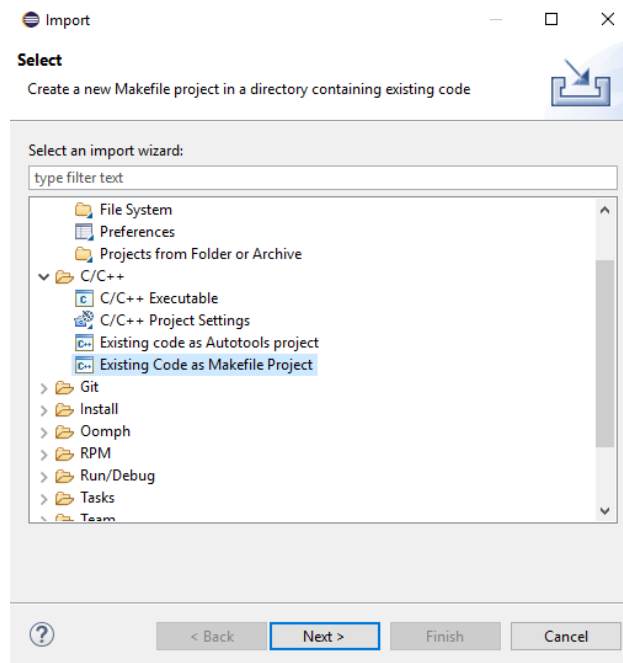
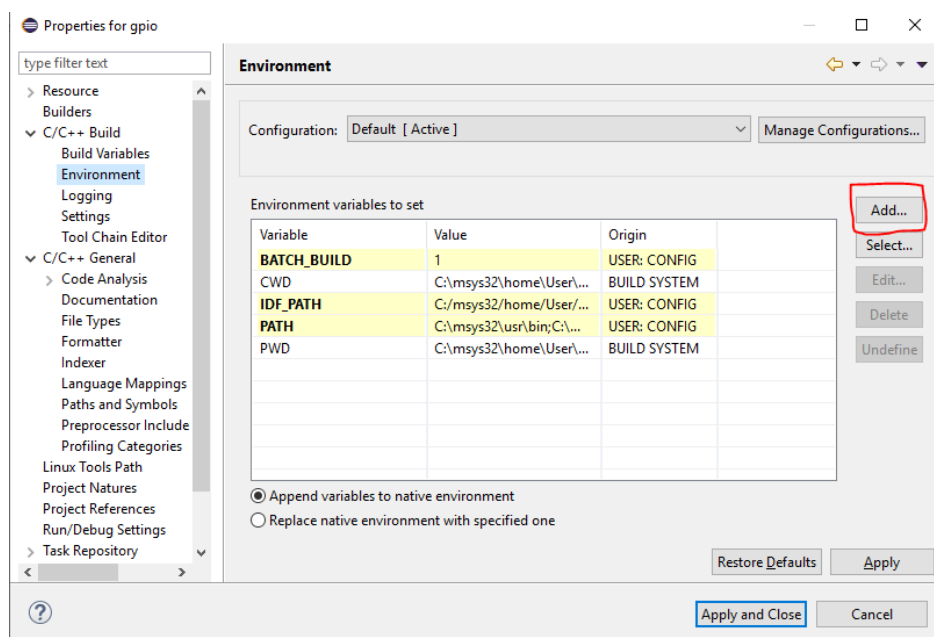


ESP8266 USAGE WITH ECLIPSE IDE

- Open Eclipse C/C++ IDE
- Follow; “File” → “Import” → “C/C++” → “Existing Code as Makefile Project”



- Choose an example project from “C:\msys32\home\User\esp\ESP8266_RTOS_SDK\examples” file.
- Check **Cross GCC** option.
- Project File has been added to Project Explorer.
- Right Click to Project File and follow; “Properties” → “C/C++ Build” → “Environment” → “Add”



Name: BATCH_BUILD, **Value:** 1

Name: IDF_PATH, **Value:** C:/msys32/home/User/esp/ESP8266_RTOS_SDK

Name: PATH, **Value:** C:\msys32\usr\bin;C:\msys32\mingw32\bin;C:\msys32\opt\xtensa-lx106-elf\bin

