

WrpBase

IOTPRJ – 2020/03/21

History

Revision	Author	Date	Comments
1.0	nguyenhtm	2019/10/06	Initial version
1.1	nguyenhtm	2020/03/21	Update setup guidelines

Overview

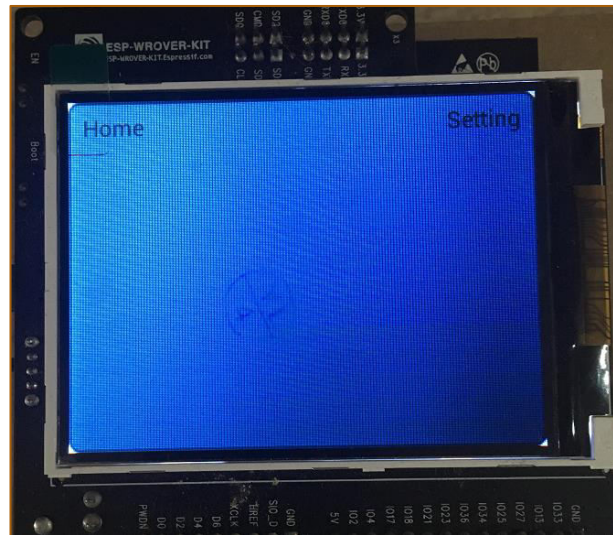
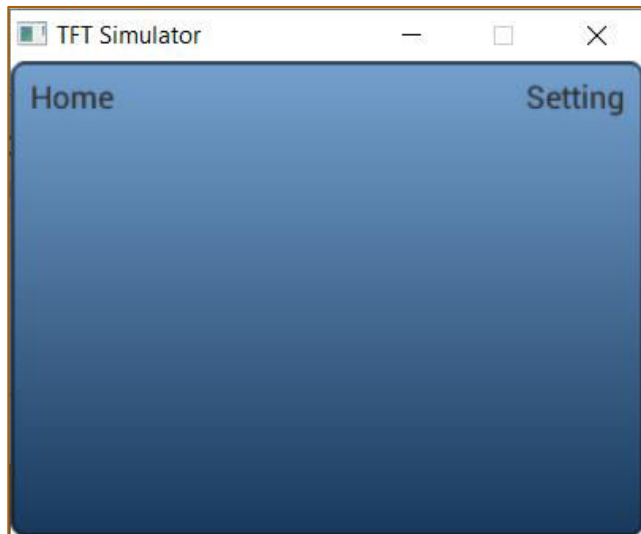
Provide a framework for developers to easily create GUI applications in IOT area with ESP32 chip



Sample App

The Sample App can run with ESP-WROVER-KIT V3 and on Windows platform using SDL Simulator

Video: <https://www.youtube.com/watch?v=e4u56qCMuuE>



SampleApp	wrphmi	wrpmidw
wrpgui	wrpsys	wrpdrv
lvgl	websocket	esp-idf
ILI9341	ESP32	Simulator

Setup

D:\iotprj\wrpbase

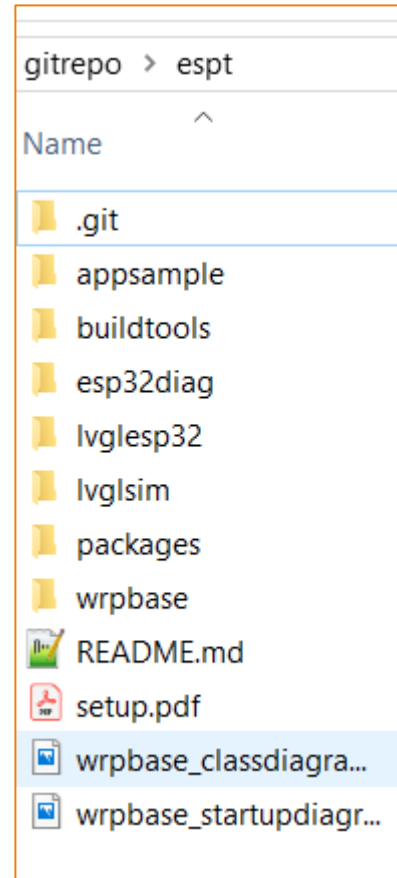
- D:\iotprj: git clone --recursive <https://github.com/nguyenhtm/espt>
- Folder Structures: D:\iotprj\appsample, D:\iotprj\lvglesp32, D:\iotprj\lvgl sim, D:\iotprj\wrpbase

D:\iotprj\buildtools

- D:\iotprj\buildtools\mingw-w64: download mingw-w64-install.exe at <http://mingw-w64.org/doku.php/download>
- D:\iotprj\buildtools\msys32: download esp32_win32_msys2_environment_and_toolchain-20181001.zip at <https://docs.espressif.com/projects/esp-idf/en/stable/get-started/windows-setup.html>
- D:\iotprj\buildtools\eclipse: download eclipse-cpp-2019-03-R-win32-x86_64.zip at <https://www.eclipse.org/downloads/packages/file/55067>

D:\iotprj\packages






- D:\iotprj\packages\esp-idf: git clone --recursive <https://github.com/espressif/esp-idf.git>
- D:\iotprj\packages\lvgl: git clone --recursive <https://github.com/littlevgl/lvgl.git>
- D:\iotprj\packages\pc_simulator_sdl_eclipse: git clone --recursive https://github.com/littlevgl/pc_simulator_sdl_eclipse.git
- D:\iotprj\packages\esp32_ili9341: git clone --recursive https://github.com/littlevgl/esp32_ili9341.git
- D:\iotprj\packages\SDL2-2.0.5: download the SDL2-devel-2.0.5-mingw.tar.gz file at <https://www.libsdl.org/release>
- D:\iotprj\packages\mongoose: git clone <https://github.com/cesanta/mongoose>
- Create D:\iotprj\packages\mongoose_lib and copy 2 D:\iotprj\packages\mongoose\mongoose.c and mongoose.h files to it



BuildTools

Mingw64

- GCC for Windows 32/64 bit which is used to compile source code for simulation
- Download mingw-w64-install.exe at <http://mingw-w64.org/doku.php/download> and install in buildtools folder

	Fedora 19	4.8.1/?
	Rolling	macOS macOS
	Rolling	Windows 7.2.0/5.0.3
	Rolling	Windows 8.2.0/trunk
	12.04 Precise Pangolin	4.6.3/2.0.1
	14.04 Trusty Tahr	4.8.2/3.1.0
	14.10 Utopic Unicorn	4.9.1/3.1.0
	15.04 Vivid Vervet	4.9.2/3.2.0
	15.10 Wily Werewolf	4.9.2/4.0.2

mingw-w64

GCC for Windows 64 & 32 bits

Mingw-builds

Installation: [Sourceforge](#)

[\[Back to top | Sitemap \]](#)

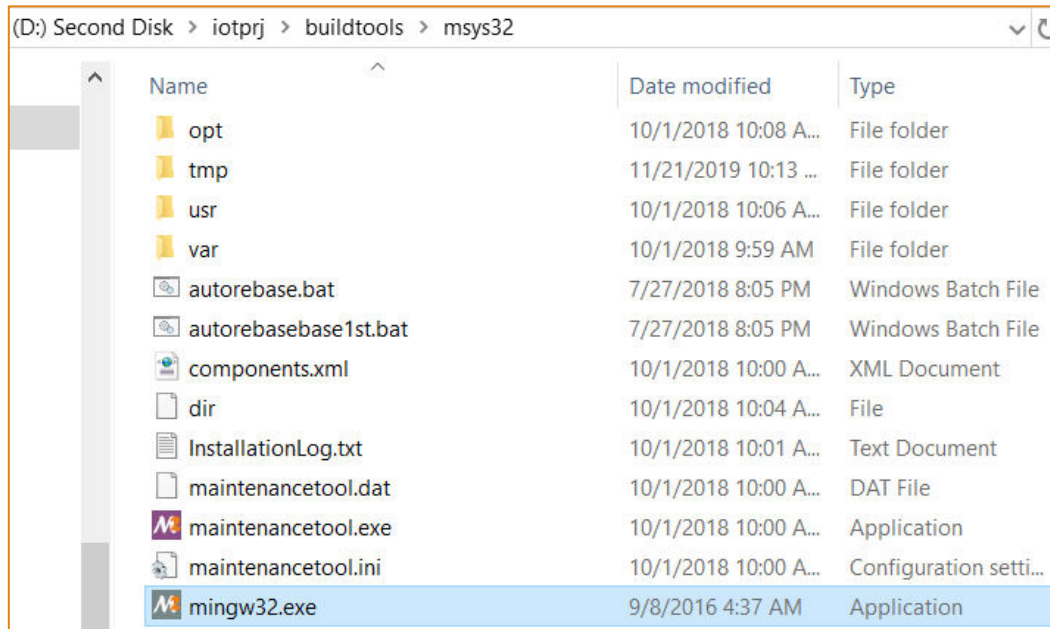
(D:) Second Disk > iotprj > buildtools > mingw-w64 > i686-8.1.0-posix-dwarf-rt_v6-rev0				
	Name	Date modified	Type	Size
	mingw32	10/11/2019 1:32 PM	File folder	
	mingw-w64.bat	9/21/2019 4:09 PM	Windows Batch File	
	mingw-w64	12/28/2015 12:30 ...	Internet Shortcut	
	uninstall.exe	9/21/2019 4:08 PM	Application	
	uninstall.ini	9/21/2019 4:09 PM	Configuration setti...	

GCC install path: D:\iotprj\buildtools\mingw-w64\i686-8.1.0-posix-dwarf-rt_v6-rev0\mingw32\bin

BuildTools 2

Msys32 with ESP32

- GCC for ESP32 which is used to compile source code for PICO target
- Download esp32_win32_msys2_environment_and_toolchain-20181001.zip at <https://docs.espressif.com/projects/esp-idf/en/stable/get-started/windows-setup.html> and extract at buildtools folder



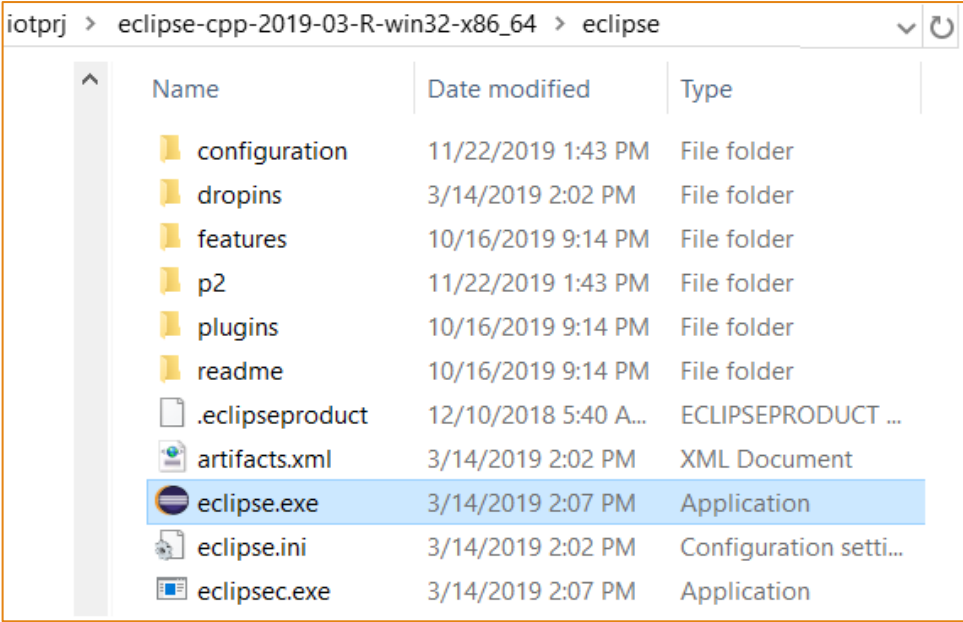
GCC install path:

D:\iotprj\buildtools\msys32\opt\xtensa-esp32-elf\bin

BuildTools 3

Eclipse

- Editor can be editor and configured to use GCC to build the sample app on PICO target and Windows
- Download eclipse-cpp-2019-03-R-win32-x86_64.zip at <https://www.eclipse.org/downloads/packages/file/55067> and extract at buildtools folder



