

# C4EV3 – PRAKTIKUM 1: ERSTELLEN EINES PROJEKTS

Geschrieben von Felix Krawczyk | 12. September 2017 | Lern- & Lehrmaterialien, Programmiersprachen | 4 🗨️ | ★★★★★

## c4ev3

### //Praktikum

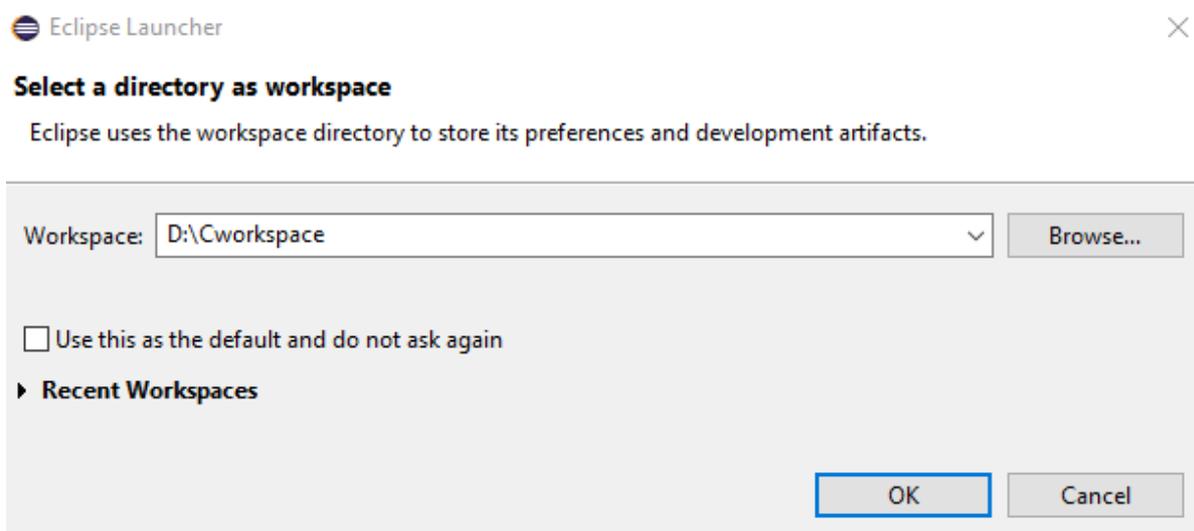
Zuerst zeigen wir dir in dieser Praktikumsaufgabe wie du Projekte erstellen kannst. Das ist wichtig für alle nachfolgenden Praktikumsaufgaben. Als erstes wird die Programmierumgebung **Eclipse**, in die wir die c4ev3-Umgebung installiert haben, geöffnet. Sollte c4ev3 noch nicht installiert sein, haben wir eine [Installationsanleitung](#) für dich zusammengestellt.

## ZWECK

Projekte sind der Grundpfeiler unserer Programme. Eclipse verwaltet darüber zum Beispiel den Programmcode und Einstellungen. Wir müssen also immer erst ein Projekt anlegen, uns mit den Einstellungen auseinandersetzen und können dann in diesem Projekt unser Programm erstellen. Daher fangen wir in diesem Praktikum mit der Projekterstellung an.

## DAS ERSTE PROJEKT IN ECLIPSE

Wenn Eclipse gestartet wird, erscheint zunächst eine Abfrage nach dem sogenannten **Workspace-Pfad** (siehe [Abbildung 1](#)). Wer möchte, kann einen eigenen Ordner auswählen in dem Eclipse alle wichtigen Daten für die Projekte hinterlegt. An sich könnt ihr den Standard-Pfad jedoch lassen und auf „OK“ klicken. Wer Eclipse zum ersten Mal öffnet, muss unter Umständen etwas Geduld mitbringen. Der erste Start nimmt etwas Zeit in Anspruch.



