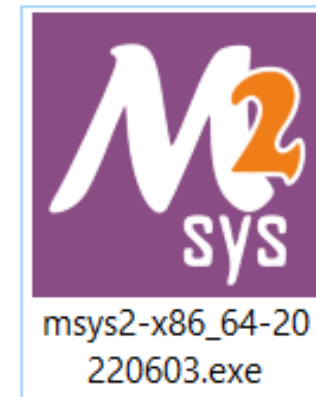




Synerduino STM Fork of INAV and the  
Synerduino Target

Synerduino STM Target  
Compilation into Hexfile

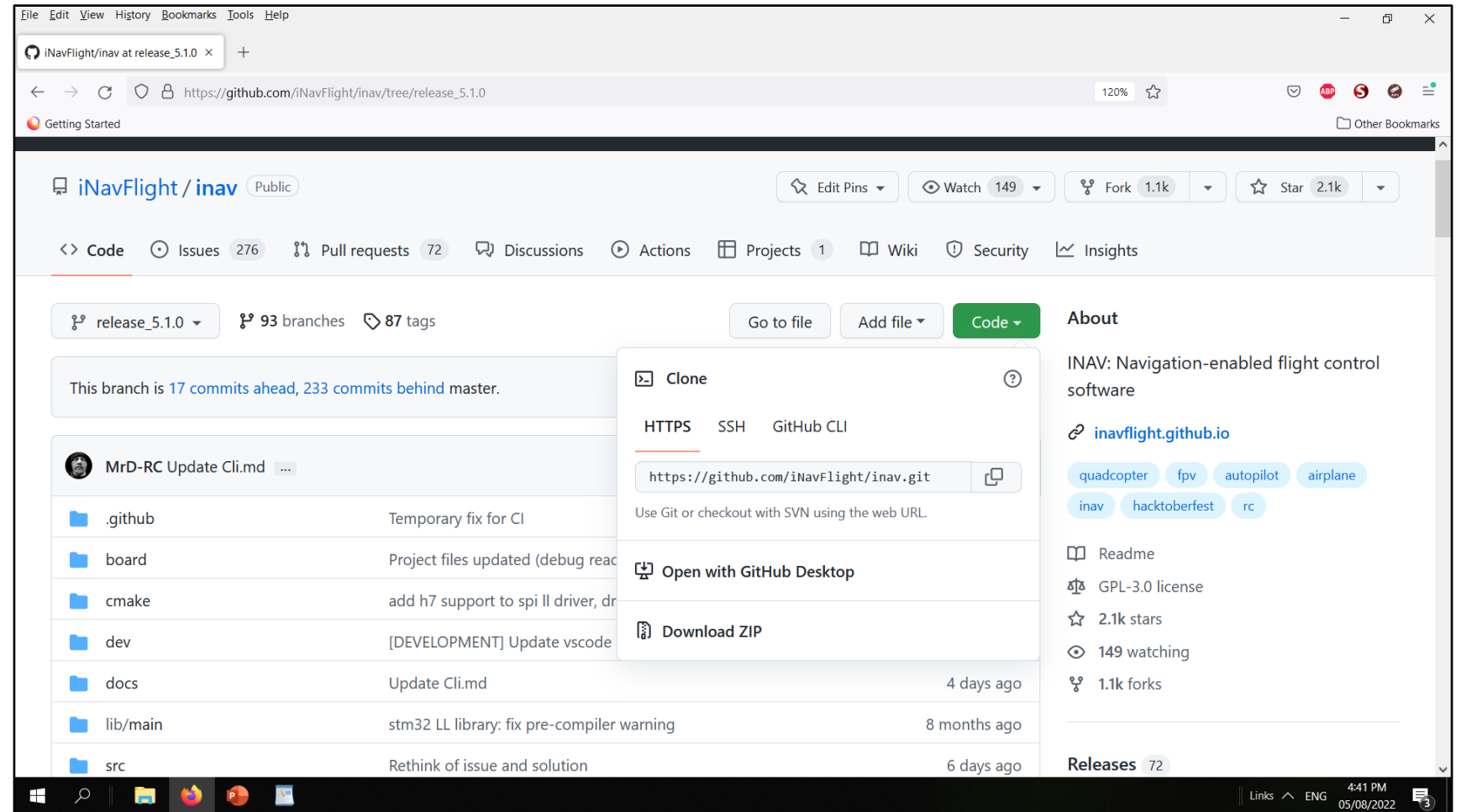


# Download the INAV Release

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/10.0.0-rc2>

INAV-5.1.0 / INAV-6.0.0

Download the Xpack  
Release 10.2.1-1.1-  
Win32-x64

INAV-7.0.0

Download the Xpack  
Release 10.3.1-2.2-  
Win32-x64

INAV-8.0.0

Download the Xpack  
Release 12.2.1-1.1-  
Win32-x64

INAV-9.0.0-10.0.0

Download the Xpack  
Release 13.2.1-1.1-  
Win32-x64

File Edit View History Bookmarks Tools Help

inav/Building in Windows 10 or x Releases · xpack-dev-tools/arm x MSYS2 Page not found · GitHub

← → ↻ 🔒 https://github.com/xpack-dev-tools/arm-none-eabi-gcc-xpack/releases/ 120% ☆ Other Bookmarks

Getting Started

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

|                                                                                                                                                                |           |              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------|
|  <a href="#">xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-arm64.tar.gz</a>     | 221 MB    | May 30, 2022 |
|  <a href="#">xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-arm64.tar.gz.sha</a> | 121 Bytes | May 30, 2022 |
|  <a href="#">xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-x64.tar.gz</a>       | 222 MB    | May 30, 2022 |
|  <a href="#">xpack-arm-none-eabi-gcc-11.2.1-1.2-darwin-x64.tar.gz.sha</a>   | 119 Bytes | May 30, 2022 |
|  <a href="#">xpack-arm-none-eabi-gcc-11.2.1-1.2-linux-arm.tar.gz</a>        | 217 MB    | May 30, 2022 |
|  <a href="#">xpack-arm-none-eabi-gcc-11.2.1-1.2-linux-arm.tar.gz.sha</a>    | 118 Bytes | May 30, 2022 |
|  <a href="#">xpack-arm-none-eabi-gcc-11.2.1-1.2-linux-arm64.tar.gz</a>      | 223 MB    | May 30, 2022 |

