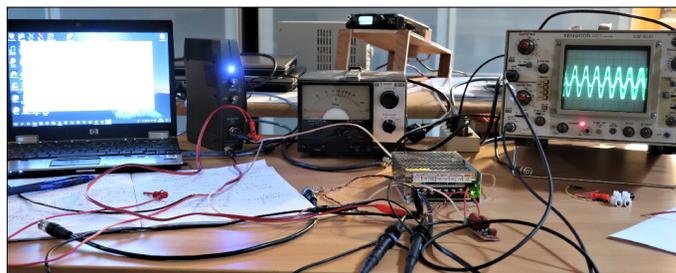
### ■ Anleitung zum Erstellen einer neuen Bin-Datei

Der wahrscheinlich aufwendigste Teil beim Aufbau eines Radios mit dem Discovery-Board ist die Software, zumindest wenn man eigene Wünsche berücksichtigen und keine fertige Bin-Datei nutzen möchte. Seit 2014 kamen diverse Versionen der Cortex-Schaltkreise heraus, die Compiler und integrierten Entwicklungsplattformen (IDE) entstanden. Außerdem haben verschiedene Anbieter neue Boards entwickelt und vermarktet. I2PHD verwendete die Version MDK5.15. Selbst, als ich Mitte 2015 meine ersten Schritte unternahm, fand ich bereits nur noch die Version MDK5.16A als kostenlosen Download.

Bild 3:  
Testgerät  
mit einem  
LTC1799 als  
Signalgenerator  
und einem  
8-stelligen  
Frequenzzähler



Bild 4:  
Aufbau während  
der Messungen  
am Ausgangsfilter



Als ich vor einigen Monaten dieses Projekt fortsetzte, gab es bereits die Version MDK5.26. Dabei werden nur die Entwicklungsplattformen betrachtet und nicht die bis dahin veränderten Bibliotheken sowie die Eigenschaften des Schaltkreises und des Boards.

Sobald Sie ein bestimmtes Programm mit einem Satz der zu diesem Zeitpunkt verfügbaren Werkzeuge zusammen haben, sind Sie zeitlich fixiert. Mit jeder neuen Bibliothek und jeder neuen Zieldatenbank müssen Sie möglicherweise Ihr ursprüngliches Design so aktualisieren, dass die neueste Version der Werkzeugpalette verwendet wird.

Sehen Sie sich im Internet nach der noch verfügbaren Versionen der benötigten Software um. Ich habe eine Liste mit sieben Punkten erstellt, die Sie in einfachen und reproduzierbaren Schritten durch diesen Prozess führt. Ich werde auch einige der Hindernisse aufzeigen, auf die Sie stoßen könnten, und wie Sie diese lösen.

Am Ende diesen Abschnitts werde ich auch Alternativen zu dieser ursprünglichen Werkzeugpalette erwähnen, die nützlich sein könnten, wenn Sie eigenen Entwicklungen auf dem Gebiet der ARM-Cortex-ICs und Entwicklungsboards beginnen möchten.

1. Suchen Sie die Teile der Werkzeugpalette zusammen und installieren Sie sie.
2. Überprüfen Sie die korrekte Installation und Position.
3. Öffnen Sie die Projektdatei und verschaffen Sie sich einen Überblick über die Organisation dieses ARM-Cortex-Projekts.

4. Ändern Sie die Projektdateipfade entsprechend Ihrer spezifischen Situation, sodass alle Bibliotheken und zusätzlichen Dateien gefunden werden.

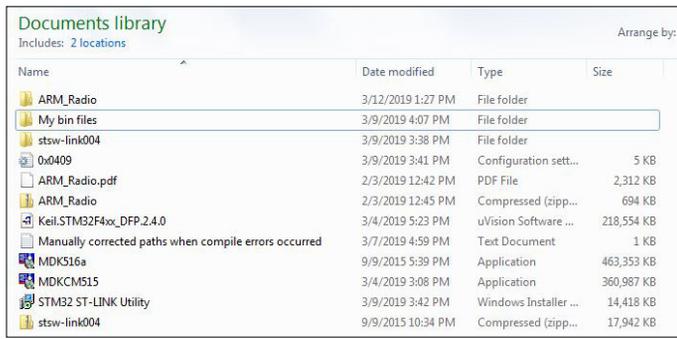
5. Ändern Sie die Datei *Presets.h* für die Sender und Frequenzen, die Sie bequem finden möchten.

