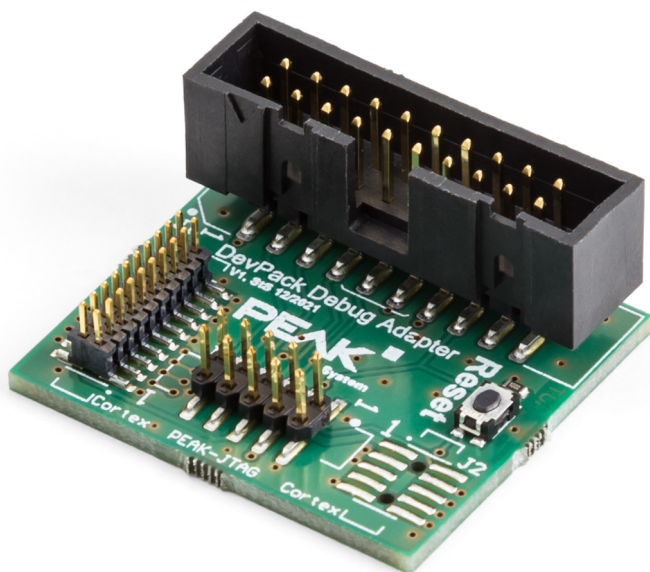


PEAK-DevPack Debug Adapter

Benutzerhandbuch



Berücksichtigtes Produkt

Produktbezeichnung	Artikelnummer
PEAK-DevPack Debug Adapter	IPEK-003026

Impressum

PCAN® ist eine eingetragene Marke der PEAK-System Technik GmbH. CiA® ist eine eingetragene EU-Marke des CAN in Automation e.V.

Andere Produktnamen in diesem Dokument können Marken oder eingetragene Marken der jeweiligen Eigentümer sein. Diese sind nicht ausdrücklich durch ™ oder ® gekennzeichnet.

© 2023 PEAK-System Technik GmbH

Die Vervielfältigung (Kopie, Druck oder in anderer Form) sowie die elektronische Verbreitung dieses Dokuments ist nur mit ausdrücklicher, schriftlicher Genehmigung der PEAK-System Technik GmbH erlaubt. Die PEAK-System Technik GmbH behält sich das Recht zur Änderung technischer Daten ohne vorherige Ankündigung vor. Es gelten die allgemeinen Geschäftsbedingungen sowie die Bestimmungen der Lizenzverträge. Alle Rechte vorbehalten.

PEAK-System Technik GmbH
Otto-Röhm-Straße 69
64293 Darmstadt
Deutschland

Telefon: +49 6151 8173-20
Telefax: +49 6151 8173-29

www.peak-system.com
info@peak-system.com

Dokumentversion 1.3.0 (2023-12-06)

Inhalt

Impressum	2
Berücksichtigtes Produkt	2
Inhalt	3
1 Einleitung	4
1.1 Eigenschaften im Überblick	4
1.2 Lieferumfang	5
1.3 Bestimmungsgemäße Verwendung	5
2 Voraussetzungen	7
2.1 Hardware	7
2.2 Software	8
3 Anschlüsse	9
3.1 PEAK-DevPack Debug Adapter	9
3.2 Programmierbare PCAN-Hardware	9
4 Hardware anschließen	13
5 Debuggen	15
5.1 Visual Studio Code einrichten	15
5.2 PCAN-Hardware debuggen	17
5.3 Sonderfall PCAN-GPS debuggen	18
Anhang A Pin-Belegung	20
Anhang B Schaltplan	21

1 Einleitung

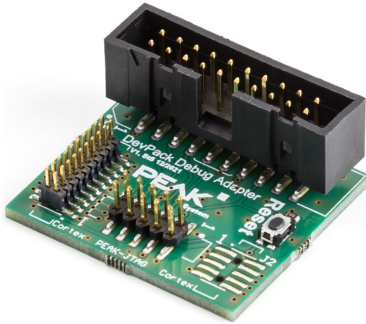
Der PEAK-DevPack Debug Adapter ermöglicht den unkomplizierten Anschluss eines Hardware-Debuggers an unterschiedliche ARM-basierende, programmierbare Hardware-Produkte von PEAK-System. Dafür verfügt der Adapter über verschiedene Anschlüsse zur Verbindung des Hardware-Debuggers mit den spezifischen JTAG-Anschlüssen der zu programmierenden Hardware.