Windows taskbar: 5:05 PM 05/08/2022

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

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

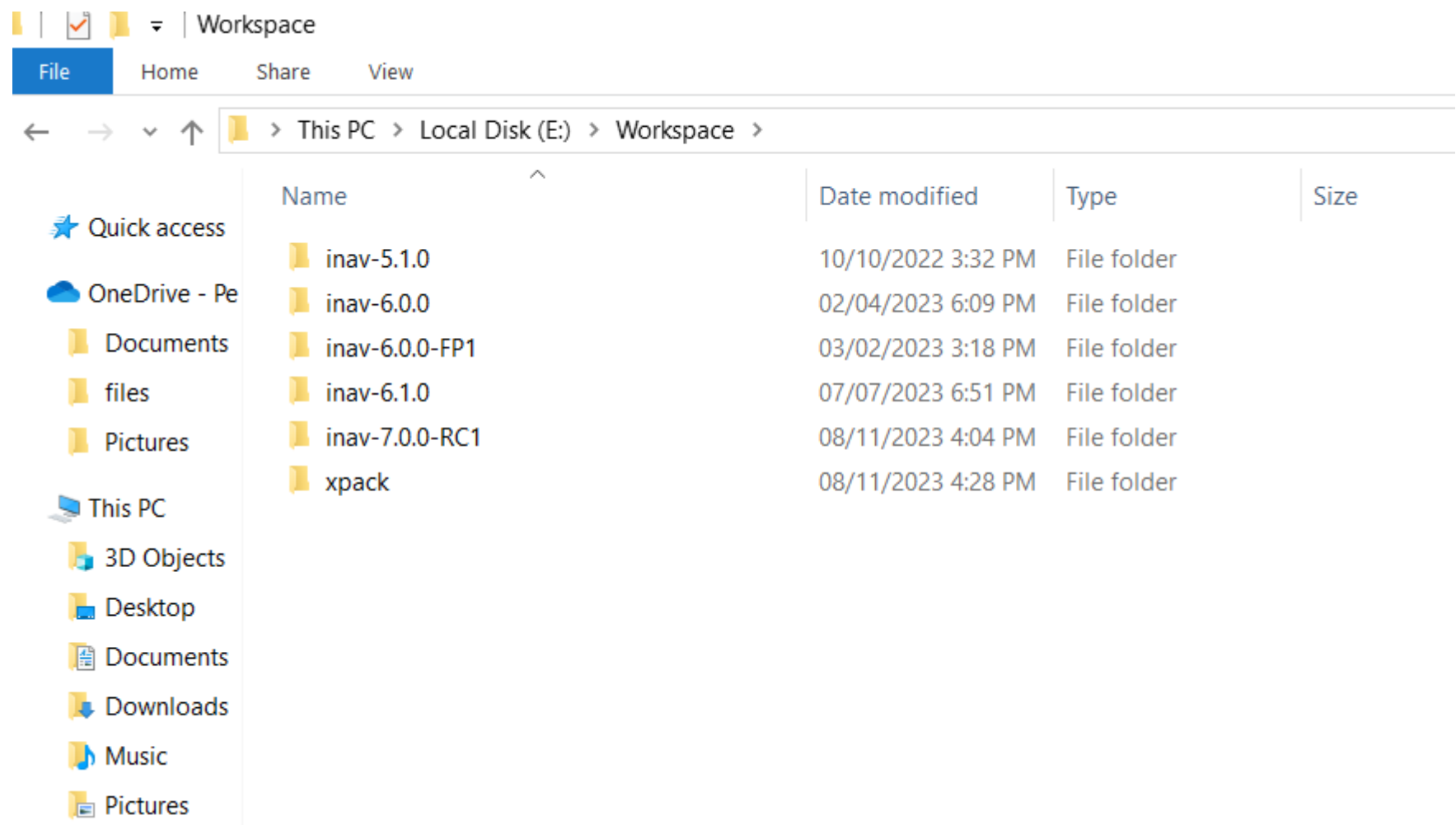
<https://github.com/xpack-dev-tools/arm-none-eabi-gcc-xpack/releases/download/v9.2.1-1.1/xpack-arm-none-eabi-gcc-9.2.1-1.1-win32-x64.zip>

<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>

<https://github.com/xpack-dev-tools/arm-none-eabi-gcc-xpack/releases/download/v10.3.1-2.2/xpack-arm-none-eabi-gcc-10.3.1-2.2-win32-x64.zip>

<https://github.com/xpack-dev-tools/arm-none-eabi-gcc-xpack/releases/download/v13.2.1-1.1/xpack-arm-none-eabi-gcc-13.2.1-1.1-win32-x64.zip>