6. Verwenden Sie die Testversion der *MDK Pro Middleware*, um eine neue Version der Bin-Datei erstellen zu können

7. Installieren Sie die Bin-Datei mithilfe der Werkzeugpalette.

Wir müssen die Programme so nah wie möglich an das ursprüngliche Setup bringen. Gehen Sie zur Website von KEIL ARM ([www.keil.com/mdk5/](http://www.keil.com/mdk5/)) und wählen Sie der rechten Seite des Bildschirms *MDK v4 Legacy Support* aus ([www.keil.com/mdk5/legacy/](http://www.keil.com/mdk5/legacy/)). Wählen Sie links die Version 5.16a für Cortex-M-Devices aus, wodurch der Download der Datei *MDKCM516a.exe* beginnt. Beachten Sie dabei, dass das Original-Tool MDK 5.15 oder MDK 5.16a auf der Keil-Website nicht mehr zu finden ist! Es lässt sich jedoch bei einer Websuche zum Beispiel auf [www.mcuzone.com/down/Software.asp?ID=10000543](http://www.mcuzone.com/down/Software.asp?ID=10000543) als Datei [www.mcuzone.com/software/mdk516a.rar](http://www.mcuzone.com/software/mdk516a.rar) finden. Die Zuhilfenahme eines Übersetzungstools ist angeraten, um die richtige Datei ausfindig zu machen.

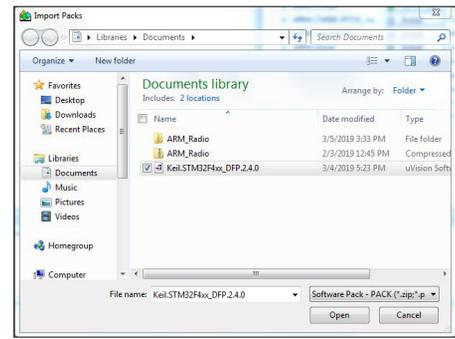
Als letztes Teil benötigen wir die ältere Version 2.4.0 des *Device Family Pack* (DFP), die auf [www.keil.com/dd2/Pack/](http://www.keil.com/dd2/Pack/) unter *ST-Microelectronics STM32F2* zu finden ist. Eine Alternative wäre die Installation des neuesten Keil-ARM-Tools, das nach einer



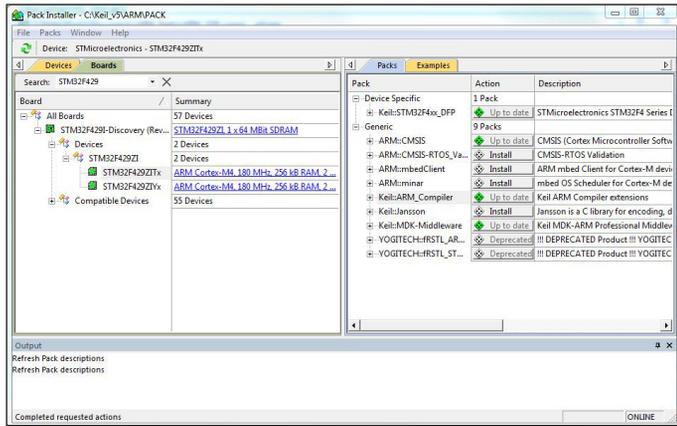
**Bild 5:**  
Ordner mit den erforderlichen Werkzeugen

Registrierung auf [www.keil.com/demo/eval/arm.htm](http://www.keil.com/demo/eval/arm.htm) erhältlich ist. Ende März 2019 waren dies die Version 5.27 und es wurden nur ältere PDF- und MDKCM-Versionen verwendet. In diesem Fall müssen Sie nichts von der asiatischen Website herunterladen. Bild 5 zeigt, wie mein Ordner aussieht, nachdem ich alle erforderlichen Teile für die Werkzeugpalette zur Softwareentwicklung heruntergeladen habe.