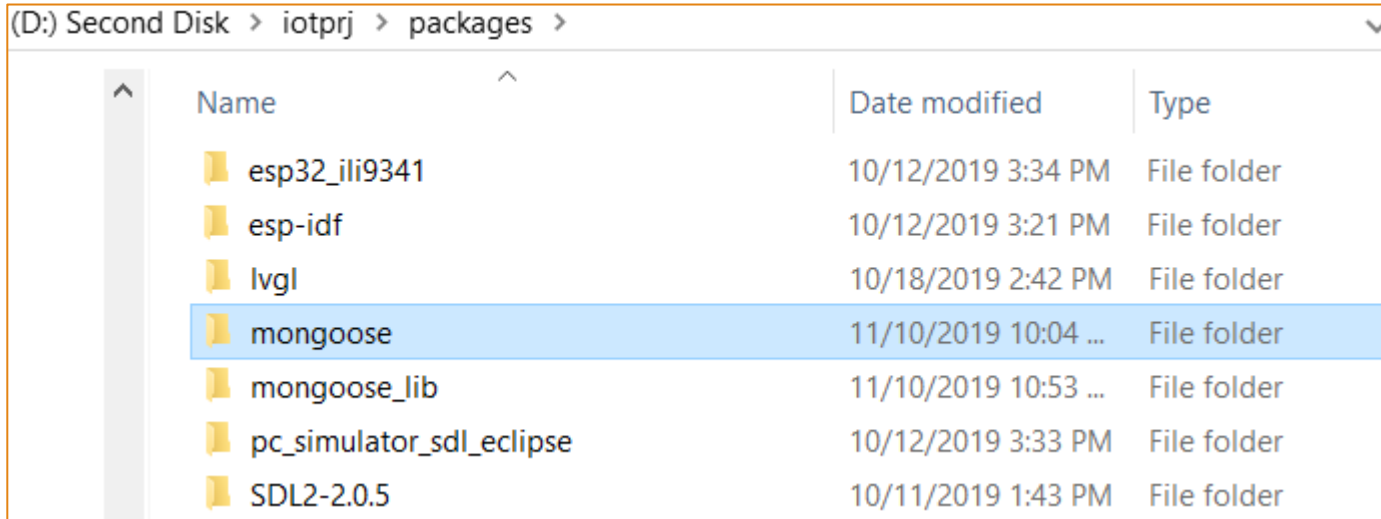
Name	Date modified	Type
configuration	11/22/2019 1:43 PM	File folder
dropins	3/14/2019 2:02 PM	File folder
features	10/16/2019 9:14 PM	File folder
p2	11/22/2019 1:43 PM	File folder
plugins	10/16/2019 9:14 PM	File folder
readme	10/16/2019 9:14 PM	File folder
.eclipseproduct	12/10/2018 5:40 A...	ECLIPSEPRODUCT ...
artifacts.xml	3/14/2019 2:02 PM	XML Document
eclipse.exe	3/14/2019 2:07 PM	Application
eclipse.ini	3/14/2019 2:02 PM	Configuration setti...
eclipsesec.exe	3/14/2019 2:07 PM	Application

Eclipse install path:

D:\iotprj\buildtools\eclipse-cpp-2019-03-R-win32-x86_64\eclipse

Packages

- The folder used to store open sources like lvgl, mongoose or esp-idf framework. Refer to Setup slide to download
- In case of mongoose, after downloading, create **mongoose_lib** folder and copy **mongoose.c** and **mongoose.h** files to it



(D:) Second Disk > iotprj > packages

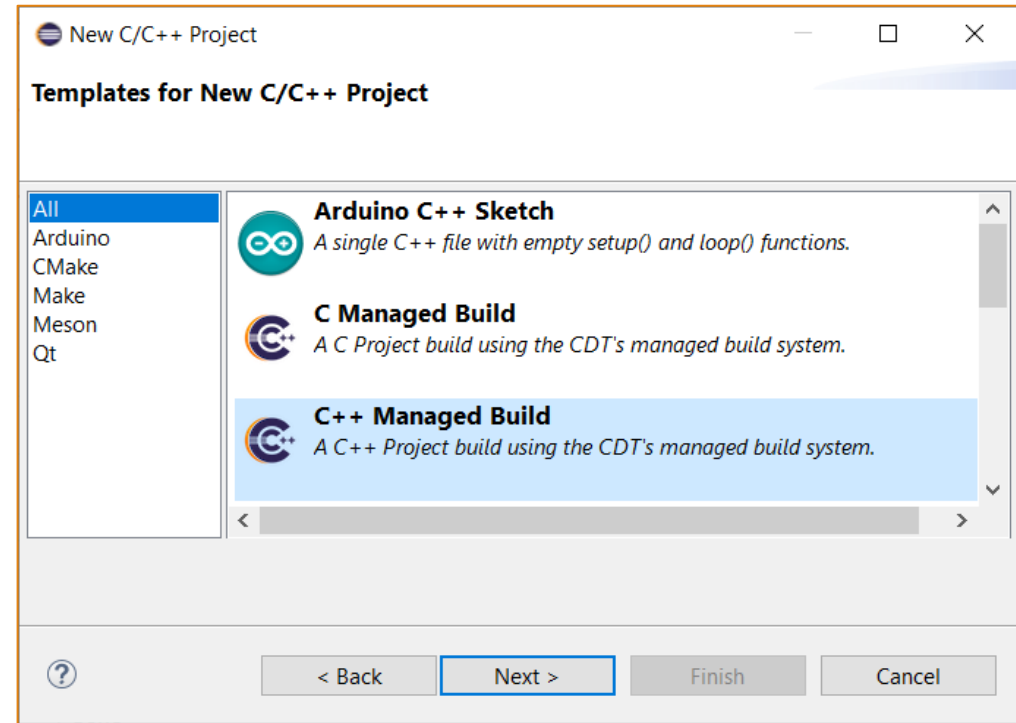
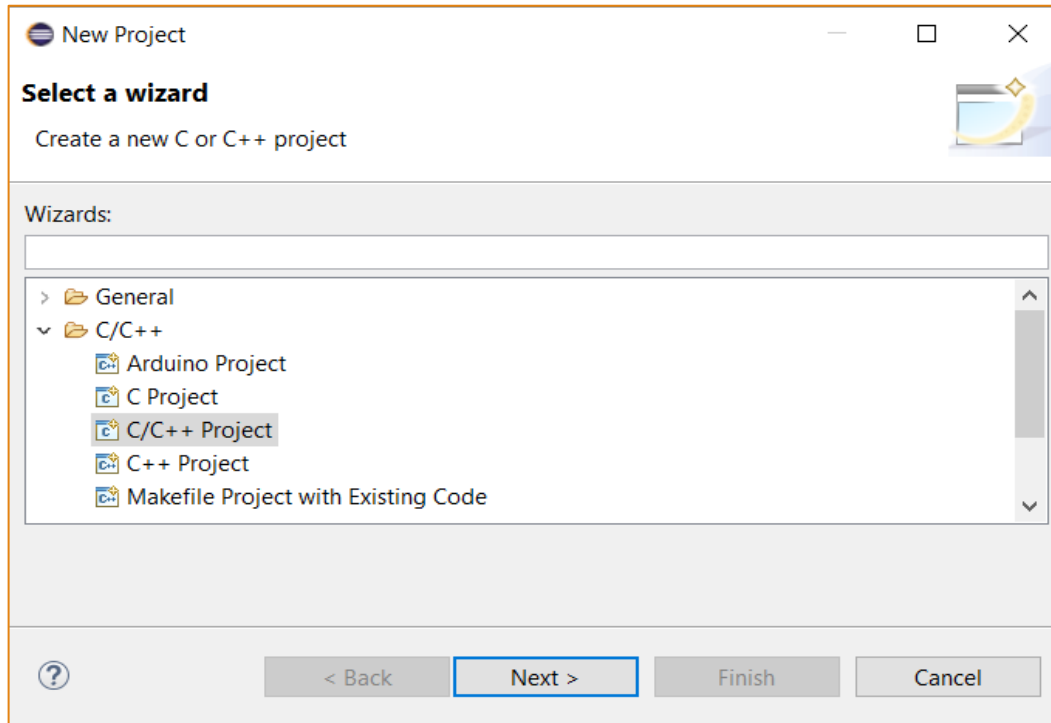
Name	Date modified	Type
esp32_ili9341	10/12/2019 3:34 PM	File folder
esp-idf	10/12/2019 3:21 PM	File folder
lvgl	10/18/2019 2:42 PM	File folder
mongoose	11/10/2019 10:04 ...	File folder
mongoose_lib	11/10/2019 10:53 ...	File folder
pc_simulator_sdl_eclipse	10/12/2019 3:33 PM	File folder
SDL2-2.0.5	10/11/2019 1:43 PM	File folder

Notes

- Set IDF_PATH: `echo 'export IDF_PATH="D:/iotprj/packages/esp-idf"' >> $HOME/.bash_profile`
 - When open MSYS32 execute file, it will use this path as default.
- Generate partition table:
 - Partition table should not same name as project name otherwise same .bin is generated
 - Generate: `python $IDF_PATH/components/partition_table/gen_esp32part.py projectname_partition_table.bin`
 - Flash: `python $IDF_PATH/tools/idf.py partition_table-flash => enter to lvgl esp32\build\partition_table`
- Execute the command to setup required software:
 - `D:/iotprj/buildtools/msys32/mingw32/bin/python.exe -m pip install --user -r D:/iotprj/packages/esp-idf/requirements.txt`

Eclipse C++ Project 1

- Create 2 Eclipse C++ projects: one for simulation and one for ESP32 with LCD
- Both demo how to setup working environment and how to use open sources like lvgl, mongoose,...



Eclipse C++ Project 2

C++ Project
Create C++ project of selected type

Project name:

☒ Use default location
Location:

Choose file system:

Project type:

- > GNU Autotools
- ▼ Executable
 - Empty Project
 - **Hello World C++ Project**
- > Shared Library
- > Static Library
- > Makefile project

Toolchains:

- Cross GCC
- Cygwin GCC

☒ Show project types and toolchains only if they are supported on the platform

Basic Settings
Basic properties of a project

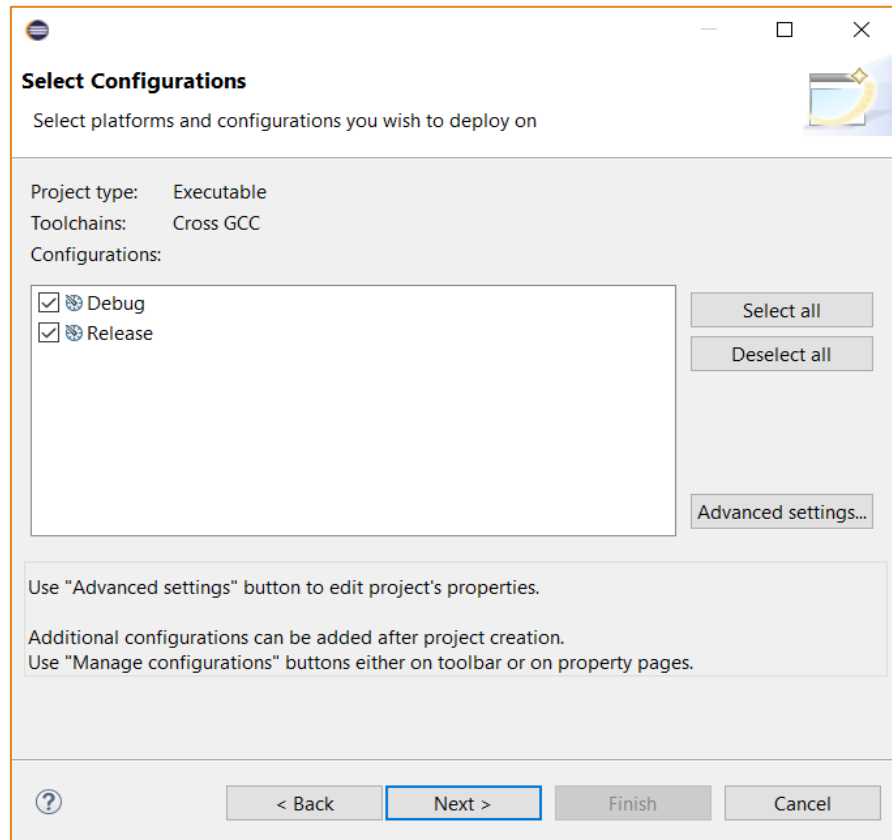
Author:

Copyright notice:

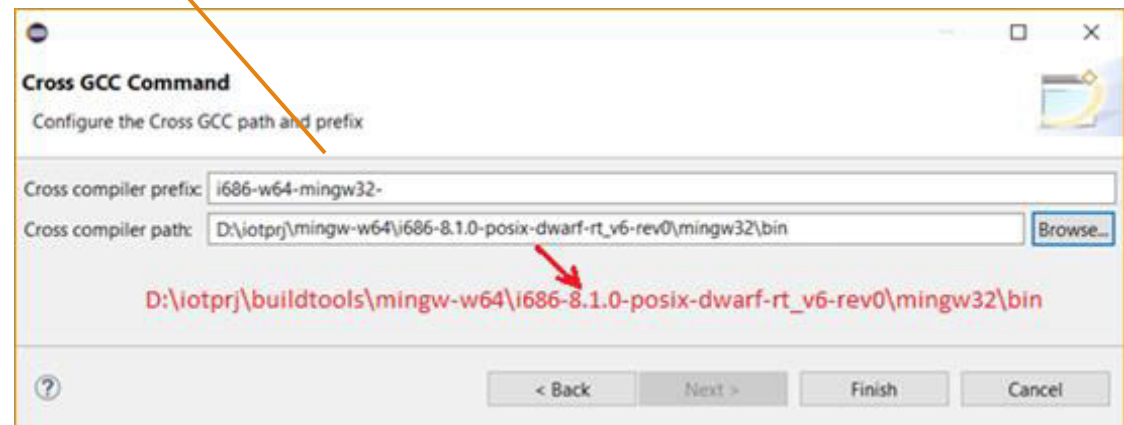
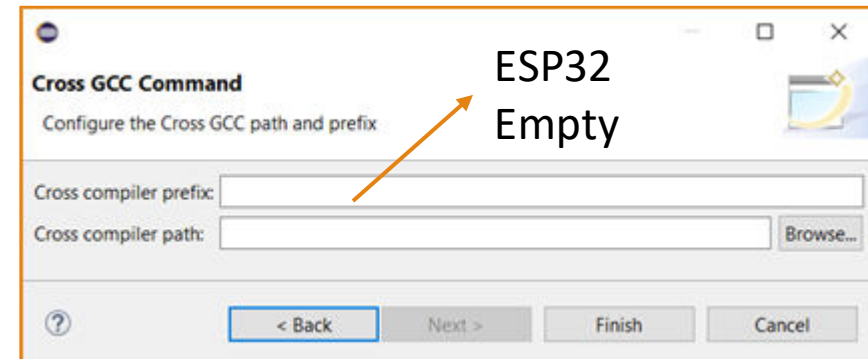
Hello world greeting:

