

```

cmake_minimum_required(VERSION 3.13)

project(MyProject)

if(NOT CMAKE_BUILD_TYPE)
    set(CMAKE_BUILD_TYPE Release)
endif()

##### create target
#####

set(WILDCARD_SOURCE *.cpp)
set(WILDCARD_HEADER *.h *.hpp *.hxx)

file(GLOB_RECURSE Headers CONFIGURE_DEPENDS src/${WILDCARD_HEADER}
intermediate/code/${WILDCARD_HEADER})
file(GLOB_RECURSE Sources CONFIGURE_DEPENDS src/${WILDCARD_SOURCE}
intermediate/code/${WILDCARD_SOURCE})
add_library(MyProject SHARED ${Headers} ${Sources})

#####

##### project include-paths
#####

target_include_directories(MyProject
    PUBLIC
        $<BUILD_INTERFACE:${CMAKE_CURRENT_SOURCE_DIR}/intermediate/code>
        $<BUILD_INTERFACE:${CMAKE_CURRENT_SOURCE_DIR}/src>
    PRIVATE

$<BUILD_INTERFACE:${CMAKE_CURRENT_SOURCE_DIR}/deploy/${ARP_DEVICE}_${_ARP_SHORT_
DEVICE_VERSION}/include>

$<BUILD_INTERFACE:${CMAKE_CURRENT_SOURCE_DIR}/deploy/${ARP_DEVICE}_${_ARP_SHORT_
DEVICE_VERSION}/opencv3>
)
#####

##### include arp cmake module path
#####

list(INSERT CMAKE_MODULE_PATH 0 "${ARP_TOOLCHAIN_CMAKE_MODULE_PATH}")

#####

##### set link options
#####
# WARNING: Without --no-undefined the linker will not check, whether all
necessary #
# libraries are linked. When a library which is necessary is not
linked, #
# the firmware will crash and there will be NO indication why it
crashed. #
#####

```

```
#####

target_link_options(MyProject PRIVATE LINKER:--no-undefined)

#####

##### add link targets
#####

find_package(ArpDevice REQUIRED)
find_package(ArpProgramming REQUIRED)

find_package(OpenCV REQUIRED)
include_directories(${OpenCV_INCLUDE_DIRS})

find_library(AVCODEC_LIBRARY avcodec)
find_library(AVDEVICE_LIBRARY avdevice)
find_library(AVFILTER_LIBRARY avfilter)
find_library(AVFORMAT_LIBRARY avformat)
find_library(AVRESAMPLE_LIBRARY avresample)
find_library(AVUTIL_LIBRARY avutil)
find_library(SWRESAMPLE_LIBRARY swresample)
find_library(SWSAMPLE_LIBRARY swscale)

target_link_libraries(MyProject PRIVATE

ArpDevice

ArpProgramming

${OpenCV_LIBS}

${AVCODEC_LIBRARY}

${AVDEVICE_LIBRARY}

${AVFILTER_LIBRARY}

${AVFORMAT_LIBRARY}

${AVUTIL_LIBRARY}

${SWRESAMPLE_LIBRARY}

${SWSAMPLE_LIBRARY})

#####

##### install
#####

string(REGEX REPLACE "^.*\${ARP_DEVICE_VERSION}([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+).*" "\${1}"
_ARP_SHORT_DEVICE_VERSION ${ARP_DEVICE_VERSION})
install(TARGETS MyProject
        LIBRARY DESTINATION ${ARP_DEVICE}_${ARP_SHORT_DEVICE_VERSION}/${CONFIG}/lib
        ARCHIVE DESTINATION ${ARP_DEVICE}_${ARP_SHORT_DEVICE_VERSION}/${CONFIG}/lib
```

```
        RUNTIME DESTINATION
${ARP_DEVICE}_${_ARP_SHORT_DEVICE_VERSION}/${<CONFIG>/bin)
unset(_ARP_SHORT_DEVICE_VERSION)
```

```
#####
#####
```