“Properties” → “C/C++ General”;

➔ “CDT Cross GCC Built in Compiler Settings” :

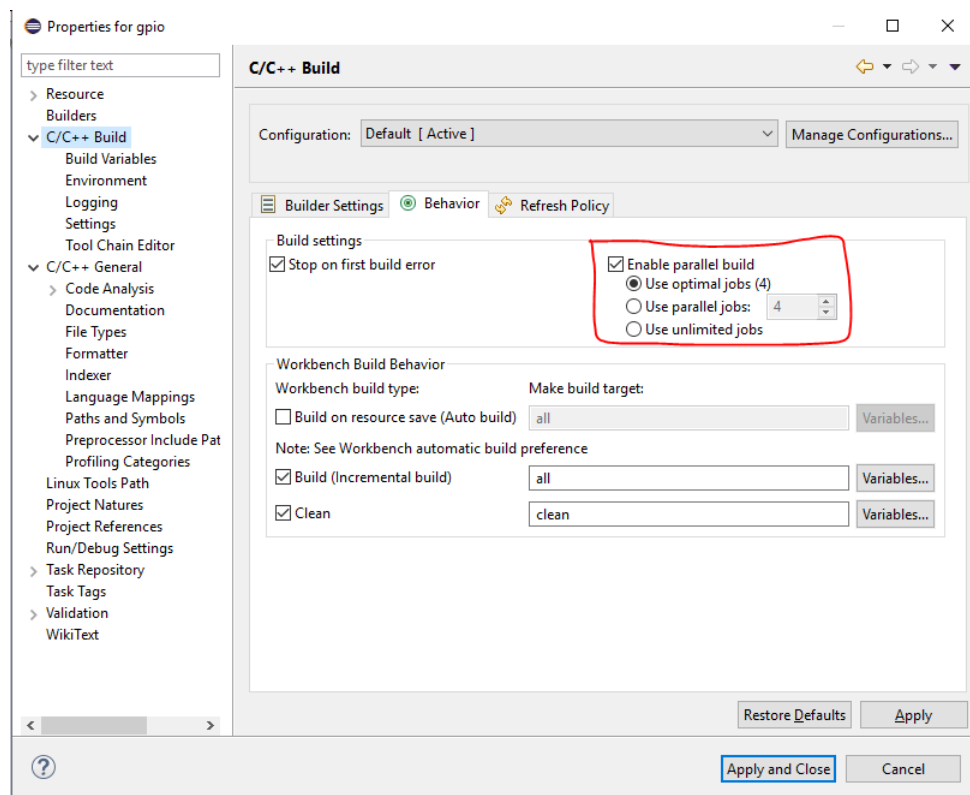
`${COMMAND} ${FLAGS} -E -P -v -dD "${INPUTS}"`

➔ “CDT GCC Build Output Parser”:

