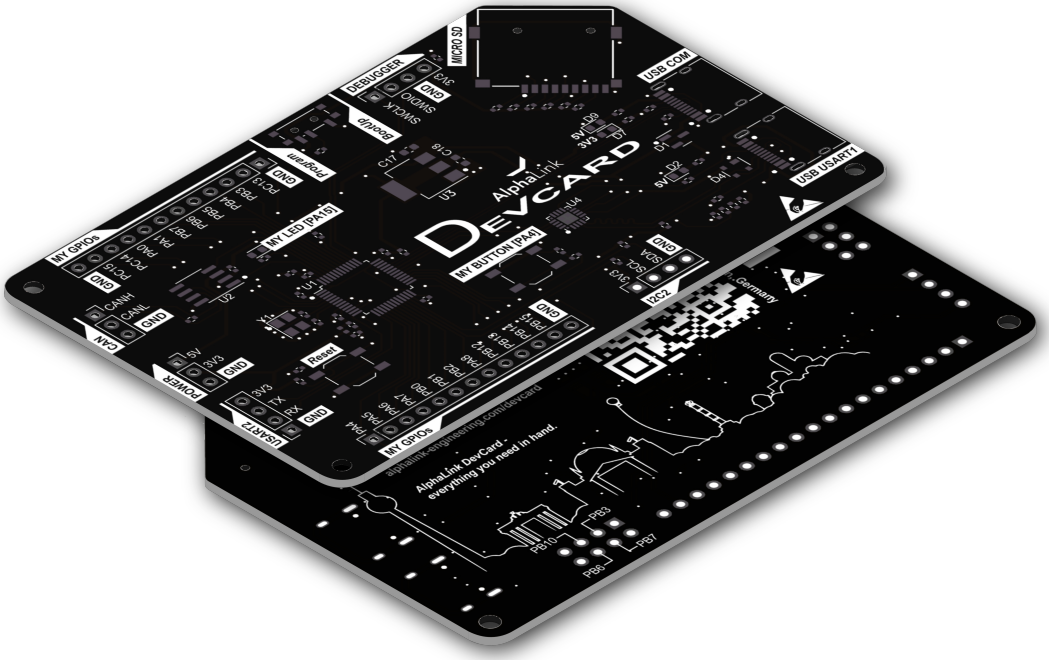
AlphaLink



DevCard

-STM32F103C8T6-

Bedienungsanleitung

Stand: 29.08.2024

AlphaLink DevCard - Everything you need in hand.

Sehr geehrte Kundin, sehr geehrter Kunde,

herzlichen Dank für den Kauf unserer DevCard. Wir möchten Dir hiermit eine kurze Einführung geben, wie Du das STM-Entwicklerboard in Betrieb nehmen und verwenden kannst. Solltest Du während der Nutzung auf unerwartete Probleme stoßen, stehen wir Dir gerne zur Verfügung und helfen Dir bei der Lösung.

Die vorliegende Bedienungsanleitung beantwortet die wichtigsten Fragen von den technischen Spezifikationen und Nutzungshinweisen bis zu Deinem ersten Projekt, mit dem Du direkt loslegen kannst.

Wir haben die Anleitung für Dich folgendermaßen gegliedert:

Inhaltsverzeichnis

Wir wünschen Dir viel Spaß mit Deiner DevCard und freuen uns über Feedback zu Deinen Projekten und den Fortschritten bei der STM-Programmierung.

DEIN ALPHALINK TEAM!

Bestimmungsgemäße Verwendung

Die DevCard ermöglicht Prototyperstellung, Entwicklung und Test von Firmware, Testen von Kommunikationsprotokollen, Sensorik und Aktorik. Entwickler:innen können schnell und effizient elektronische Schaltungen und Systeme erstellen und verschiedene Schnittstellen testen. Die DevCard eignet sich auch für Lernzwecke und Ausbildungsprogramme.

Sicherheitshinweise

Bitte beachte, dass die DevCard über jeden der beiden USB-C Anschlüsse mit 5 V DC an einen Computer oder eine ähnliche Spannungsquelle angeschlossen werden kann. Alternativ kann die DevCard über Debugger Pins mit 3,3 V DC versorgt werden. Dies sollte nur geschehen, wenn die DevCard nicht bereits von einer anderen Spannungsquelle versorgt wird. Der verwendete Prozessor STM32F103C8T6 darf nicht höher als mit 3,3 V DC betrieben werden. Bitte berücksichtige die spezifizierten Stromstärken für die I/O-Pins sowie die benötigte und maximale Stromstärke des STM32 selbst. Der STM32F103C8T6 Mikrocontroller hat eine maximale I/O-Stromstärke von 25 mA pro Pin. Die DevCard wurde für eine Stromstärke von 4,26 mA bei einer Spannung von 5 V DC entwickelt und getestet. Die Stromstärke beträgt 3,16 mA bei einer Spannung von 3 V DC. Bitte beachte auch die benötigte und maximale Stromstärke des STM32F103C8T6 Mikrocontrollers selbst, die je nach Aktivität und Konfiguration variiert. Die maximale Stromstärke des STM32F103C8T6 beträgt typischerweise 60 mA. Bitte halte während des Betriebs die Stromstärken für die I/O-Pins der DevCard sowie die benötigte und maximale Stromstärke des STM32F103C8T6 Mikrocontrollers ein. Ein Übertakten des STM32 ist nicht empfohlen und kann zu Schäden führen. Eine Berücksichtigung dieser Sicherheitshinweise gewährleistet nicht nur eine ordnungsgemäße Funktion der DevCard, sondern auch die Sicherheit und Zuverlässigkeit des gesamten Systems.

Bei Nichteinhaltung der Sicherheitshinweise kann es zu Fehlfunktionen, Hitzeentwicklung und der Gefahr eines elektrischen Schlages kommen.

Spannungsversorgung	USB 5 Volt (DC)
Betriebsspannung	3,0 –3,6 Volt (DC)

1. Produktübersicht

Die DevCard STM32F103C8T6 ist ein Prototyping Board, das sich über die STM32CubeIDE programmieren lässt. Es verfügt über USB, USART sowie CAN. Ebenfalls ist ein SD-Kartenslot für Micro SD verbaut.
Die DevCard kommuniziert über USART1 ohne weiteren benötigten Chipsatz. Dies gilt auch für die CAN-Schnittstelle.