1.1 Eigenschaften im Überblick

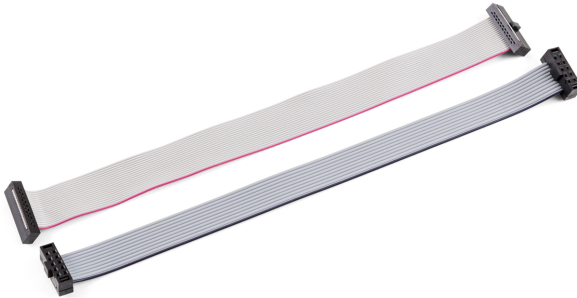
- Adapter zur Verbindung eines Hardware-Debuggers mit dem JTAG-Anschluss einer PCAN-Hardware
- Unterstützte PCAN-Hardware-Produkte:
 - PCAN-Router Pro FD IPEH-002220 / IPEH-002222
 - PCAN-Router FD IPEH-002214 / IPEH-002215
 - PCAN-MicroMod FD IPEH-003080
 - PCAN-GPS IPEH-002110
 - PCAN-GPS FD IPEH-003110
- Reset-Taste zum Neustarten der zu programmierenden PCAN-Hardware

1.2 Lieferumfang

- PEAK-DevPack Debug Adapter



- Flachbandkabel mit farblich gekennzeichnetem Pin 1
 - 1 x 20-Pin mit 1,27 mm Rastermaß
 - 1 x 10-Pin mit 2 mm Rastermaß



1.3 Bestimmungsgemäße Verwendung

Der PEAK-DevPack Debug Adapter ist als Verbindungsadapter zwischen ARM-basierender, programmierbarer PCAN-Hardware und einem Hardware-Debugger konzipiert.

Der PEAK-DevPack Debug Adapter wurde mit dem Hardware-Debugger ST-Link V2, der Software Visual Studio Code und dem PEAK-DevPack Debug Add-on getestet. Eine Verwendung anderer Hardware-Debugger und/oder Software ist möglich, aber nicht Bestandteil dieser Anleitung.

2 Voraussetzungen

2.1 Hardware

Hardware-Debugger ST-Link V2

Für das Debuggen einer PCAN-Hardware wird ein Hardware-Debugger benötigt. Der PEAK-DevPack Debug Adapter wurde mit dem Hardware-Debugger ST-Link V2 getestet. Dieser Hardware-Debugger ist nicht im Lieferumfang enthalten.

Mehr Informationen zum ST-Link V2 finden Sie auf der Webseite:

<https://www.st.com/en/development-tools/st-link-v2.html>

PCAN-Hardware

Verschiedene ARM-basierende, programmierbare Hardware-Produkte von PEAK-System unterstützten das Debuggen mit einem Hardware-Debugger und dem PEAK-DevPack Debug Adapter. Nehmen Sie Ihre PCAN-Hardware, wie im dazugehörigen Handbuch beschrieben, in Betrieb. Die PCAN-Hardware muss über eine separate Spannungsquelle versorgt werden.

PCAN-Interface

Für die Übertragung einer Firmware per CAN wird ein CAN-Interface von PEAK-System benötigt. Nehmen Sie das CAN-Interface, wie im dazugehörigen Handbuch beschrieben, in Betrieb.

2.2 Software

Für das Debuggen benötigen Sie die folgende Software:

- Visual Studio Code mit den folgenden installierten Plug-ins:



C/C++ Plug-in

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools>



Cortex-Debug Plug-in

<https://marketplace.visualstudio.com/items?itemName=marus25.cortex-debug>

- Installierter ST-Link-V2-Treiber:

<https://www.st.com/en/development-tools/stsw-link009.html#overview>

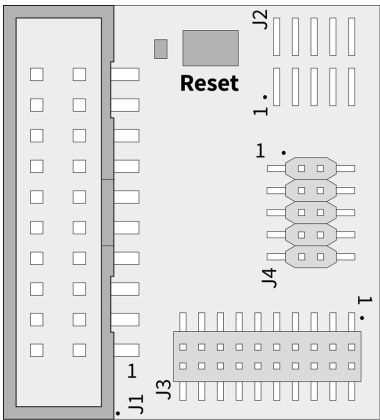
- PEAK-DevPack mit enthaltenem Debug-Add-on:

<https://www.peak-system.com/quick/DLP-DevPack>

1. Speichern und entpacken Sie das *PEAK-DevPack*-Archiv.
2. Führen Sie das Script `SetPath_for_VSCode.vbs` aus.
3. Wechseln Sie danach in das Unterverzeichnis *PEAK-DevPack/Debug*.
4. Führen sie das Skript `SetDebug_for_VSCode.vbs` aus.