`(g?cc)|([gc]\+)|clang`

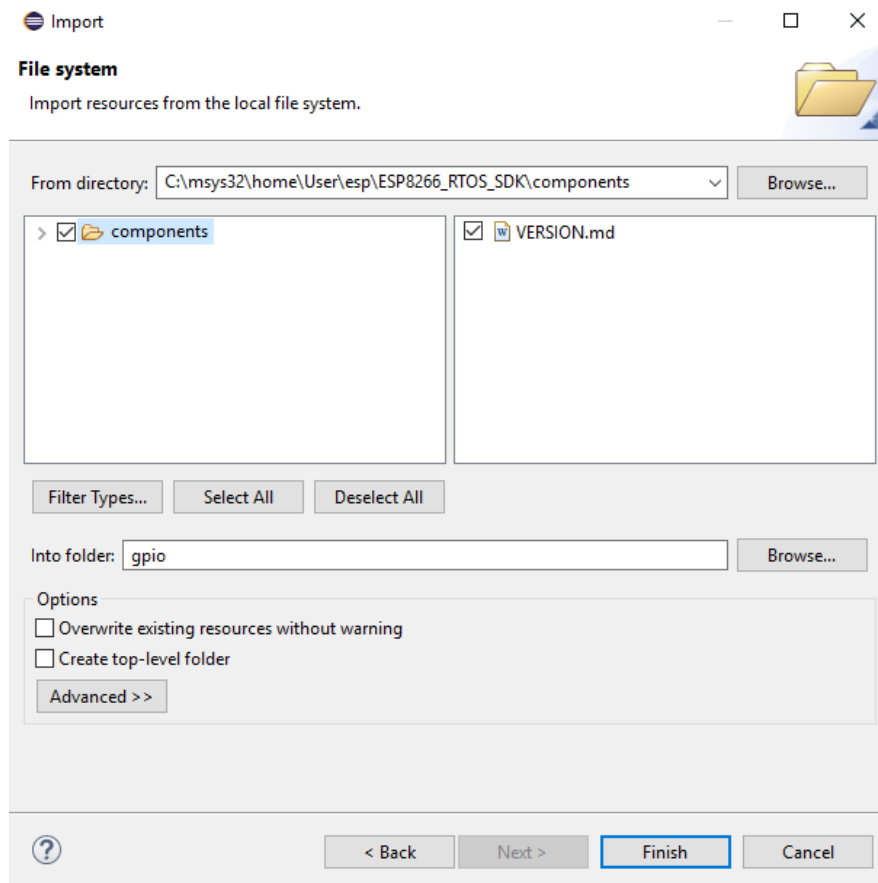
additions were made.

C/C++ Build: Behavior → Check “Enable parallel build”

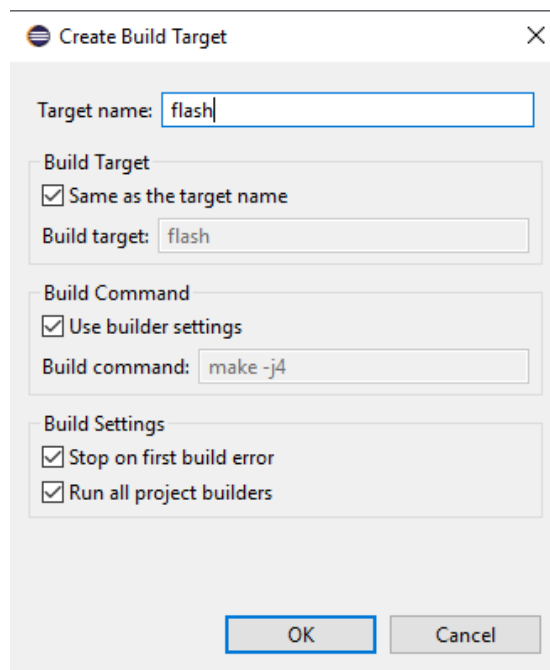


- Right click to Project File → “Add Folder” → Add a file which name is “libraries”
- Right click to “libraries” → Import: “General”: “File System” → Browse:
“C:\msys32\home\User\esp\ESP8266_RTOS_SDK\components”

This step provides adding ESP8266 Libraries to Eclipse IDE.



- Flash Configuration: Project File (Right click) → “Build Targets” → “create” → “flash”



Other Files which ones are needed for compiling are on the follows;

- Firstly we should go to **“.settings”**, **“.cproject”** ve **“.project”** files location. These files exist in below location;

“C:\msys32\home\User\esp\ESP8266_RTOS_SDK\examples\get-started\project_template”

- These files should be copy to project file.
- The “settings” file will remain intact and no changes will be made.
- The sections in the html format (“project_template”) in the “.cproject” and “.project” files which ones are given on the follow should be replaced with the name of the opened project.