Übersicht der Pinbelegungen

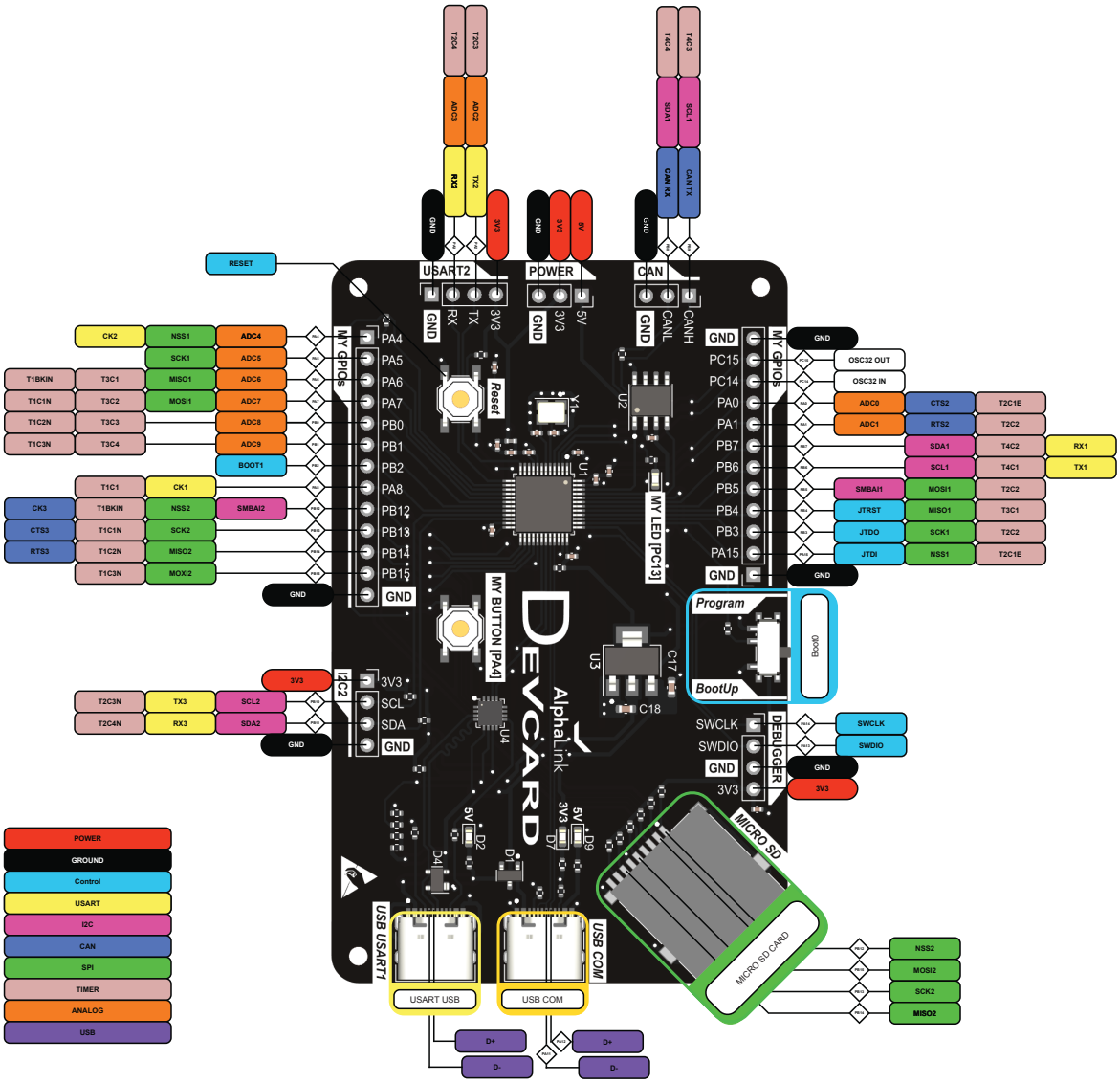


Abbildung 1: Schematische Pin-Übersicht DevCard (v1.0).

Im Verlauf der Entwicklung gab es weiter Iterationen der DevCard welche das Pinout verändert haben. Die Änderungen sind in den unteren Abbildungen 1 & 2 dargestellt.

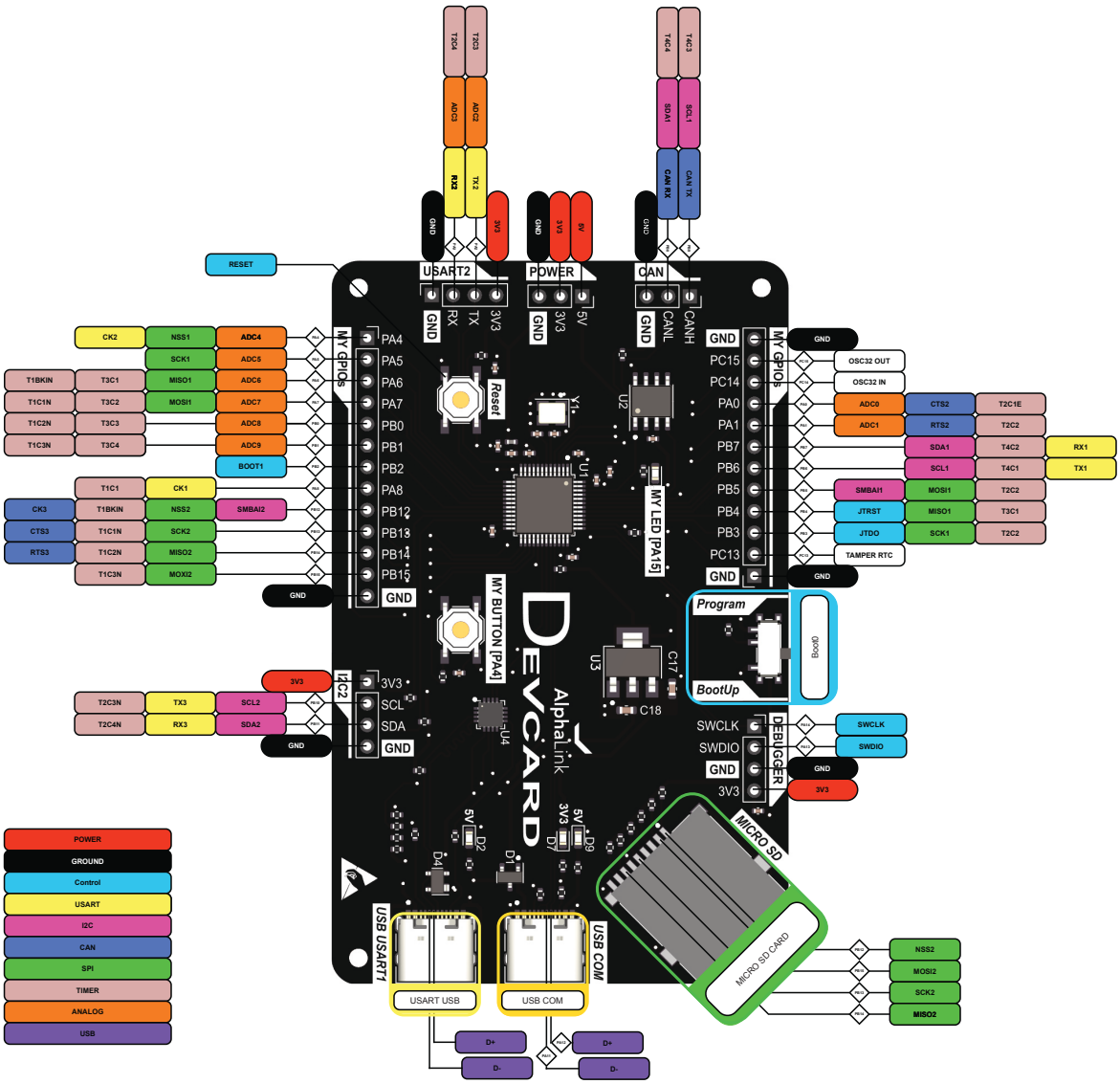


Abbildung 2: Schematische Pin-Übersicht DevCard (v1.2 & v1.3).