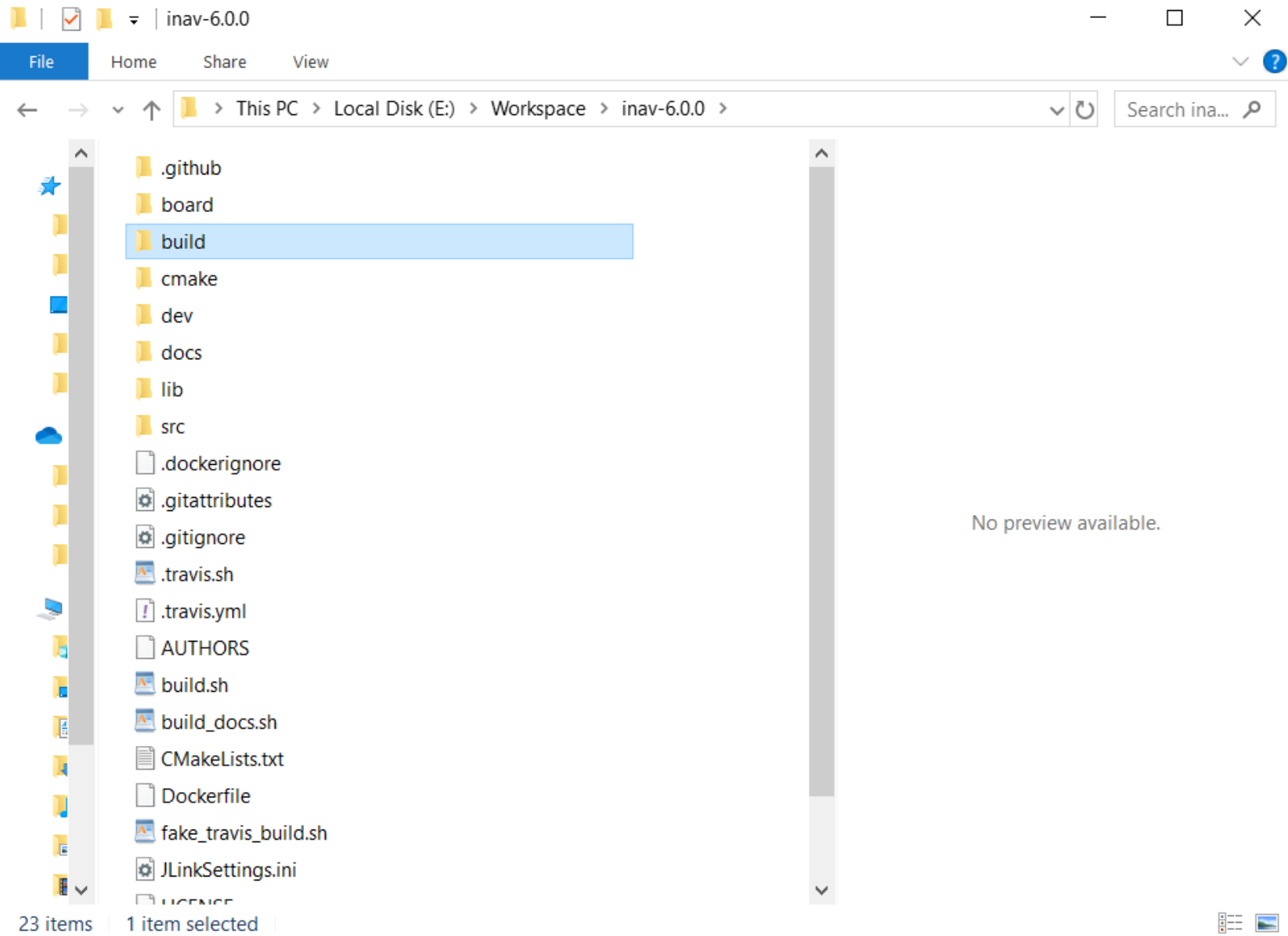
After installation Go Create a Workspace folder in your local Drive in my case Drive E , and extract and place the following directories [INAV 5.0.0 – INAV 10.0.0 Folder](#) , [Xpack folder](#)



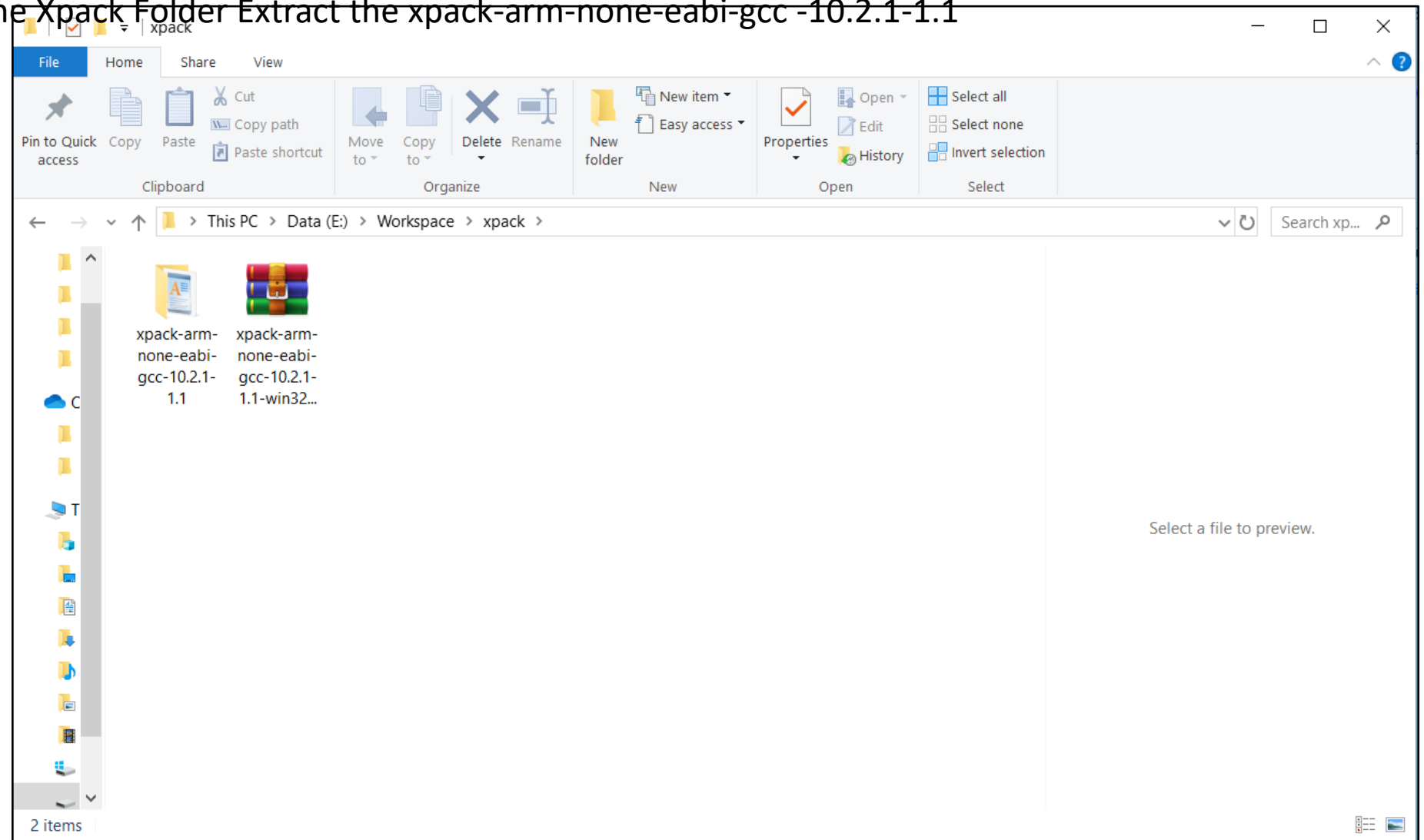
Note: I place my Workspace Folder on Drive E: as I was using SSD on my Drive C so not to fill it up