Installieren Sie die drei Hauptkomponenten: MDK, MDKCM und DFP (über das Pack-Installationsprogramm des ersten Tools). Die Bilder 6 und 7 zeigen diese Zwischenschritte. Der Geräte-Baum und der Board-Baum zeigen die große Anzahl verfügbarer Schaltkreise und Entwicklungsboards. Es gibt buchstäblich Hunderte von jedem. Öffnen Sie das uV5-Programm und überprüfen Sie den Info-Bildschirm.

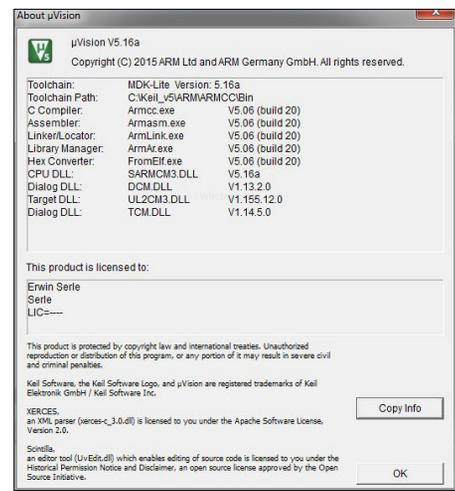


**Bild 6:**  
Import des DFP-Pakets Version 2.4.0

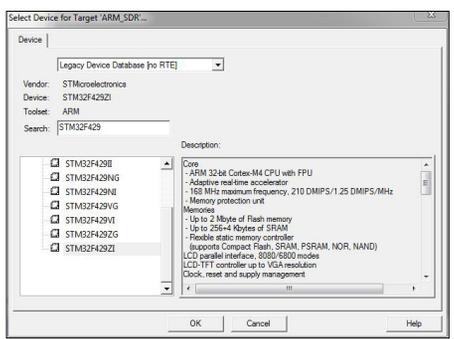


**Bild 7:**  
Das Discovery-Board ist im Keil-Compiler verfügbar.

Website auswählte, konnte ich nicht mehr in den ursprünglichen Zustand zurückkehren. In der Projektdatei selbst ist das Build-Ziel bereits registriert und es ist offensichtlich nicht notwendig, es zu verändern. Nun öffnen wir die Projektdatei und sehen, was passiert, wenn wir unseren ersten *Build* über die Taste *F7* versuchen. Bild 11 zeigt,



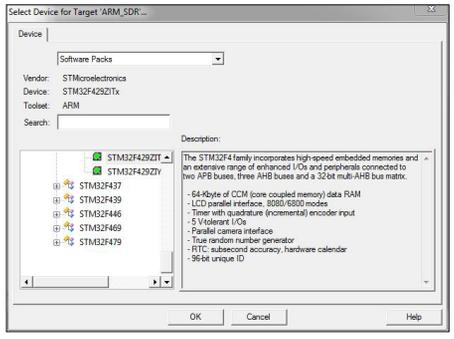
**Bild 8:**  
uVision, Hinweisfenster der Version V5.16a



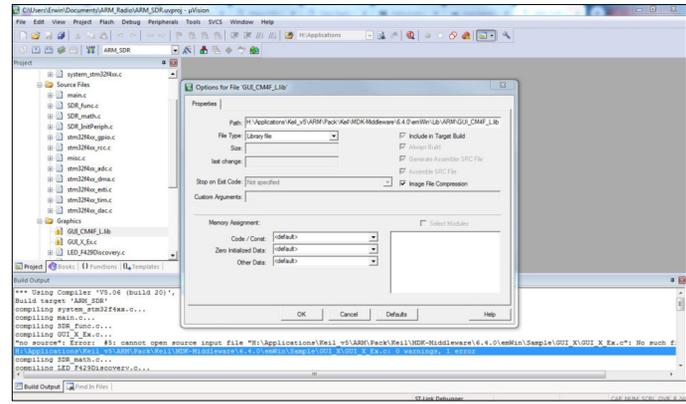
**Bild 9:** Fenster zur Auswahl des Zielgeräts unter Legacy Device Database