Source:

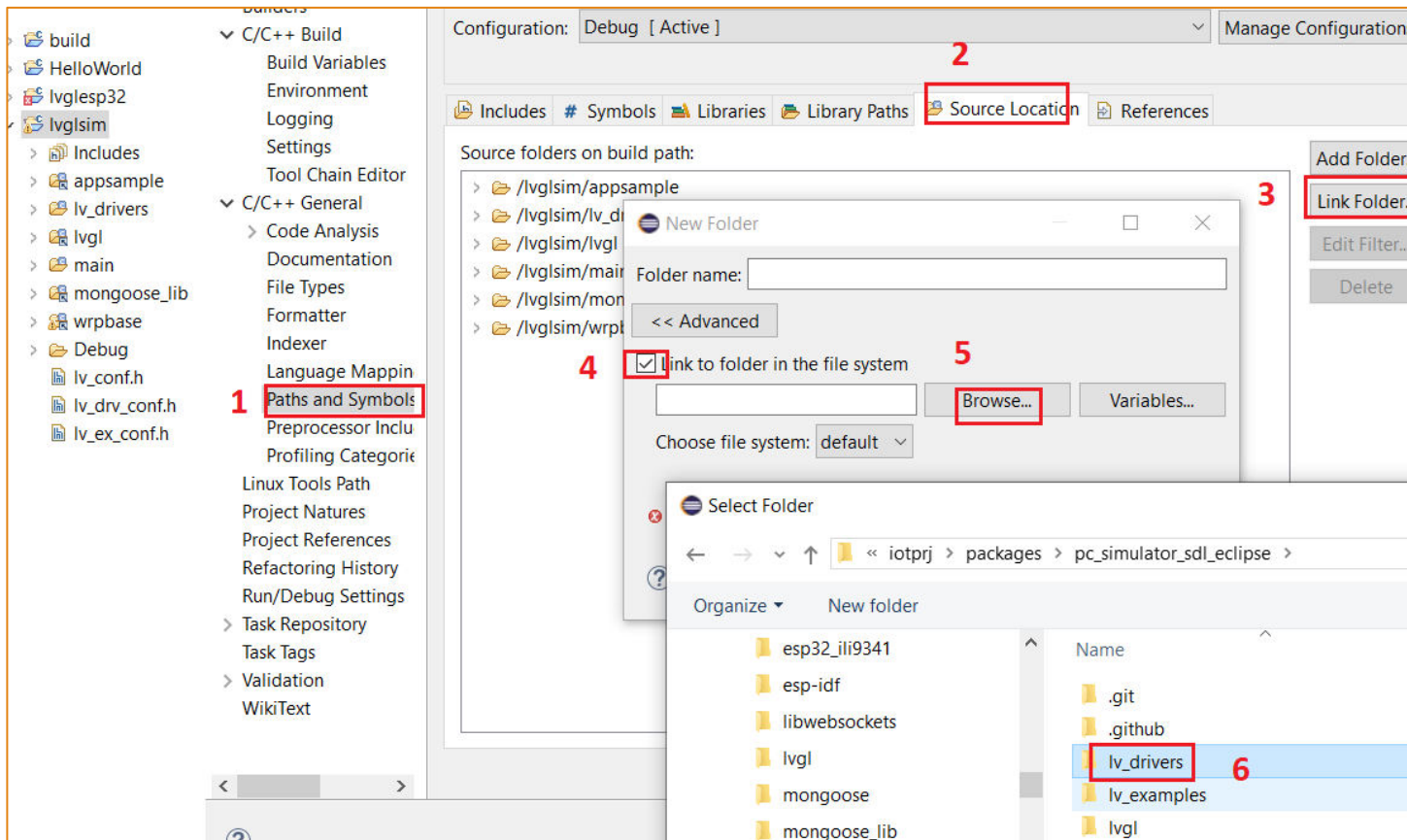
Eclipse C++ Project 3



Simulator

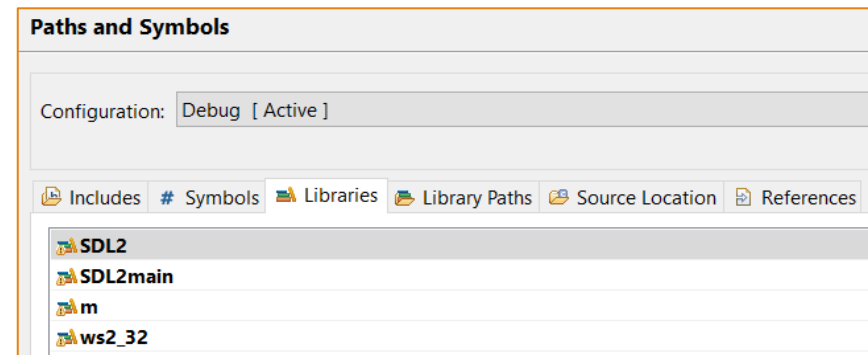
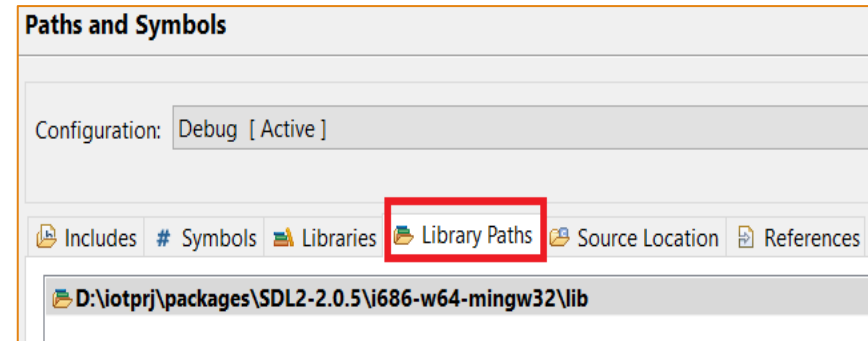
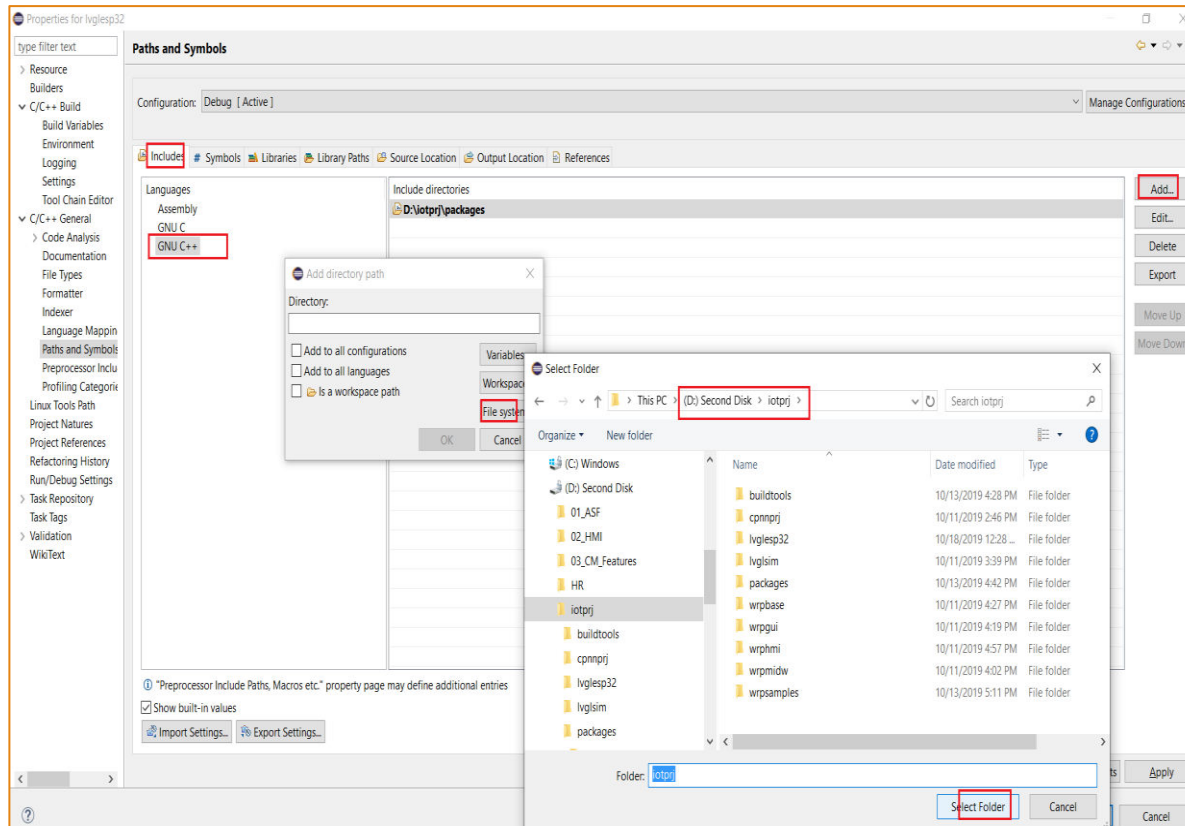


LVGL SIM Eclipse 1



Configuration of Lvgl Simu Eclipse Project:
Link source folders such as appsample,
Lvgl, mongoose_lib, and wrpbases

LVGL SIM Eclipse 2



LVGL SIM Eclipse 3

Paths and Symbols

Configuration: Debug [Active]

Includes # Symbols Libraries Library Paths Source Location

Languages	Symbol	Value
GNU C	# LV_CONF_INCLUDE_SIMPLE	1
GNU C++	# LVGL_PC_SIMU	1

Includes # Symbols Libraries Library Paths Source Location

Source folders on build path:

- > /lvgl_sim/appsample
- > /lvgl_sim/lv_drivers
- > /lvgl_sim/lvgl
- > /lvgl_sim/main
- > /lvgl_sim/mongoose_lib
- > /lvgl_sim/wrpbse

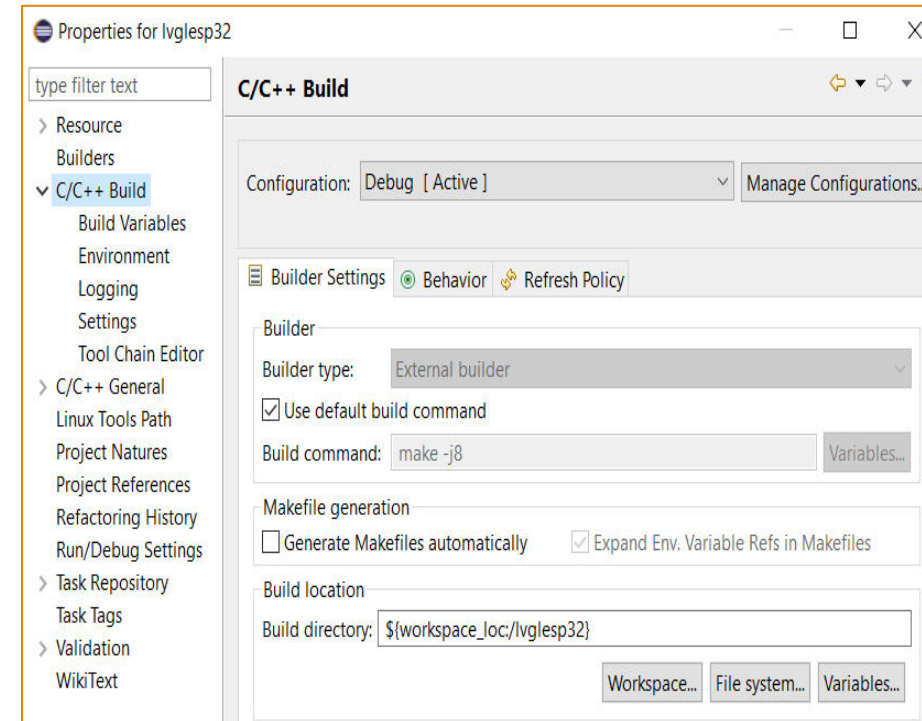
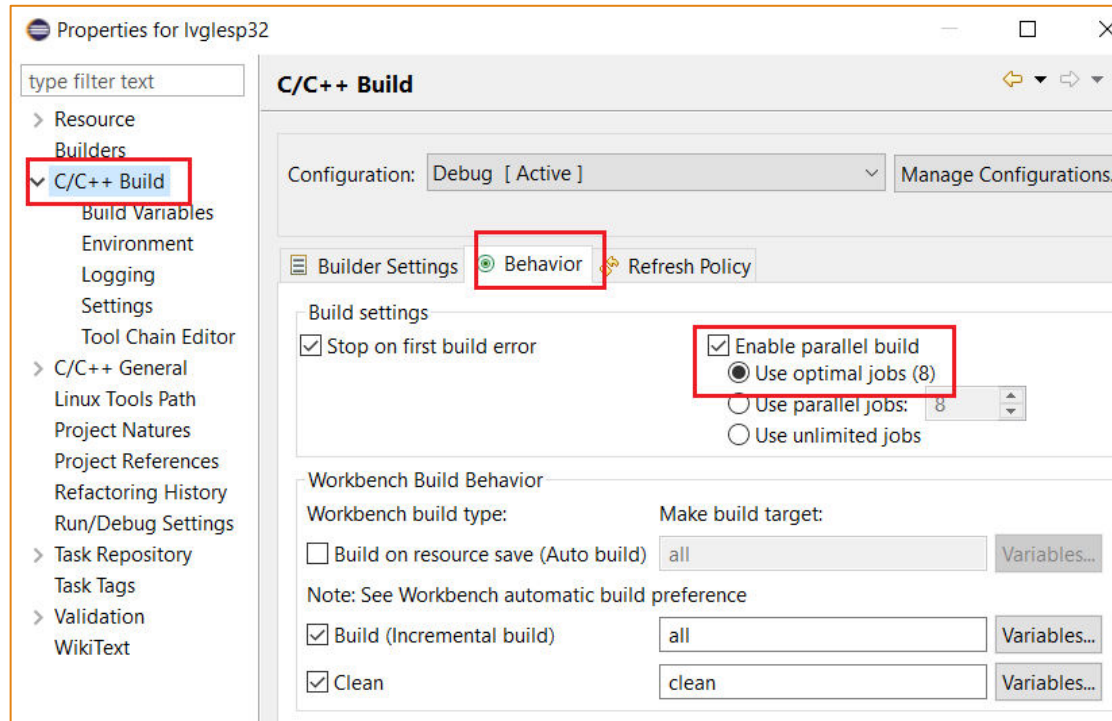
Paths and Symbols

