



Synerduino STM Fork of INAV and the
Synerduino Target

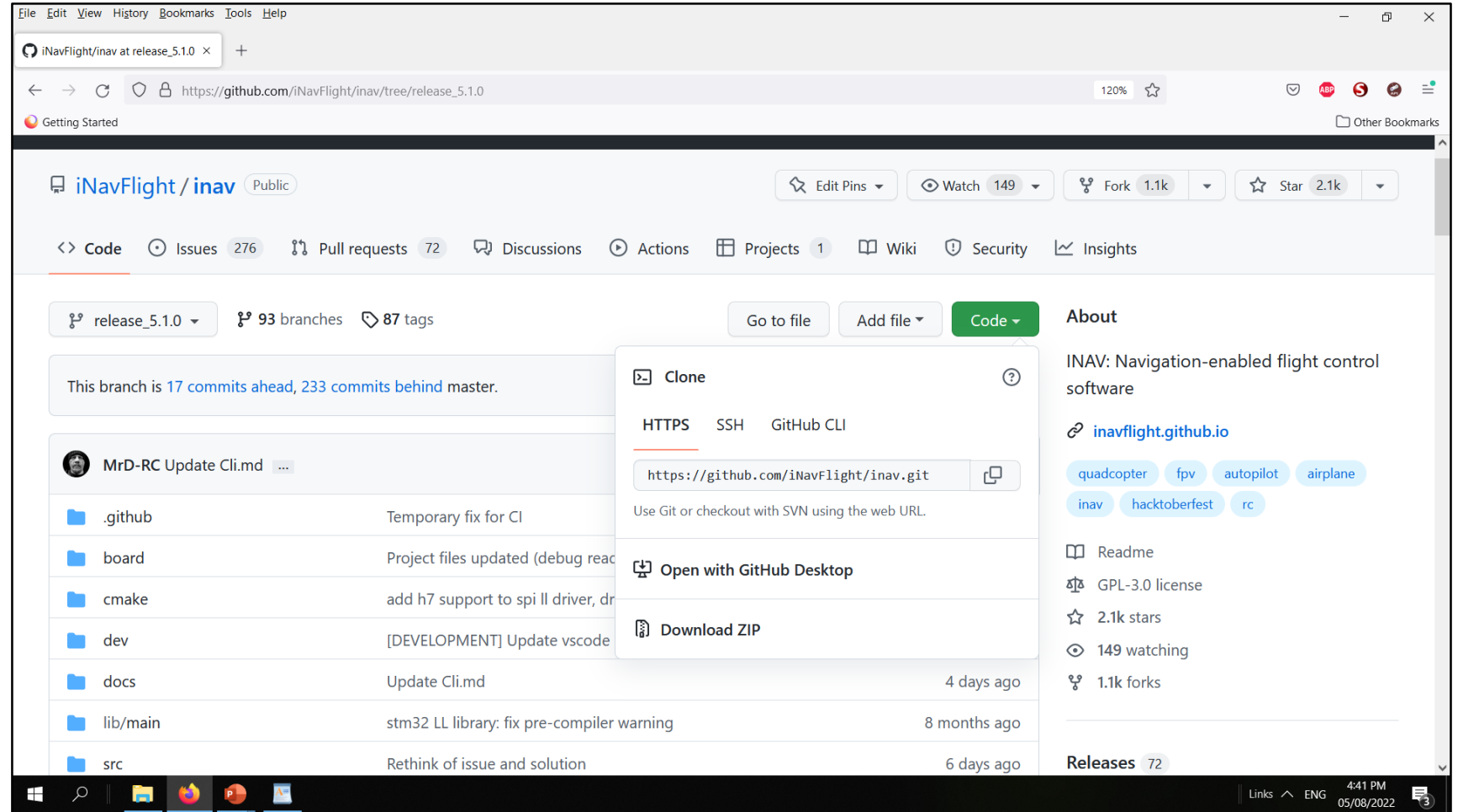
Synerduino STM Target
Compilation into Hexfile

Download the INAV Release 5.1.0 Release 6.0.0-FP1

First you need to download
the INAV release version

These are available in the
official github page or
Synerflight SynerduinoSTM in
download tab

Prepare to Build the Hex File



https://github.com/iNavFlight/inav/tree/release_5.1.0

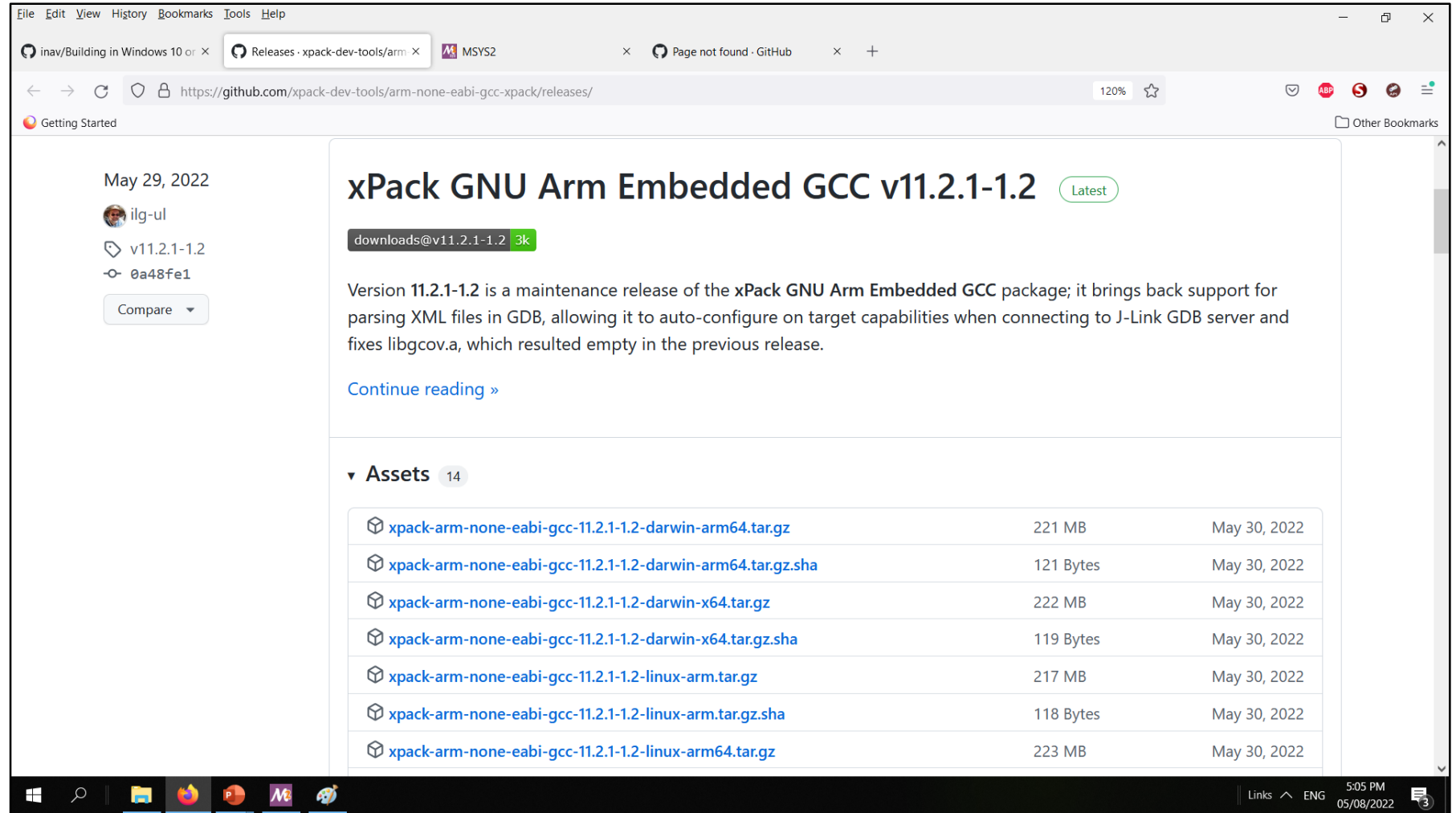
<https://github.com/iNavFlight/inav/tree/6.0.0-FP1>

Download the Xpack
Release 9.2.1-1.1-Win32-
x64 or higher

Im using
INAV-5.1.0
INAV-6.0.0-FP1

xpack-arm-none-eabi-gcc -10.2.1-1.1

These are available in the
official github page or
Synerflight SynerduinoSTM in
download tab



May 29, 2022
ilg-ul
v11.2.1-1.2
0a48fe1
Compare

xPack GNU Arm Embedded GCC v11.2.1-1.2 Latest

downloads@v11.2.1-1.2 3k

Version 11.2.1-1.2 is a maintenance release of the xPack GNU Arm Embedded GCC package; it brings back support for parsing XML files in GDB, allowing it to auto-configure on target capabilities when connecting to J-Link GDB server and fixes libgconv.a, which resulted empty in the previous release.