Er sollte dem in Bild 8 ähneln. Damit sind die ersten beiden Schritte des 7-Phasen-Ansatzes abgeschlossen. Sie können sich nun frei in den reichhaltigen Menüs und Optionen des uV5-Programms bewegen. Es gibt nur eine Warnung, die ich Ihnen mitteilen möchte. Ändern Sie nicht das Ziel für das Erstellen der Software! Die Bilder 9 und 10 zeigen Bereiche, in denen dies leicht passieren könnte. Die STM23F429-Schaltkreise und -Boards finden Sie hier. Als ich diese jedoch vor dem Öffnen der Projektdatei von I2PHDs

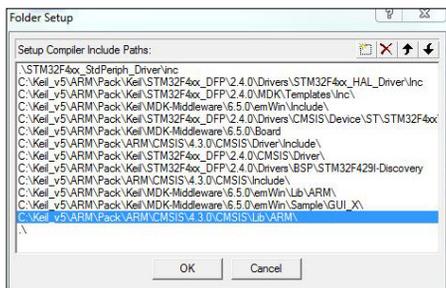
dass die Software die benötigten Bibliotheken, die wir gerade installiert haben, nicht finden kann. Das bringt uns von Schritt 3 zu Schritt 4. Wir müssen die Dateipfade aus dem ursprünglichen Projekt korrigieren, um die Speicherorte dieser Dateien auf unserem eigenen System zu finden. Dies kann über *Alt-F7*, Tab *C/C++* erfolgen. Sie müssen jede Zeile korrigieren, um auf die richtigen Ordner auf Ihrem System zu zeigen. Bild 12 zeigt das Endergebnis, das zu meinem persönlichen Setup passt. Beachten Sie, dass Sie auch eine der



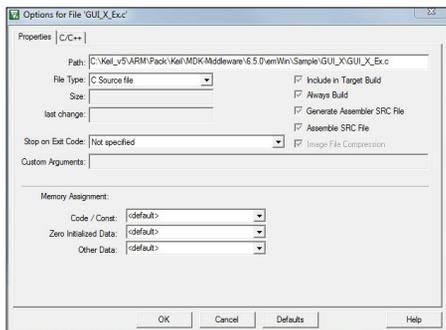
**Bild 20:** Fenster zur Auswahl des Software-Pakets



**Bild 11:**  
Fehlermeldung zu nicht korrekt zugeordneten Verzeichnissen



**Bild 12: Alle Verzeichnisbezüge wurden korrigiert und der Compiler läuft ordnungsgemäß durch.**



**Bild 13: Fehler über nicht vorhandene Archiv-Datei 2 beseitigt**

Versionsnummern ändern müssen, da die ältere Version nicht verfügbar ist: *MDK Middleware 6.5.0* in allen Zeilen mit dieser Referenz.

Bild 13 bestätigt nun, dass alle benötigten Dateien gefunden werden, wenn der *Building*-Prozess angefordert wird. Ein neuer Fehler, der häufig auftritt, ist: L9937E. Wir werden sehen, wie wir das in Schritt 6 lösen können, aber zuerst gehen wir zum fünften Schritt.

Es stehen Bin-Dateien auf Albertos Website und im Download-Bereich auf [www.funkamateur.de](http://www.funkamateur.de) zur Verfügung. Warum ist es dann noch notwendig, eine neue Bin-Dateien zu erstellen? Weil Sie vielleicht Änderungen an der Datei *Presets.h* vornehmen möchten, in der die Radiostationen und Frequenzen enthalten sind, zu denen Sie durch Drücken der blauen Taste auf der Entwicklungsplatine gelangen können.