.cproject

```
47 </storageModule>
48 <storageModule moduleId="cdtBuildSystem" version="4.0.0">
49     <project id="pwm.null.1880056647" name="pwm"/>
50 </storageModule>
51 <storageModule moduleId="org.eclipse.cdt.core.LanguageSettingsProviders"/>
52 <storageModule moduleId="org.eclipse.cdt.make.core.buildtargets"/>
53 <storageModule moduleId="refreshScope" versionNumber="2">
54     <configuration configurationName="Default">
55         <resource resourceType="PROJECT" workspacePath="/pwm"/>
56     </configuration>
57 </storageModule>
```

.project

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <projectDescription>
3     <name>pwm</name>
4     <comment></comment>
```

After the additions given above, the contents of the file should be as follows;

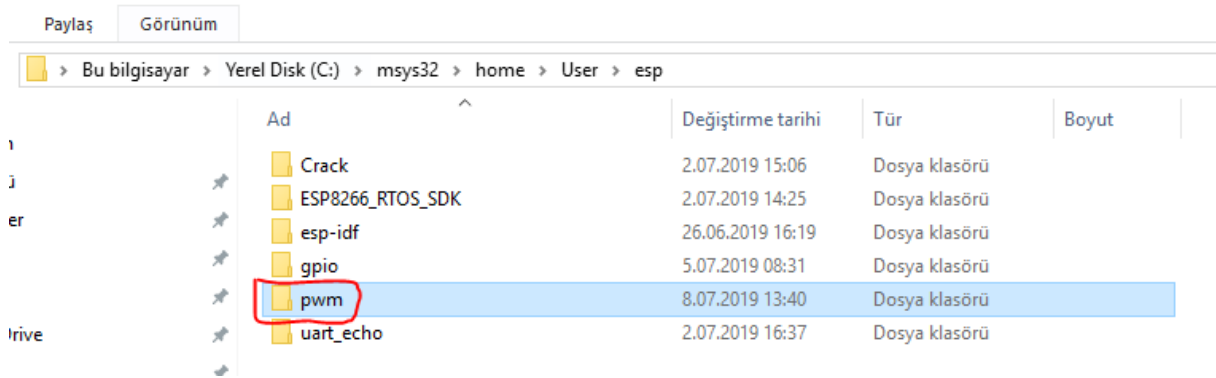
| Bu bilgisayar > Yerel Disk (C:) > msys32 > home > User > esp > pwm | | | | | |
|--|----------------|-------------------|------------------|-------|--|
| | Ad | Değiştirme tarihi | Tür | Boyut | |
| | .settings | 8.07.2019 13:28 | Dosya klasörü | | |
| | .vscode | 8.07.2019 13:33 | Dosya klasörü | | |
| | build | 8.07.2019 14:01 | Dosya klasörü | | |
| | main | 8.07.2019 13:28 | Dosya klasörü | | |
| | .cproject | 8.07.2019 13:29 | CPROJECT Dosyası | 6 KB | |
| | .project | 8.07.2019 13:29 | PROJECT Dosyası | 1 KB | |
| | CMakeLists.txt | 28.06.2019 09:21 | Metin Belgesi | 1 KB | |
| | Makefile | 28.06.2019 09:21 | Dosya | 1 KB | |
| | README.md | 28.06.2019 09:21 | Markdown File | 2 KB | |
| | sdkconfig | 8.07.2019 13:40 | Dosya | 10 KB | |
| | sdkconfig.old | 8.07.2019 13:40 | OLD Dosyası | 10 KB | |
| | wave.png | 28.06.2019 09:21 | PNG Dosyası | 2 KB | |

Finally, you can build and flash the project.

- Project File (Right click) → “Build Project” . (“make -j4” command is used)
- If there is no error has seen on the output screen, code can be flashed to the ESP8266 Card.
“Build Targets” → “Flash”

ESP8266 USAGE WITH VISUAL STUDIO CODE

- Project file has copied from “examples” folder to “esp” folder.



Some changes should be done in the project file. (For this example it is "pwm", you can select another project file). Steps followed;

- The file which name is “.vscode” has created. (Name should be same!)
 - c_cpp_properties.json
 - settings.json
 - tasks.json

“.JSON” files which are above have been added to “.vscode” folder. The contents of these files have been opened and changed with an application such as Notepad ++ or CodeWriter.

c_cpp_properties.json