[Continue reading »](#)

Assets 14

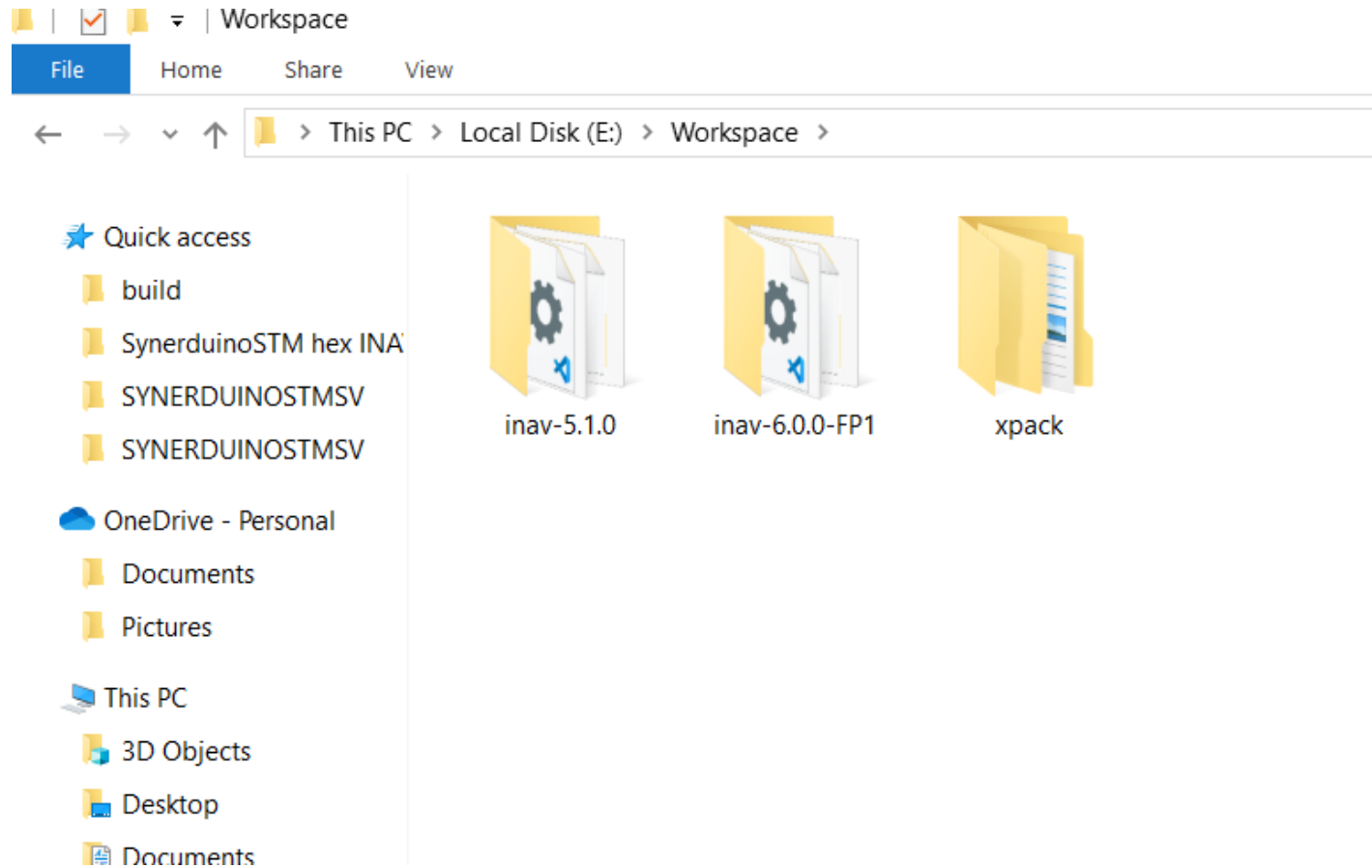
xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-arm64.tar.gz	221 MB	May 30, 2022
xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-arm64.tar.gz.sha	121 Bytes	May 30, 2022
xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-x64.tar.gz	222 MB	May 30, 2022
xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-x64.tar.gz.sha	119 Bytes	May 30, 2022
xpack-arm-none-eabi-gcc-11.2.1-1.2-linux-arm.tar.gz	217 MB	May 30, 2022
xpack-arm-none-eabi-gcc-11.2.1-1.2-linux-arm.tar.gz.sha	118 Bytes	May 30, 2022
xpack-arm-none-eabi-gcc-11.2.1-1.2-linux-arm64.tar.gz	223 MB	May 30, 2022

<https://github.com/xpack-dev-tools/arm-none-eabi-gcc-xpack/releases/>

<https://github.com/xpack-dev-tools/arm-none-eabi-gcc-xpack/releases/download/v9.2.1-1.1/xpack-arm-none-eabi-gcc-9.2.1-1.1-win32-x64.zip>

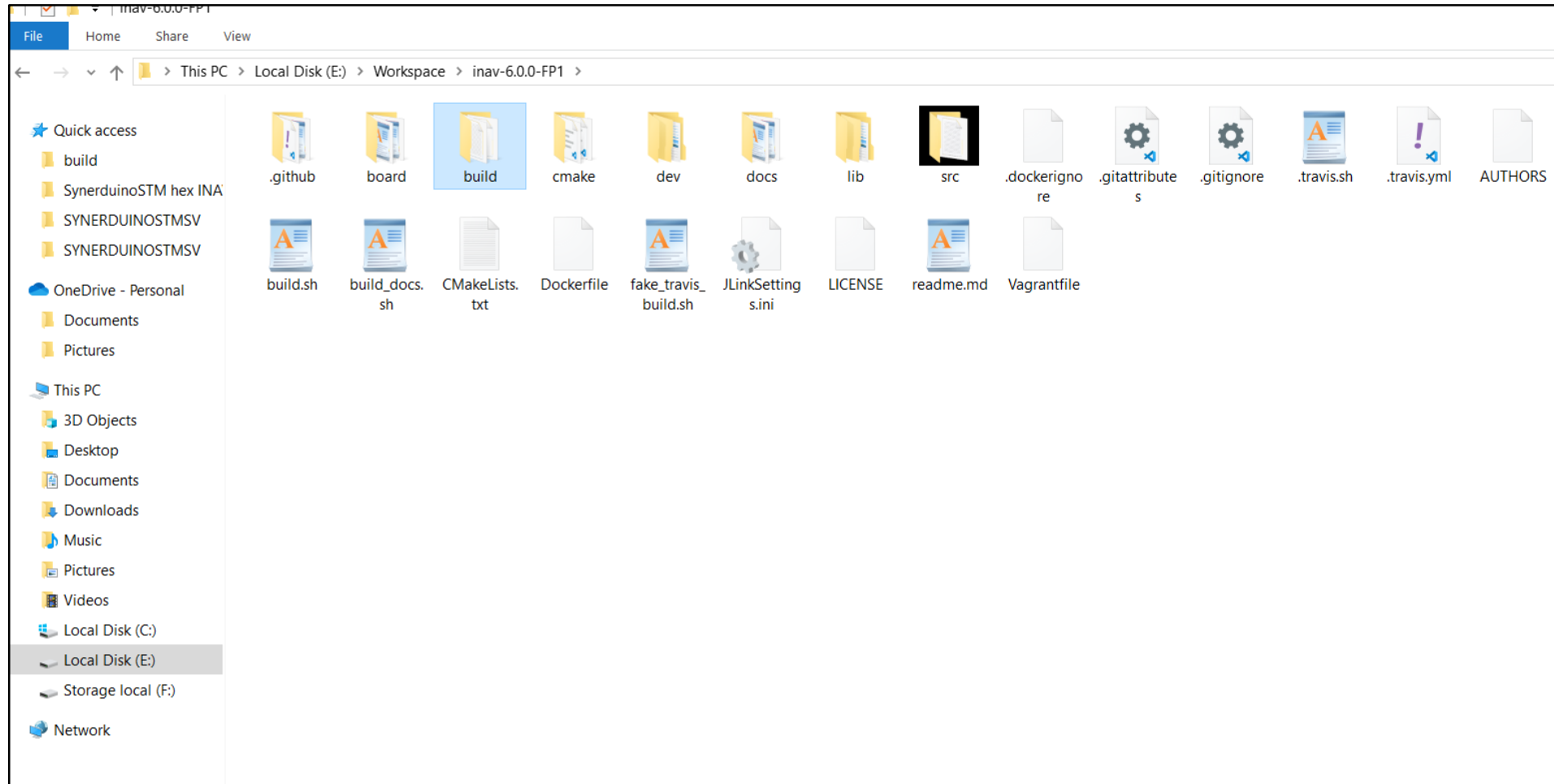
<https://github.com/xpack-dev-tools/arm-none-eabi-gcc-xpack/releases/download/v10.2.1-1.1/xpack-arm-none-eabi-gcc-10.2.1-1.1-win32-x64.zip>