STM32CubeIDE

Unter folgendem Link findest Du die Informationen zu der Entwicklungsumgebung:
`https://www.st.com/en/development-tools/stm32cubeide.html#get-software`

Bemaßung der Devcard

2. Transport

Die DevCard wird in einer geschlossenen ESD-Hülle, nach Norm *EN 61340-5-3*, geliefert. Um Beschädigungen am Board zu vermeiden, sollte das Produkt bei weiterem Transport in der

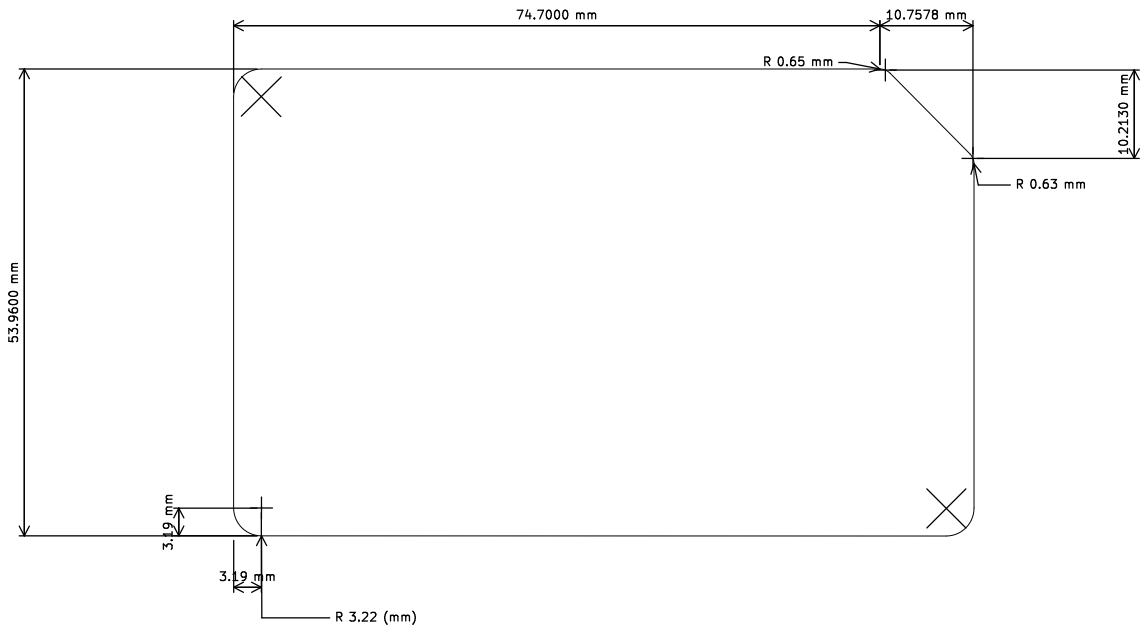


Abbildung 3: Bemaßung der DevCard.

Hülle bleiben. Achte darauf, Dich vor der Entnahme zu erden, um die DevCard und Dich vor elektrischen Phänomenen zu schützen.

Lieferumfang

Im Lieferumfang ist ausschließlich die mit den Standardkomponenten bestückte DevCard enthalten.

3. Inbetriebnahme

Anschließen

Die DevCard ist ein vielseitiges Entwicklungsboard, das eine Reihe von Möglichkeiten bietet, um den STM32-Mikrocontroller zu programmieren und mit ihm zu arbeiten. Hier sind die Schritte zur Inbetriebnahme der DevCard erklärt. Wir unterscheiden dabei

- Anschließen über USB COM: Die DevCard kann über einen COM-USB-Anschluss mit einem Computer verbunden werden. Verbinde das eine Ende des USB-Kabels mit dem COM-USB-Anschluss auf der DevCard und das andere Ende mit einem USB-Anschluss Deines Computers. Durch diese Verbindung wird das Board mit Strom versorgt.
- Anschließen über USB USART1: Alternativ kann die DevCard über den USB-zu-USART1-Anschluss mit einem Computer verbunden werden. Schließe das eine Ende des USB-Kabels an den USB-zu-USART1-Anschluss auf der DevCard und das andere Ende an einen USB-Anschluss Deines Computers an. Auch durch diese Verbindung wird das Board mit Strom versorgt.

Beide Anschlussmöglichkeiten ermöglichen die Stromversorgung der DevCard mit 5 V; dies ist für den Betrieb des Boards erforderlich.

Programmieren mit STM32CubeIDE

Der STM32 auf der DevCard kann über einen ST-Link/V2 und die *STM32CubeIDE* programmiert werden. Dazu wird eine Stromversorgung über die Debugger-Pins hergestellt (Spannungsversorgung mit 3,3 V). So wird die DevCard erfolgreich für die Programmierung in Betrieb genommen und ist bereit für die Entwicklung und weitere Arbeiten mit dem STM32-Mikrocontroller. Eine Erklärung zur STM32CubeIDE findest Du im nächsten Kapitel zu Deinem ersten eigenen Projekt.
Mit diesen zusätzlichen Schritten ist die DevCard für die Programmierung einsatzbereit, das Programm kann auf den STM32 geladen werden und das Board kann wieder mit den entsprechenden USB-Anschlüssen verbunden werden, um das Programm auszuführen. Das Programm sollte nun auf dem STM32-Mikrocontroller laufen und ordnungsgemäß funktionieren.

Hinweis: Die Programmdatei, die auf den STM32 geladen wird, muss den Dateityp `.bin` oder `.elf` haben. Stelle sicher, dass die richtige Datei ausgewählt und übertragen wird, um einen erfolgreichen Programmablauf auf dem STM32 zu gewährleisten.

Alternative: CubeProgrammer

Die DevCard kann auch über den USB USART1-Anschluss mit dem *CubeProgrammer* programmiert werden. Dafür muss der STM32 in den **Programmiermodus** versetzt werden. Hier-

zu wird auf dem Board der Schalter in den `Program`-Modus gestellt. Nachdem der STM32 in den Programmiermodus versetzt wurde, muss das Board vom Strom getrennt und wieder verbunden werden. Außerdem muss ein Male-to-Male-Kabel von Pin PB2 zu einem beliebigen GND-Pin verlegt werden, um die Programmierung zu ermöglichen.

Nach Abschluss der Programmierung sind folgende Schritte erforderlich:

1. Wenn der CubeProgrammer den Programmierprozess abgeschlossen hat, muss die DevCard über den Reset-Taster neu gestartet werden. Anschließend sollte das Board von der Stromversorgung getrennt werden.
2. Der Modus des STM32 muss über den Schalter in den `BootUp`-Modus zurückgesetzt werden.
3. Nun kann die DevCard wieder über den USB COM oder USB USART1 Anschluss mit dem Computer verbunden werden.

4. Dein erstes Projekt

In diesem Kapitel wird ein Beispielprojekt vorgestellt, um Dir den Einstieg in die Nutzung der DevCard zu erleichtern und erste Erfahrungen in der Programmierung des STM32-Mikrocontrollers zu sammeln. Dieses Projekt bietet eine praktische Einführung in grundlegende Funktionen der DevCard.
Durch die Umsetzung dieses Beispielprojektes erhältst Du grundlegende Kenntnisse über die Verwendung der DevCard und kannst diese als Ausgangspunkt für Deine eigenen Entwicklungen nutzen. Jedes weitere Projekt ermöglicht Dir, verschiedene Aspekte der Hardware und der Programmierung zu erkunden und Deine Fähigkeiten weiter auszubauen.

