

## Code::Blocks

### 1. Download the AlfES® library

- Got to: [https://github.com/Fraunhofer-IMS/AlfES\\_for\\_Arduino](https://github.com/Fraunhofer-IMS/AlfES_for_Arduino)
- Download AlfES® as ZIP archive using "Download ZIP"

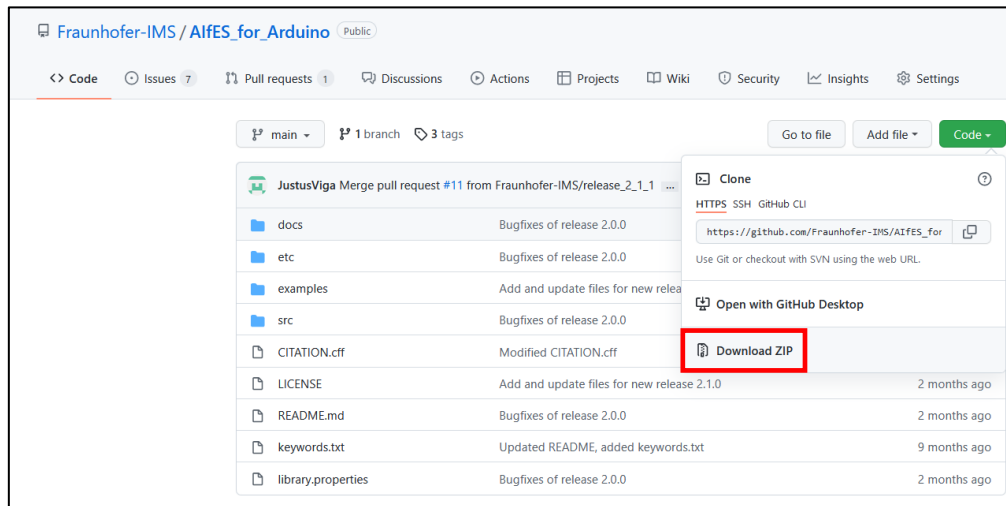


Figure 1: Download AlfES, by using the "Download Zip" Function

### 2. Prepare AlfES®

We have different configuration files for AlfES® to make it compatible with different IDEs. In the default configuration it is of course optimized for the Arduino IDE. Arduino specific functions like `"serial.print()"` are used. By changing the configuration files, you can make AlfES® universal. Then the classic `"printf()"` functions are used. You can then use AlfES® almost everywhere, on the PC, in a microcontroller IDE of your choice, even in a hardware simulator.

How this works is explained here step by step:

- Extract the ZIP file
- Navigate to the `"src/"` folder
- Replace the files `"aifes_config.cpp"` and `"aifes_config.h"` in the `"src/"` directory by the ones found in `"etc/aifes_configurations/pc/"`
- Note that the new config file has the extension `*.c` and not `*.cpp`
- That's it

### 3. Use Arm CMSIS

Of course, you can also use CMSIS in the universal version if you use e.g. another microcontroller IDE. For the installation please follow this [guide](#).

The only difference is that you don't make the changes in the file structure of the Arduino IDE, but in your just downloaded copy.



## Code::Blocks

---

### 4. New "main.c" for PC and Code::Blocks online tutorial

For Code::Blocks there is also an online [tutorial](#) on Hackster.

Especially for PC IDEs we have created a new "main.c":

- Download the **main.c** from the tutorial
- Copy the "**main.c**" into the "**src/**" folder

The **main.c** contains one example each for AlfES and AlfES-Express.

### 5. Code::Blocks installation

Download the [Code::Blocks IDE](#) and install it (Note: Download the "*mingw*" version, it includes the GCC/G++/GFortran compiler)

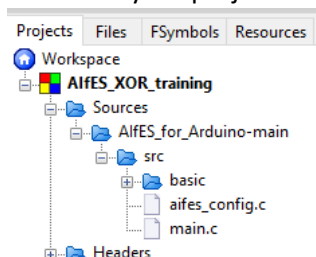
### 6. Create a new project

Create a new project in Code::Blocks:

- Create a new Project: File -> new -> Project...
- Select "*Console application*"
- Select "*C*" as language
- Think of a project name. We have chosen "*AlfES\_XOR\_training*"
- Compiler: GNU GCC Compiler
- Create the project

### 7. Integrate AlfES® in Code::Blocks

- Delete the automatically generated *main.c* file from the project: right-click on **main.c** -> remove file from project
- Navigate to: *Project* -> *Add files recursively...*
- Choose the **AlfES\_for\_Arduino-main\src** folder (The copied **main.c** should also be located there)
- Confirm everything
- After that your project should look like this:



- Navigate to: *Project* -> *Build options* -> *Search directories* -> *Compiler*
- Press the Button Add and navigate to the **AlfES\_for\_Arduino-main\src** folder



## Code::Blocks

---

- Confirm Keep this as a relative path? with yes
- Save the project
- Now AlfES is integrated into your project

Code::Blocks separates the header and source code files by itself.

## 8. Legal Disclaimer

The author of this installation guide assumes no responsibility or liability for any errors or omissions in the contents of this document that may result in damage to any kind of hardware or software. The information contained in this manual is provided without warranty of completeness, functionality, accuracy, usefulness or timeliness...