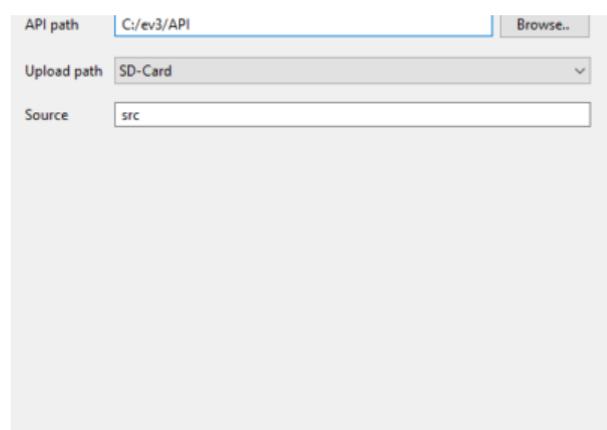
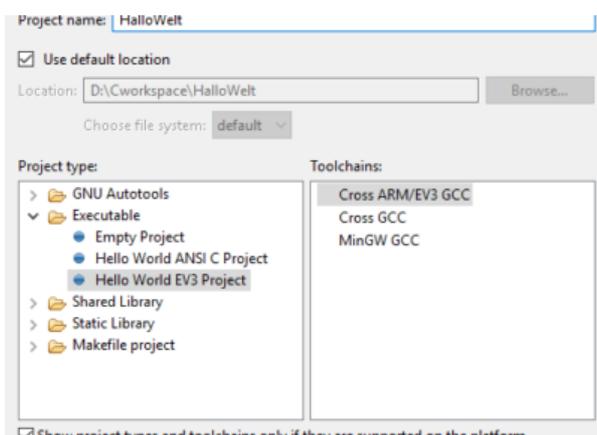
Um das erste Projekt anzulegen, klickt bitte in Eclipse oben in der Menüleiste auf **File > New > C Project**. Daraufhin öffnet sich ein neues Fenster mit Titel **C Project**, in der einige Einstellungen vorzunehmen sind (siehe [Abbildung 2](#)). In diesem Fenster könnt ihr oben unter **Project name** den Namen eures Projekts festlegen. Bitte beachtet, dass der Projektname keine Leerzeichen enthalten darf! Wir haben zum Beispiel den Namen **HalloWelt** für unser erstes Testprojekt genommen. In der linken Auswahlliste mit dem Titel **Project type** müsst ihr im Ordner Executable den vorgefertigten Projekttyp **Hello World EV3 Project** auswählen. Zum Auswählen reicht es aus, den Eintrag einmal anzuklicken, wodurch er grau hinterlegt wird. Wichtig ist, dass in der rechten Auswahlliste mit dem Titel **Toolchains** der Eintrag **Cross ARM/EV3 GCC** ausgewählt wird. Diese Auswahl legt fest, dass die Zielplattform die ARM-Plattform für den **LEGO® Mindstorms®**

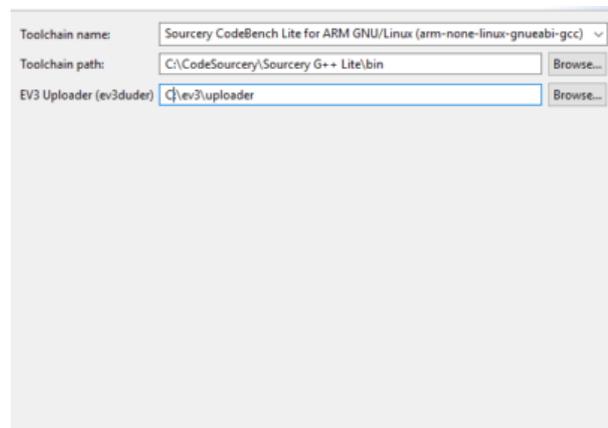
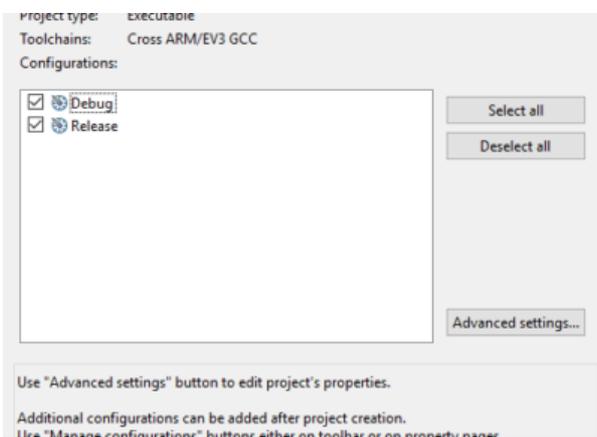
EV3 ist. Denn dafür wollen wir ja Projekte erstellen. Ist alles eingegeben geht es weiter auf Next.