Konfiguration von Projekten

Um die DevCard zu Programmieren, wird die Verwendung des Programms *STM32CubeIDE* empfohlen (Download hier).[3]
Wurde die benötigte Software heruntergeladen und installiert, kannst Du Dein erstes Projekt konfigurieren. Dafür legst Du zuerst einen Workspace an.

[3] Hinweis: Es ist ebenfalls möglich die DevCard mit der *ArduinoIDE* oder *VSCode* zu Programmieren.

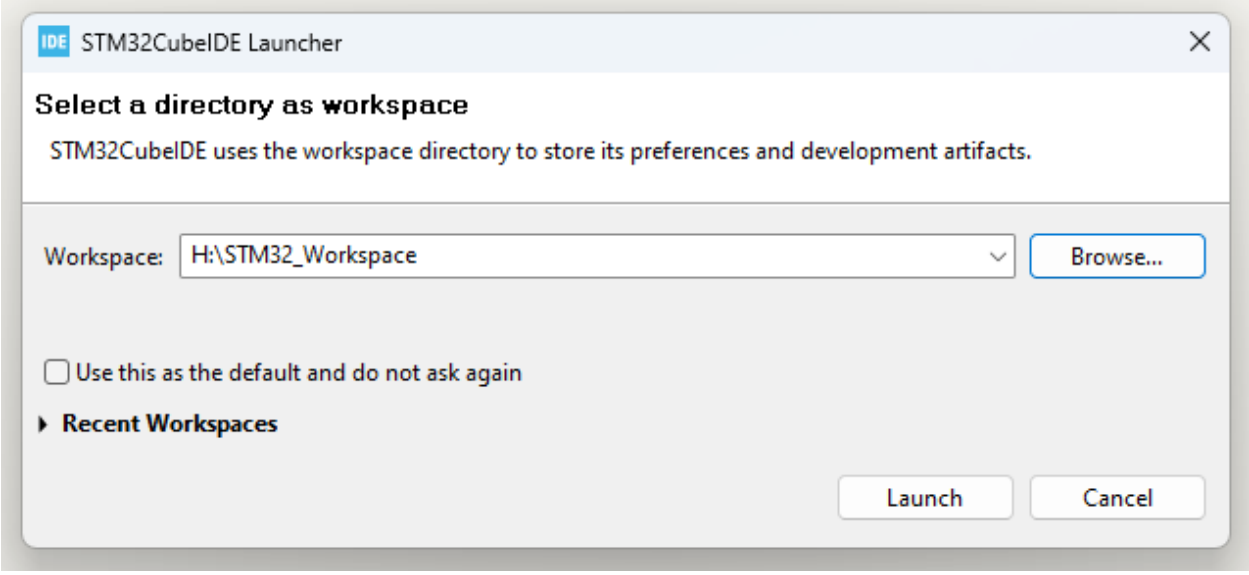


Abbildung 4: Erstellen eines Workspaces.

Wurde der Workspace wie in Abb. 4 erstellt, so öffnet sich das `Information Center` bzw. die STM32CubeIDE Home Seite. In dieser wählst Du `Start new STM32 project` aus.

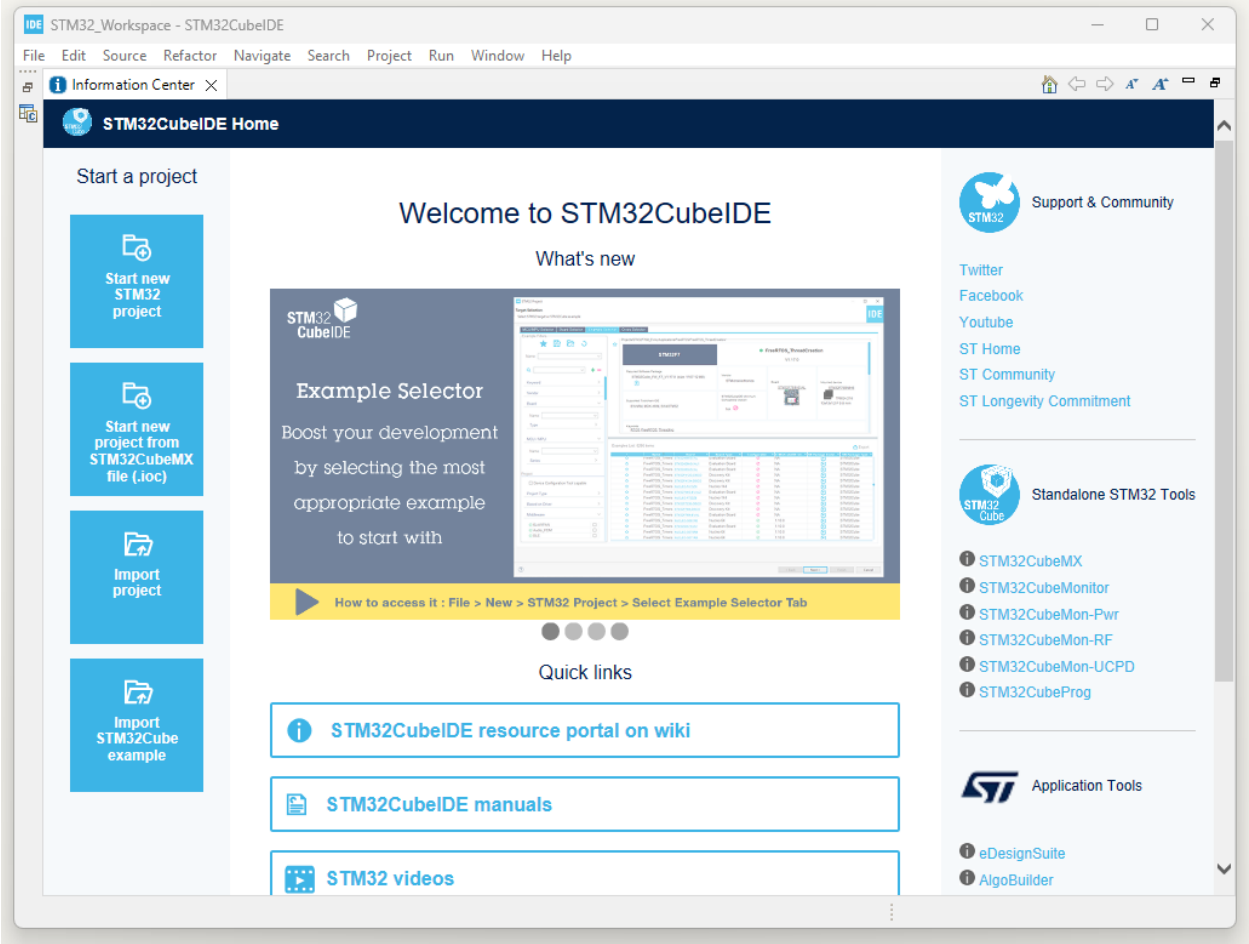


Abbildung 5: Erstellen eines neuen Projektes.