After installation Go Create a Workspace folder in your local Drive in my case Drive E , and extract and place the following directories [INAV-5.1.0 or INAV-6.0.0-FP1 Folder](#) , [Xpack folder](#)

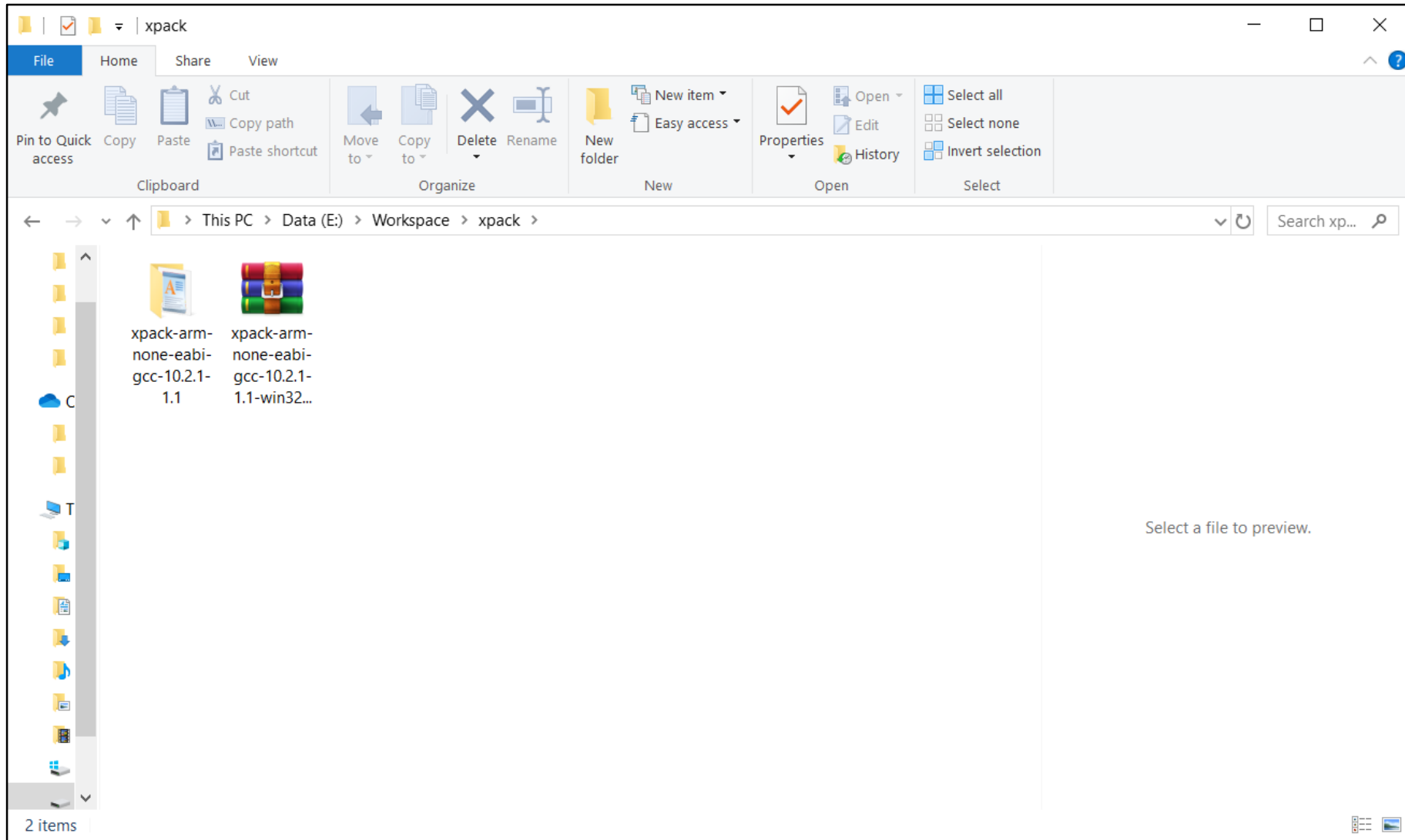


Note: I place my Workspace Folder on Drive E: as I was using SSD on my Drive C so not to fill it up

Create new folder inside INAV 5.1.0 or INAV 6.0.0-FP1 Directory and Name it build



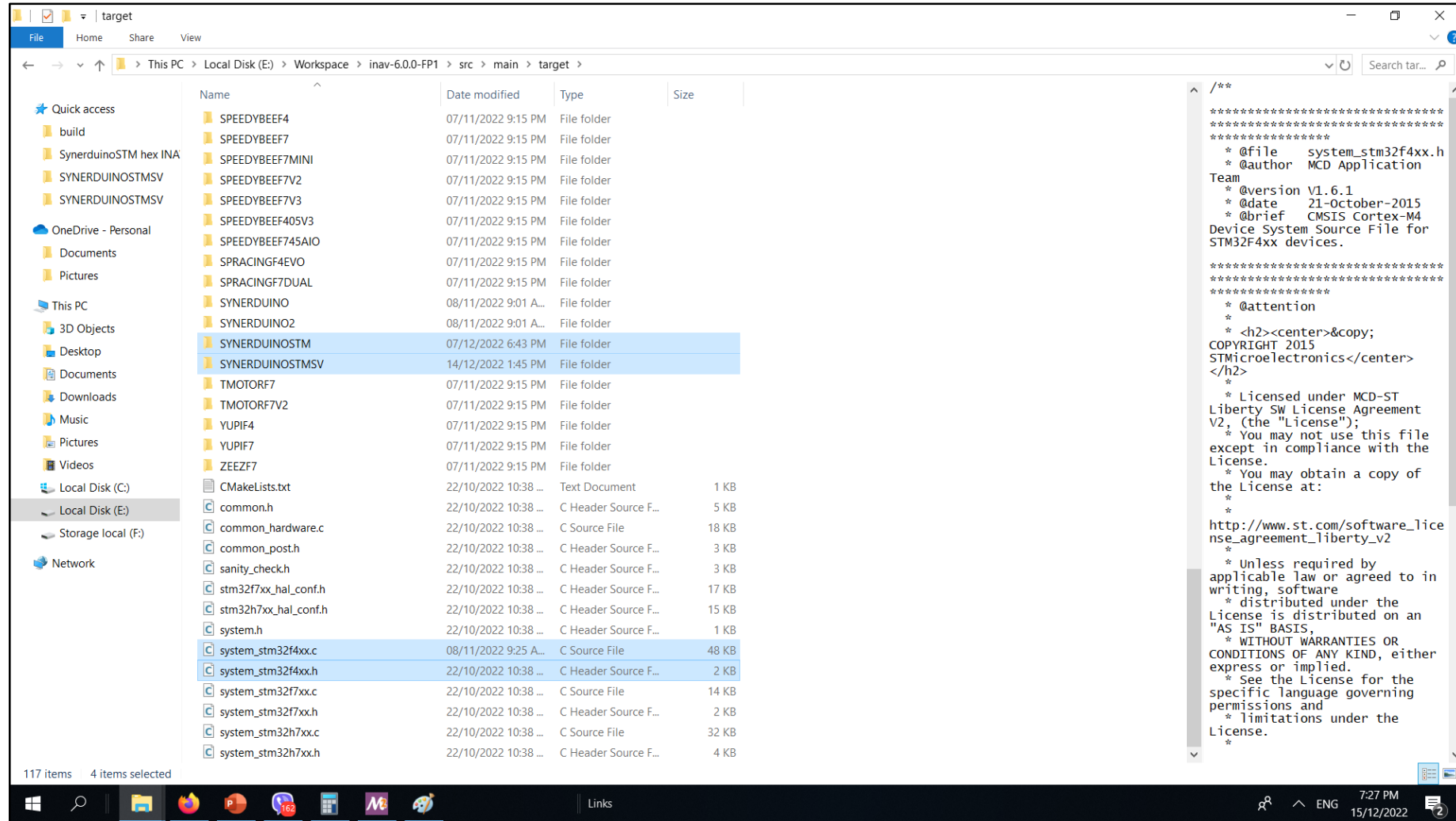
Inside the Xpack Folder Extract the xpack-arm-none-eabi-gcc -10.2.1-1.1