```
{
  "configurations": [
    {
      "name": "ESP32-Win",
      "includePath": [
        "${workspaceRoot}",
        "${workspaceRoot}/build/include",
        "${workspaceRoot}/build/",
        "C:/msys32/home/User/esp/ESP8266_RTOS_SDK/components/**",
        "C:/msys32/opt/xtensa-lx106-elf/include",

```

```

        "C:/msys32/opt/xtensa-lx106-elf/xtensa-lx106-elf/include",
        "C:/msys32/opt/xtensa-lx106-elf/xtensa-lx106-elf/include/c++/5.2.0",
        "C:/msys32/opt/xtensa-lx106-elf/lib/gcc/xtensa-lx106-elf/5.2.0/include",
        "C:/msys32/opt/xtensa-lx106-elf/lib/gcc/xtensa-lx106-elf/5.2.0/include-fixed"
    ],
    "intelliSenseMode": "clang-x64",
    "browse": {
        "path": [
            "${workspaceRoot}",
            "${workspaceRoot}/build/include",
            "${workspaceRoot}/build/"
        ]
    },
    "C:/msys32/home/User/esp/ESP8266_RTOS_SDK/components/**",

    "C:/msys32/opt/xtensa-lx106-elf/include",
    "C:/msys32/opt/xtensa-lx106-elf/xtensa-lx106-elf/include",
    "C:/msys32/opt/xtensa-lx106-elf/xtensa-lx106-elf/include/c++/5.2.0",
    "C:/msys32/opt/xtensa-lx106-elf/lib/gcc/xtensa-lx106-elf/5.2.0/include",
    "C:/msys32/opt/xtensa-lx106-elf/lib/gcc/xtensa-lx106-elf/5.2.0/include-fixed"
    ],
    "limitSymbolsToIncludedHeaders": true,
    "databaseFilename": "${workspaceRoot}/.vscode/browse.vc.db"
},
"cStandard": "c11",
"cppStandard": "c++17"
}
],
"version": 4
}

```

settings.json

```

{
    "terminal.integrated.shell.windows": "C:/msys32/mingw32.exe",
    //C:/msys32/usr/bin/bash.exe",

    "terminal.integrated.shellArgs.windows": [
        "--login",
    ],
    "terminal.integrated.env.windows": {
        "CHERE_INVOKING": "1",
        "MSYSTEM": "MINGW32",
    },
    "files.associations": {
        "task.h": "c",
        "freertos.h": "c"
    },
}

```

tasks.json

```
{
  // See https://go.microsoft.com/fwlink/?LinkId=733558
  // for the documentation about the tasks.json format
  "version": "2.0.0",
  "tasks": [
    {
      "label": "F4",
      "group": "build",
      "command": "make",
      "type": "shell",
      "args": [
        "-j8",
        "app"
      ],
      "presentation": {
        "reveal": "always",
      },
    },
    {
      "label": "F5",
      "command": "make",
      "type": "shell",
      "args": [
        "-j8",
        "flash"
      ],
      "presentation": {
        "reveal": "always",
      },
    },
    {
      "label": "F8",
      "command": "make",
      "type": "shell",
      "args": [
        "-j8",
        "flash",
        "monitor"
      ],
      "presentation": {
        "reveal": "always",
      },
    },
    {
      "label": "F9",
      "command": "make",
      "type": "shell",
      "args": [
        "clean"
      ],
      "presentation": {
        "reveal": "always",
      },
    },
    {
      "label": "F12",
      "type": "shell",
      "windows": {
```

