```
######################################################################
##########
# \file Makefile
# \version 1.0
#
# \brief
# Top-level application make file.
#
######################################################################
##########
# \copyright
# Copyright 2018-2022, Cypress Semiconductor Corporation (an Infineon
company)
# SPDX-License-Identifier: Apache-2.0
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
# See the License for the specific language governing permissions and
# limitations under the License.
######################################################################
##########


######################################################################
##########
# Basic Configuration
######################################################################
##########

# Type of ModusToolbox Makefile Options include:
#
# COMBINED    -- Top Level Makefile usually for single standalone
application
# APPLICATION -- Top Level Makefile usually for multi project
application
# PROJECT     -- Project Makefile under Application
#
MTB_TYPE=COMBINED

# Target board/hardware (BSP).
# To change the target, it is recommended to use the Library manager
# ('make library-manager' from command line), which will also update
```

```
Eclipse IDE launch
# configurations.
TARGET=APP_CY8CKIT-062-WIFI-BT

# Name of application (used to derive name of final linked file).
#
# If APPNAME is edited, ensure to update or regenerate launch
# configurations for your IDE.
APPNAME=OurProj3

# Name of toolchain to use. Options include:
#
# GCC_ARM -- GCC provided with ModusToolbox software
# ARM     -- ARM Compiler (must be installed separately)
# IAR     -- IAR Compiler (must be installed separately)
#
# See also: CY_COMPILER_PATH below
TOOLCHAIN=GCC_ARM

# Default build configuration. Options include:
#
# Debug -- build with minimal optimizations, focus on debugging.
# Release -- build with full optimizations
# Custom -- build with custom configuration, set the optimization flag
in CFLAGS
#
# If CONFIG is manually edited, ensure to update or regenerate launch
configurations
# for your IDE.
CONFIG=Custom

# If set to "true" or "1", display full command-lines when building.
VERBOSE=


######################################################################
##########
# Advanced Configuration
######################################################################
##########

# Enable optional code that is ordinarily disabled by default.
#
# Available components depend on the specific targeted hardware and
firmware
# in use. In general, if you have
#
#    COMPONENTS=foo bar
#
# ... then code in directories named COMPONENT_foo and COMPONENT_bar
```

```
will be
# added to the build
#
COMPONENTS=FREERTOS WICED_BLE LWIP MBEDTLS SECURE_SOCKETS

# Like COMPONENTS, but disable optional code that was enabled by
default.
DISABLE_COMPONENTS=

# Choose the core where the application and Bluetooth LE Host stack
must run
# CM4 -- Application and AIROC BTSTACK(Host stack) run on CM4
# CM0P -- Application and AIROC BTSTACK(Host stack) run on CM0P
ifneq ($(TARGET), $(filter $(TARGET), APP_CYW920829M2EVB-01))
CORE=CM4
endif

# By default the build system automatically looks in the Makefile's
directory
# tree for source code and builds it. The SOURCES variable can be used
to
# manually add source code to the build process from a location not
searched
# by default, or otherwise not found by the build system.
SOURCES=

# Like SOURCES, but for include directories. Value should be paths to
# directories (without a leading -I).
INCLUDES=./configs ./configs/COMPONENT_$(CORE)

# Custom configuration of mbedtls library.
MBEDTLSFLAGS = MBEDTLS_USER_CONFIG_FILE='"mbedtls_user_config.h"'
# Add additional defines to the build process (without a leading -D).
DEFINES=$(MBEDTLSFLAGS) CYBSP_WIFI_CAPABLE
CY_RETARGET_IO_CONVERT_LF_TO_CRLF CY_RTOS_AWARE
DISABLE_MBEDTLS_ACCELERATION
#DEFINES+=COMPONENT_43907
DEFINES+=NDEBUG
#DEFINES+=DISABLE_MBEDTLS_ACCELERATION
#DEFINES+=MBEDTLS_USER_CONFIG_FILE='"mbedtls_user_config.h"'

CY_IGNORE=
CY_IGNORE+= $(SEARCH_aws-iot-device-sdk-embedded-C)/libraries/
standard/coreHTTP
CY_IGNORE+= libs/aws-iot-device-sdk-embedded-C/libraries/standard/
coreHTTP

# CY8CPROTO-062-4343W board shares the same GPIO for the user button
(USER BTN1)
# and the CYW4343W host wake up pin. Since this example uses the GPIO
```

```
for
# interfacing with the user button, the SDIO interrupt to wake up the
host is
# disabled by setting CY_WIFI_HOST_WAKE_SW_FORCE to '0'.
DEFINES+=CY_WIFI_HOST_WAKE_SW_FORCE=0

# Select softfp or hardfp floating point. Default is softfp.
VFP_SELECT=

# Additional / custom C compiler flags.
#
# NOTE: Includes and defines should use the INCLUDES and DEFINES
variable
# above.
CFLAGS=-Os

# Additional / custom C++ compiler flags.
#
# NOTE: Includes and defines should use the INCLUDES and DEFINES
variable
# above.
CXXFLAGS=-Os

# Additional / custom assembler flags.
#
# NOTE: Includes and defines should use the INCLUDES and DEFINES
variable
# above.
ASFLAGS=

# Additional / custom linker flags.
LDFLAGS=

# Additional / custom libraries to link in to the application.
LDLIBS=

# Path to the linker script to use (if empty, use the default linker
script).
LINKER_SCRIPT=

# Custom pre-build commands to run.
PREBUILD=

# Custom post-build commands to run.
POSTBUILD=

# To change the default policy
CY_SECURE_POLICY_NAME=policy_single_CM0_CM4_smif_swap

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