3 Anschlüsse

3.1 PEAK-DevPack Debug Adapter



Anschluss	Pins und Rastermaß	Anschluss für
J1	20 Pins 2,54 mm	ST-Link V2 Debugger
J2	10 Pins 1,27 mm	Nicht bestückt
J3	20 Pins 1,27 mm	PCAN-Router Pro FD PCAN-MicroMod FD PCAN-GPS FD
J4	10 Pins 2,00 mm	PCAN-Router FD PCAN-GPS

Pin 1 der Anschlussfelder ist auf der Platine jeweils mit einer 1 markiert.
Die Reset-Taste ist direkt an den Reset-Pin der PCAN-Hardware-CPU angeschlossen.

3.2 Programmierbare PCAN-Hardware

Das Debuggen erfolgt über den JTAG-Anschluss der PCAN-Hardware. Bei den meisten PCAN-Hardware-Produkten ist dieser Anschluss bei Auslieferung nicht bestückt. Anwender müssen die Stiftleiste eigenständig auflöten.

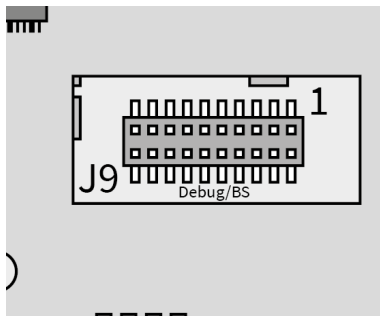


Kurzschlussgefahr! Lötarbeiten an der PCAN-Hardware dürfen nur durch Fachpersonal der Elektrotechnik erfolgen.



Achtung! Elektrostatische Entladung (ESD) kann Komponenten auf der Platine beschädigen oder zerstören. Treffen Sie Vorkehrungen zur Vermeidung von ESD.

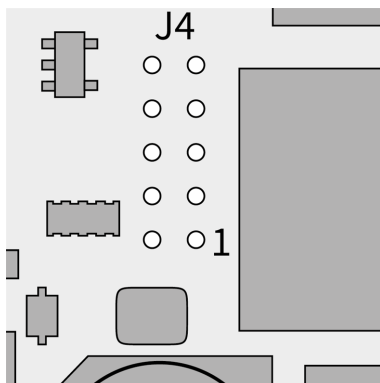
PCAN-Router Pro FD



JTAG-Anschluss	Pins und Rastermaß	Pin 1-Kennung
J9	20 Pins 1,27 mm	1

Der JTAG-Anschluss des PCAN-Router Pro FD ist bei Auslieferung bestückt.

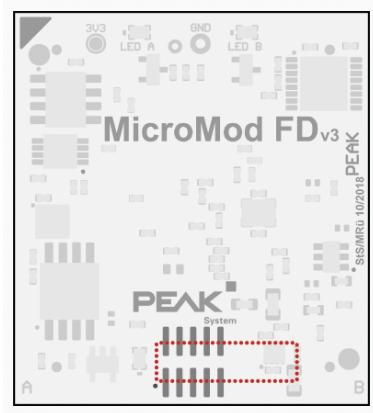
PCAN-Router FD



JTAG-Anschluss	Pins und Rastermaß	Pin 1-Kennung
J4	10 Pins 2,00 mm	1

Der JTAG-Anschluss des PCAN-Router FD ist bei Auslieferung nicht bestückt.

