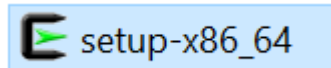This method is using Makefile to build the project. In term of flexibility, this method is much better.

1. Double click the Cygwin installer to begin the installation. Select location on where to put the cygwin files. In my case, I put in Drive C
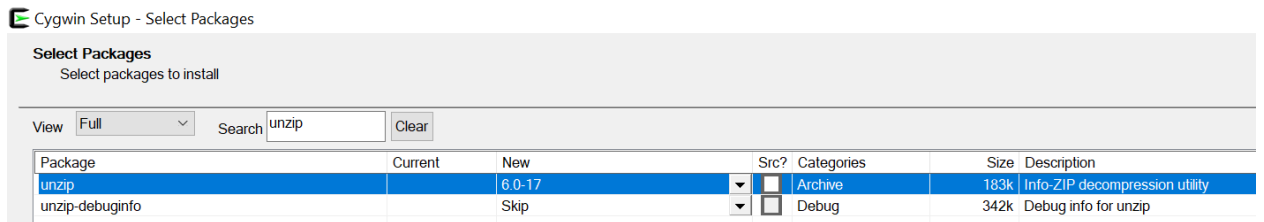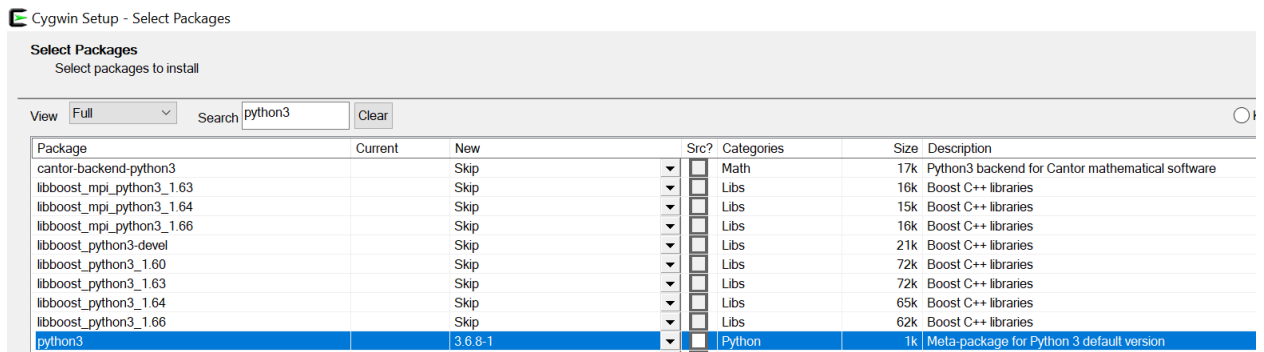
   https://cygwin.com/install.html

2. Select make, gcc-g++, python3, unzip

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Full   Search: python3   Clear

Package	Current	New	Src?	Categories	Size	Description
cantor-backend-python3		Skip		Math	17k	Python3 backend for Cantor mathematical software
libboost_mpi_python3_1.63		Skip		Libs	16k	Boost C++ libraries
libboost_mpi_python3_1.64		Skip		Libs	15k	Boost C++ libraries
libboost_mpi_python3_1.66		Skip		Libs	16k	Boost C++ libraries
libboost_python3-devel		Skip		Libs	21k	Boost C++ libraries
libboost_python3_1.60		Skip		Libs	72k	Boost C++ libraries
libboost_python3_1.63		Skip		Libs	72k	Boost C++ libraries
libboost_python3_1.64		Skip		Libs	65k	Boost C++ libraries
libboost_python3_1.66		Skip		Libs	62k	Boost C++ libraries
python3		3.6.8-1		Python	1k	Meta-package for Python 3 default version

