Lerne programmieren mit der Tello-EDU

Ein paar faszinierende Beispiele, um Programmieren zu lernen

Erstellt von

Lukas Seglias
Christian Nussbaum
Luca Berger
Luca Ritz

Fachhochschule
Bern

2020
PUBLISHER

Inhaltsverzeichnis

Kapitel 1

Einführung

1.1 Installation

Das folgende Setup wird vorausgesetzt:

- Windows 10
- IDE Visual Studio Community 2019 oder höher mit C++ [1]
- CMake in der Version 3.18.2 [2]
- Python in der Version 3 [3] (Füge Python der Pfad-Variable hinzu)

Als Alternative kann auch CLion [4] als IDE verwendet werden. Diese ist jedoch nicht frei erhältlich. Achtung, da auch CLion den MSVC++ verwendet, muss Visual Studio Community 2019 ebenfalls installiert werden.

1.1.1 MS Visual Studio 2019 konfigurieren

Sobald sichergestellt ist, dass obig aufgeführte Voraussetzungen gegeben sind, kann die IDE MS Visual Studio 2019 gestartet werden. Führe die folgenden Schritte durch:

- Auf der rechten Seite siehst du diverse Optionen, um ein Projekt zu öffnen oder zu erstellen. Wähle dort 'Lokalen Ordner öffnen' aus.
- Navigiere zu deinem Verzeichnis, in das du das Projekt gespeichert hast.
- Öffne den Projektornder (meistens 'learn_with_tello').
- Nun wird das Projekt geöffnet und die CMake-Generierung gestartet, welche eventuell fehlschlagen wird. Sollte dies der Fall sein, so ist die CMake-Version des Visual-Studios zu gering und wir müssen diese manuell einstellen. Öffne in dem Fall den Explorer und navigiere in deinen Projektordner. Dort siehst du die Datei 'CMakeSettings.json'. Öffne diese in einem Editor und füge die folgende Zeile hinzu:

```
1    {
2        ...
3        "cmakeExecutable": "D:\\Prog\\CMake-3.18.2\\bin\\cmake.exe"
4        ...
5    }
```

[1] https://visualstudio.microsoft.com/de/downloads/
[2] https://cmake.org/download/
[3] https://www.python.org/downloads/windows/
[4] https://www.jetbrains.com/de-de/clion/download/

- Sobald diese Änderungen gemacht sind und du zurück ins Visual Studio wechselst, siehst du, dass CMake bereits die Generierung gestartet hat. CMake macht nun in einem ersten Schritt das Basissetup, um der IDE mitzuteilen, wie das Projekt aufgebaut und wie die 'targets' generiert werden sollen.

Sobald CMake fertig ist, kannst du nun die nachfolgend aufgeführten 'Startelemente' erstellen. Um dies zu bewerkstelligen, verschaffst du dir zuerst einen kleinen Überblick der IDE. Im Zentrum steht das Code-Fenster, welches den meisten Platz einnimmt. Auf der rechten Seite siehst du den Projektmappen-Explorer. Dort werden die verschiedenen Verzeichnisse sowie der Code gelistet. Oben kannst du einen grünen Play-Button sehen, daneben steht zu Beginn 'Startelement auswählen'. Klappe dieses auf und du solltest mindestens folgende Startelemente sehen:

00_base_module.dll Beinhaltet Basiseinstellungen für die Drohne(n)

01_keyboard_module.dll Die ersten paar Übungen.

01_keyboard_module_solution.dll Die Lösungen zu den Übungen.

99_template.dll Dies ist eine Vorlage für neue Module, das kann ignoriert werden.

app_basic.exe Die Hauptapplikation (Um diese zu starten, drücke den grünen Play-Button hinter dem Dropdown mit den 'targets')

app_common_video.dll Eine Bibliothek mit Code, den Module verwenden können.