Im using xpack-arm-none-eabi-gcc -
10.2.1-1.1

xpack-arm-none-eabi-gcc-10.2.1-1.1 is extracted to the Xpack folder

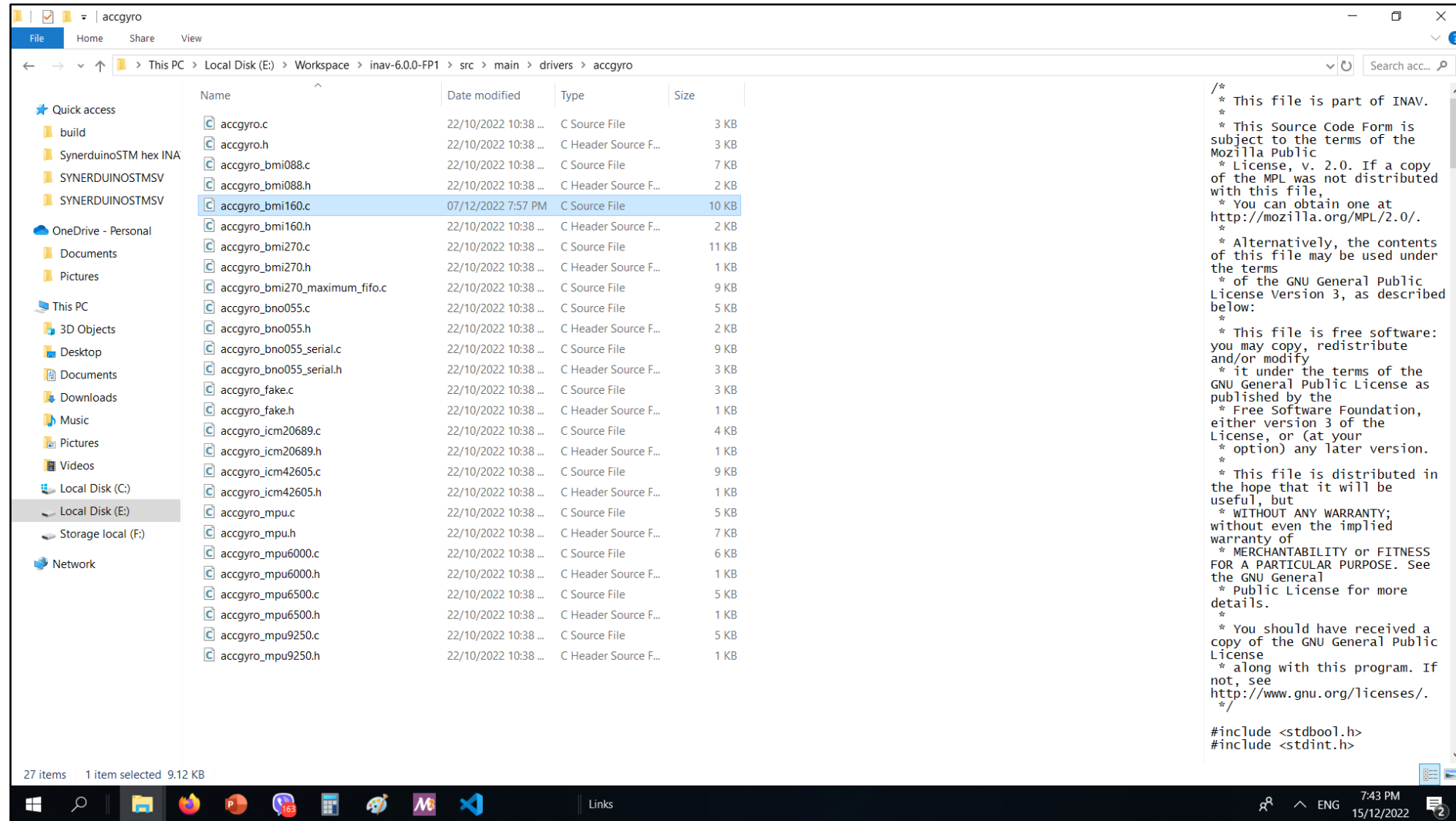
Download the Synerduino Target Folder and Extract to Local drive and place it in the following (E:/Workspace/inav-5.1.0/scr/main/target) or (E:/Workspace/inav-6.0.0-FP1/scr/main/target)
Readme file is also available in the download



Other Download of the Synerduino Target Folder is the sensor driver this is also Extract to Local drive and place it in the following

(E:/Workspace/inav-5.1.0/src/main/drivers/accgyro) or (E:/Workspace/inav-6.0.0-FP1/src/main/drivers/accgyro)

Readme file is also available in the download (accgyro_BMI160) is also modify to use with the SynerduinoSTM shield



General Info

This is a guide on how to use Windows MSYS2 distribution and building platform to build iNav firmware. This environment is very simple to manage and does not require installing docker for Windows which may get in the way of VMWare or any other virtualization software you already have running for other reasons. Another benefit of this approach is that the compiler runs natively on Windows, so performance is much better than compiling in a virtual environment or a container.

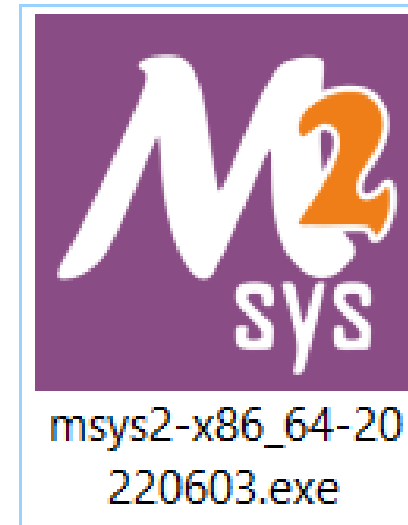
Download Arm Embedded GCC toolkit from The xPack Project