PCAN-MicroMod FD

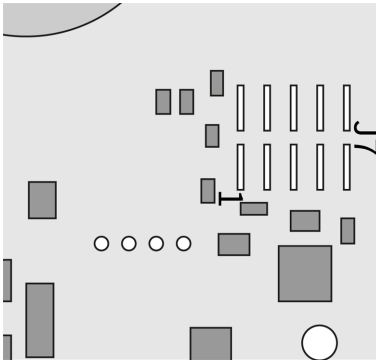


JTAG-Anschluss	Pins und Rastermaß	Pin 1-Kennung
ohne Beschriftung	10 Pins 1,27 mm	weißer Punkt

Der JTAG-Anschluss des PCAN-MicroMod FD ist bei Auslieferung nicht bestückt.

Bei Verwendung des 20-Pin-Flachbandkabels mit 1,27 mm Rastermaß überlappt der Stecker, wie in der Abbildung rot markiert. Stellen Sie sicher, dass Pin 1 des Kabels und der Platine verbunden sind.

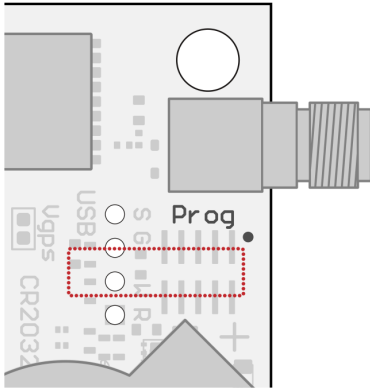
PCAN-GPS



JTAG-Anschluss	Pins und Rastermaß	Pin 1-Kennung
J7	10 Pins 2,00 mm	1

Der JTAG-Anschluss des PCAN-GPS ist bei Auslieferung nicht bestückt.

PCAN-GPS FD



JTAG-Anschluss	Pins und Rastermaß	Pin 1-Kennung
Prog	10 Pins 1,27 mm	weißer Punkt

Der JTAG-Anschluss des PCAN-GPS FD ist bei Auslieferung nicht bestückt.

Bei Verwendung des 20-Pin-Flachbandkabels mit 1,27 mm Rastermaß überlappt der Stecker, wie in der Abbildung rot markiert. Stellen Sie sicher, dass Pin 1 des Kabels und der Platine verbunden sind.

4 Hardware anschließen

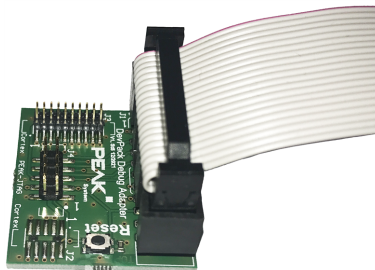
1. Schließen Sie das ST-Link an Ihren Computer an. Die LED leuchtet rot.



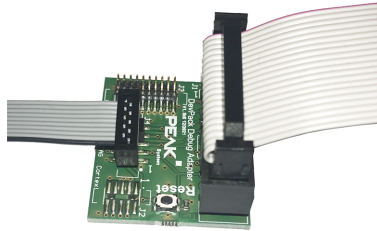
2. Schließen Sie das 20-Pin-Flachbandkabel an den STM32-Anschluss des ST-Link an.



3. Schließen Sie das andere Ende des 20-Pin-Flachbandkabels an den PEAK-DevPack Debug Adapter an.

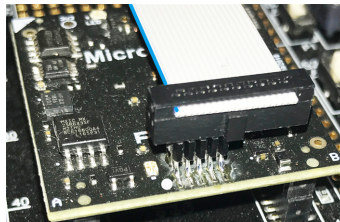


- Schließen Sie das Flachbandkabel, das zu Ihrer PCAN-Hardware passt, an den PEAK-DevPack Debug Adapter an.



Achten Sie darauf, dass Pin 1 des Flachbandkabels mit Pin1 des PEAK-DevPack Debug Adapters verbunden ist.

- Schließen Sie das Flachbandkabel an den JTAG-Anschluss Ihrer PCAN-Hardware an.



Dargestellt am Beispiel PCAN-MicroMod FD

Achten Sie darauf, dass Pin 1 des Flachbandkabels mit Pin 1 der PCAN-Hardware verbunden ist.

Hinweis: Pin 1 ist bei den Flachbandkabeln farblich markiert und auf den Platinen der PCAN-Hardware und beim Adapter mit einer 1 oder einem Punkt gekennzeichnet.