Um ein Startelement zu kompilieren, muss dieses ausgewählt werden. Danach drückst du den Play-Button, um den Build zu starten.
Für den Anfang reicht es, wenn die folgenden Startelemente erstellt werden (Achtung, nach dem Build einer .dll wird ein Fehler angezeigt, dass diese nicht ausgeführt werden kann. Das ist ganz normal, klicke den Fehler einfach weg und schliesse die automatisch geöffnete Datei launch.vs.json ohne zu speichern):

- 00_base_module.dll
- 01_keyboard_module.dll
- app_common_video.dll
- app_basic.exe

Schliesse die nun geöffnete Anwendung.
Wenn die Builds allesamt ohne Fehler durchlaufen, dann kannst du deine Tello-EDU-Drohne starten. Du solltest sie zuerst mit der offiziellen App kalibrieren. Stelle sie danach auf den Boden, wo genug Platz drumherum und nach oben ist. Die Drohne sollte gelb blinken, das heisst, sie wartet auf eine Verbindung. Wähle in den WLAN-Einstellungen das Tello-EDU WLAN aus.
Wähle nun das Startelement 'app_basic' aus und drücke den Play-Button. Um deine Umgebung zu testen, drücke den Button 'Take off' in der Applikation. Wenn die Drohne fliegt und du das Kamerabild der Drohne siehst, hast du alles richtig gemacht und kannst nun mit den Übungen starten. Drücke den Knopf nochmals, um die Drohne zu landen und schliesse die Applikation.

1.1.2 CLion konfigurieren

Wenn CLion als IDE verwendet wird, so müssen die folgenden Schritte durchgeführt werden. In einem ersten Schritt wird CLion geöffnet und über 'File - New Project' ein neues Projekt ersetllt. Nun muss das höchste CMakeLists.txt file des Repositorys 'learn_with_tello' ausgewählt werden. Danach kann die Build-Umgebung eingerichtet werden.

- Öffne 'File - Settings'
- Navigiere zu 'Build, Execution, Deployment'
- Wähle 'Toolchains' aus
- Prüfe, ob eine Toolchain namens 'Visual Studio' verfügbar ist. Wenn ja, erübrigen sich die weiteren Schritte.
- Drücke den 'Add'-Button in der oberen linken Ecke
- Nun muss die Umgebung ausgewählt werden. Beachte das nachfolgende Bild. Es muss mindestens eine CMake-Version 3.18.2 verfügbar sein.

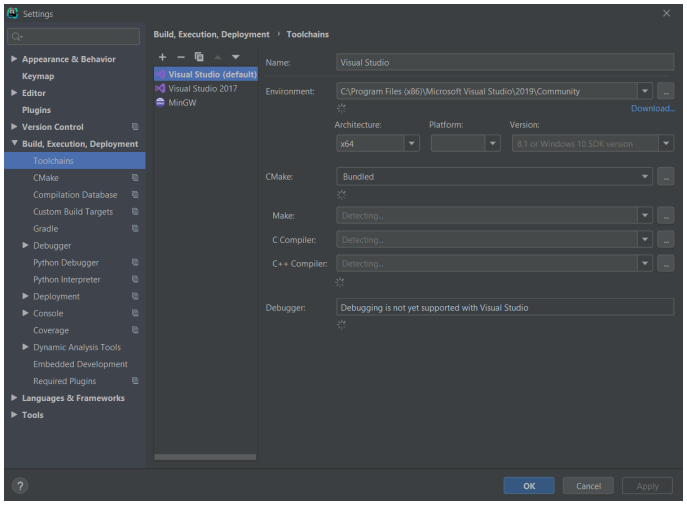

Abbildung 1.1: CLion toolchain Konfiguration

Nun bist du in der Lage, das Projekt im 'Debug'-Modus zu erstellen. Damit ist gemeint, dass zu jeder Library oder jeder ausführbaren-Datei (executable) zusätzlicher Code der Applikation hinzugefügt wird. Dadurch wird das Binary ungefähr zehn mal langsamer. Wenn du die Applikation im Release-Modus erstellen willst, dann musst du die folgenden Schritte durchführen. Im Release-Modus kann eine Applikation nicht mehr schrittweise durchlaufen werden, dafür ist sie viel performanter. Dieser Modus wird verwendet, wenn eine Applikation fertig ist und sich nicht mehr in der Entwicklungsphase befindet.

- Öffne 'File - Settings'
- Navigiere nach 'Build, Execution, Deployment'
- Wähle 'CMake' aus
- Prüfe, ob ein Profil namens 'Release' verfügbar ist. Wenn ja, dann ist bereits alles eingerichtet.
- Drücke den 'Add'-Button in der oberen linken Ecke.
- Ein neuer Eintrag wird erstellt. Benenne ihn 'Release', wenn dies nicht schon der Fall ist.