Jetzt sind die Basic Settings an der Reihe (siehe [Abbildung 3](#)). Zuerst kontrollieren wir, dass der API-Pfad auf den Ordner mit dem API verweist. Wer unsere Installationsanleitung genutzt hat, sollte den Pfad „**C:\ev3\API**“ hier eintragen.

- C:\ev3\API

Mit einem API, ausgeschrieben **Application Programming Interface**, wird eine Programmierschnittstelle gemeint. Diese Schnittstelle können Programme nutzen, um zum Beispiel Hardware anzusteuern. In diesem Fall ist das der LEGO® Mindstorms® EV3. Danach können wir den späteren Speicherort des zukünftigen HalloWelt Programms auf dem EV3 festlegen. Hierfür muss der Roboter **nicht** angeschlossen sein, da das Laden auf den Roboter später kommt. Der Source-Ordner **src** müsst ihr nicht ändern. Aktuell wird als Speicherort auf dem EV3 nur die Option SD-Card unterstützt (Stand: **September 2017**). Darum müssen wir für den EV3 aktuell eine microSD-Karte nutzen, um c4ev3-Programme auf dem EV3 auszuführen. Über die Schaltfläche Next kommt ihr anschließend zu dem Fenster mit dem Titel **Select Configurations** (siehe [Abbildung 4](#)). In der Auswahlliste sollten die **Debug** und die **Release Configurations** ausgewählt sein.





Das Fenster mit dem Titel Cross GNU EV3 Toolchain (siehe [Abbildung 5](#)) erreicht ihr mit einem weiteren Klick auf die Schaltfläche Next. Wichtig ist hier nun, dass das c4ev3-Toolchain in das Projekt eingebunden wird. Dazu sollte als erstes kontrolliert werden, dass der Toolchain Name **Sourcery CodeBench Lite for ARM GNU/Linux (arm-none-linux-gnueabi-gcc)** lautet. Sollte die c4ev3-Umgebung nach [unserer Anleitung](#) installiert worden sein, ist unter Toolchain Path der Pfad „**C:\Program Files (x86)\CodeSourcery\Sourcery G++ Lite**“ und beim EV3 Uploader der Pfad „**C:\ev3\uploader**“ einzugeben beziehungsweise auszuwählen.

- C:\Program Files (x86)\CodeSourcery\Sourcery G++ Lite
- C:\ev3\uploader

Ist dies geschehen, kann das Projekt nun mit einem Klick auf Finish angelegt werden. Dadurch öffnet sich automatisch das HalloWelt-Projekt (siehe [Abbildung 6](#)).

```
2+ \file      HalloWelt.c
7
8 #include <ev3.h>
9
10 int main(void)
11 {
12     //TODO Place here your variables
13
14     InitEV3();
15
16     //TODO Place here your code
17     LcdPrintf(1, "Hello World!\n");
18     Wait(SEC_2);
19
20     FreeEV3();
21     return 0;
22 }
23
```