<https://xpack.github.io/arm-none-eabi-gcc/>

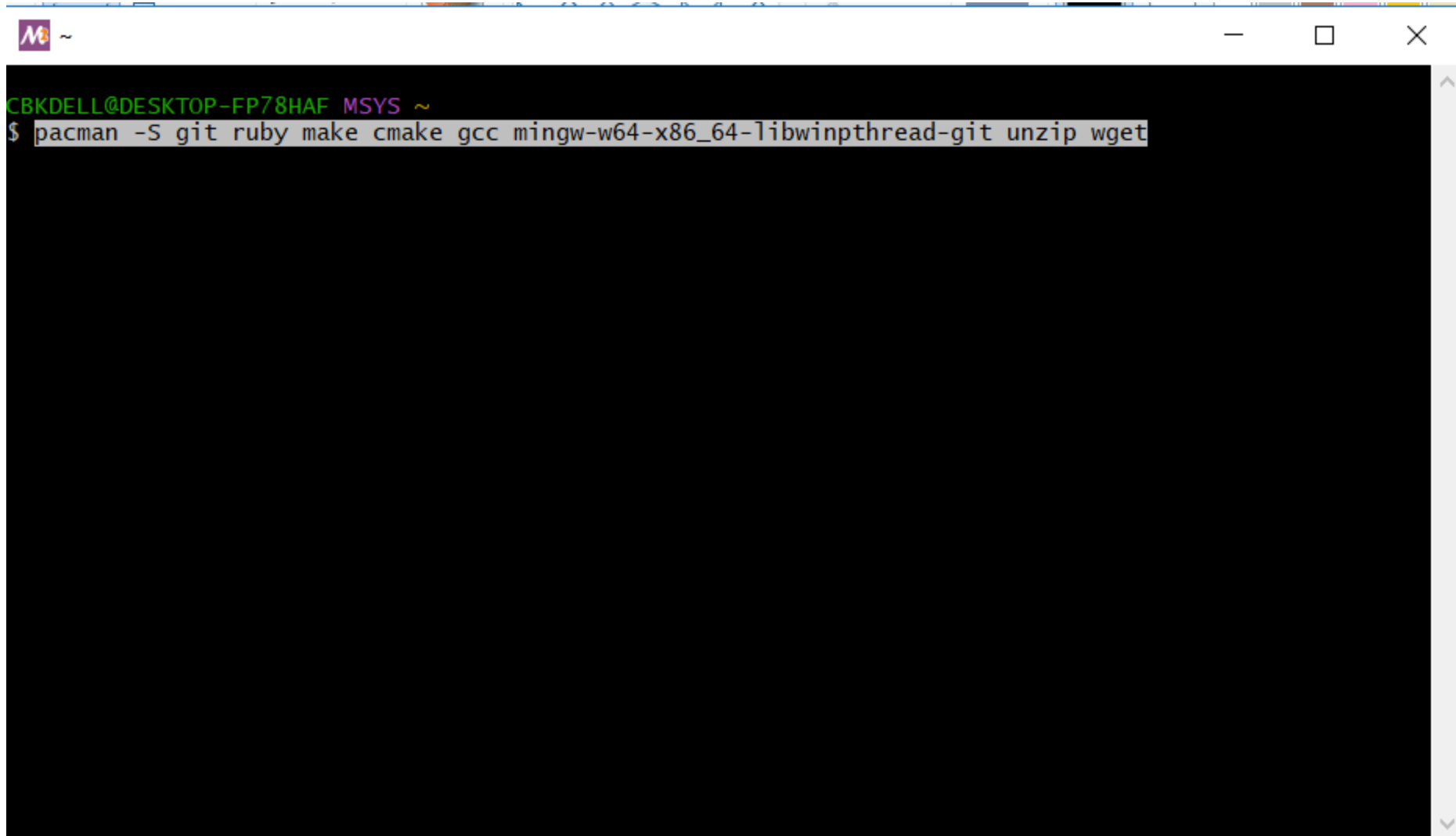
Download MSYS2 for your architecture (most likely 64-bit)

<https://www.msys2.org/wiki/MSYS2-installation/>

Install and Run msys2-x86



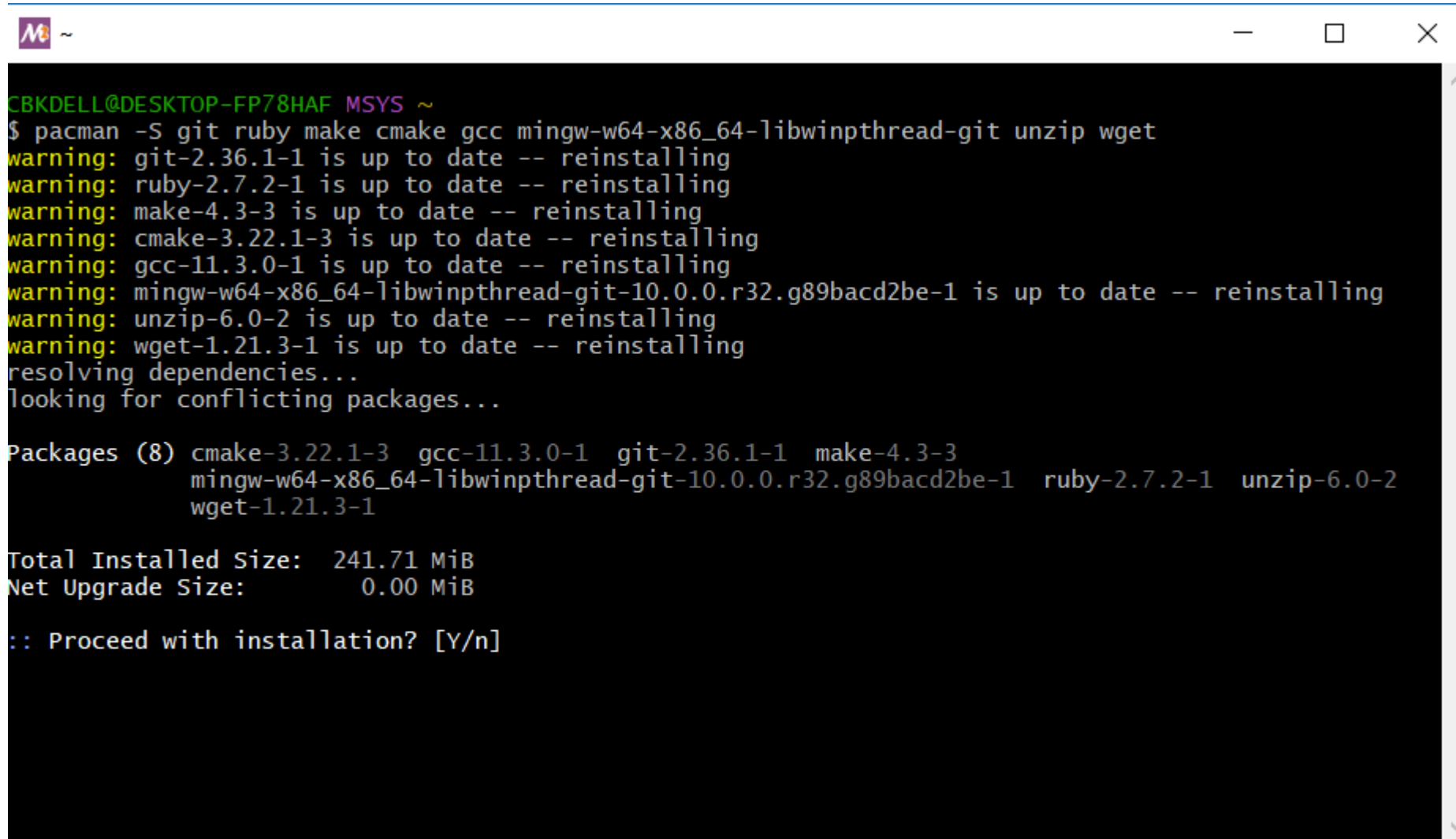
Copy and paste this script to install Ruby ,Cmake , Make and Gcc

A screenshot of a terminal window with a black background. The window title bar shows a purple icon and a tilde symbol. The prompt is 'CBKDELL@DESKTOP-FP78HAF MSYS ~'. The command 'pacman -S git ruby make cmake gcc mingw-w64-x86_64-libwinpthread-git unzip wget' is entered and highlighted with a light blue selection box. The terminal has a scrollbar on the right side.

```
CBKDELL@DESKTOP-FP78HAF MSYS ~  
$ pacman -S git ruby make cmake gcc mingw-w64-x86_64-libwinpthread-git unzip wget
```

```
pacman -S git ruby make cmake gcc mingw-w64-x86_64-libwinpthread-git unzip wget
```