Nun werden die CMake-Konfigurationen neu erstellt. Nach diesem Vorgang, kannst du einen Blick in die obere rechte Ecke von CLion werfen. Dort siehst du ein grünes Hammersymbol. Daneben befindet sich ein Dropdown, wo die verschiedenen 'targets' (Ziele) ausgewählt werden können. Die Konfiguration wird dahinter ausgegeben. Dies ist entweder 'Debug' oder 'Release'. Wenn du das Dropdown ausklappst, kannst du sehen, dass du zwischen diesen Konfigurationen 'Debug' und 'Release' sowie allen 'targets' wechseln kannst. Ein 'target' wird in diesem Kontext als library oder executable-file verstanden. Es kann aber auch etwas spezielleres sein wie zum Beispiel ein Vorgang, um Dateien von einem Verzeichnis in ein anderes zu kopieren.
Die folgenden 'targets' sollten vorhanden sein (Es werden nicht alle aufgeführt):

00_base_module Beinhaltet Basiseinstellungen für die Drohne(n)

01_keyboard_module Die ersten paar Übungen.

01_keyboard_module_solution Die Lösungen zu den Übungen.

99_template Dies ist eine Vorlage für neue Module, das kann ignoriert werden.

app_basic Die Hauptapplikation (Um diese zu starten, drücke den grünen Play-Button hinter dem Dropdown mit den 'targets')

app_common Eine Bibliothek mit Code, den alle Module verwenden.

app_common_video Eine Bibliothek mit Code, den Module verwenden können.

Um ein 'target' zu kompilieren, muss dieses im Dropdown ausgewählt werden. Danach wird der grüne Hammer daneben geklickt. Für den Anfang reicht es, wenn die folgenden 'targets' erstellt werden:

- app_basic
- 00_base_module
- 01_keyboard_module

Wenn die Builds allesamt ohne Fehler durchlaufen, dann kannst du deine Tello-EDU-Drohne starten. Du solltest sie zuerst mit der offiziellen App kalibrieren. Stelle sie danach auf den Boden, wo genug Platz drumherum und nach oben ist. Die Drohne sollte gelb blinken, das heisst, sie wartet auf eine Verbindung. Wähle in den WLAN-Einstellungen das Tello-EDU WLAN aus.
Wähle nun das 'target' 'app_basic' aus und drücke den Play-Button. Um deine Umgebung zu testen, drücke den Button "Take off' in der Applikation. Wenn die Drohne fliegt und du ein Bild siehst, hast du alles richtig gemacht und kannst nun mit den Übungen starten. Drücke den Knopf nochmals, um die Drohne zu landen und schliesse die Applikation.

1.2 Architektur

Das Projekt besteht aus einer ausführbaren Applikation (app_basic), welche grundsätzlich nur alle Plugins lädt. Die Plugins bilden die eigentlich wichtigen Teile der Applikation, sie beinhalten die Übungen und Lösungen. Jedes Plugin ist eine in sich abgeschlossene Anwendung, welche für einen spezifischen Use-Case erstellt wurde, z.B. das Fliegen der Drohne mithilfe der Tastatur oder das Fliegen der Drohne über Objekterkennung.

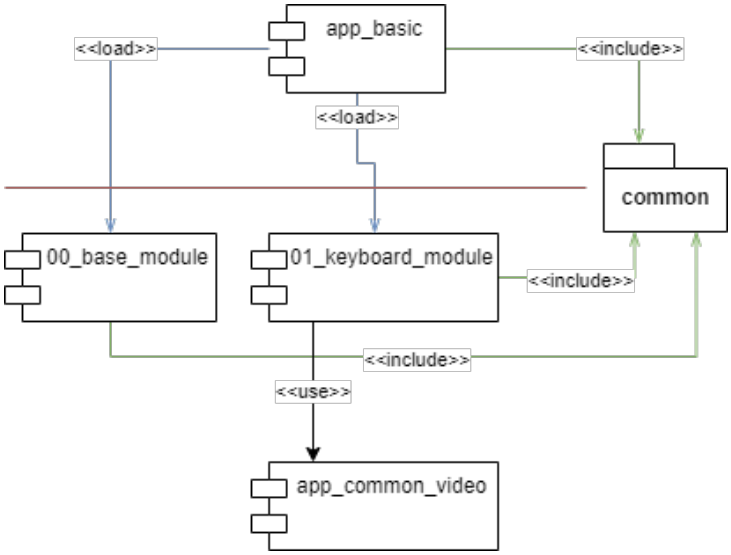

Abbildung 1.2: Architektur

Wenn du in einem der Module (Plugin) Änderungen gemacht hast, dann musst du dieses Modul neu kompilieren. Wähle dazu dieses Modul als Startlement in Visual Studio 2019 oder als 'target' in CLion aus und führe die Kompilierung durch. Wenn du anschliessend die Hauptapplikation startest, wirst du deine Änderungen sehen. Wenn du bei einer Aufgabe nicht weiter wissen solltest, kannst du einen Blick in die Referenzimplementation werfen. Diese enden mit der Bezeichnung '_solution'. Jeder Übungsabschnitt beinhaltet die entsprechenden Lösungen am Ende des Kapitels. Und nun, viel Spass bei den Übungen.