Also Öffnen wir die Datei *Presets.h* mit einem Texteditor. Da nicht alle Texteditoren vollständig Linux-kompatibel sind (Windows und Linux behandeln CR/LF am Ende einer Zeile unterschieden), bevorzuge ich Notepad++ ([www.notepad-plus-plus.org](http://www.notepad-plus-plus.org)). Die Datei *Presets.h* lässt sich auch in der IDE bearbeiten, wenn Sie dies möchten, Bild 14.

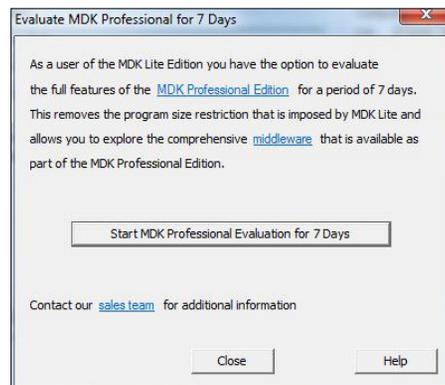
In dieser Datei werden die Voreinstellungen mit Namen, Frequenzen, Bandbreite usw. registriert. Sie können dies nach Belieben ändern und werden feststellen, dass Sie maximal 14 verschiedene Stationen eingeben können. Der Parameter *MAXPRESETS* verhindert eine größere Anzahl. Bild 15 zeigt die Datei *main.h*, in der dieser Parameter

eingestellt ist – Sie können den Wert dort selbst ändern.

Mit der aktualisierten Datei *Presets.h* möchten wir jetzt über die Taste *F7* ein neues *Building* ablaufen lassen. Daher müssen wir den lästigen L9937E-Fehler beseitigen. Es ist möglich, eine kostenlose Testversion der recht teuren *MDK Pro Middleware* zu erhalten. Bild 16 zeigt, wie. Die Testversion wird nur eine Woche nutzbar sein. Aber jetzt können wir tatsächlich einen neuen Build erstellen, siehe Bild 17, der das gute Funktionieren der gesamten Werkzeugpalette zeigt. Bild 18 zeigt, wie schnell ich eine neue Bin-Datei generieren konnte, nachdem ich nur einige Teile der Datei *Presets.h* das zweite Mal geändert hatte. Bereinigen Sie die dazwischen liegenden Build-Dateien nicht. Wenn Sie dies tun, wird der gesamte Build erneut erstellt.

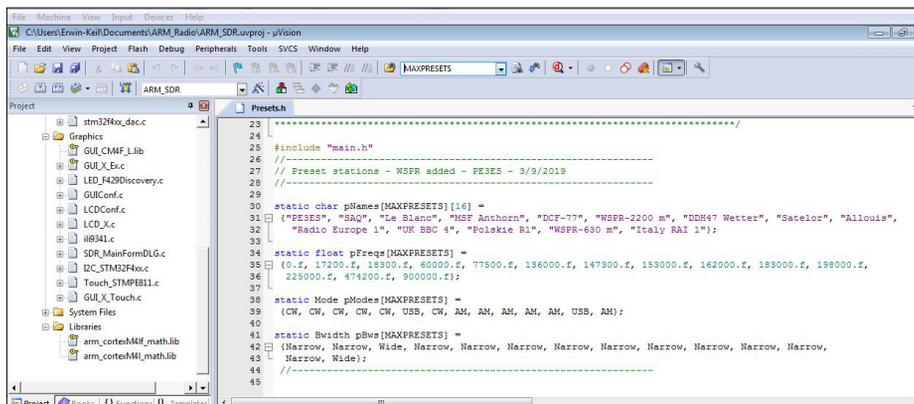
Wir gehen von Schritt 6 zu Schritt 7. Die neu generierte Bin-Datei können wir auf das STM32F429-Board laden und danach das ARM-Radio mit unserer eigenen Liste von Stationsvoreinstellungen genießen. Viel Spaß!