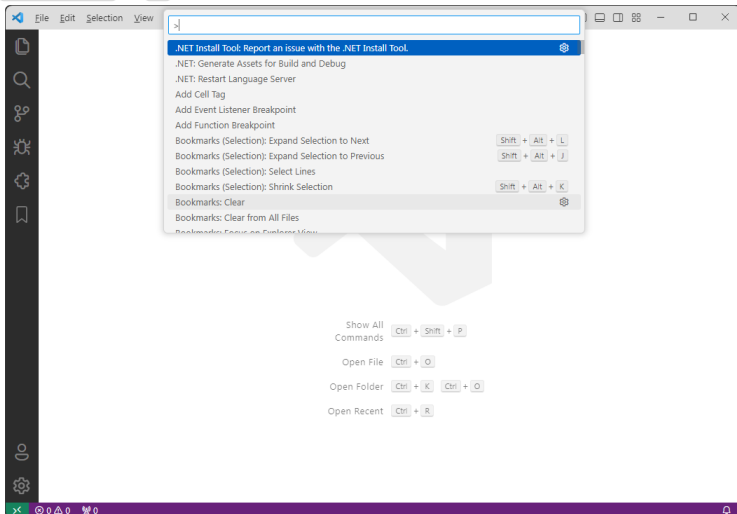
Hinweis: Beim PCAN-MicroMod FD und PCAN-GPS FD ist die Verwendung des 20-Pin-Flachbandkabels mit 1,27 mm Rastermaß vorgesehen. Der Anschluss des Flachbandkabels wird dabei überlappen, da der JTAG-Anschluss nur 10 Pins hat, wie im Beispielbild oben dargestellt.

5 Debuggen

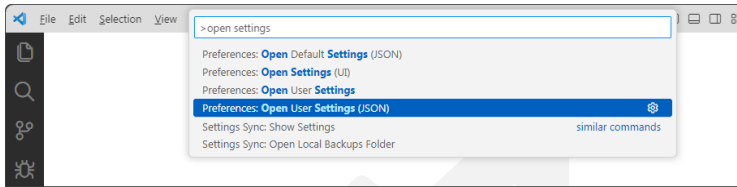
Dieses Kapitel demonstriert die Verwendung von Visual Studio Code mit dem ST-Link V2 Hardware-Debugger.

5.1 Visual Studio Code einrichten

1. Starten Sie Visual Studio Code.
2. Öffnen Sie die Kommandozeile mit **F1** oder alternativ mit **Strg** + **Umschalt** + **P**.



3. Geben Sie *open settings* ein.



Visual Studio Code bietet unter anderen den folgenden Befehl zur Auswahl an:
Preferences: Open User Settings (JSON)

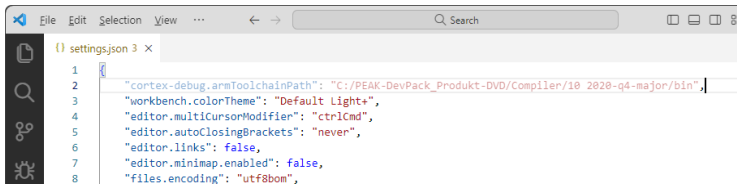
4. Bestätigen Sie die Auswahl mit einem Klick.
Die Datei `settings.json` wird geöffnet.



5. Fügen Sie den Pfad zur Datei `arm-none-eabi-gdb.exe` in den Code ein.

```
"cortex-debug.armToolchainPath":
".../PEAK-DevPack/Compiler/10 2020-q4-major/bin/",
```

Der Platzhalter „...“ muss durch Ihren Verzeichnispfad ersetzt werden. Damit stellen Sie sicher, dass das Cortex-Debug Plug-in den Pfad zum Compiler kennt.