Create new folder inside  
INAV 5.0.0 – INAV 10.0.

Directory and Name it **build**



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



Create an xpack folder  
in Workspace folder  
and place it in there

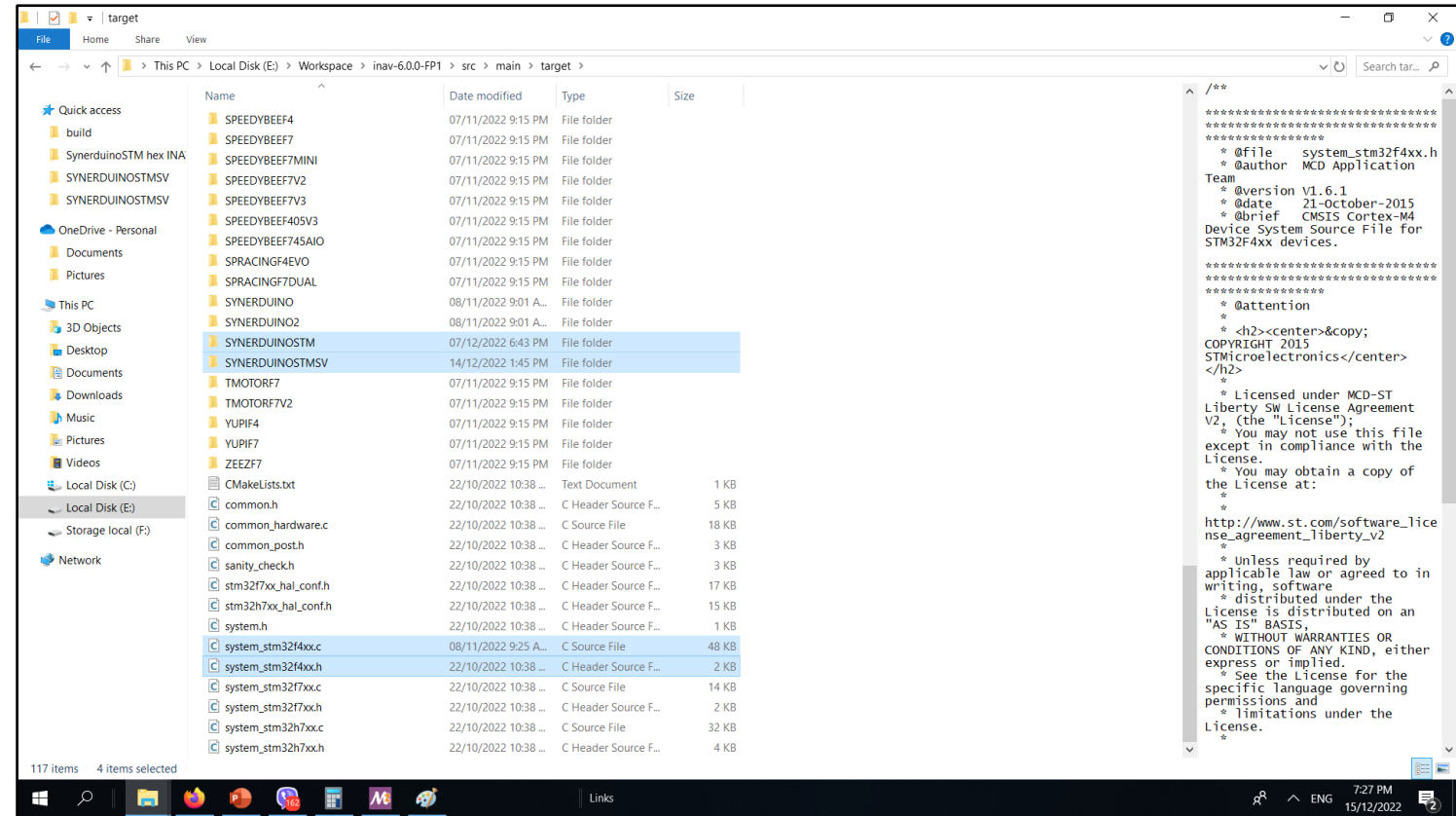
INAV 5.1.0 INAV 6.0.0  
INAV 7.0.0  
INAV 8.0.0  
INAV 8.0.0-10.0.0

xpack-arm-none-eabi-gcc-10.2.1-1.1 is extracted to the Xpack folder  
xpack-arm-none-eabi-gcc-10.3.1-2.2 is extracted to the Xpack folder  
xpack-arm-none-eabi-gcc-12.2.1-2.2 is extracted to the Xpack folder  
xpack-arm-none-eabi-gcc-13.2.1-1.1 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)  
(E:/Workspace/inav-6.0.0/scr/main/target)  
(E:/Workspace/inav-7.0.0/scr/main/target)  
(E:/Workspace/inav-8.0.0/scr/main/target)  
(E:/Workspace/inav-9.0.0/scr/main/target)