Y to Proceed installation

A terminal window with a dark background and light-colored text. The window title bar shows a small icon and the text '~'. The terminal content shows a user running a pacman command to install several packages. The output indicates that all packages are already up to date and will be reinstalled. The user is prompted to proceed with the installation, and they have entered 'Y'.

```
pacman -S git ruby make cmake gcc mingw-w64-x86_64-libwinpthread-git unzip wget
```

After installation

```
resolving dependencies...
looking for conflicting packages...

Packages (8) cmake-3.22.1-3 gcc-11.3.0-1 git-2.36.1-1 make-4.3-3
             mingw-w64-x86_64-libwinpthread-git-10.0.0.r32.g89bacd2be-1 ruby-2.7.2-1 unzip-6.0-2
             wget-1.21.3-1

Total Installed Size: 241.71 MiB
Net Upgrade Size:      0.00 MiB

:: Proceed with installation? [Y/n]
(8/8) checking keys in keyring [#####] 100%
(8/8) checking package integrity [#####] 100%
(8/8) loading package files [#####] 100%
(8/8) checking for file conflicts [#####] 100%
(8/8) checking available disk space [#####] 100%
:: Processing package changes...
(1/8) reinstalling git [#####] 100%
(2/8) reinstalling ruby [#####] 100%
(3/8) reinstalling make [#####] 100%
(4/8) reinstalling cmake [#####] 100%
(5/8) reinstalling gcc [#####] 100%
(6/8) reinstalling mingw-w64-x86_64-libwinpthread-git [#####] 100%
(7/8) reinstalling unzip [#####] 100%
(8/8) reinstalling wget [#####] 100%
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

CBKDELL@DESKTOP-FP78HAF MSYS ~
```

```
pacman -S git ruby make cmake gcc mingw-w64-x86_64-libwinpthread-git unzip wget
```

Set a Path for the Exported GCC to the Environment Pls make sure the directory is same name as the folder

```

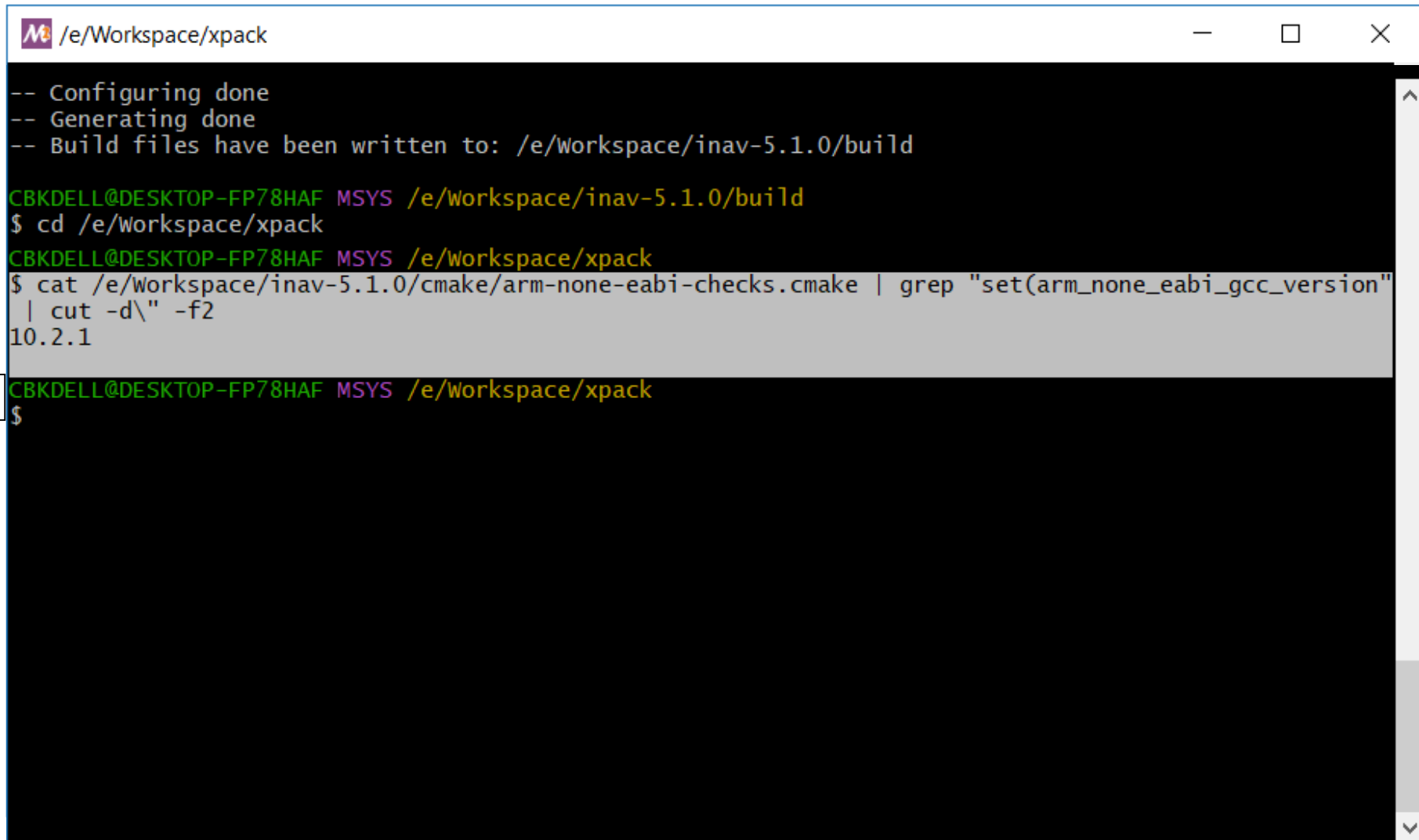
M ~
:: Proceed with installation? [Y/n]
(8/8) checking keys in keyring [#####] 100%
(8/8) checking package integrity [#####] 100%
(8/8) loading package files [#####] 100%
(8/8) checking for file conflicts [#####] 100%
(8/8) checking available disk space [#####] 100%
:: Processing package changes...
(1/8) reinstalling git [#####] 100%
(2/8) reinstalling ruby [#####] 100%
(3/8) reinstalling make [#####] 100%
(4/8) reinstalling cmake [#####] 100%
(5/8) reinstalling gcc [#####] 100%
(6/8) reinstalling mingw-w64-x86_64-libwinpthread-git [#####] 100%
(7/8) reinstalling unzip [#####] 100%
(8/8) reinstalling wget [#####] 100%
:: Running post-transaction hooks...
(1/1) Updating the info directory file...

CBKDELL@DESKTOP-FP78HAF MSYS ~
$ export PATH=/e/Workspace/xpack/xpack-arm-none-eabi-gcc-10.2.1-1.1/bin:$PATH
```

export PATH=/e/Workspace/xpack/xpack-arm-none-eabi-gcc-10.2.1-1.1/bin:\$PATH

get the toolkit version you need for your INAV version

for INAV version 5.0.0, 5.1.0 , 6.0.0 toolchain version needed is 10.2.1



```
/e/Workspace/xpack

-- Configuring done
-- Generating done
-- Build files have been written to: /e/Workspace/inav-5.1.0/build

CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-5.1.0/build
$ cd /e/Workspace/xpack
CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/xpack
$ cat /e/Workspace/inav-5.1.0/cmake/arm-none-eabi-checks.cmake | grep "set(arm_none_eabi_gcc_version"
| cut -d\" -f2
10.2.1
CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/xpack
$
```

