ARSetupHelp

TechStack:
Windows
Godot + GDExtension API
Python (for Scons)
OpenCV

Remark. Here are some tips and quick version to the Tutorial given in the official documentations..
it is not complete and many things will probably be missing!

Godot + CPP

1. Download Godot Engine - Free and open source 2D and 3D game engine
2. refer to GDExtension — Godot Engine (stable) documentation in English or GDExtension C++ example — Godot Engine (stable) documentation in English

Python.org Stuff for SCons: A software construction tool - SCons

1. If not installed, install python since it is the recommended way to get SCons.
    1. There is another way online on the Scons website, but I did not tested it!
2. During the installation enable everything with path!
3. open a console with pip
4. `pip install SCons` see SCons · PyPI
5. if python is setupped correctly one can use the `scons -v` command
   1. if not one need to set the path variable!
   2. type in `python -mSCons --version` or `pyVERSION -mSCons --version`
   3. It will give you a path where scons is installed (in the best case)
   4. Try to follow this path and open the respective folder of PythonVERSION
   5. And add the path of the `Scripts` folder to your windows environment stuff
   6. Why systempath stuff? if not your command could look like this :D
   `C:\Users\YOURNAME\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-packages\Python310\Scripts\scons.exe plattforms=windows`
   instead of `scons plattforms=windows`
6. try again with `scons -v` and it should work

1. if not possible, have fun

GDExtension C++ example — Godot Engine (stable) documentation in English

First Get a copy of the repository.

If you are versioning your project using Git, it is recommended to add it as a Git submodule:

```
mkdir gdextension_cpp_example
cd gdextension_cpp_example
git init
git submodule add -b 4.x https://github.com/godotengine/godot-cpp
cd godot-cpp
git submodule update --init


cd gdextension_cpp_example
git submodule update --init
```

Alternatively, you can also clone it to the project folder:

```
mkdir gdextension_cpp_example
cd gdextension_cpp_example
git clone -b 4.x https://github.com/godotengine/godot-cpp
```

Afterwards the command `godot --dump-extension-api` does not work in windows, except you added it as systemvariable as well (no tested).
The thing that works is to `cd` to the path where the godot executable is and use windows powershell and type `.\Godot_v4.4.1-stable_win64.exe --dump-extension-api`, normally just type godot and press `TAB` will autocomplete the name.

this will generate a `extension_api.json` file WHERE THE EXECUTABLE IS!
move it somewhere closer to the project or into the `gdextension_cpp_example` folder

in the `gdextension_cpp_example` folder

```
cd godot-cpp
scons platform=windows custom_api_file=<PATH_TO_FILE>
cd ..
```

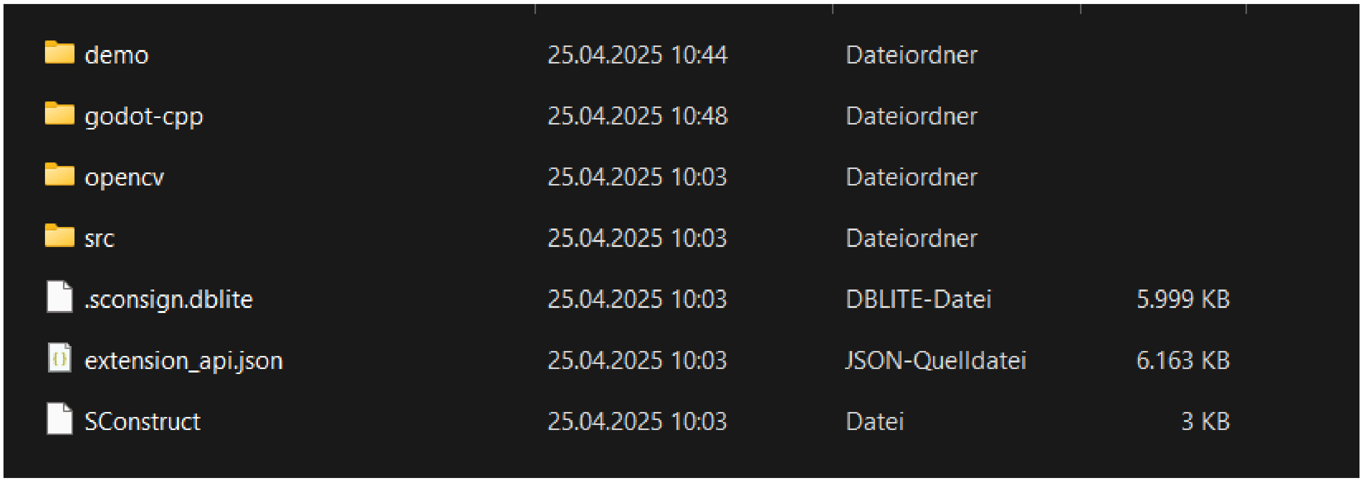
now just follow the documentation link

On each change off the src files you need to once again compile everything with `scons platform=windows` make sure to run the command in `Gdextension_cpp_example` and not in the sub directory `godot-cpp`

GODOT Opencv

Pls. follow the things written in the official document first!

the opencv link extracts an folder and stuff and insert it into the project next to demo, godot-cpp, src
paste them into your project



demo	25.04.2025 10:44	Dateiordner	
godot-cpp	25.04.2025 10:48	Dateiordner	
opencv	25.04.2025 10:03	Dateiordner	
src	25.04.2025 10:03	Dateiordner	
.sconsign.dblite	25.04.2025 10:03	DBLITE-Datei	5.999 KB
extension_api.json	25.04.2025 10:03	JSON-Quelldatei	6.163 KB
SConstruct	25.04.2025 10:03	Datei	3 KB

At least my setup looks like this.

Go to `gdextension_cpp_example\opencv\build\x64\vc16`
copy the 2 lib files from `v16\lib` to `demo\bin` and (optional??) copy the dll files from the `v16\bin` to `demo\bin`
depending of your version update the SConstruct file with the correct opencv version.

if correctly the line `camera.open(0, 1920, 1080);` should work
the line to open a video file e.g `camera.open_file("MarkerCube.mp4")` NEEDS THE ABSOLUTE PATH!!! e.g
`camera.open_file("C:\Users\YOURNAME\...\gdextension_cpp_example\demo\MarkerCube.mp4")
or if you put the markercube into the SAME FOLDER as your ar.tscn scene use

```
var real_path = ProjectSettings.globalize_path("res://MarkerCube.mp4")
camera.open_file(real_path)
```