Nun sollte sich ein weiteres Fenster öffnen, das `STM32 Project. Target Selection` heißt. Hier suchst Du unter dem Tab `MCU/MPU Filters` das Modell („STM32F103C8T6“) aus (siehe Abb. 6). Dieser STM32 ist auf der DevCard verbaut.

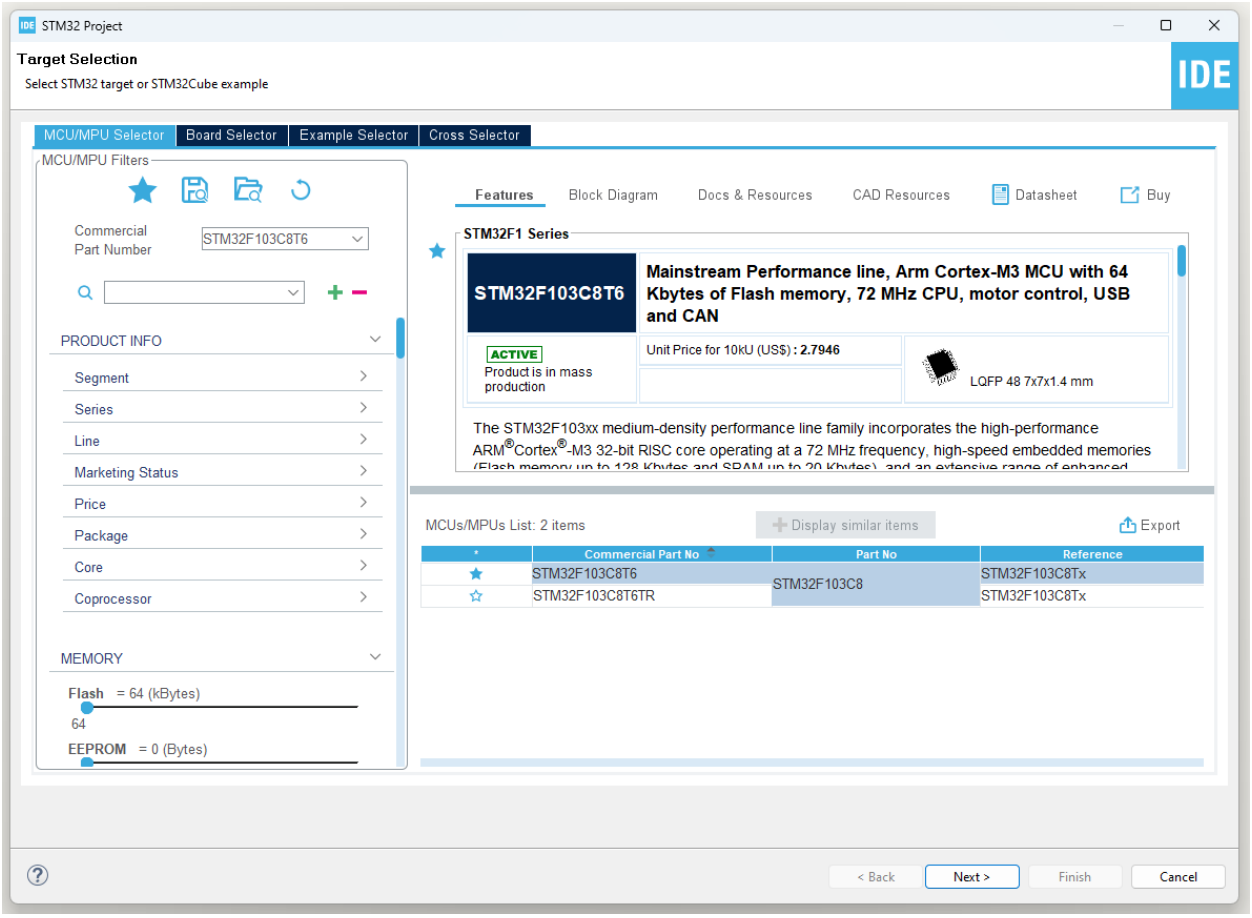


Abbildung 6: STM32 Modell auswählen.

Wurde der entsprechende STM32 ausgewählt, gibst du Deinem Projekt einen Namen. Wie Du in Abb. 7 sehen kannst, wurde im Beispiel der Name *„First_Project“* gewählt.

Nach kurzem Warten, öffnet sich die `.ioc`-Datei. In dieser Datei kannst Du über die Benutzeroberfläche bereits erste Konfigurationen an dem Projekt vornehmen (siehe Abb. 8. In den nächsten Schritten wird weiter auf diese Konfigurationen eingegangen.

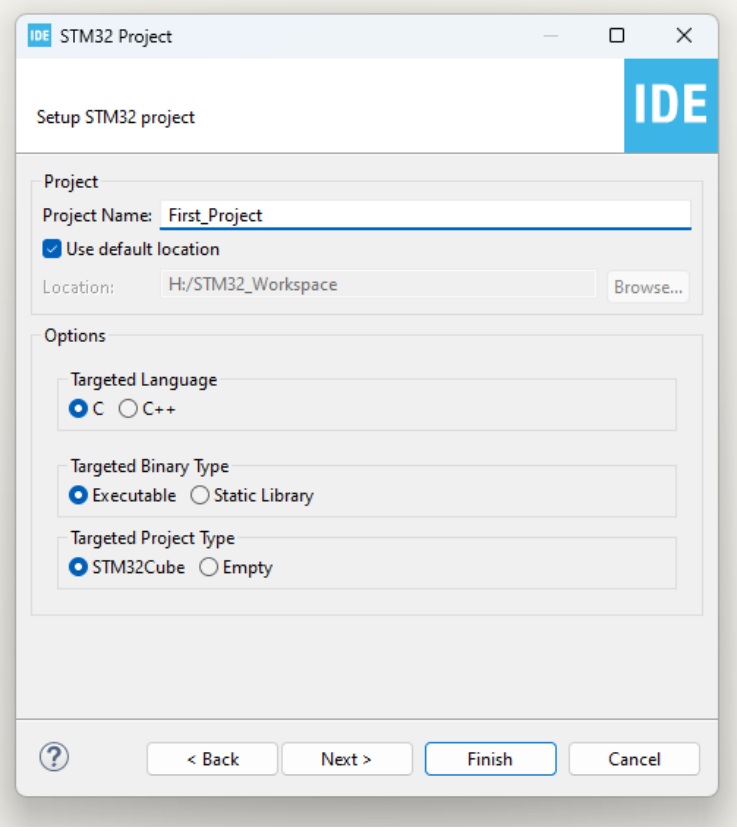


Abbildung 7: Benennung eines STM32 Projektes.