Configuration: Debug [Active]

Includes # Symbols Libraries Library Paths Source Location References

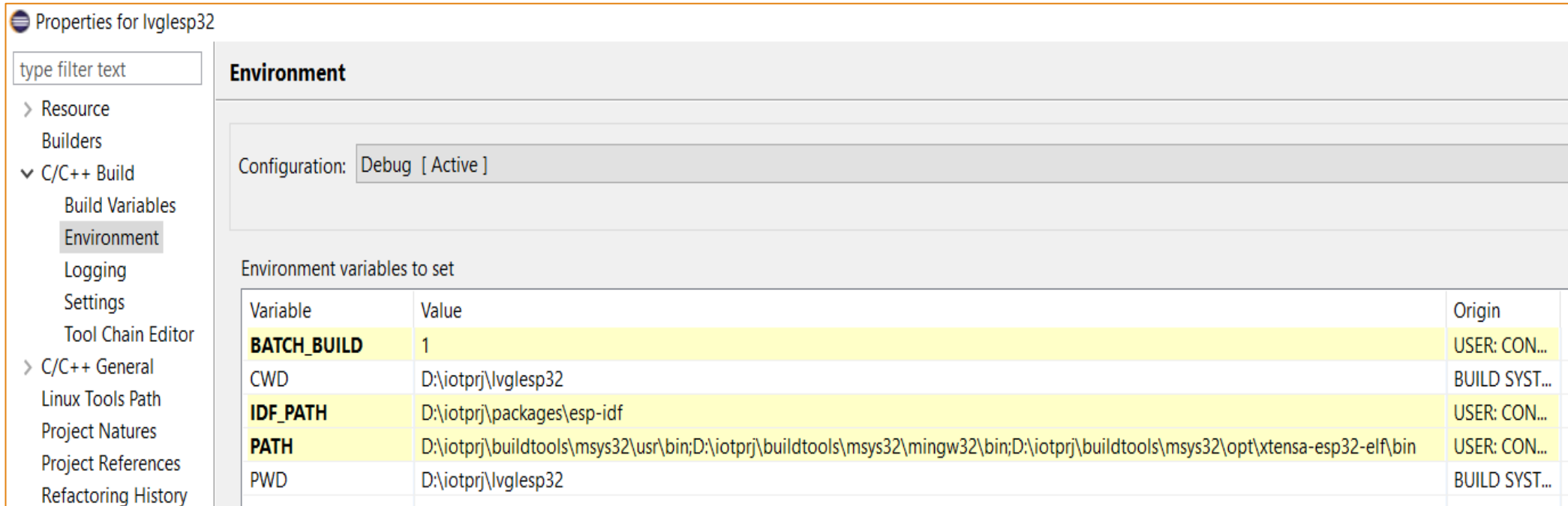
Languages	Include directories
Assembly	/\${ProjName}
GNU C	D:\iotprj
GNU C++	D:\iotprj\packages
	D:\iotprj\packages\mongoose_lib
	D:\iotprj\packages\SDL2-2.0.5\i686-w64-mingw32\include

LVGL ESP32 Eclipse



LVGL ESP32 Eclipse 1

Configure ESP-IDF framework and GCC for ESP32 compiler path as below



Properties for lvglesp32

type filter text

- > Resource
- Builders
- ✓ C/C++ Build
 - Build Variables
 - Environment
 - Logging
 - Settings
 - Tool Chain Editor
- > C/C++ General
- Linux Tools Path
- Project Natures
- Project References
- Refactoring History

Environment

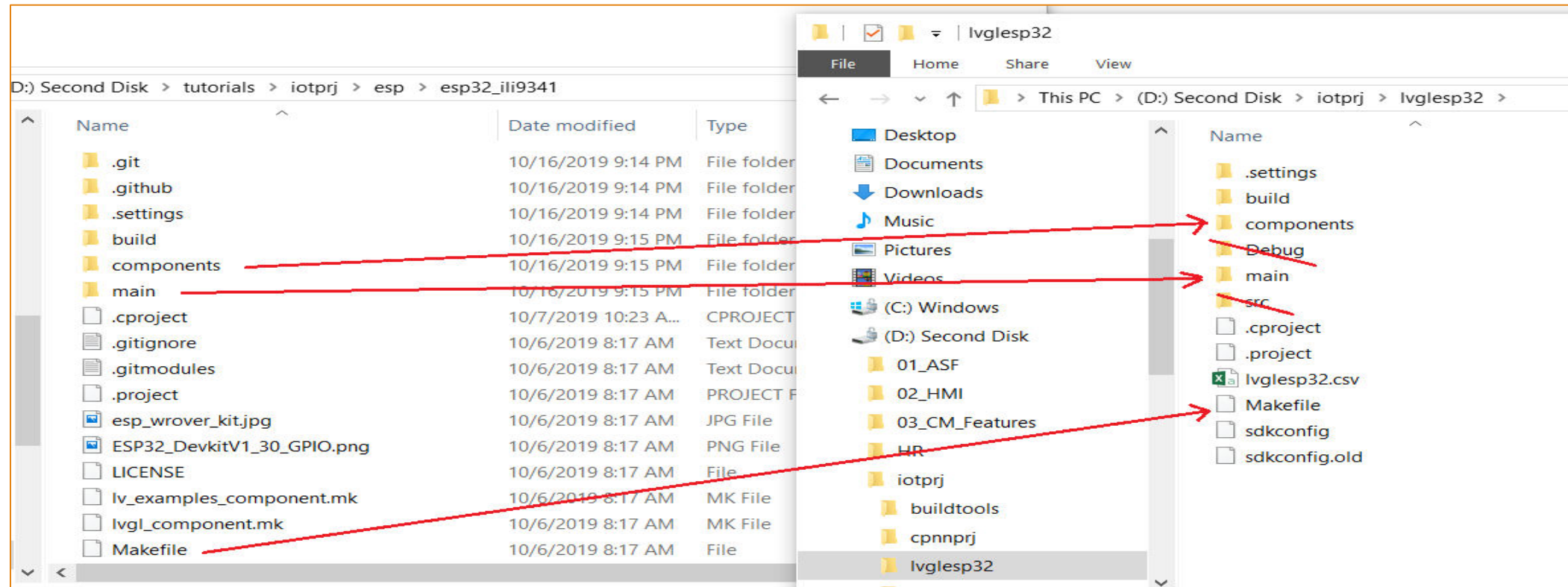
Configuration: Debug [Active]

Environment variables to set

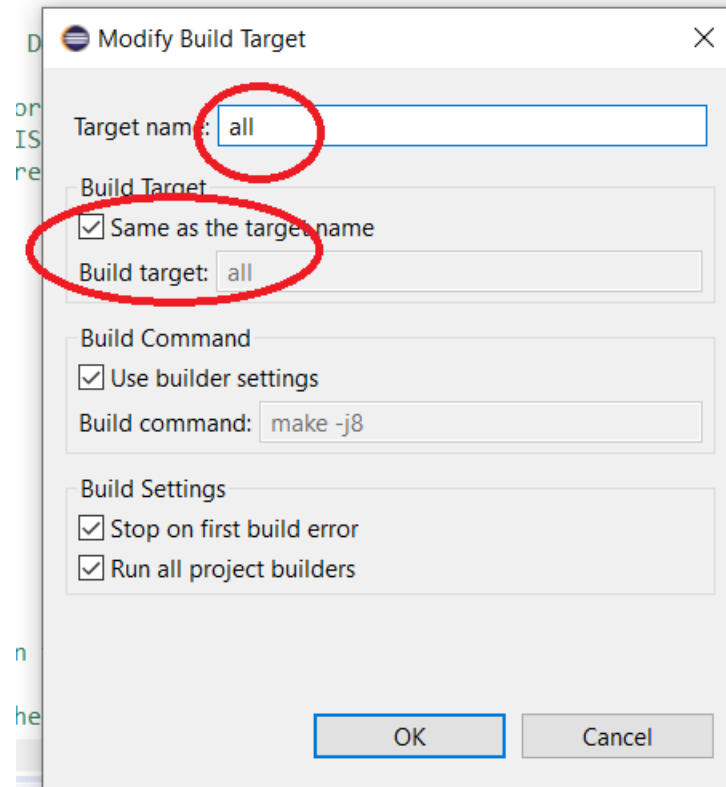
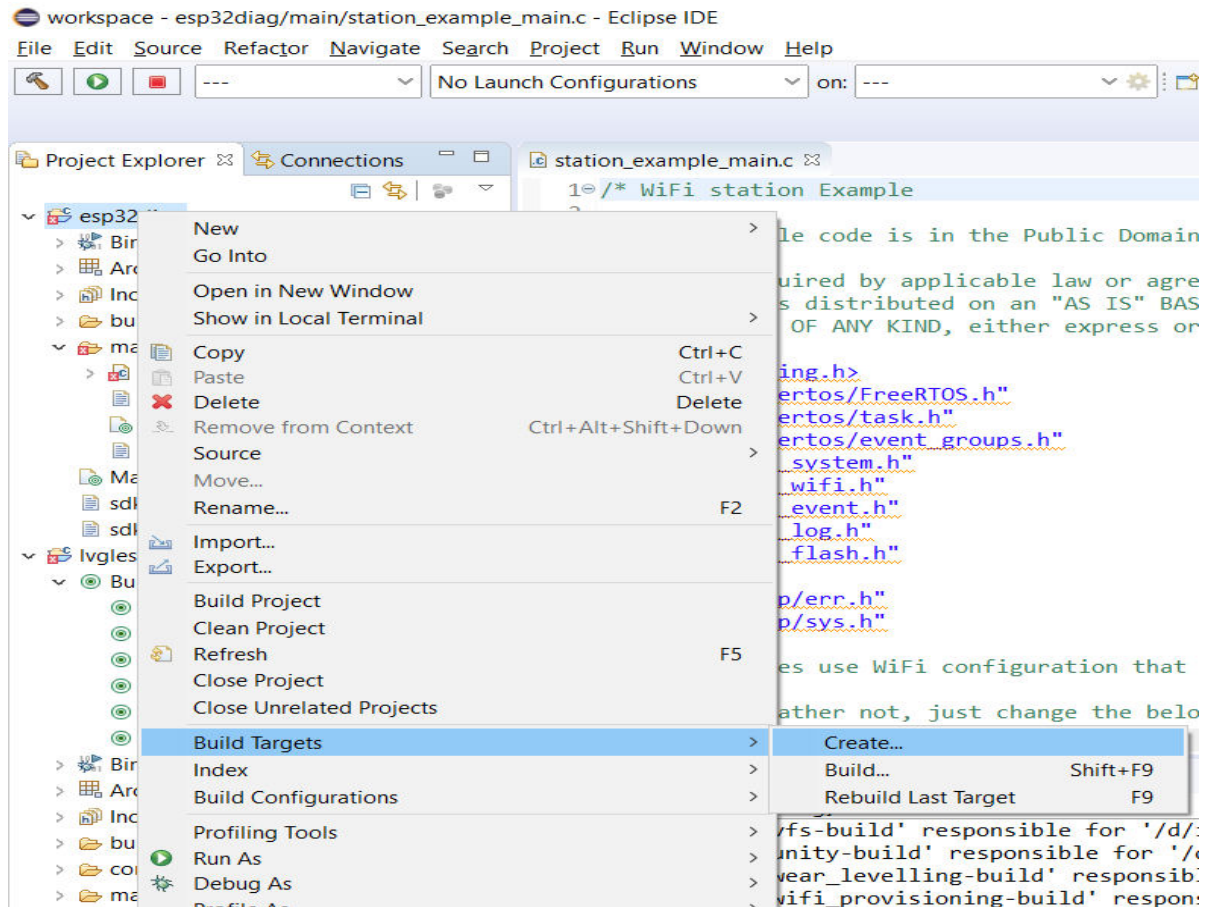
Variable	Value	Origin
BATCH_BUILD	1	USER: CON...
CWD	D:\iotprj\lvglesp32	BUILD SYST...
IDF_PATH	D:\iotprj\packages\esp-idf	USER: CON...
PATH	D:\iotprj\buildtools\msys32\usr\bin;D:\iotprj\buildtools\msys32\mingw32\bin;D:\iotprj\buildtools\msys32\opt\xtensa-esp32-elf\bin	USER: CON...
PWD	D:\iotprj\lvglesp32	BUILD SYST...