```
##########
# Paths
######################################################################
##########

# Relative path to the project directory (default is the Makefile's
directory).
#
# This controls where automatic source code discovery looks for code.
CY_APP_PATH=

# Relative path to the shared repo location.
#
# All .mtb files have the format, <URI>#<COMMIT>#<LOCATION>. If the
<LOCATION> field
# begins with $$ASSET_REPO$$, then the repo is deposited in the path
specified by
# the CY_GETLIBS_SHARED_PATH variable. The default location is one
directory level
# above the current app directory.
# This is used with CY_GETLIBS_SHARED_NAME variable, which specifies
the directory name.
CY_GETLIBS_SHARED_PATH=../

# Directory name of the shared repo location.
#
CY_GETLIBS_SHARED_NAME=mtb_shared

# Absolute path to the compiler's "bin" directory.
#
# The default depends on the selected TOOLCHAIN (GCC_ARM uses the
ModusToolbox
# software provided compiler by default).
CY_COMPILER_PATH=


# Locate ModusToolbox helper tools folders in default installation
# locations for Windows, Linux, and macOS.
CY_WIN_HOME=$(subst \,/,$(USERPROFILE))
CY_TOOLS_PATHS ?= $(wildcard \
    $(CY_WIN_HOME)/ModusToolbox/tools_* \
    $(HOME)/ModusToolbox/tools_* \
    /Applications/ModusToolbox/tools_*)

# If you install ModusToolbox software in a custom location, add the
path to its
# "tools_X.Y" folder (where X and Y are the version number of the
tools
# folder). Make sure you use forward slashes.
CY_TOOLS_PATHS+=
# Default to the newest installed tools folder, or the users override
(if it's
# found).
CY_TOOLS_DIR=$(lastword $(sort $(wildcard $(CY_TOOLS_PATHS))))

ifeq ($(CY_TOOLS_DIR),)
$(error Unable to find any of the available CY_TOOLS_PATHS -- $
(CY_TOOLS_PATHS). On Windows, use forward slashes.)
endif

$(info Tools Directory: $(CY_TOOLS_DIR))

include $(CY_TOOLS_DIR)/make/start.mk
```