Readme file is also available in the download  
System\_stm32f4xx.c – is modify to suit the  
Synerduino STM

INAV 7.0.0 when F411 BP became official)  
INAV 8.0.0 F411 discontinued due to size  
limitation



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/scr/main/drivers/accgyro) ,

(E/Workspace/inav-6.0.0/scr/main/drivers/accgyro) ,

(E/Workspace/inav-7.0.0/scr/main/drivers/accgyro)

(E/Workspace/inav-7.1.0/scr/main/drivers/accgyro)

(E/Workspace/inav-8.0.0/scr/main/drivers/accgyro)

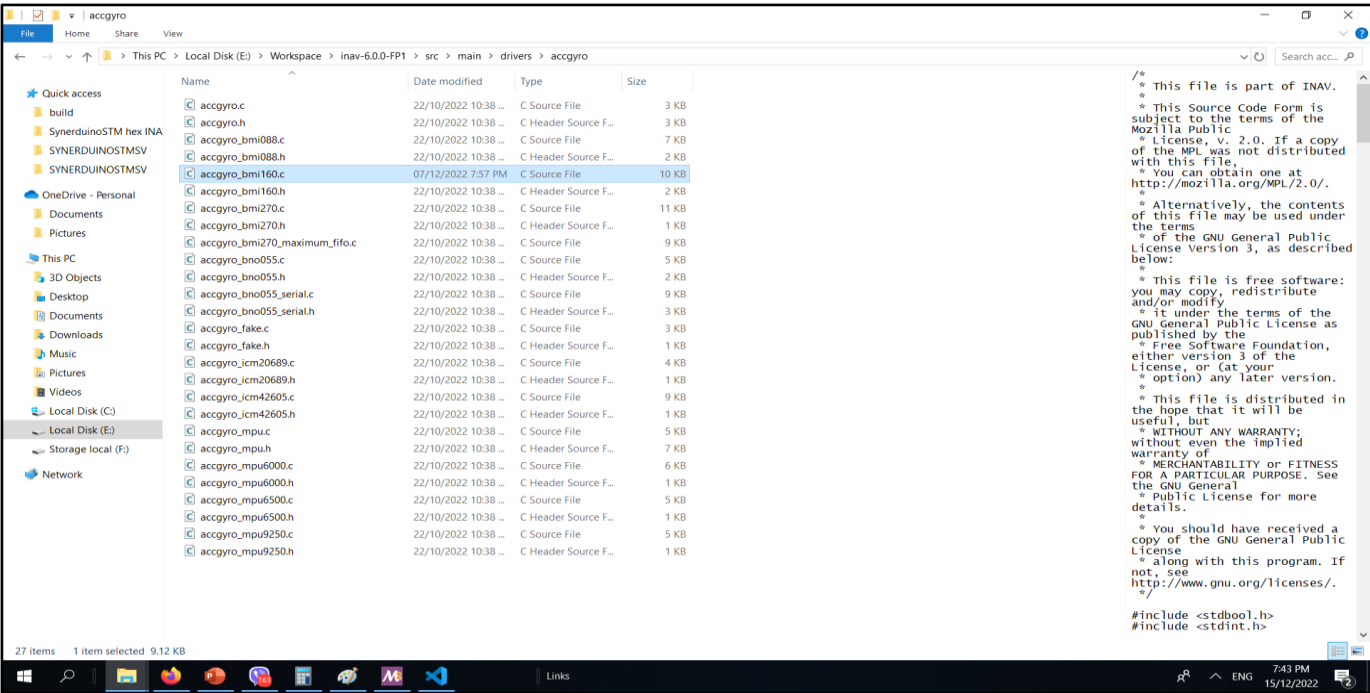
(E/Workspace/inav-9.0.0/scr/main/drivers/accgyro)

(E/Workspace/inav-10.0.0/scr/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.

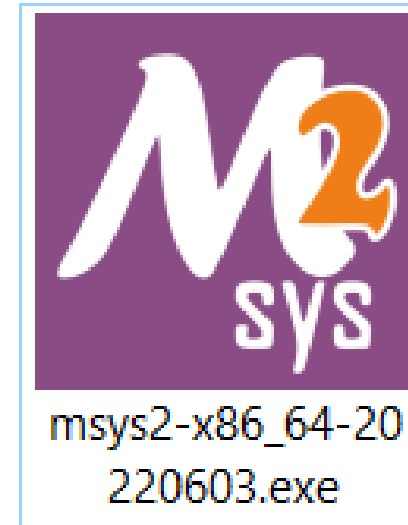
Download Arm Embedded GCC toolkit from The xPack Project

