# Generated Cmake Pico project file

cmake\_minimum\_required(VERSION 3.13)

set(CMAKE\_C\_STANDARD 11)

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_EXPORT\_COMPILE\_COMMANDS ON)

# Initialise pico\_sdk from installed location

# (note this can come from environment, CMake cache etc)

# == DO NOT EDIT THE FOLLOWING LINES for the Raspberry Pi Pico VS Code Extension to work ==

if(WIN32)

set(USERHOME $ENV{USERPROFILE})

else()

set(USERHOME $ENV{HOME})

endif()

set(sdkVersion 2.1.1)

set(toolchainVersion 14\_2\_Rel1)

set(picotoolVersion 2.1.1)

set(picoVscode ${USERHOME}/.pico-sdk/cmake/pico-vscode.cmake)

if (EXISTS ${picoVscode})

include(${picoVscode})

endif()

# ====================================================================================

set(PICO\_BOARD pico\_w CACHE STRING "Board type")

# Pull in Raspberry Pi Pico SDK (must be before project)

include(pico\_sdk\_import.cmake)

project(exemplo1\_superloopLed C CXX ASM)

# Initialise the Raspberry Pi Pico SDK

pico\_sdk\_init()

# Add executable. Default name is the project name, version 0.1

add\_executable(exemplo1\_superloopLed exemplo1\_superloopLed.c )

pico\_set\_program\_name(exemplo1\_superloopLed "exemplo1\_superloopLed")

pico\_set\_program\_version(exemplo1\_superloopLed "0.1")

# Modify the below lines to enable/disable output over UART/USB

pico\_enable\_stdio\_uart(exemplo1\_superloopLed 0)

pico\_enable\_stdio\_usb(exemplo1\_superloopLed 1)

# Add the standard library to the build

target\_link\_libraries(exemplo1\_superloopLed

pico\_stdlib)

# Add the standard include files to the build

target\_include\_directories(exemplo1\_superloopLed PRIVATE

${CMAKE\_CURRENT\_LIST\_DIR}

)

# Add any user requested libraries

target\_link\_libraries(exemplo1\_superloopLed

)

pico\_add\_extra\_outputs(exemplo1\_superloopLed)