Alternativen für die beschriebene Werkzeugpalette sind: CubeMX (offene uVision5-Werkzeugpalette, github), Eclipse-

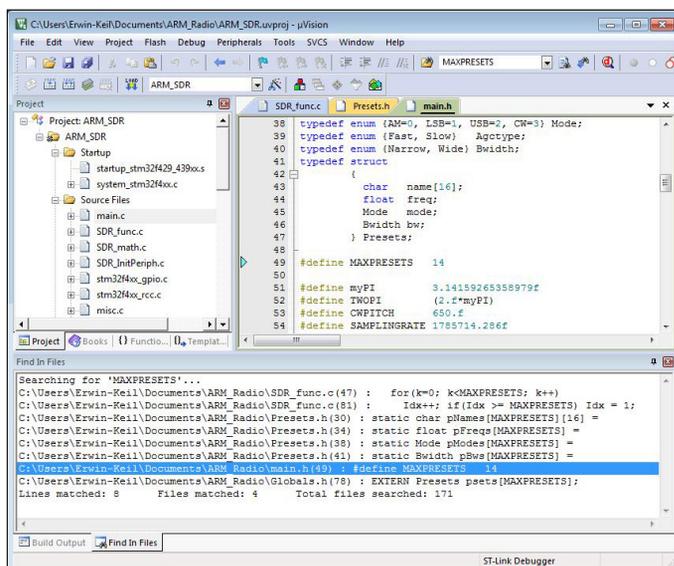


**Bild 27: Beseitigung des Fehlers L9937E für einige Tage**

CDT für GNU ARM, EWARM von IAR, Coccox. Alle sind (teilweise) Optionen. Wie Alberto mir vor einiger Zeit schrieb: „Sie können immer versuchen, den Code auf einen kostenlosen Compiler zu portieren, wie dem von STM, dem Hersteller des ARM-ICs selbst, veröffentlichten System *Workbench*. Die Informationen sind auf [www.openstm32.org/HomePage](http://www.openstm32.org/HomePage) zu finden. Oder alternativ zum kostenlosen TruStudio-Compiler von Atollic sind Informationen auf [www.atollic.com/trustudio/](http://www.atollic.com/trustudio/) verfügbar. Ich kann Ihnen jedoch nicht sagen, wieviel