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

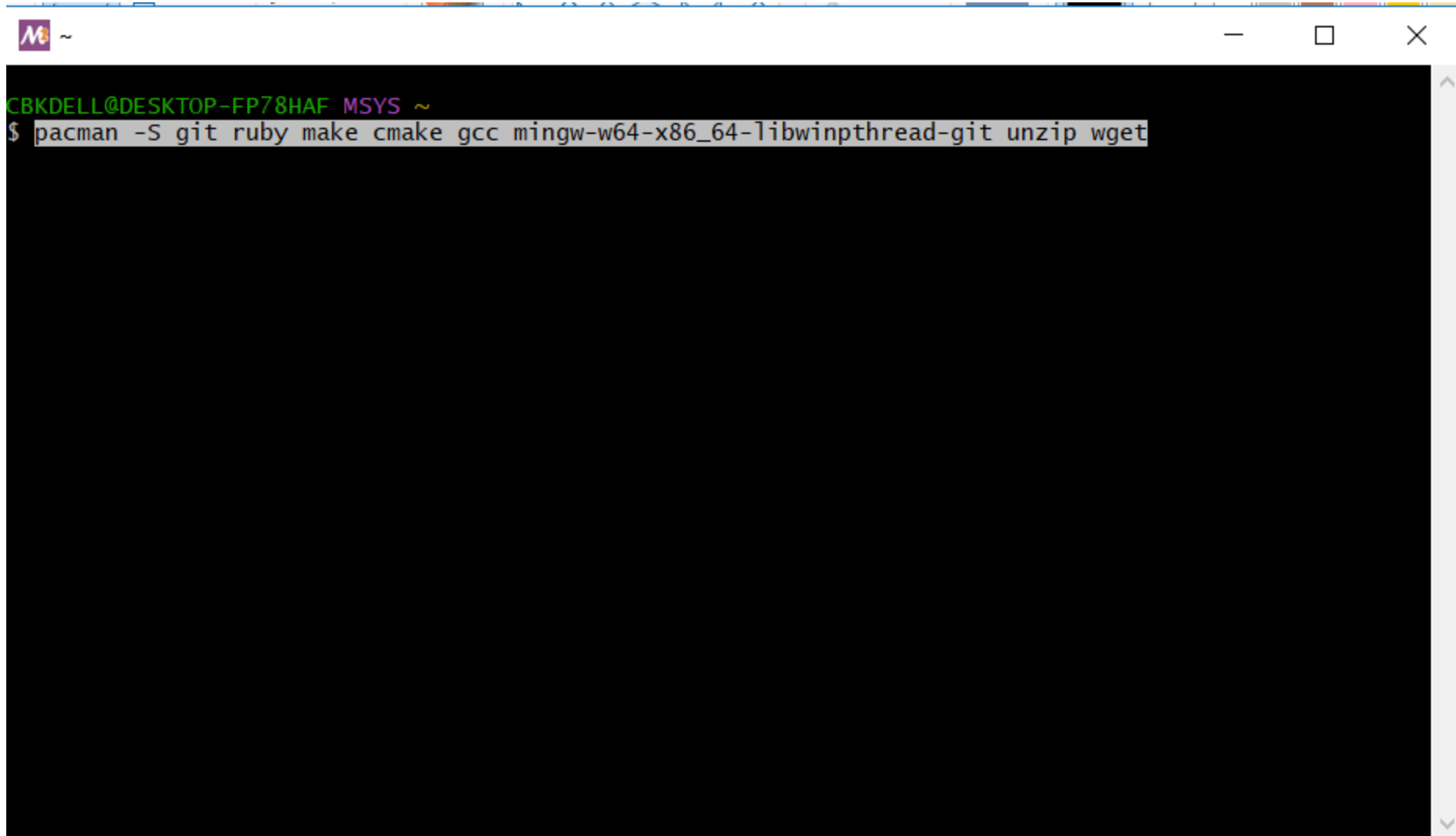
Download MSYS2 for your architecture (most likely 64-bit)

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

Install and Run msys2-x86



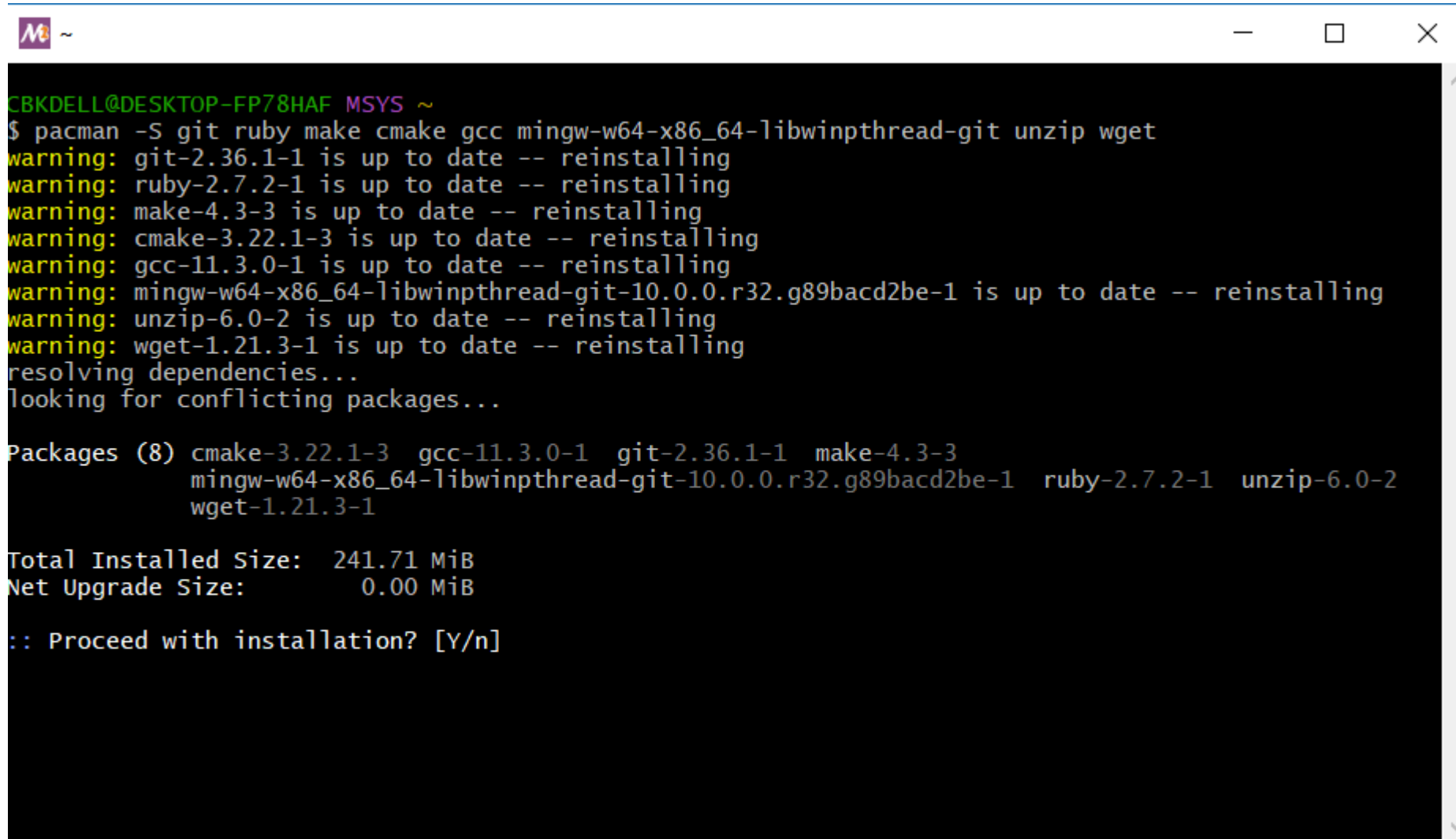
Copy and paste this script to install Ruby ,Cmake , Make and Gcc