LVGL ESP32 Eclipse 2

Copy esp32_ili9341/components, main folder and **Makefile** to lvglesp32



LVGL ESP32 Eclipse 3



LVGL ESP32 Eclipse 4

Then edit Makefile to connect to WrpBase framework as below:

ESP32 Diagnosis Project Name

PROJECT_NAME := esp32diag

#For C++ language flags

EXTRA_CPPFLAGS := -DLV_CONF_INCLUDE_SIMPLE -DILI9341_BCKL_ACTIVE_LVL=0

#Use LVGL library and ILI9341 driver

EXTRA_COMPONENT_DIRS := D:/iotprj/packages/lvgl D:/iotprj/packages/esp32_ili9341/components/drv

#Include headers of lvgl component and headers of ili9341 driver

COMPONENT_EXTRA_INCLUDES := D:/iotprj/packages D:/iotprj/packages/esp32_ili9341/components

#Use include macros for wrpbase

CPPFLAGS += -DLVGL_PC_SIMU=0 -DLVGL_ESP32_ILI9341=1 -DUSE_ESP_IDF=1

#Use wrpbase component

EXTRA_COMPONENT_DIRS += D:/iotprj/wrpbase/wrpdrrv D:/iotprj/wrpbase/wrpsys D:/iotprj/wrpbase/wrpgui

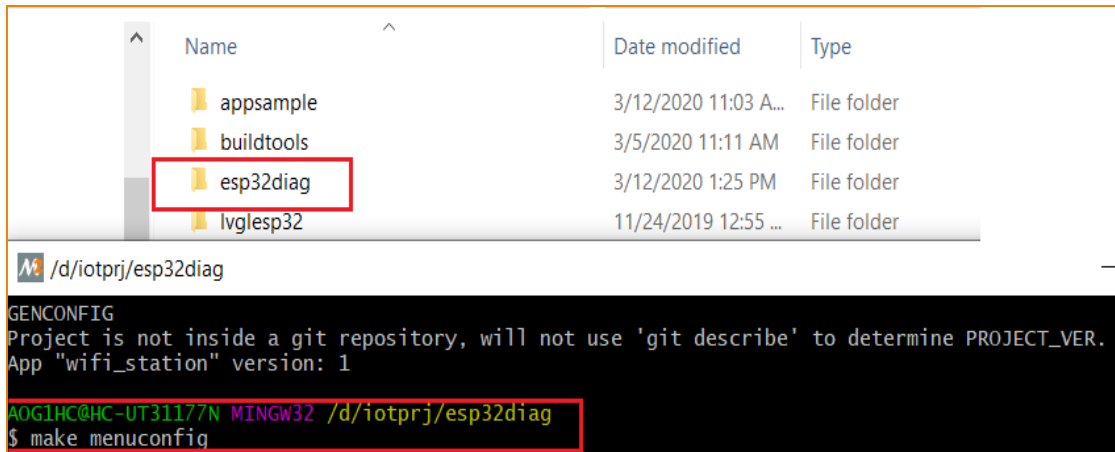
D:/iotprj/wrpbase/wrprmidw D:/iotprj/wrpbase/wrphmi D:/iotprj/wrpbase/wrpres

#esp-idf platform makefile

include \$(IDF_PATH)/make/project.mk

Example of New ESP32 Project 1

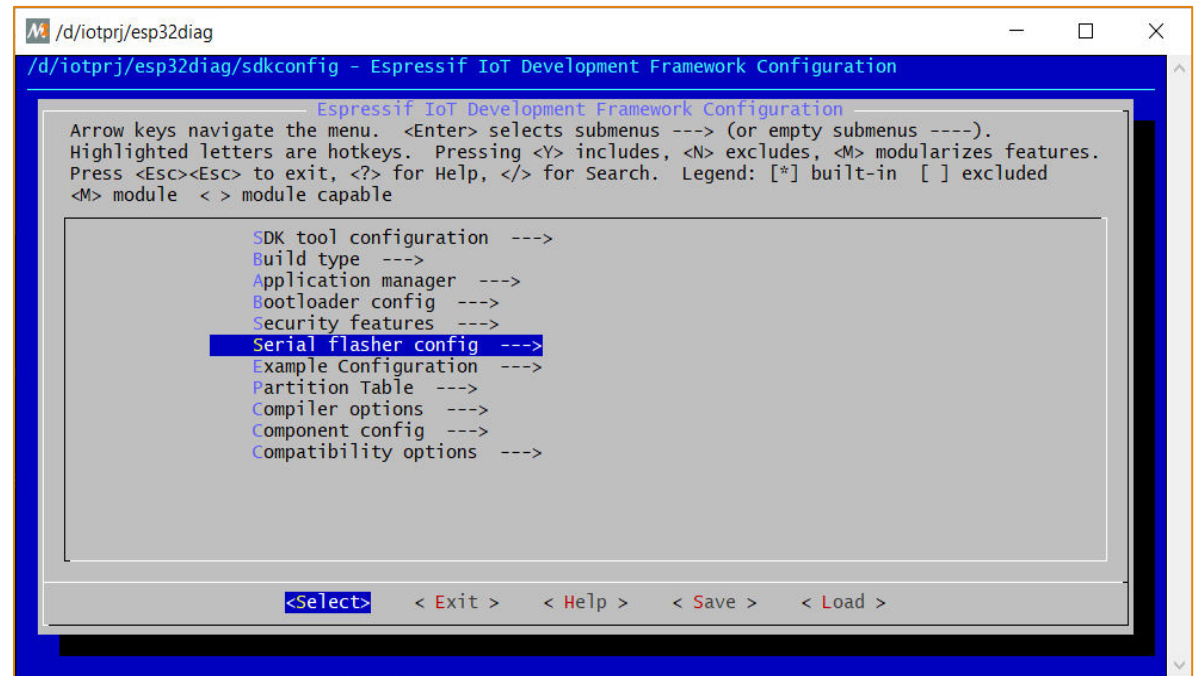
Example of the **esp32diag** project creation and WrpBase usage



Name	Date modified	Type
appsample	3/12/2020 11:03 A...	File folder
buildtools	3/5/2020 11:11 AM	File folder
esp32diag	3/12/2020 1:25 PM	File folder
lvgl esp32	11/24/2019 12:55 ...	File folder

```
/d/iotprj/esp32diag
GENCONFIG
Project is not inside a git repository, will not use 'git describe' to determine PROJECT_VER.
App "wifi_station" version: 1
AOGIHC@HC-UT31177N MINGW32 /d/iotprj/esp32diag
$ make menuconfig
```

- Copy D:\iotprj\packages\esp-idf\examples\wifi\getting_started\station to D:\iotprj folder and rename to esp32diag
- Do “make menuconfig” and then “make”
- Now add it into Eclipse: create a workspace folder eg: D:\iotprj\workspace



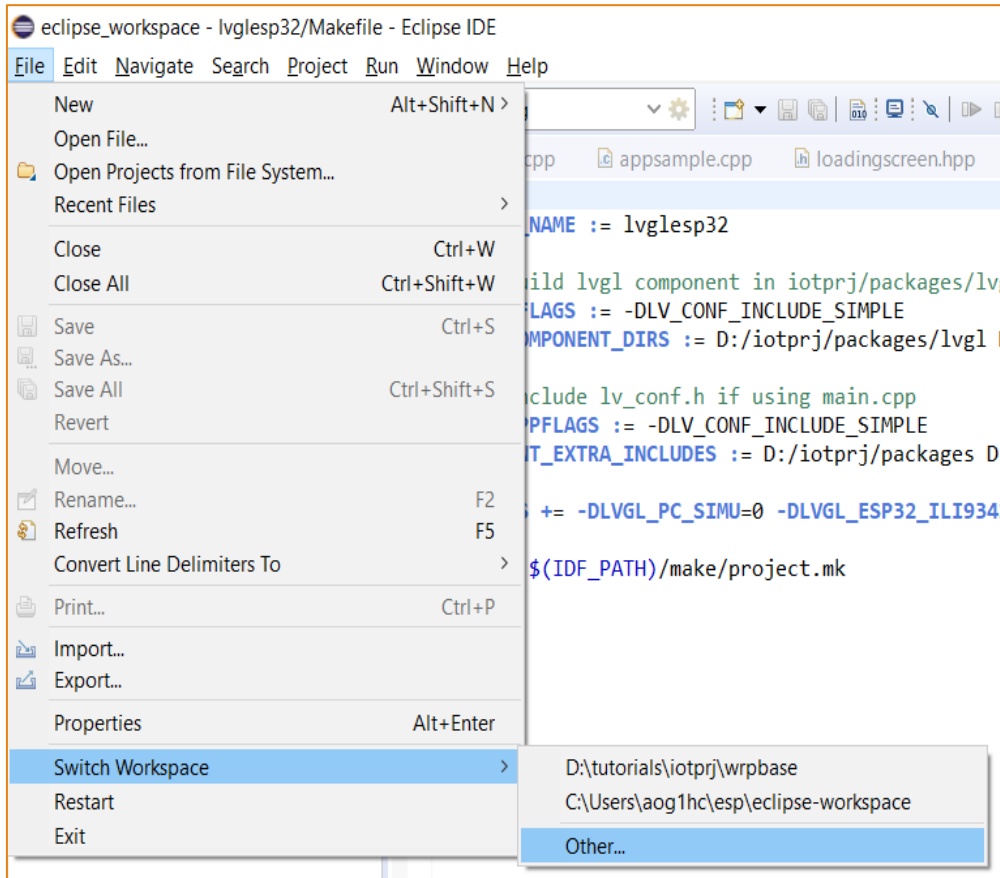
```
/d/iotprj/esp32diag
/d/iotprj/esp32diag/sdkconfig - Espressif IoT Development Framework Configuration

Espressif IoT Development Framework Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded
<M> module <> module capable

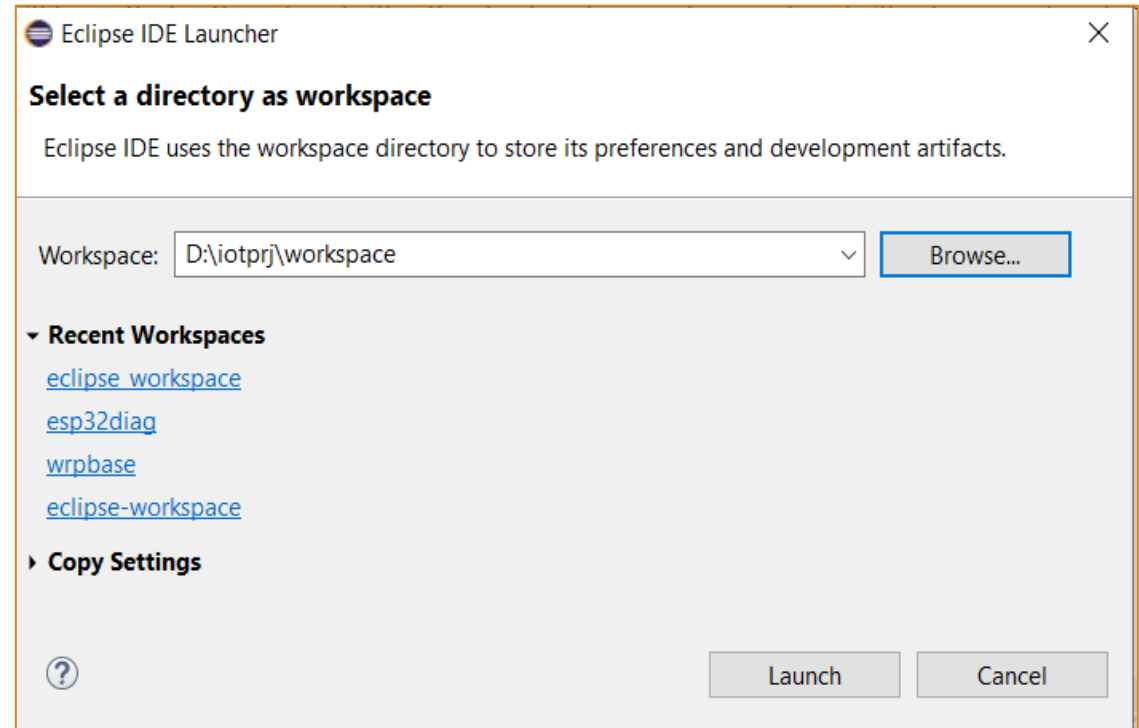
  SDK tool configuration --->
  Build type --->
  Application manager --->
  Bootloader config --->
  Security features --->
  Serial flasher config ---->
  Example Configuration ---->
  Partition Table ---->
  Compiler options --->
  Component config --->
  Compatibility options --->

  <Select>  < Exit >  < Help >  < Save >  < Load >
```