**Bild 14: Datei Presets.h in uV5 geöffnet**



**Bild 15: Die maximale Anzahl von Voreinstellungen (Presets) ist in der Datei main.h definiert.**

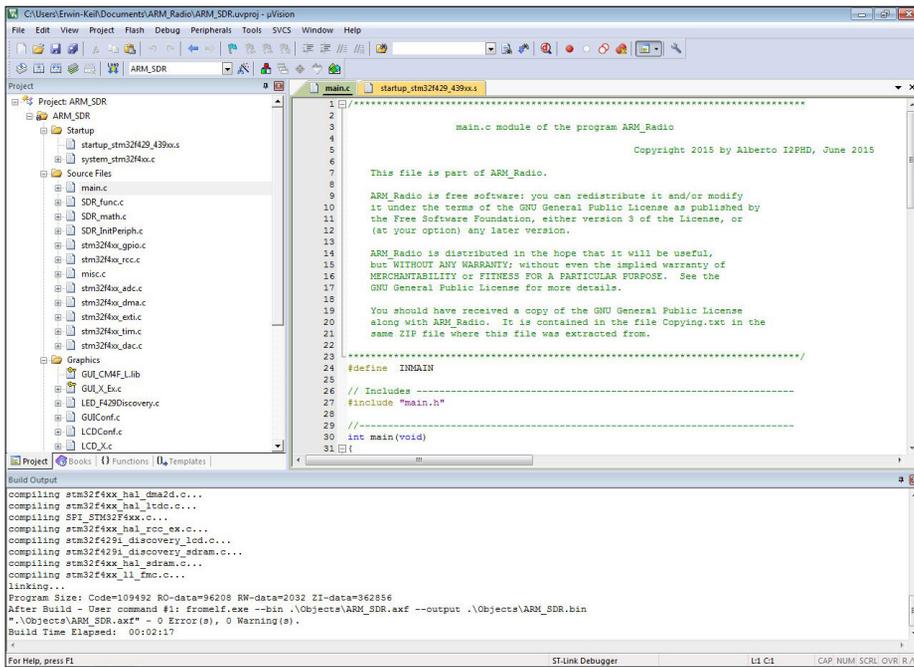


Bild 16: Nach dem erfolgreichen Ablauf des **Building**-Prozesses

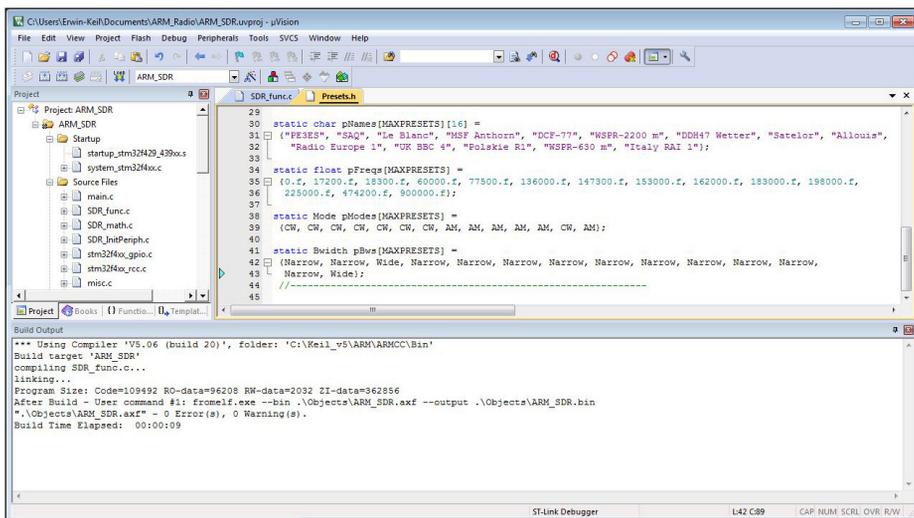


Bild 17: Wenn nur die Datei **Presets.h** geändert wurde, ist **Build** sehr effizient und schnell.

### ARM-Architektur

ARM entwirft Schaltkreise wie CPUs und SoCs. Sie stellt sie jedoch nicht her, sondern verkauft das geistige Eigentum an Produktionsorganisationen. Die ARM-Prozessoren werden beispielsweise in Smartphones verwendet, da sie sehr energieeffizient sind. Die deutsche Firma Keil stellt Software-Entwicklungstools für die ARM-Architektur her und wurde 2005 von ARM aufgekauft. Keil entwirft auch Entwicklungsboards. Einige anderer Hersteller wie das französisch-italienische Unternehmen ST Microelectronics produziert einige der ARM-Designs und stellt Entwicklungsboards für diese Technologie zur Verfügung. Obwohl diese Boards sehr gute Ei-

genschaften besitzen, können sie preislich recht günstig sein. Dies macht eine solche hochentwickelte Technologie für den Amateur- und Hobbymarkt und die Maker-Gemeinschaft zugänglich, besonders in Kombination mit den frei verfügbaren Tools, die für die Softwareentwicklung erforderlich sind, wie den Werkzeugen von Keil. Nach der Installation Ihrer eigenen Werkzeugpalette benötigen Sie die neuesten Updates. Dabei wird deutlich, wie viele verschiedene Schaltkreise und Platinenvarianten hergestellt wurden. In meinem Setup finde ich 18 Hersteller mit 2184 verschiedenen Schaltkreisen und 882 unterschiedlichen Boards. M4, ein 32-Bit-

Arbeit nötig wäre, um meinen Code in eine dieser Entwicklungsumgebungen zu portieren – das habe ich nie versucht.“ Die gleiche Bemerkung trifft für mich zu. Ich habe eine funktionierende Lösung gefunden. Da Zeit knapp ist, kann ich mich jetzt wieder auf andere interessante Projekte konzentrieren. *pe3es@veron.nl*