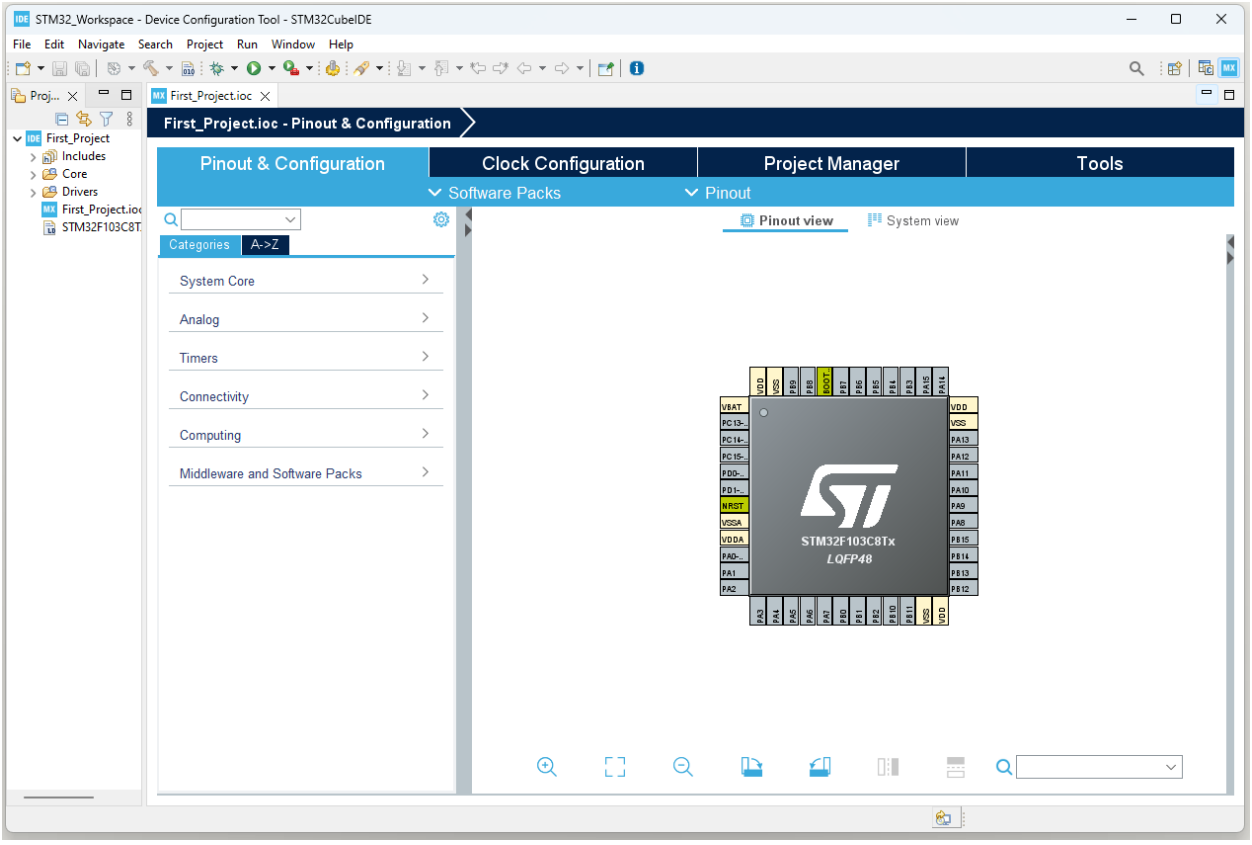
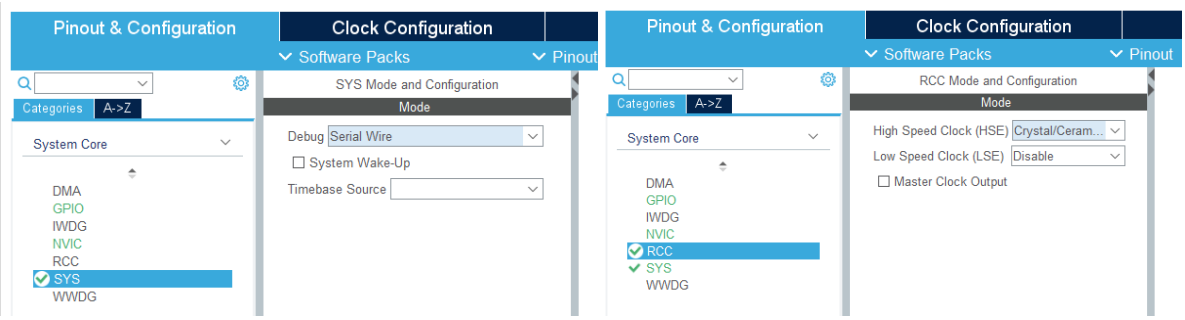


Abbildung 8: Standard `.ioc`-Datei.

STM32 Konfigurationen

In diesem Kapitel wird erklärt, wie der STM32 der DevCard für die meisten Projekte konfiguriert wird. Dafür klickst Du als erstes in dem Tab `Pinout & Configuration` auf `Categories` und darunter auf `System Core`. In diesen werden unter anderem die Debug-Art sowie die Clock eingestellt. Diese beiden Schritte werden in Abb. 9a und 9b gezeigt.



(a) Debugger Konfiguration. (b) Konfiguration einer externen Clock.

Abbildung 9: Konfiguration, erste Schritte.

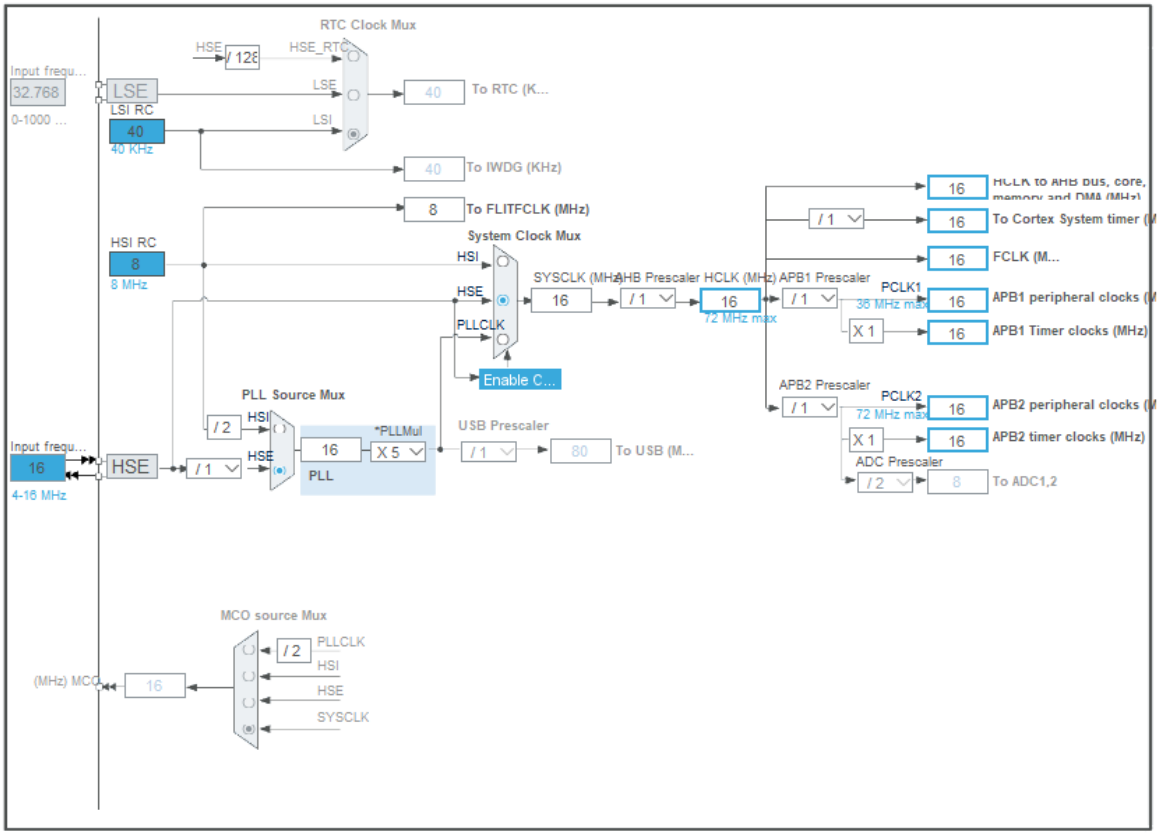


Abbildung 10: Detail-Einstellungen in der Clock Konfiguration.