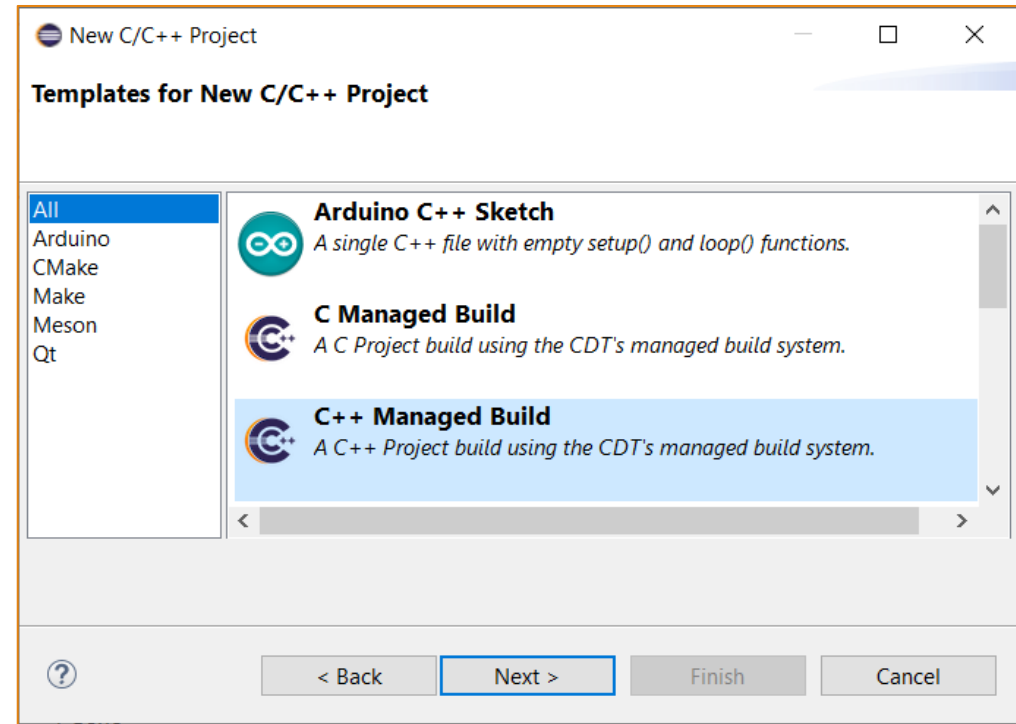
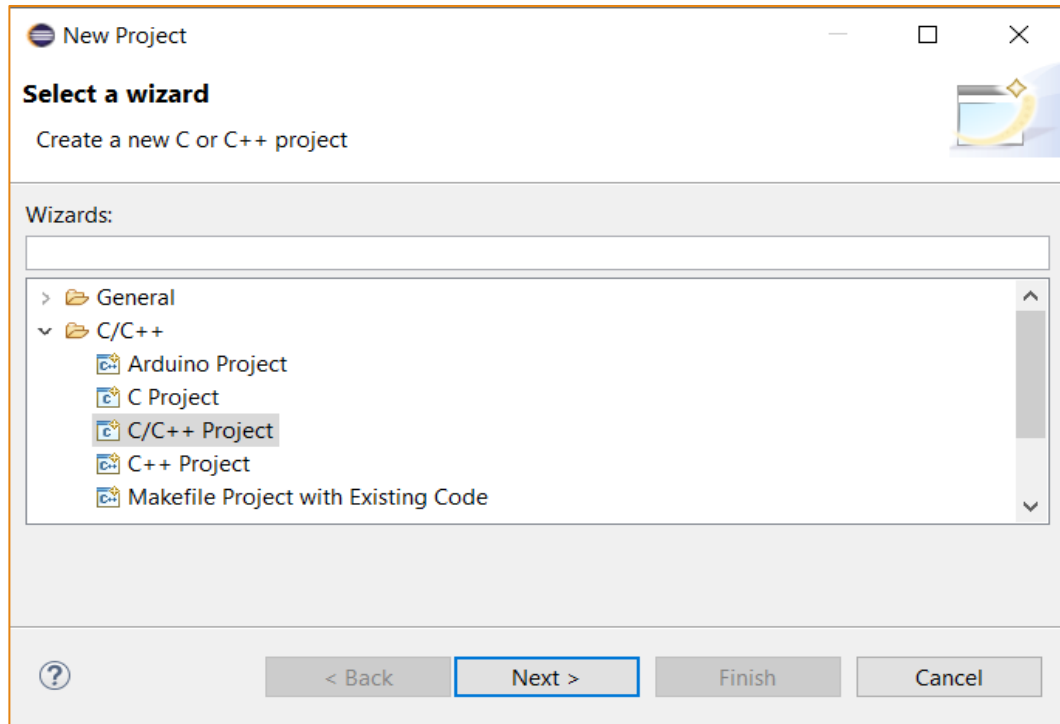

Example of New ESP32 Project 2



Open Eclipse and swith to D:\iotprj\workspace



Example of New ESP32 Project 3



Example of New ESP32 Project 4

C++ Project

⚠ Directory with specified name already exists.

Project name:

☐ Use default location

Location:

Choose file system:

Project type:

- > GNU Autotools
- > Executable
 - Empty Project
 - **Hello World C++ Project**
- > Shared Library
- > Static Library
- > Makefile project

Toolchains:

- Cross GCC

☒ Show project types and toolchains only if they are supported on the platform

Basic Settings

Basic properties of a project

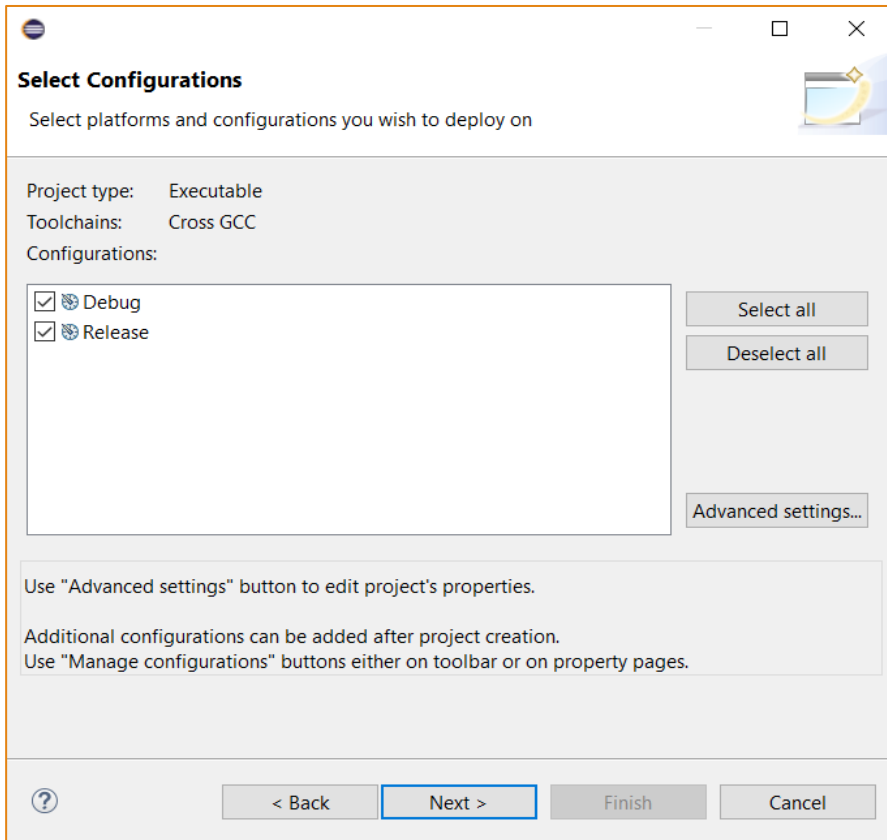
Author:

Copyright notice:

Hello world greeting:

Source:

Example of New ESP32 Project 5



Select Configurations
Select platforms and configurations you wish to deploy on

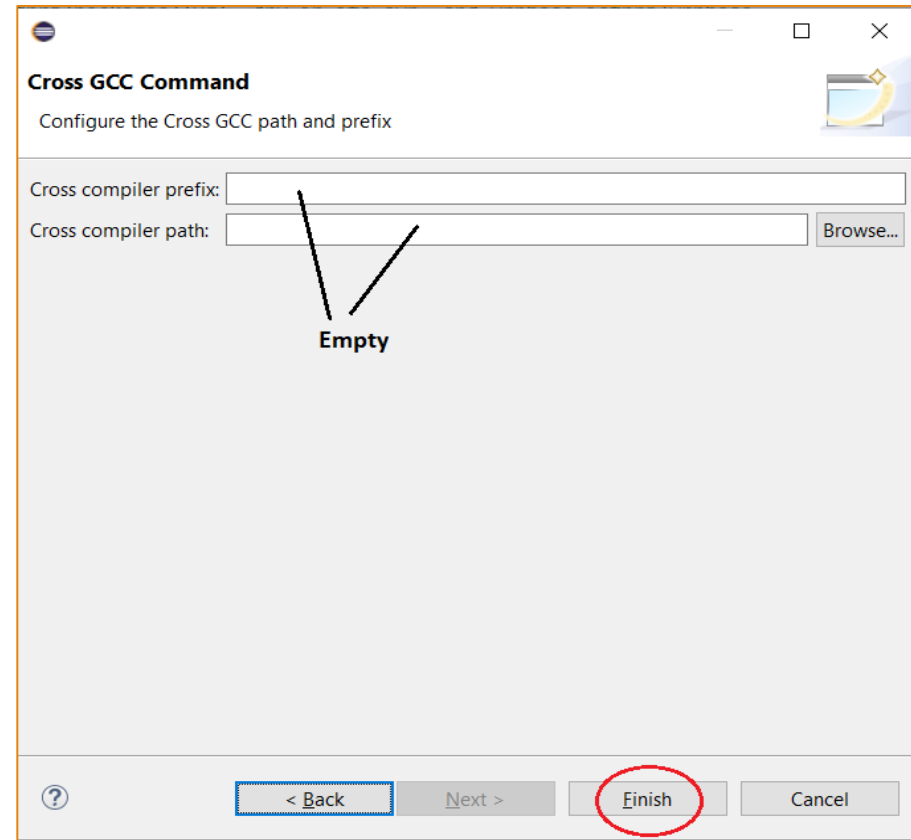
Project type: Executable
Toolchains: Cross GCC
Configurations:

☒ Debug
☒ Release

Select all
Deselect all
Advanced settings...

Use "Advanced settings" button to edit project's properties.
Additional configurations can be added after project creation.
Use "Manage configurations" buttons either on toolbar or on property pages.