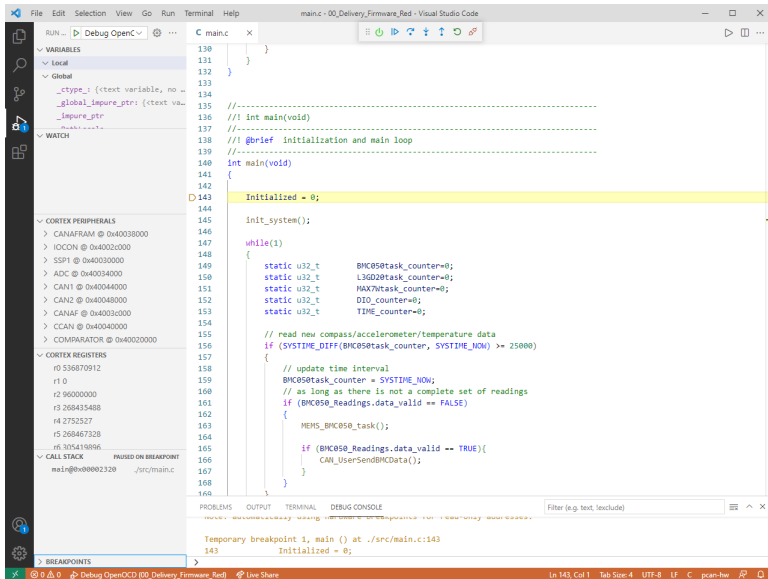


6. Speichern und schließen Sie die Datei.

Hinweis: Die Datei `settings.json` wird zumeist im folgenden Pfad gespeichert: `C:\Users\User\AppData\Roaming\Code\User\`. Der Platzhalter `User` muss durch Ihren Benutzernamen ersetzt werden.

5.2 PCAN-Hardware debuggen

1. Öffnen Sie ein Beispielprojekt für ihre PCAN-Hardware in Visual Studio Code.
2. Kompilieren Sie den Code vor der Ausführung mit *Make Debug* (**Strg** + **Umschalt** + **B**).
3. Starten Sie den Debugger mit **F5** . Die zuvor kompilierte Firmware wird dabei auf Ihre PCAN-Hardware übertragen.



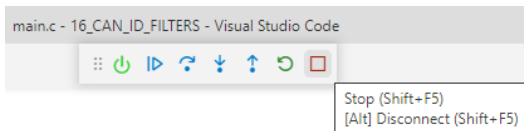
```
130 }
131 }
132 }
133
134
135 //-----
136 //! int main(void)
137 //-----
138 //! @brief Initialization and main loop
139 //-----
140 int main(void)
141 {
142
143     Initialized = 0;
144
145     init_system();
146
147     while(1)
148     {
149         static u32_t BHCBS0task_counter=0;
150         static u32_t L3020task_counter=0;
151         static u32_t MAX7task_counter=0;
152         static u32_t DIO_counter=0;
153         static u32_t TIME_counter=0;
154
155         // read new compass/accelerometer/temperature data
156         if (SYSTIME_DIFF(BHCBS0task_counter, SYSTIME_NOW) >= 25000)
157         {
158             // update time interval
159             BHCBS0task_counter = SYSTIME_NOW;
160             // as long as there is not a complete set of readings
161             if (BHCBS0_Readings.data_valid == FALSE)
162             {
163                 MEMS_BHCBS0_task();
164             }
165             if (BHCBS0_Readings.data_valid == TRUE){
166                 CAN_UserSendBHCData();
167             }
168         }
169     }
170 }
```

Temporary breakpoint 1, main () at ./src/main.c:143
143 Initialized = 0;

Für die folgenden Befehle sind Tastaturkürzel verfügbar:

Continue	F5
Step Over	F10
Step Into	F11
Step Out	Umschalt + F11
Restart	Strg + Umschalt + F5
Stop	Strg + F5
Disconnect	Alt + Strg + F5

Mit *Disconnect* wird der Debugger verlassen.



5.3 Sonderfall PCAN-GPS debuggen



Risiko einer Beschädigung! Der Versuch, die Firmware mit der Visual Studio Code-Debug-Toolchain zu übertragen, kann das PCAN-GPS irreversibel beschädigen.

Die Firmware-Übertragung zum PCAN-GPS kann nicht direkt mit Visual Studio Code erfolgen. Die entsprechende Funktion ist in der Datei `launch.json` für den PCAN-GPS bereits abgeschaltet. Die Übertragung der von Ihnen kompilierten Firmware muss nach dem PCAN-GPS-Benutzerhandbuch mit PEAK-Flash erfolgen. Nach dem erfolgreichen Firmware-Upload kann das PCAN-GPS mit Visual Studio Code debuggt werden.