Nach diesen Schritten kann die Clock noch weiter konfiguriert werden. Dafür wählst Du den Tab `Clock Configuration` aus. In Abb. 10 wird gezeigt, wie diese einzustellen ist. Dafür

gibst Du bei der `Input Frequenz` die Frequenz des externen Oszillators an. Diese beträgt hier 16 MHz, da der verbaute Crystal-Oszillator 16 MHz besitzt.

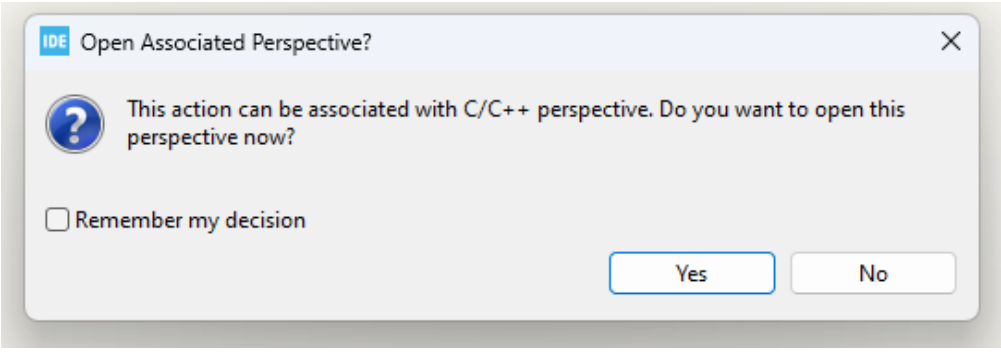


Abbildung 11: Bestätigung, die Projekt-Entwicklungs-Umgebung zu öffnen.

Als nächstes klickst Du auf das Icon mit dem kleinen Zahnrad ( ) in der oberen Werkzeugleiste der STM32CubeIDE. Das ermöglicht Dir, aus denen über die Benutzeroberfläche getroffenen Einstellungen eine `main.c` Datei zu erzeugen. Nach einem Klick auf dieses Icon öffnet sich das Fenster `Open Associated Perspective`, in dem Du auf `Yes` klickst (siehe Abb. 11). Es ist auch möglich einen Haken bei `Remember my decision` zu setzen, der dafür sorgt, dass sich dieses Fenster nicht bei jedem Übersetzen der `.ioc`-Datei öffnet.
Es sollte sich folglich die `main.c`-Datei öffnen. In dieser werden die Programme geschrieben. Im nächsten Abschnitt wird erklärt, wie in dieser Datei das erste Programm entwickelt wird.

EXKURS: LED zum Leuchten bringen (Beispielprojekt)

Wir möchten Dich herzlich zum ersten Projekt der DevCard-Dokumentation begrüßen. In diesem Projekt konzentrieren wir uns darauf, die auf dem Board gekennzeichnete `MY LED` zum Leuchten und Blinken zu bringen. Die `MY LED` bietet eine hervorragende Möglichkeit, die Grundlagen der LED-Steuerung kennenzulernen und einen ersten Einblick in die Programmierung des STM32-Mikrocontrollers zu erhalten.
Wir werden Schritt für Schritt vorgehen, um die auf dem Board verbaute LED, genauer gesagt die `MY LED`, zum Leuchten und Blinken zu bringen. Die LED ist mit Pin PA15[4] (siehe Beschriftung auf der DevCard) des STM32 verbunden und wird über die GPIO-Pin-Steuerung aktiviert. Wir werden uns darauf konzentrieren, den entsprechenden Code zu erstellen, die LED korrekt mit dem GPIO-Pin mittels des Codes zu verbinden und den entwickelten Code erfolgreich auf den STM32-Mikrocontroller zu übertragen.

„Hello World“ für die STM32 Programmierung

Wer schon einmal eine Programmiersprache lernen wollte, ist bestimmt auf den Begriff „Hello World“ gestoßen. In der STM32 Programmierung lassen wir als erstes Programm aber nicht

[4] **Hinweis:** Bei Version v1.0 ist die LED mit Pin PC13 verbunden.

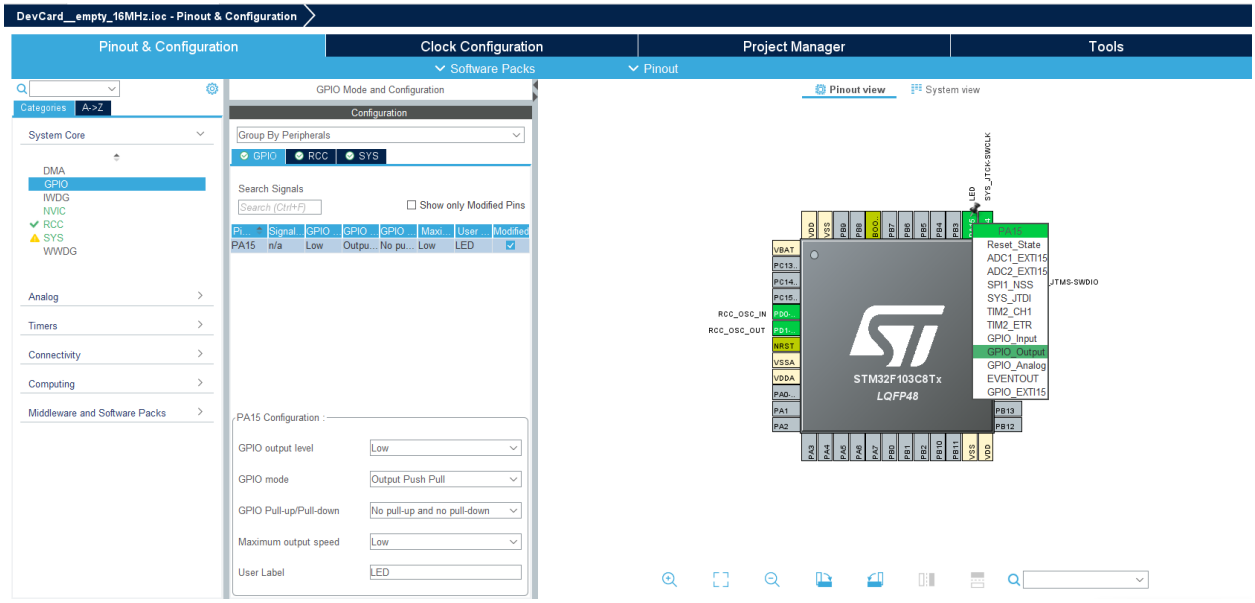


Abbildung 12: Benennung eines GPIO-Pins.

einfach eine simple Zeile Text in der Konsole ausgeben, die *„Hello World“* sagt, sondern wir bringen gleich eine LED zum Leuchten.