? < Back Next > Finish Cancel



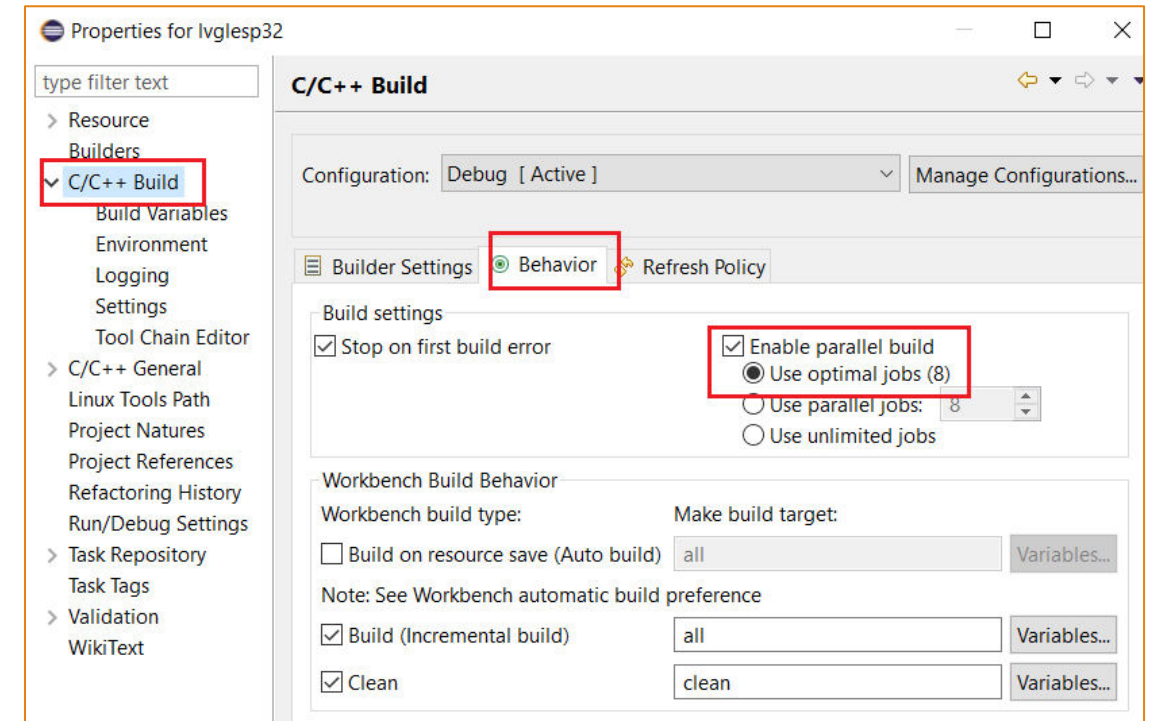
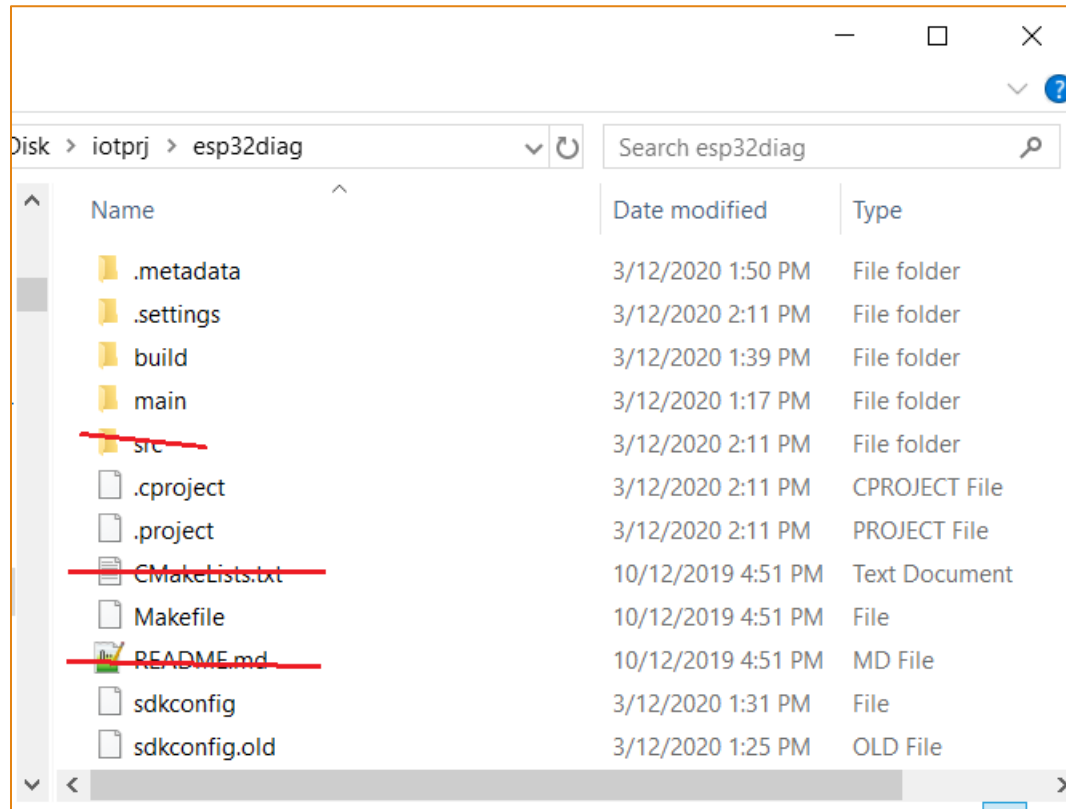
Cross GCC Command
Configure the Cross GCC path and prefix

Cross compiler prefix:
Cross compiler path: Browse...

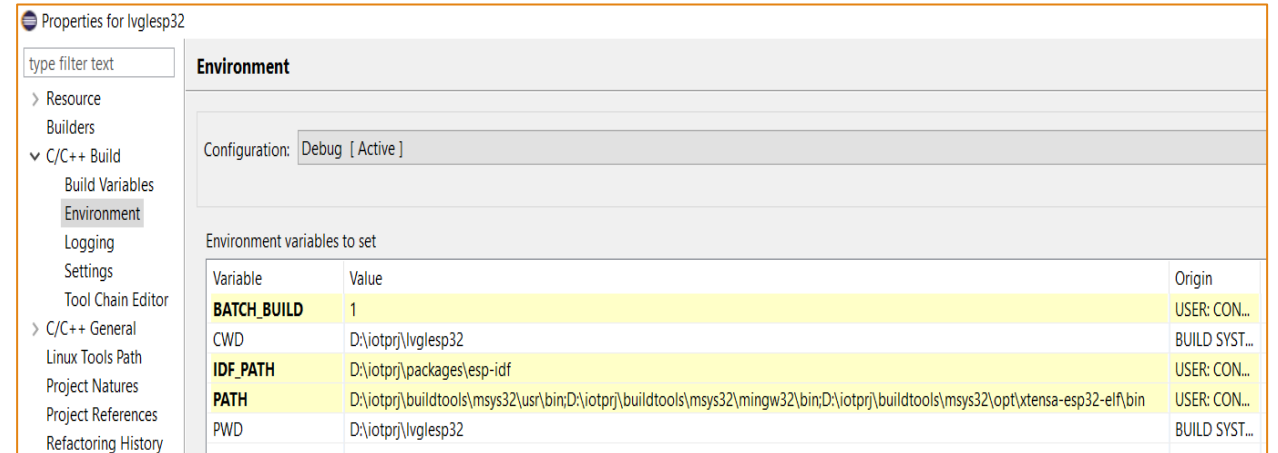
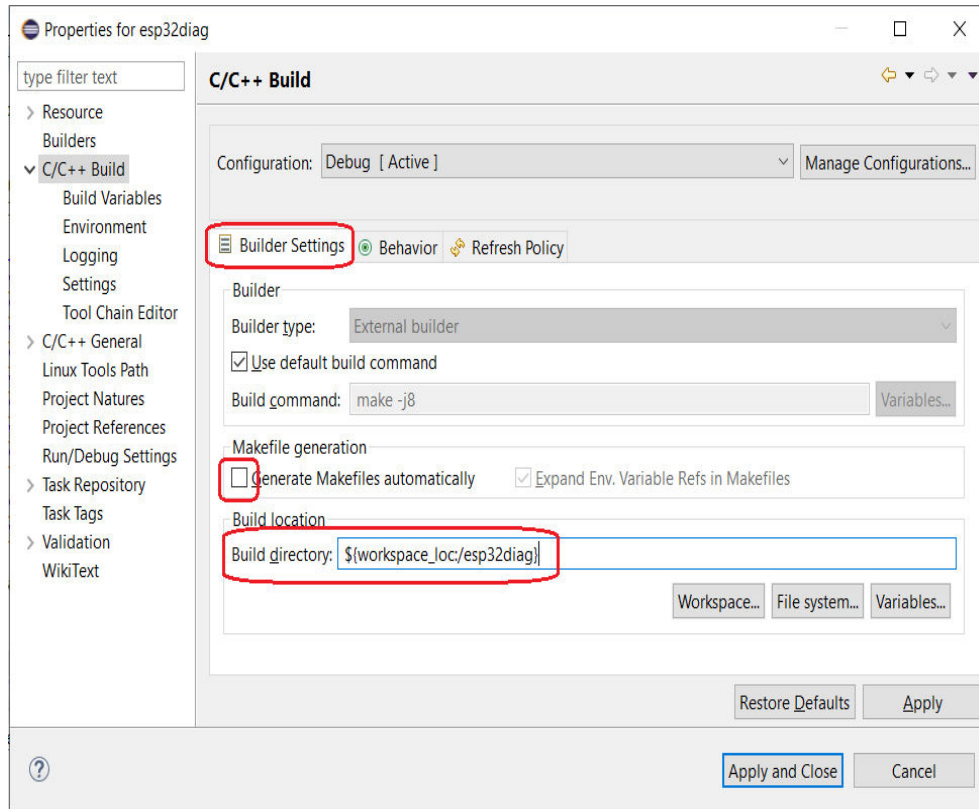
Empty