```

        "command": "C:/msys32/mingw32.exe",
        "args": [
            "make",
            "menuconfig"
        ]
    },
    "presentation": {
        "reveal": "always",
    },
    "problemMatcher": []
}
]
}

```

The purpose of the above file changes;

- ➔ Showing the path to the required ESP8266 Libraries (Components)
- ➔ Running the "MINGW32 - bash" as terminal

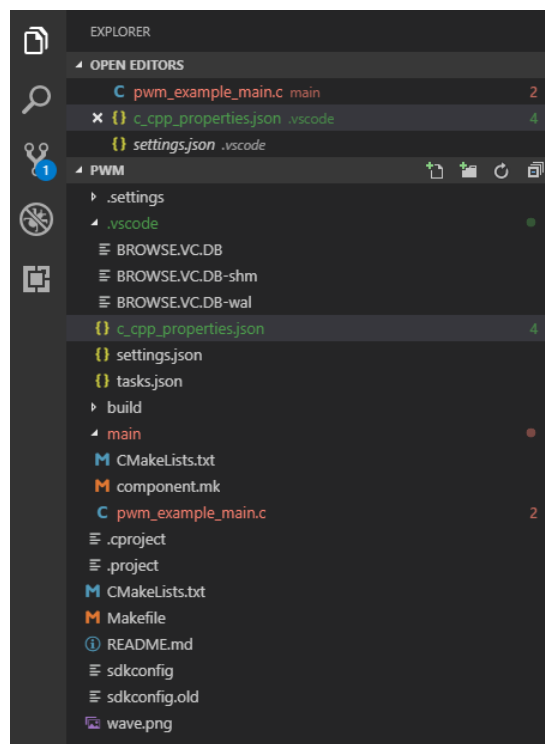
```

User@DESKTOP-NAGU1VE MINGW32 ~/esp/pwm
$ printenv IDF_PATH
C:\msys32\home\User\esp\ESP8266_RTOS_SDK

User@DESKTOP-NAGU1VE MINGW32 ~/esp/pwm
$ make build
DEFCONFIG
#
# configuration written to /home/User/esp/pwm/sdkconfig
#
MENUCONFIG

```

- ➔ Showing location of the "xtensa-lx106f-elf-gcc" compiler
- ➔ When you open a project file with Visual Studio Code, it should look like the following;



- **CmakeList** and **MakeFile** files should absolutely exist.
- “Terminal” → “New Terminal” → MINGW32 bash will be opened
- Serial Port configuration can be set with “make menuconfig” command.
- Project can be build with “make build” command.

```
User@DESKTOP-NAGU1VE MINGW32 ~/esp/pwm
$ make build
Toolchain path: /opt/xtensa-lx106-elf/bin/xtensa-lx106-elf-gcc
Toolchain version: crosstool-ng-1.22.0-92-g8facf4c0
Compiler version: 5.2.0
```

- Compiled project (code) can be flashed to the ESP8266 Card

```
User@DESKTOP-NAGU1VE MINGW32 ~/esp/pwm
$ make flash
Toolchain path: /opt/xtensa-lx106-elf/bin/xtensa-lx106-elf-gcc
Toolchain version: crosstool-ng-1.22.0-92-g8facf4c0
Compiler version: 5.2.0
CC build/bootloader/main/bootloader_start.o
AR build/bootloader/main/libmain.a
CC build/bootloader/bootloader_support/src/bootloader_random.o
CC build/bootloader/bootloader_support/src/flash_encrypt.o
CC build/bootloader/bootloader_support/src/bootloader_sha.o
CC build/bootloader/bootloader_support/src/esp_image_format.o
CC build/bootloader/bootloader_support/src/flash_partitions.o
CC build/bootloader/bootloader_support/src/secure_boot_signatures.o
CC build/bootloader/bootloader_support/src/bootloader_clock.o
CC build/bootloader/bootloader_support/src/secure_boot.o
CC build/bootloader/bootloader_support/src/bootloader_common.o
CC build/bootloader/bootloader_support/src/bootloader_utility.o
CC build/bootloader/bootloader_support/src/bootloader_init.o
CC build/bootloader/bootloader_support/src/efuse.o
CC build/bootloader/bootloader_support/src/flash_qio_mode.o
CC build/bootloader/bootloader_support/src/bootloader_flash.o
AR build/bootloader/bootloader_support/libbootloader_support.a
CC build/bootloader/spi_flash/src/spi_flash.o
CC build/bootloader/spi_flash/src/spi_flash_raw.o
CC build/bootloader/spi_flash/port/port.o
```

ATALAY PABUŞÇU