Go to Xpack directory
by typing

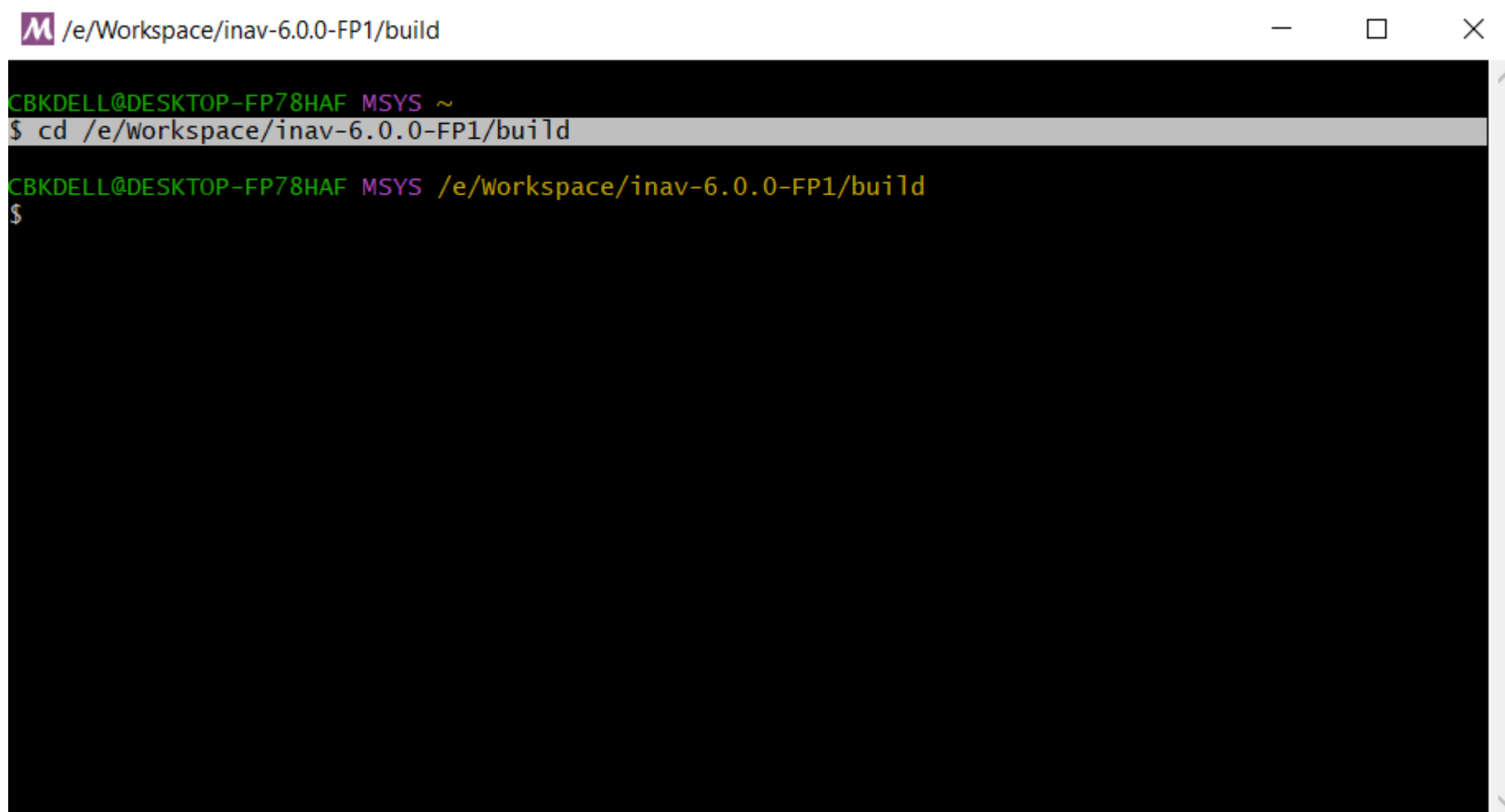
cd /e/Workspace/xpack

Source the version number that needs to be downloaded

cat /e/Workspace/inav-5.1.0/cmake/arm-none-eabi-checks.cmake | grep "set(arm_none_eabi_gcc_version" | cut -d\" -f2

cat /e/Workspace/inav-6.0.0-FP1/cmake/arm-none-eabi-checks.cmake | grep "set(arm_none_eabi_gcc_version" | cut -d\" -f2

Go to the build directory by entering this command



The screenshot shows a Windows command prompt window with the title bar "/e/Workspace/inav-6.0.0-FP1/build". The prompt is "CBKDELL@DESKTOP-FP78HAF MSYS ~". The command "\$ cd /e/Workspace/inav-6.0.0-FP1/build" has been entered and is highlighted. The prompt now shows the current directory as "/e/Workspace/inav-6.0.0-FP1/build".

```
CBKDELL@DESKTOP-FP78HAF MSYS ~  
$ cd /e/Workspace/inav-6.0.0-FP1/build  
CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-6.0.0-FP1/build  
$
```

cd /e/Workspace/inav-5.1.0/build

cd /e/Workspace/inav-6.0.0-FP1/build

This will extract the Cmake into the build folder creating the Environment

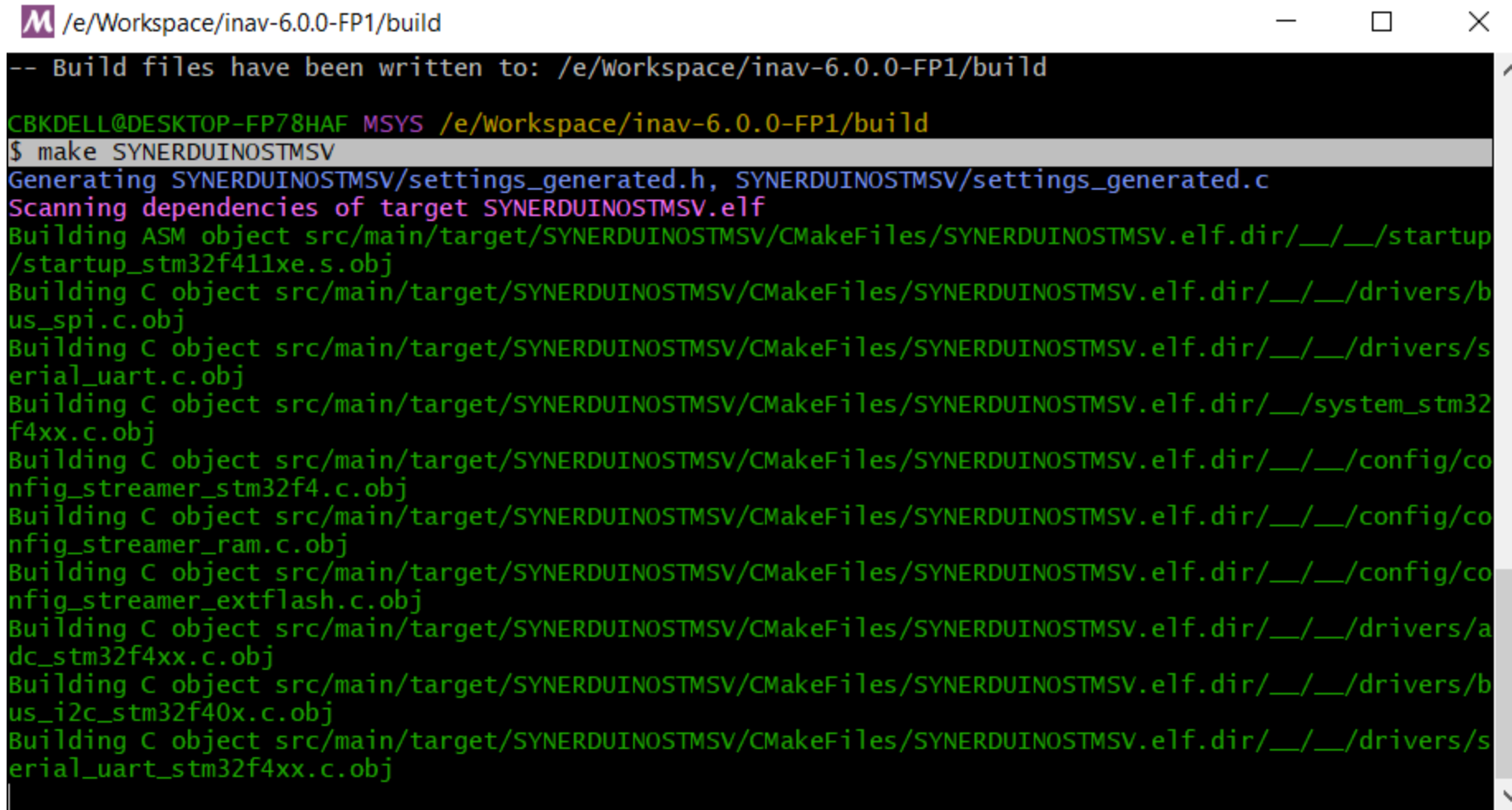
```
M /e/Workspace/inav-6.0.0-FP1/build
CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-6.0.0-FP1/build
$ export PATH=/e/Workspace/xdpack/xdpack-arm-none-eabi-gcc-10.2.1-1.1/bin:$PATH

CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-6.0.0-FP1/build
$ Cmake ..
-- found arm-none-eabi-gcc 10.2.1 at /e/Workspace/xdpack/xdpack-arm-none-eabi-gcc-10.2.1-1.1/bin/arm-n
one-eabi-gcc.exe
-- The C compiler identification is GNU 10.2.1
-- The CXX compiler identification is GNU 10.2.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /e/Workspace/xdpack/xdpack-arm-none-eabi-gcc-10.2.1-1.1/bin/arm-none-
eabi-gcc.exe - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /e/Workspace/xdpack/xdpack-arm-none-eabi-gcc-10.2.1-1.1/bin/arm-non
e-eabi-g++.exe - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- The ASM compiler identification is GNU
-- Found assembler: /e/Workspace/xdpack/xdpack-arm-none-eabi-gcc-10.2.1-1.1/bin/arm-none-eabi-gcc.exe
-- toolchain: arm-none-eabi, WARNINGS_AS_ERRORS: OFF
-- Could not find openocd, debugging won't be available
-- DEBUG_HARDFAULTS: OFF, SEMIHOSTING: OFF
```