Kapitel 2

Grundverständnis

In diesem Kapitel befassen wir uns mit den ersten paar Aufgaben und wollen ein Grundlagenverständnis des Computers und seiner Funktionsweise aufbauen, welches wir später verwenden, um allfällige Probleme oder Verhaltensweisen besser verstehen zu können. Hierbei stellt sich also die Frage, wie so ein Gerät aufgebaut ist und wie die einzelnen Komponenten zusammenspielen, die wir hauptsächlich verwenden. Wir starten daher erstmal mit einigen Komponenten und lösen kleinere Übungen dazu.

2.1 Der Arbeiter

Die CPU bildet das Herzstück eines Computers. Sie ist zuständig, um verschiedene Instruktionen auszuführen. Wenn z.B. zwei Zahlen addiert werden sollen, dann ist die CPU das Stück Hardware, welches diese Arbeit erledigt.

Abbildung 2.1: Intel CPU

Nun fragt man sich zurecht, wie die CPU denn solche Aufgaben bewältigen kann? Dazu muss man wissen, dass die CPU selbst wiederum aus mehreren Teilen besteht, die alle eine bestimmte Aufgabe erfüllen. Es sind dies im Wesentlichen:

Control unit Diese Einheit steuert den ganzen Ablauf der CPU. Sie sagt den anderen Komponenten im Gerät und in der CPU selbst, was sie wann tun sollen und steuert so das ganze Gerät.[Unb20c]

Arithmetic logic unit Wenn arithmetische oder logische Operationen durchgeführt werden sollen, dann ist diese Einheit wichtig. Wenn z.B. 4 + 5 gerechnet werden soll, dann wird die ALU damit beauftragt.[Unb20b]

Address generation unit Will die CPU auf den Hauptspeicher zugreifen, dann wird in dieser Einheit die korrekte Adresse berechnet. Was das genau bedeutet wird im nächsten Abschnitt erklärt.[Unb20a]

Die CPU beinhaltet noch mehr solcher Einheiten, auf diese wird aber nun nicht mehr weiter eingegangen.

2.1.1 Übung 1 - Zahlenspiel

Es wird Zeit für die erste kleine Übung. Wir wollen darin das Verständnis stärken, wie Zahlen überhaupt in einem Computer dargestellt werden und wie wir in der Lage sind, diese zu addieren. Du hast sicher schon gehört, dass ein Computer mit 1en und 0en arbeitet. TODO: Go on

2.2 Übung 1 - Grundverständnis

2.3 Exercise 1 - Let the drone fly

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Abbildungsverzeichnis

Literatur

[Unb20a] Unbekannt. *Address Generation Unit*. 2020. URL: https://en.wikipedia.org/wiki/Central_processing_unit#Address_generation_unit (besucht am 05. 10. 2020).

[Unb20b] Unbekannt. *Arithmetic Logic Unit*. 2020. URL: https://en.wikipedia.org/wiki/Central_processing_unit#Arithmetic_logic_unit (besucht am 05. 10. 2020).

[Unb20c] Unbekannt. *Control Unit*. 2020. URL: https://en.wikipedia.org/wiki/Central_processing_unit#Control_unit (besucht am 05. 10. 2020).

Glossar

IDE Eine integrierte Entwicklungsumgebung stellt eine Sammlung verschiedener Werkzeuge zur Softwareentwicklung zur Verfügung..

CMake CMake generiert Makefiles, welche Anweisungen zum Erstellen einer Software enthalten..

Python Python ist eine Programmiersprache.

Plugin Ein Plugin ist eine Softwarekomponente, die eine bestehende Software um Funktionalität erweitert. Diese wird zur Laufzeit eingebunden und kann nicht ohne die Hauptanwendung ausgeführt werden..

CPU Central Processing Unit.