Wechseln Sie dazu zurück in Ihr Visual Studio Code-Projekt und starten Sie das Debuggen durch zweimaliges Drücken der Taste **F5**. Der erste Druck startet die Debug-Umgebung und stoppt die PCAN-Hardware an einem zufälligen Punkt in der Software. Der zweite Druck bewirkt einen Sprung zum Breakpoint, der automatisch an den Anfang der Main-Funktion gesetzt wurde.

Anhang A Pin-Belegung

Minimal für SWD-Debuggen benötigte Pin-Belegung auf den jeweiligen Steckern:

	J1 ARM JTAG 20 2,54 mm	J2 Cortex 2x5 1,27 mm	J3 Cortex 2x10 1,27 mm	J4 PEAK-JTAG 2x5 2,00 mm
Hardware	ST-Link V2 Debugger	Nicht bestückt	PCAN-Router Pro FD PCAN-MicroMod FD PCAN-GPS FD	PCAN-Router FD PCAN-GPS
Single Wire Signals				
SWO	13	6	6	7
SWDIO	7	2	2	6
SWCLK	9	4	4	5
V _{CC}	1/2	1	1	4
GND	4/6/8/.../20	3/5/9	3/5	1/2
SRST	15	10	10	3

Die Pin-Belegung der Stecker des PEAK-DevPack Debug Adapters ist im folgenden Schaltplan abgebildet.

1

2

3

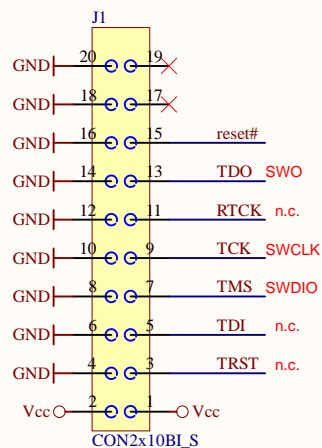
4

A

B

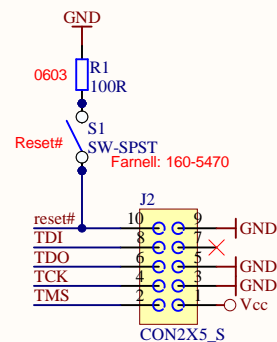
C

D



ARM JTAG 20 2.54mm
(for. e.g.: ST-Link V2)

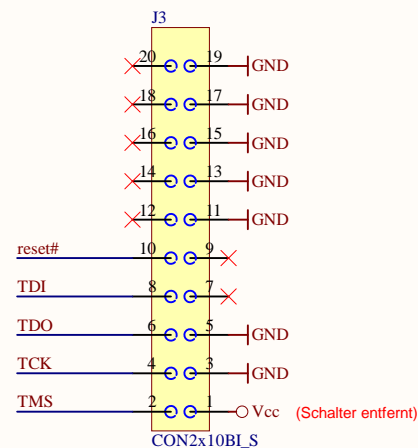
Amphenol 52601-S20-4LF
Mouser: 649-52601-S20-4LF



Cortex 2x5 1.27mm

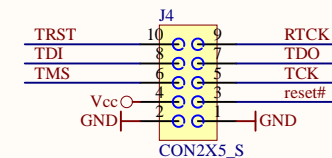
Samtec FTSH-105-01-L-DV-A-P
Mouser: 200-FTSH10501LDVAP

pin header SMT for debug target (if not populated):
Samtec FTSH-105-01-L-DV-P
Mouser 200-FTSH10501LDVP



Cortex 2x10 1.27mm

Samtec FTSH-110-01-L-DV-A-P
Mouser: 200-FTSH11001LDVAP



PEAK-JTAG 2x5 2mm

Amphenol 98424-F52-10ULF
Mouser: 649-98424-F52-10ULF

pin header THT for debug target (if not populated):
Hirose A3C-10P-2DSA(30)
Mouser 798-A3C-10P-2DSA30

© PEAK-System Technik GmbH

PEAK
System

PEAK-System Technik GmbH
Otto-Röhm-Str. 69
D-64293 Darmstadt

Engineer: StS
Date: 26.06.2022

File name: PEAK-DevPack-DebugAdapter_V1.SchDoc

Title:
DevPack-DebugAdaptor

Sheet: Main

Customer:

*

Version: 1.1

Variant:
[No Variations]

Page 1 of 1