Um eine LED in der STM32-Programmierung zum Leuchten zu bringen, wählst Du zunächst den Pin aus, an dem die LED angeschlossen ist. Die DevCard, ein Lernwerkzeug, verfügt bereits über eine solche LED. Diese ist mit `MY LED [PA15]`[5] gekennzeichnet. Die Bezeichnung *[PA15]* gibt an, an welchem Pin die LED angeschlossen ist.
In der `.ioc`-Datei klickst Du zunächst auf den entsprechenden Pin des STM32 und definierst diesen als *„GPIO_Output“*. GPIO bedeutet dabei *General Purpose Input/Output*. Anschließend wird der Pin unter `System Core > GPIO` aufgelistet. Klickst Du diesen an, kannst du weitere Eigenschaften des GPIO Pins verändern. Im Feld `User Label` kannst du dem Pin außerdem einen eigenen Namen zuweisen, unter welchem Du ihn später im Programmcode ansprechen kannst. Abbildung 12 zeigt Dir dies für die `MY LED`, dieser wird hier der Name „LED“ gegeben.

Nach dem Speichern und dem Generieren des Codes öffnet sich die `main.c`-Datei. In dieser scrollst Du zu nachfolgendem Code-Teil. Dabei ist zu beachten, dass der Code der ausgeführt werden soll zwischen *„/* USER CODE BEGIN WHILE */“* und *„/* USER CODE END WHILE */“* stehen muss. Sollte der Code nicht zwischen diesen Kommentaren stehen, so wird bei erneutem Generieren von Code aus der `.ioc`-Datei Dein selbst geschriebener Code gelöscht.

[5] **Hinweis:** Bei der Version v1.0 ist die LED stattdessen mit dem Pin PC13 verbunden.

```
/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {
       /*
       The LED, located at Pin PC13 (for v1.0) or at PA15 (for v1.2/v1.3),
       will be set to HIGH, which means it will be active. In this example,
       the name "LED" serves as a reference to the pin it is connected to.

       Alternatively, you could use the following code:
       HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, SET);
       */

       HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, SET);

   /* USER CODE END WHILE */

   /* USER CODE BEGIN 3 */
 }
 /* USER CODE END 3 */
```

Um das Programm aus der STM32CubeIDE auf die DevCard zu flashen, ist es notwendig, diese über den ST-Link anzuschließen.[6] Für Anfänger wird empfohlen, einen ST-Link zu verwenden. Über einen Klick auf das Icon mit dem weißen Dreieck in einem grünen Kreis () in der oberen Werkzeugleiste der STM32CubeIDE, wird zuerst das aktuell geöffnete Projekt geprüft, dann kompiliert und zuletzt auf die DevCard geflasht.

Weitere Beispielprojekte

Unser Team arbeitet laufend daran, weitere Projekte zusammenzustellen, um Dir die Arbeit mit der DevCard zu erleichtern. Schau einfach regelmäßig auf unserer Webseite nach (`alphalink-engineering.com/devcard`) oder schreib uns, was dich interessiert.

[6] *Flashen* bei STM32 bezieht sich auf den Prozess des Programmierens der Firmware oder des Programmcodes in den eingebetteten Flash-Speicher des STM32-Mikrocontrollers, um die gewünschte Funktionalität auf dem Mikrocontroller auszuführen.

5. Support

Auch nach dem Kauf stehen wir Dir gerne zur Seite. Falls Du offene Fragen oder Probleme hast, kontaktiere uns doch ganz einfach per E-Mail. Für weitere Informationen scanne gerne den QR-Code auf der Rückseite der DevCard.

E-Mail:	devcard@alphalink-engineering.com
Internet:	alphalink-engineering.com/devcard

6. Technische Daten

CPU:	STM32F103C8T6
CPU frequency:	72 MHz
Schnittstellen:	- USB zu USART1 - USB COM - USART2/I2C2/CAN - Serial Wire Debugger - GPIO/PWM - Micro SD
Eingangsspannung:	5 VDC über via USB-C / 3,3 VDC via Debugger Pins
Temperaturbereich:	-40 °C bis +80 °C
Abmessungen (LxBxH):	85,6 x 54 x 11,7 mm
Gewicht:	19 g

7. Sonstige Hinweise

Informationen zur Entsorgung gemäß Elektrogesetz (ElektroG)

Symbol auf Elektro- und Elektronikgeräten:

Das durchgestrichene Mülltonnen-Symbol weist darauf hin, dass Elektro- und Elektronikgeräte nicht im Hausmüll entsorgt werden dürfen. Du musst Altgeräte an einer dafür vorgesehenen

Erfassungsstelle abgeben. Vor der Abgabe trenne bitte Altbatterien und Altakkumulatoren, die nicht fest mit dem Altgerät verbunden sind.

Rückgabemöglichkeiten:
Als Endnutzer hast Du die Möglichkeit, Dein Altgerät kostenlos zur Entsorgung abzugeben, wenn Du ein neues Gerät erwirbst, das im Wesentlichen die gleiche Funktion wie Dein Altgerät erfüllt. Kleingeräte mit einer äußeren Abmessung von weniger als 25 cm können unabhängig vom Kauf eines Neugeräts in haushaltsüblichen Mengen abgegeben werden.

Rückgabe an unserem Firmenstandort:
AlphaLink Engineering GmbH
Bismarkstraße 10-12 (MEL 206-209)
10625 Berlin

Rückgabe in Deiner Nähe:
Wir stellen Dir eine Paketmarke zur Verfügung, mit der Du das Gerät kostenlos an uns zurücksenden kannst. Bitte kontaktiere uns per E-Mail unter *devcard@alphalink.aero* um die Rücksendung zu arrangieren.

Verpackungsinformationen:
Verpacke Dein Altgerät bitte sicher für den Transport. Solltest Du kein geeignetes Verpackungsmaterial zur Verfügung haben oder es bevorzugen, stellen wir Dir gerne eine geeignete Verpackung zur Verfügung.

Wir danken Dir für die Zusammenarbeit bei der ordnungsgemäßen Entsorgung von Elektrogeräten gemäß den Bestimmungen des Elektrogesetzes (ElektroG). Bei weiteren Fragen stehen wir Dir gerne zur Verfügung.

EU-Konformitätserklärung

Der Hersteller/Inverkehrbringer
AlphaLink Engineering GmbH
Bismarkstraße 10-12 - 10625 Berlin

erklärt hiermit, dass sich das folgende Produkt

Produktbezeichnung: **Devcard®**
Modellbezeichnung: **Devcard STM32F103C8T6**
Beschreibung/Verwendungszweck: **Prototyping/Versuchsaufbau**

bei bestimmungsgemäßer Verwendung in Übereinstimmung mit den grundlegenden Anforderungen der folgenden Richtlinien befindet:

Richtlinien: RoHS-Richtlinie 2011/65/EU

Folgende harmonisierte Normen wurden angewandt:

IEC 61191-1 bis IEC 61191-4

EN 61340-5-3

EN 82079-1

Berlin, 23.06.23
-Erstfassung-

Daniel Cracau
Geschäftsführung
AlphaLink Engineering GmbH