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 background. 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 black background and white text. The window title bar shows a purple icon and a tilde symbol. The terminal output shows a pacman command being executed, followed by several warning messages indicating that the packages are up to date and will be reinstalled. The output also shows the packages being installed, the total installed size, and the net upgrade size. The prompt asks to proceed with installation, with [Y/n] as the default response.

```
CBKDELL@DESKTOP-FP78HAF MSYS ~  
$ pacman -S git ruby make cmake gcc mingw-w64-x86_64-libwinpthread-git unzip wget  
warning: git-2.36.1-1 is up to date -- reinstalling  
warning: ruby-2.7.2-1 is up to date -- reinstalling  
warning: make-4.3-3 is up to date -- reinstalling  
warning: cmake-3.22.1-3 is up to date -- reinstalling  
warning: gcc-11.3.0-1 is up to date -- reinstalling  
warning: mingw-w64-x86_64-libwinpthread-git-10.0.0.r32.g89bacd2be-1 is up to date -- reinstalling  
warning: unzip-6.0-2 is up to date -- reinstalling  
warning: wget-1.21.3-1 is up to date -- reinstalling  
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]
```

```
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
```

```
Do you want to delete it? [Y/n] y
error: libarchive: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/libarchive-3.6.1-2-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: libhash: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/libhash-1.4.2-1-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: libuv: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/libuv-1.42.0-1-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: pkgconf: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/pkgconf-1.8.0-1-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: cmake: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/cmake-3.23.2-1-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: binutils: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/binutils-2.37-5-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: isl: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/isl-0.25-2-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: mpc: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/mpc-1.2.1-1-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: msys2-runtime-devel: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/msys2-runtime-devel-3.3.5-3-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: msys2-w32api-headers: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/msys2-w32api-headers-10.0.0.r16.g49a56d453-1-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: msys2-w32api-runtime: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/msys2-w32api-runtime-10.0.0.r16.g49a56d453-1-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: windows-default-manifest: signature from "Alexey Pavlov (Alexpux) <alexpux@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/windows-default-manifest-6.4-1-x86_64.pkg.tar.xz is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: gcc: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/gcc-11.3.0-2-x86_64.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: mingw-w64-x86_64-libwinpthread-git: signature from "Christoph Reiter (MSYS2 development key) <reiter.christoph@gmail.com>" is invalid
:: File /var/cache/pacman/pkg/mingw-w64-x86_64-libwinpthread-git-10.0.0.r72.g1dd2a4993-1-any.pkg.tar.zst is corrupted (invalid or corrupted package (PGP signature)).
Do you want to delete it? [Y/n] y
error: failed to commit transaction (invalid or corrupted package)
Errors occurred, no packages were upgraded.
y
yy
CBKDELL@DESKTOP-FP78HAF MSYS ~
$ y
-bash: y: command not found
CBKDELL@DESKTOP-FP78HAF MSYS ~
$ yy
-bash: yy: command not found
CBKDELL@DESKTOP-FP78HAF MSYS ~
$ |
```

Note: If some fails to download, use the following command to install the rest without reinstalling everything:

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

## To get the toolkit version you need for your INAV version

```
# for INAV version 5.0.0, 6.1.0 toolchain version needed is 10.2.1
# for INAV version 7.0.0 , 7.1.2 toolchain version needed is 10.3.1
# for INAV version 8.0.0 , 10.0.0 toolchain version needed is 13.2.1
```

```
-- 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
$
```

Source the version number that  
needs to be downloaded

Go to Xpack directory  
by typing

```
cd /d/Workspace/xpack
```

```
cat /d/Workspace/inav-10.0.0-rc2/cmake/arm-none-eabi-checks.cmake | grep "set(arm_none_eabi_gcc_version" | cut -d\" -f2
```

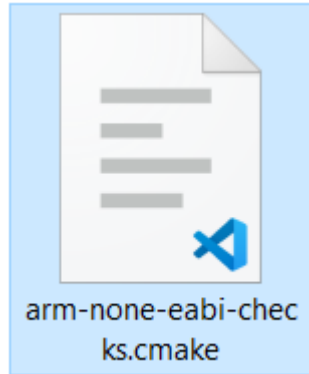
```
cat /d/Workspace/inav-9.1.0/cmake/arm-none-eabi-checks.cmake | grep "set(arm_none_eabi_gcc_version" | cut -d\" -f2
```

```
cat /d/Workspace/inav-8.0.0/cmake/arm-none-eabi-checks.cmake | grep "set(arm_none_eabi_gcc_version" | cut -d\" -f2
```

```
cat /d/Workspace/inav-7.1.2/cmake/arm-none-eabi-checks.cmake | grep "set(arm_none_eabi_gcc_version" | cut -d\" -f2
```