? < Back Next > Finish Cancel

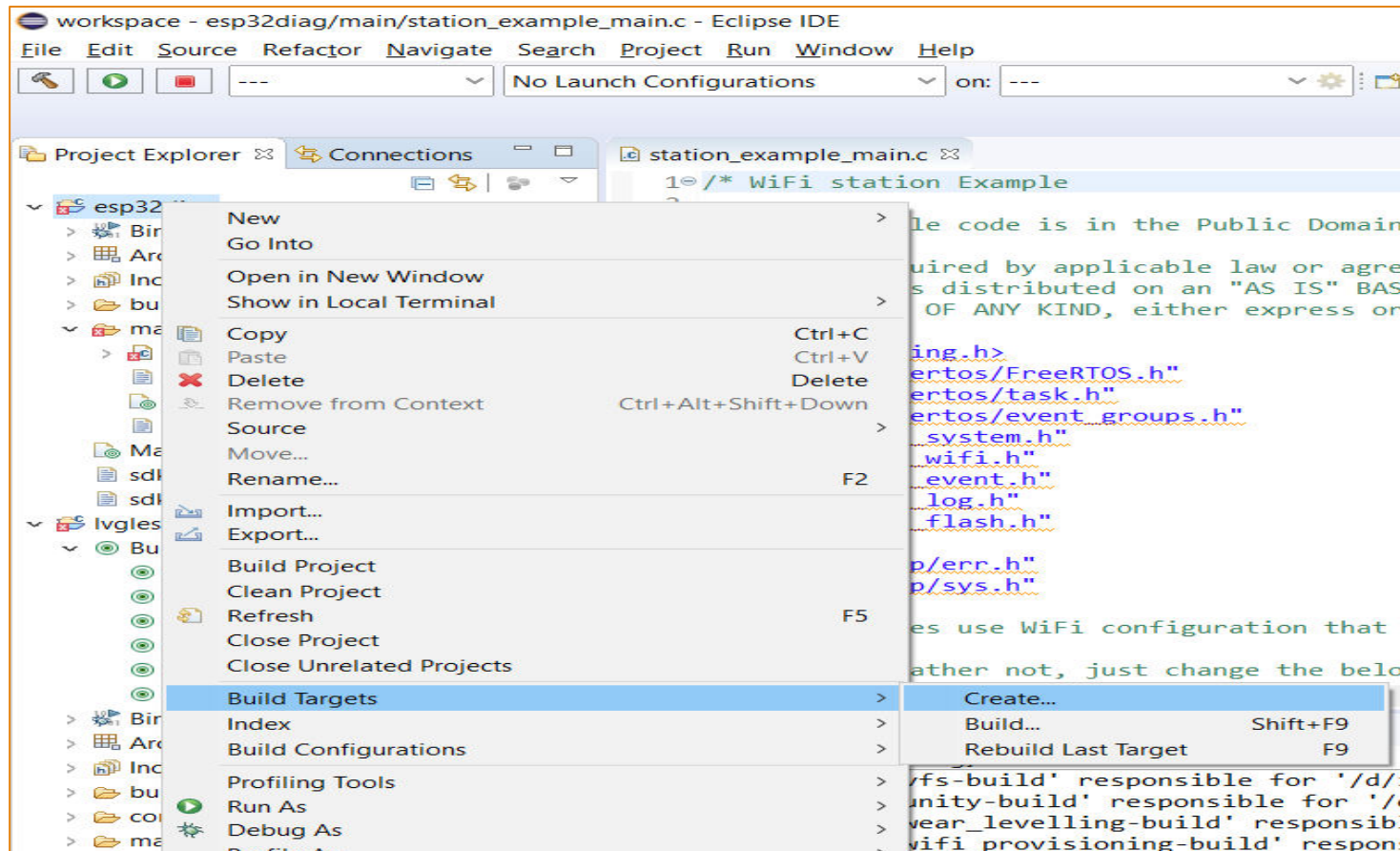
Example of New ESP32 Project 6



Example of New ESP32 Project 7



Example of New ESP32 Project 8



Create Build Target Icon

Example of New ESP32 Project 9

Modify Build Target

Target name: all

Build Target

☒ Same as the target name

Build target: all

Build Command

☒ Use builder settings

Build command: make -j8

Build Settings

☒ Stop on first build error

☒ Run all project builders

OK Cancel

Create Build Target

Target name: app

Build Target

☒ Same as the target name

Build target: app

Build Command

☒ Use builder settings

Build command: make -j8

Build Settings

☒ Stop on first build error

☒ Run all project builders

OK Cancel

Similar for app-flash, clean, flash and monitor

Example of New ESP32 Project 10

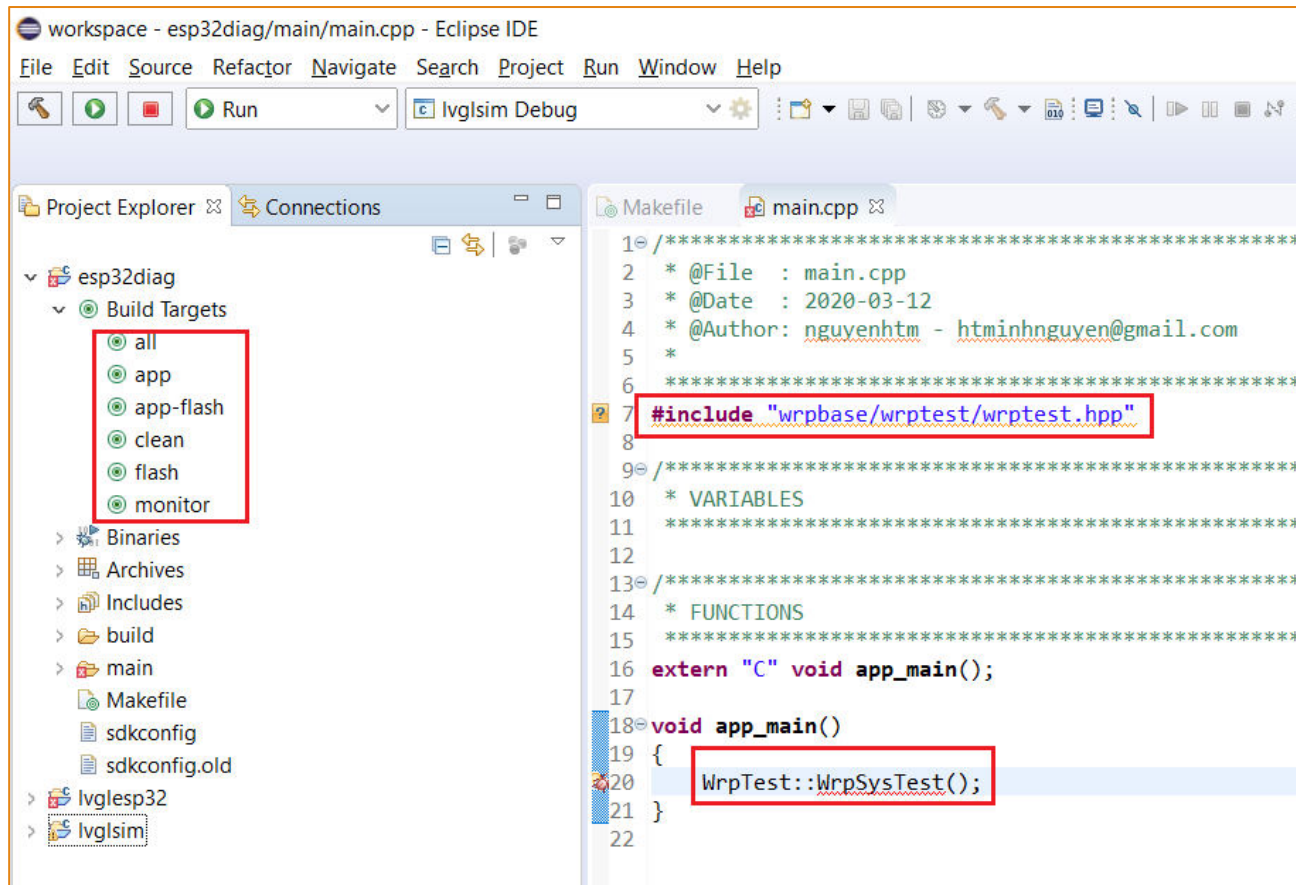
```
Makefile  Makefile  wrpnetwork.cpp  wrpsys.hpp  wrpmidwappfsm.cpp
1# ESP32 Diagnosis Project Name
2PROJECT_NAME := esp32diag
3
4#For C++ language flags
5EXTRA_CPPFLAGS := -DLV_CONF_INCLUDE_SIMPLE -DILI9341_BCKL_ACTIVE_LVL=0
6
7#Use LVGL library and ILI9341 driver
8EXTRA_COMPONENT_DIRS := D:/iotprj/packages/lvgl D:/iotprj/packages/esp32_ili9341/components/drv
9
10#Include headers of lvgl component and headers of ili9341 driver
11COMPONENT_EXTRA_INCLUDES := D:/iotprj/packages D:/iotprj/packages/esp32_ili9341/components
12
13#Use include macros for wrpbase
14CPPFLAGS += -DLVGL_PC_SIMU=0 -DLVGL_ESP32_ILI9341=1 -DUSE_ESP_IDF=1
15#Use wrpbase component
16EXTRA_COMPONENT_DIRS += D:/iotprj/wrpbase/wrpdvr D:/iotprj/wrpbase/wrpsys D:/iotprj/wrpbase/wrpgui D:/iotprj/wrpbase/wrpmidw D:/iotprj/wrpbase/wrphmi D:/iotprj/wrpbase/wrpres
17
18#esp-idf platform makefile
19include $(IDF_PATH)/make/project.mk
20
```

For GUI

For Midw

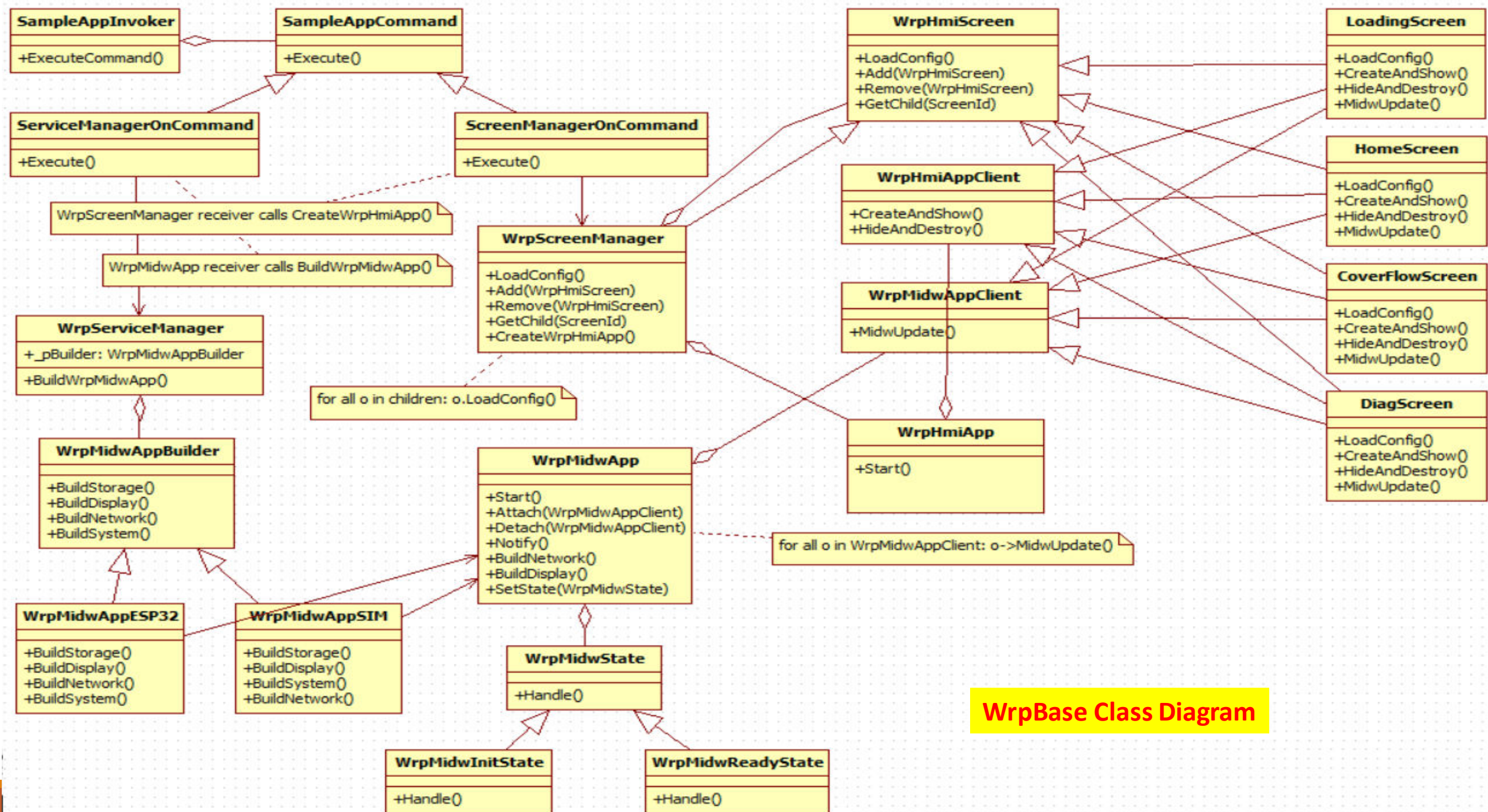
Edit Makefile to connect to WrpBase library

Example of New ESP32 Project 11



- Edit main.cpp to execute the system test function
- Click Build Targets -> app-flash to build and download into ESP32 target

DONE



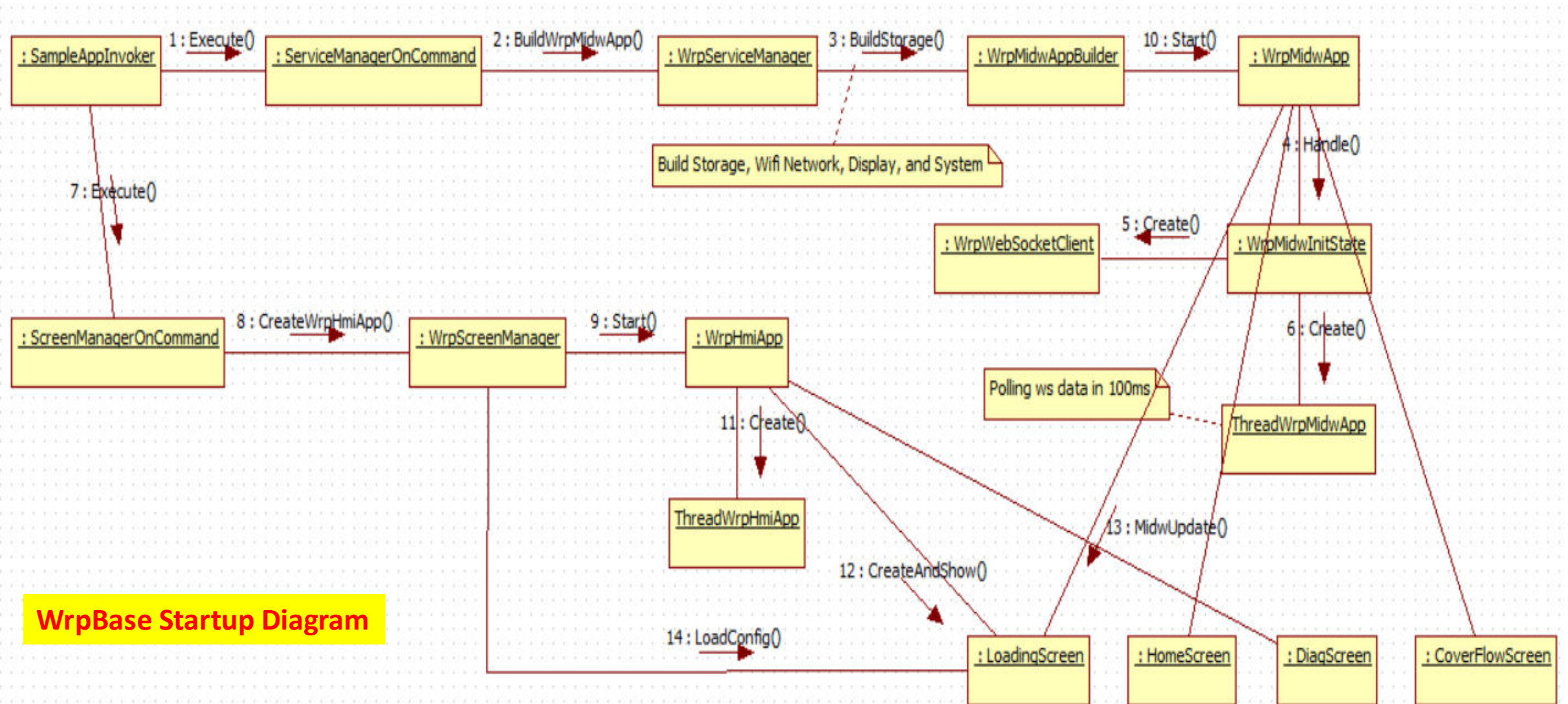


Image Converter

- <https://littlevgl.com/image-converter>: the current configuration for image converter

Image file

D:\iotprj\wrpbase\resources Browse...

Name

background ×

Color format

True color chroma keyed ▼

Alpha byte

Add a 8 bit Alpha value to every pixel

Chroma keyed

Make LV_COLOR_TRANSP (lv_conf.h) pixels to transparent

Output format

Binary RGB565 ▼

Dithering

☒ Dithering of True color images

Convert

References

LVGL: <https://github.com/littlevgl/lvgl>

ESP-IDF: <https://docs.espressif.com/projects/esp-idf/en/latest/index.html>

Patterns: Command, Singleton, Observer, Factory Method, Builder, State, Composite