Cygwin Setup - Select Packages

Select Packages
Select packages to install

View: Full   Search: unzip   Clear

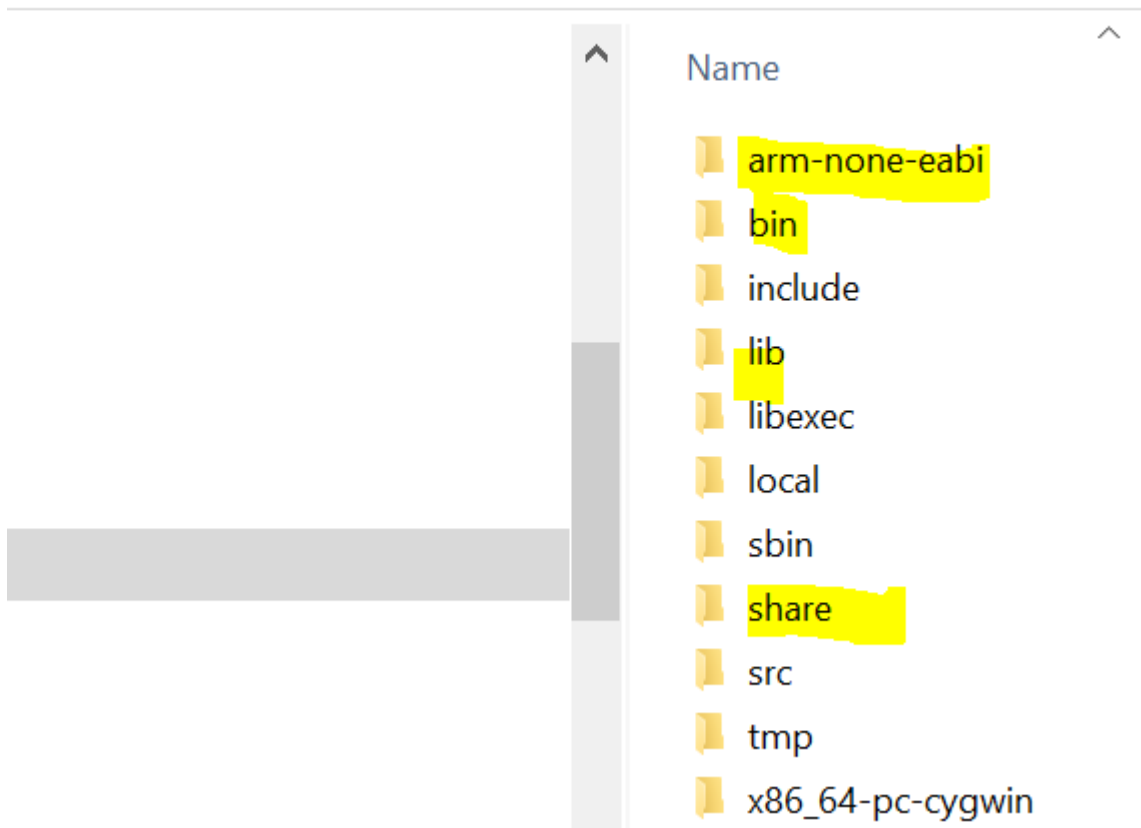| Package	Current	New	Src?	Categories	Size	Description
unzip		6.0-17		Archive	183k	Info-ZIP decompression utility
unzip-debuginfo		Skip		Debug	342k	Debug info for unzip

3. Click Next and wait until installation is completed.

4. Download the gcc-arm-none-aebi (e.g. **gcc-arm-none-eabi-10-2020-q4-major-win32.zip**)
   https://developer.arm.com/tools-and-software/open-source-software/
   developer-tools/gnu-toolchain/gnu-rm/downloads

5. Extract the downloaded gcc-arm-none-aebi

gcc-arm-none-eabi-10-2020-q4-major

Name

- arm-none-eabi
- bin
- lib
- share
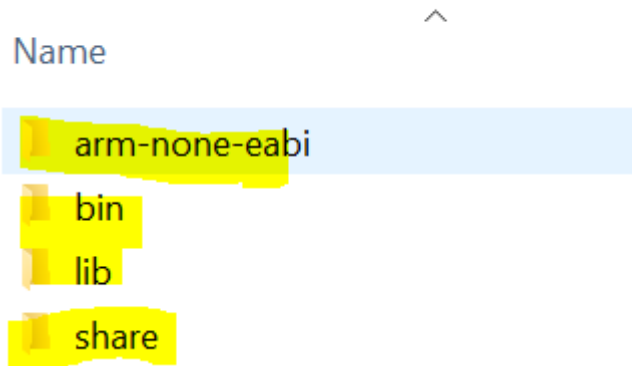
6. Copy and paste the extracted file inside C:/cygwin64/usr

> This PC > OS (C:) > cygwin64 > usr

Name

arm-none-eabi

bin

include

lib

libexec

local

sbin

share

src

tmp

x86_64-pc-cygwin

I copy them in "usr" because on my Ubuntu the gcc-arm is also inside "usr" folder.

7. Compile the project. Make sure inside "main.mak" update the path as follows
TOOLCHAIN_ROOT := **c:/cygwin64/usr**

```
/cygdrive/c/Rudy/mc_sw

User@DESKTOP-ESERESB /cygdrive/c/Rudy/mc_sw
$ make -j8 BUILDSET="bs_4wd"
[CC] main.c
[CC] system_stm32f4xx.c
[CC] syscalls.c
[CC] port.c
[CC] queue.c
[CC] list.c
[CC] croutine.c
[CC] tasks.c
```

```
/cygdrive/c/Rudy/mc_sw

[CC] stm32f4xx_ltdc.c
[CC] stm32f4xx_qspi.c
[CC] stm32f4xx_rng.c
[CC] stm32f4xx_rng.c
[CC] stm32f4xx_sdio.c
[CC] stm32f4xx_tim.c
[CC] stm32f4xx_usart.c
[CC] stm32f4xx_spdifrx.c
[CC] stm32f4xx_sdio.c
[CC] stm32f4xx_wwdg.c
[CC] stm32f4xx_spi.c
[CC] stm32f4xx_syscfg.c
[CC] stm32f4xx_syscfg.c
[CC] SEGGER_SYSVIEW_Config_FreeRTOS.c
[CC] SEGGER_SYSVIEW_FreeRTOS.c
[CC] SEGGER_RTT.c
[CC] SEGGER_SYSVIEW.c
[AS] startup_stm32.s
[LD] mambabot_4wd.elf
[HEX] mambabot_4wd.hex
[BIN] mambabot_4wd.bin
```

To clean-up the generated objects and binaries

```
/cygdrive/c/Rudy/mc_sw

User@DESKTOP-ESERESB /cygdrive/c/Rudy/mc_sw
$ make clean
[RM] OBJ
[RM] BIN
```