Cmake ..

You may need to run `rm -rf *` in build directory if you had any failed previous runs now run cmake

This will build the Hexfile Targeted the Synerduino board and its settings to the Build Folder the Hexfile is use as Firmware when loading the configurator



```
M /e/Workspace/inav-6.0.0-FP1/build
-- Build files have been written to: /e/Workspace/inav-6.0.0-FP1/build
CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-6.0.0-FP1/build
$ make SYNERDUINOSTMSV
Generating SYNERDUINOSTMSV/settings_generated.h, SYNERDUINOSTMSV/settings_generated.c
Scanning dependencies of target SYNERDUINOSTMSV.elf
Building ASM object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/startup/startup_stm32f411xe.s.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/bus_spi.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/serial_uart.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/system_stm32f4xx.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/config/config_streamer_stm32f4.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/config/config_streamer_ram.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/config/config_streamer_extflash.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/adc_stm32f4xx.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/bus_i2c_stm32f40x.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/serial_uart_stm32f4xx.c.obj
```

make SYNERDUINOSTM

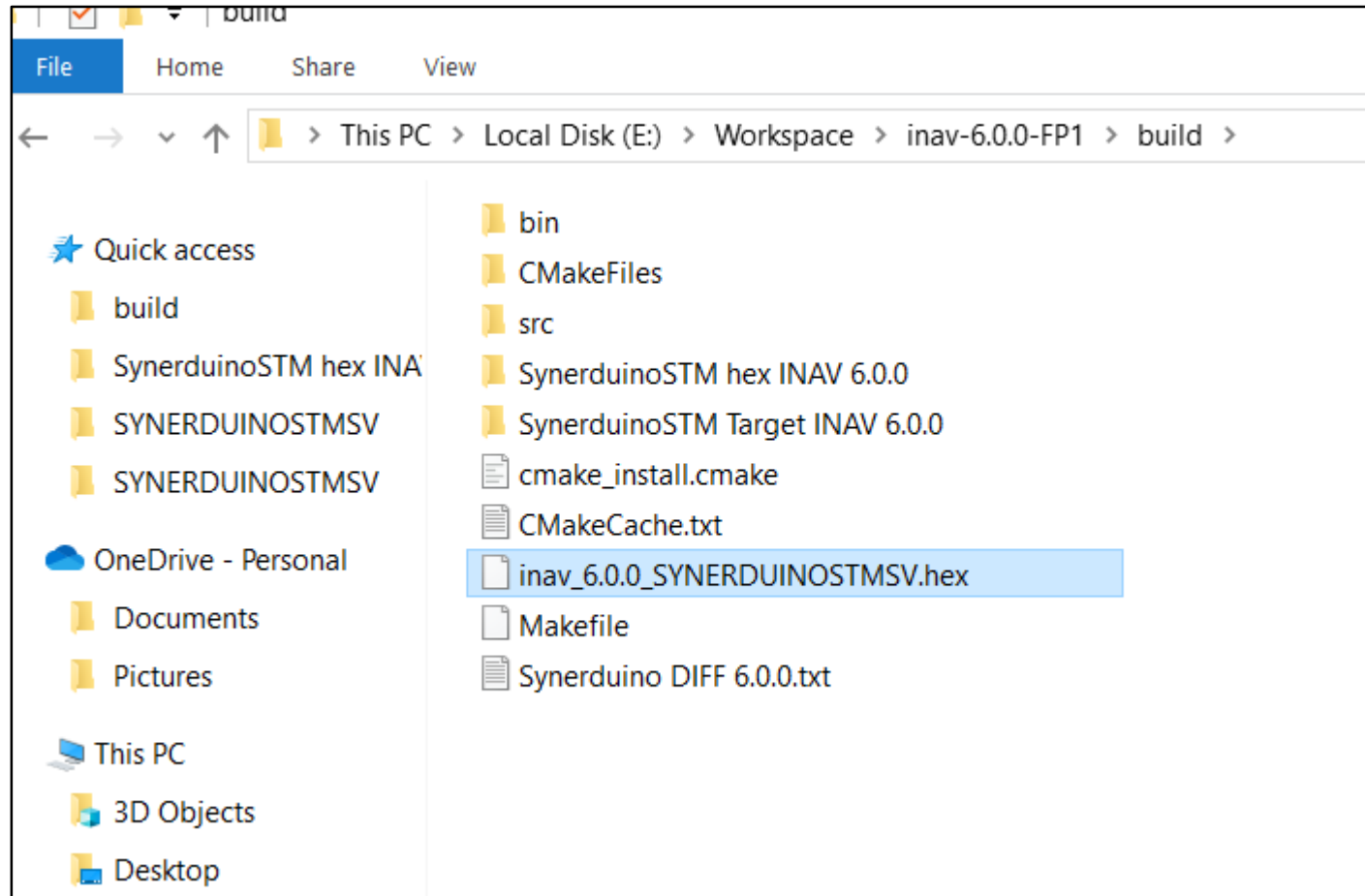
make SYNERDUINOSTMSV

Once completed the Hexfile can be found on the Build folder

```
M /e/Workspace/inav-6.0.0-FP1/build
/ibus_shared.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/telemetry
/ibus.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/telemetry
/ltm.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/telemetry
/mavlink.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/telemetry
/msp_shared.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/telemetry
/smartport.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/telemetry
/sim.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/telemetry
/telemetry.c.obj
Linking C executable ../../../../bin/SYNERDUINOSTMSV.elf
Memory region      Used Size  Region Size  %age Used
      FLASH:         1108 B       16 KB       6.76%
    FLASH_CONFIG:         0 GB       16 KB       0.00%
      FLASH1:    458570 B       480 KB      93.30%
        RAM:     92056 B       128 KB      70.23%
    MEMORY_B1:         0 GB        0 GB
Built target SYNERDUINOSTMSV.elf
Built target SYNERDUINOSTMSV

CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-6.0.0-FP1/build
$
```

And... its here . The INAV_5.1.0 SYNERDUINOSTM.hex or INAV_6.0.0_SYNERDUINOSTM.hex
INAV_5.1.0 SYNERDUINOSTMSV.hex INAV_6.0.0_SYNERDUINOSTMSV.hex



Open this file in the Configurator as
this is the firmware
Find it in the Build folder of the INAV
directory

```
cd /e/Workspace/inav-5.1.0/build
```

```
cd /e/Workspace/inav-6.0.0-FP1/build
```