## DER ERSTE TEST UNSERES PROJEKTS

Das fertige Projekt wollen wir im nächsten Schritt auf den EV3 bringen. Dazu muss der EV3 angeschaltet und eine microSD-Karte eingelegt sein. Wichtig ist jetzt noch, dass der EV3 mit dem PC über USB-Kabel verbunden ist. Ist dies geschehen, muss als nächstes das aktuelle Projekt gespeichert werden. Entweder auf dem Symbol in der oberen Schnellzugriff-Leiste oder mit der Tastenkombination **Strg + S**. Danach müssen wir den Code kompilieren. Das geht mit dem Hammer-Symbol in der oberen Schnellzugriff-Leiste oder mit der Tastenkombination **Strg + B**. Nun ist der Code bereit, um auf den EV3 geladen zu werden. Dafür ist zunächst darauf zu achten, dass das aktuelle Projekt im **Projekt Explorer** ausgewählt (grau hinterlegt) ist. Danach kann man in der Schnellzugriff-Leiste auf das EV3-Symbol oder im Reiter **EV3 > Upload** klicken. Ein Fortschrittsbalken erscheint. Ist dieser durchgelaufen findet man das Programm **HalloWelt** auf dem EV3-Brick auf der microSD-Karte im Ordner **myapps**. Geschafft! Starten wir nun das Programm. Wenn alles richtig funktioniert hat, sollte für zwei Sekunden in der oberen linken Ecke „Hello World!“ stehen. Das Projekt ist nun erstellt und das vorgefertigte **HalloWelt**-Projekt ist geöffnet.

## LÖSUNG

Nun ist es an der Zeit sich etwas mit der Syntax von der Programmiersprache C zu befassen. Wie schon anfangs erwähnt, ist es hilfreich, Vorkenntnisse in einer textuellen Programmiersprache mitzubringen. Denn im Vordergrund steht in diesem Tutorial die Besonderheiten von c4ev3 und kein Tutorial zur Programmiersprache C. Die Befehle, die für dieses Praktikum benutzt wurden, sind alle in der **EV3-API** zu finden auf [GitHub](#). In den folgenden Praktika wird die API zum wichtigen Begleiter. Ihr könnt das API-Dokument herunterladen. Wir bieten extra keinen eigenen Download an, da sich das Dokument

vermutlich häufig ändert wird und wir wollen nicht aus versehen eine veraltete Version zum Download anbieten. Das nachfolgende **Listing 1** zeigt den Quelltext des **HalloWelt**-Projekts.

Listing 1: Das erste "Hallo Welt" Programm

```
1  #include <ev3.h>
2
3  int main(void)
4  {
5
6      InitEV3();
7
8      LcdPrintf(1, "Hello World!");
9      Wait(SEC_2);
10
11     FreeEV3();
12     return 0;
13 }
```

TEILEN:          

BEWERTEN:

< ZURÜCK

NÄCHSTER >

[c4ev3 – Auftakt zum Praktikum](#)

[WRO 2018: Die Neuigkeiten im Überblick](#)

## ÜBER DEN AUTOR



### Felix Krawczyk

Felix interessiert sich leidenschaftlich für Informatik und Elektrotechnik, die Kombination dieser Komponenten lebt er in seinem Studium und seinem Job voll aus. Dieses Wissen, an Schüler und Lehrer sowie Freunde und Bekannte zu vermitteln hat er sich als Ziel gesetzt.





## ÄHNLICHE BEITRÄGE



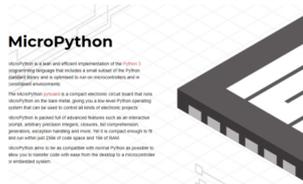
### c4ev3 – Auftakt zum Praktikum

12. September 2017



### MicroPython – Installationsanleitun g

11. Januar 2020



### MicroPython – LEGO® Education öffnet seine Plattform

24. Oktober 2019



### c4ev3 – Installationsanleitun g

25. August 2017

