

---

# 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

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Installation . . . . .	2
1.1.1	Configure CLion . . . . .	2
1.2	Architecture . . . . .	5
<b>2</b>	<b>Fly with keyboard</b>	<b>6</b>
2.1	Exercise 1 - Let the drone fly . . . . .	6
2.2	Exercise 1 - Let the drone fly . . . . .	6
	<b>Abbildungsverzeichnis</b>	<b>7</b>
	<b>Glossar</b>	<b>8</b>

# Kapitel 1

## Introduction

### 1.1 Installation

To start your programming adventure, you must have the following setup:

- Windows 10
- IDE Visual Studio 2019 mit C++
- CMake in der Version 3.18.2

We recommend CLion as IDE. Later, we will also provide the possibility to use Microsoft Visual-Studio (MSVC).

#### 1.1.1 Configure CLion

Once you've installed the prerequisites, you have to configure CLion to be able to build the project. In a first step, open CLion and press 'File - New Project'. Select the top level CMakeLists.txt file and open it as a project. Now, we can start with the build environment.

- Open 'File - Settings'
- Navigate to 'Build, Execution, Deployment'
- Select 'Toolchains'
- Check if a toolchain called something like 'Visual Studio' is available. If yes, you don't have to go further with these steps.
- Press the 'Add' Button on the upper left corner
- Now you have to select the environment. Take a look at the following picture. You require at least a CMake-Version of 3.15 (The bundled one should be fine).

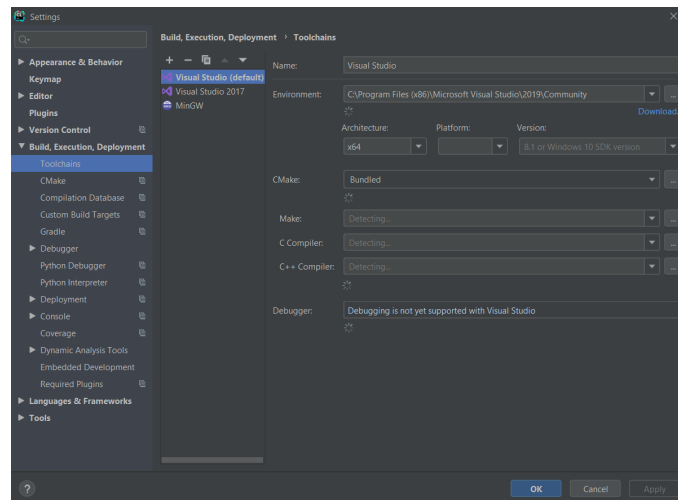


Abbildung 1.1: CLion toolchain configuration

Now you are able to build the project in 'Debug' mode. This means, that in every library or executable file some additional code gets added to debug your application. This makes your binaries approximately ten times slower. If you want to build a project in 'Release' mode, you have to do the following steps.

- Open 'File - Settings'
- Navigate to 'Build, Execution, Deployment'
- Select 'CMake'
- Check if a profile called 'Release' is available. If yes, you don't have to go further with these steps.
- Press the 'Add' Button on the upper left corner
- Name it 'Release' if it is not already named like that.

Wait till CMake has imported the changes. After the CMake import, have a look on the upper right corner of the application. You see a green hammer symbol. Next to that, there is a dropdown where you can select the 'targets'. The release mode is printed behind the target. Per default, this is 'Debug' (you can change that in the settings). If you open this dropdown, you can see, that you're able to switch between these modes and you can select all targets. A target in this context means in the most cases a build of a library or executable file. But it can also be some special stuff like copying files to directory. You should see at least the following targets:

**app\_basic** The main application (You have to press the green play-button behind the dropdown.)

**copy\_libraries** Copies the external shared libs to the correct place.

**copy\_modules** Copies the modules to the correct place.

**00\_base\_module** Contains all basic setup stuff.

**01\_keyboard\_module** The first set of exercises.

**01\_keyboard\_module\_solution** The first set of solutions.

**gmock** Test libs used by external project.

**gmock\_main** Test libs used by external project.

**gtest** Test libs used by external project.

**gtest\_main** Test libs used by external project.

**spdlog** Logging libs used by external project.

**tello** Library to interact with your Tello-EDU drone.

**tello\_test** Test executable used by external project.

**copy\_shared\_test\_libraries** Target of external project

**shared\_linking\_test** Target of external project.

To compile a target, select it in the dropdown and press the green hammer next to it. For the first time, you should build the following targets:

- app\_basic
- 00\_base\_module
- 01\_keyboard\_module

The builds should run without any exceptions. Now you can start your Tello-EDU drone. Put it somewhere on the ground, where it has some space around it. Then select the WIFI-Settings on your computer and change into your Tello-EDU WLAN. The drone should blink yellow, this means, it waits to a connection.

Then select the target 'app\_basic' and press the play button. To test your environment, press the 'Take off' button on the center of the screen. If the drone is flying, you've done all correctly and you are able to start (press the button again to land the drone). Congratulations, you can close the application.

## 1.2 Architecture

The project consists of an executable (app\_basic), which basically just loads all the plugins. The plugins are the essential parts. They contain the exercises and solutions. Each plugin is a self-contained application, which is specialized for a specific use-case e.g. flying with arrow keys, fly by recognitions etc.

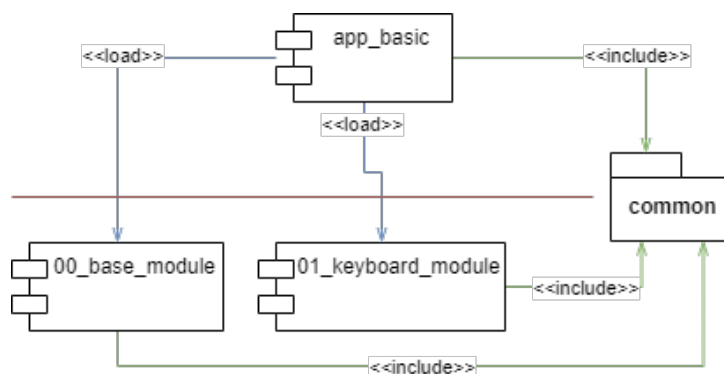


Abbildung 1.2: Architecture  
usw.

When you added your changes to one of the modules, select that module in the target dropdown, rebuild it and then start the target 'app\_basic' to see your changes. For a reference implementation, have a look at the solutions. They are postfix with '\_solution'. Each exercise section in this document has a solution section at the end of a chapter. If you're surce you have the correct solution, take a look.

And now we wish you a lot of fun.

# Kapitel 2

## Fly with keyboard

In this first chapter you will learn how you can fly and control the drone with your keyboard. Furthermore, you will see how different mathematical functions take effect in the real world. We will map the time a key was pressed to the speed the tello has to flight in a direction by these functions. Beside these things, you will have your first look into C++ code.

### 2.1 Exercise 1 - Let the drone fly

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur 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.

### 2.2 Exercise 1 - Let the drone fly

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur 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

1.1	CLion toolchain configuration . . . . .	3
1.2	Architecture . . . . .	5



# 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..