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

To get the toolkit version you need for your INAV version



Cmake/arm-none-eabi-checks.cmake

If older versions of arm\_none\_eabi\_gcc\_version doesn't build correctly due to depreciated dependency you can change the Version number corresponding to the Latest Xpack Versions

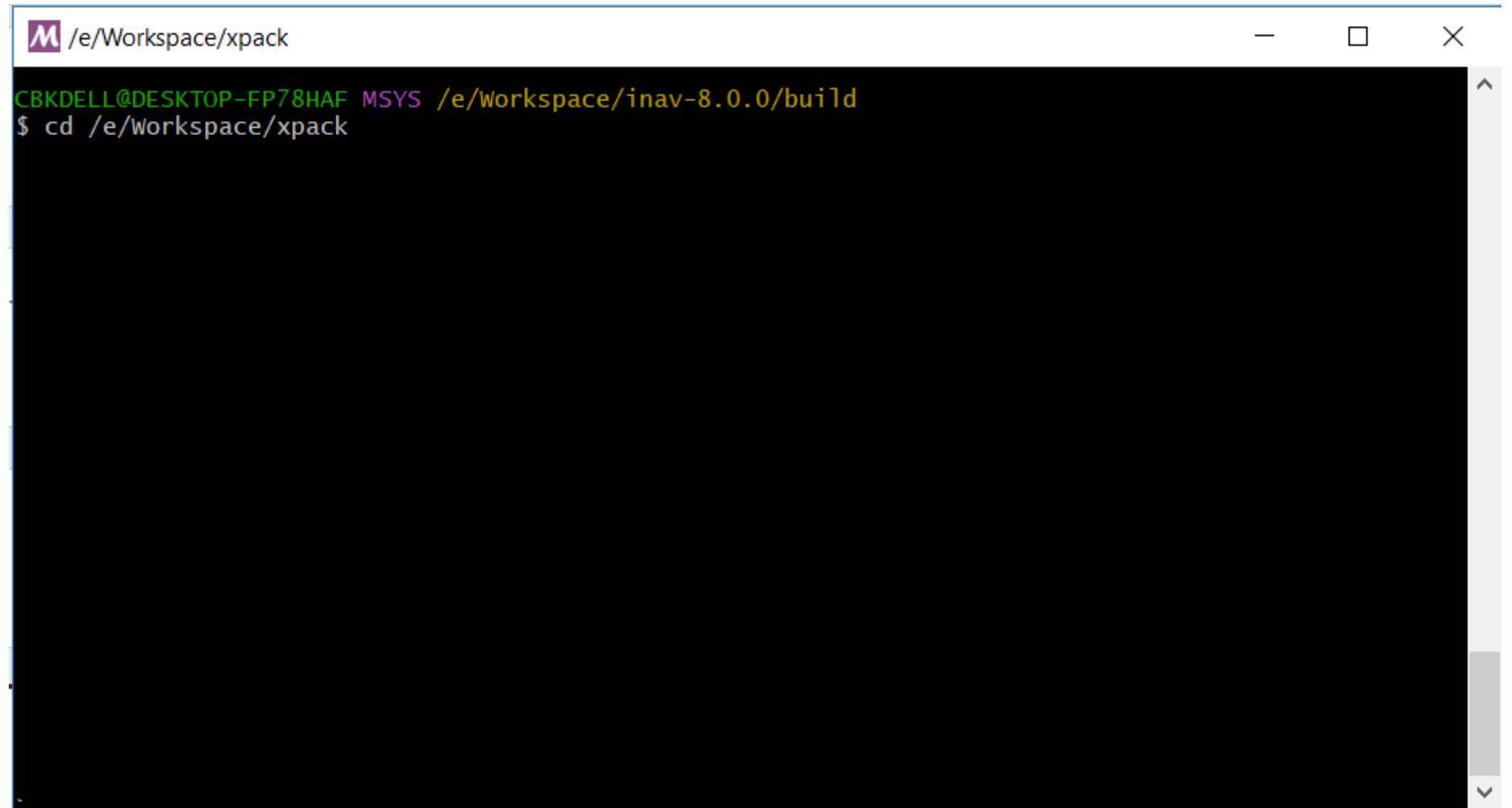
From the link provided in the file

arm-none-eabi-checks.cmake X

D: > Workspace > inav-7.1.2-1 > cmake > arm-none-eabi-checks.cmake

```
1  include(gcc)
2  set(arm_none_eabi_triplet "arm-none-eabi")
3
4  # Keep version in sync with the distribution files below
5  set(arm_none_eabi_gcc_version "13.2.1")
6  set(arm_none_eabi_base_url "https://developer.arm.com/-/media/Files/downloads/gnu-rm/10.3-2021.10/gcc-arm-none-eabi-10.3-2021.10")
7  # suffix and checksum
8  set(arm_none_eabi_win32 "win32.zip" 2bc8f0c4c4659f8259c8176223eeafc1)
9  set(arm_none_eabi_linux_amd64 "x86_64-linux.tar.bz2" 2383e4eb4ea23f248d33adc70dc3227e)
10 set(arm_none_eabi_linux_aarch64 "aarch64-linux.tar.bz2" 3fe3d8bb693bd0a6e4615b6569443d0d)
11 set(arm_none_eabi_gcc_macos "mac.tar.bz2" 7f2a7b7b23797302a9d6182c6e482449)
12
```

Go to Xpack directory  
by typing  
`cd /d/Workspace/xpack`



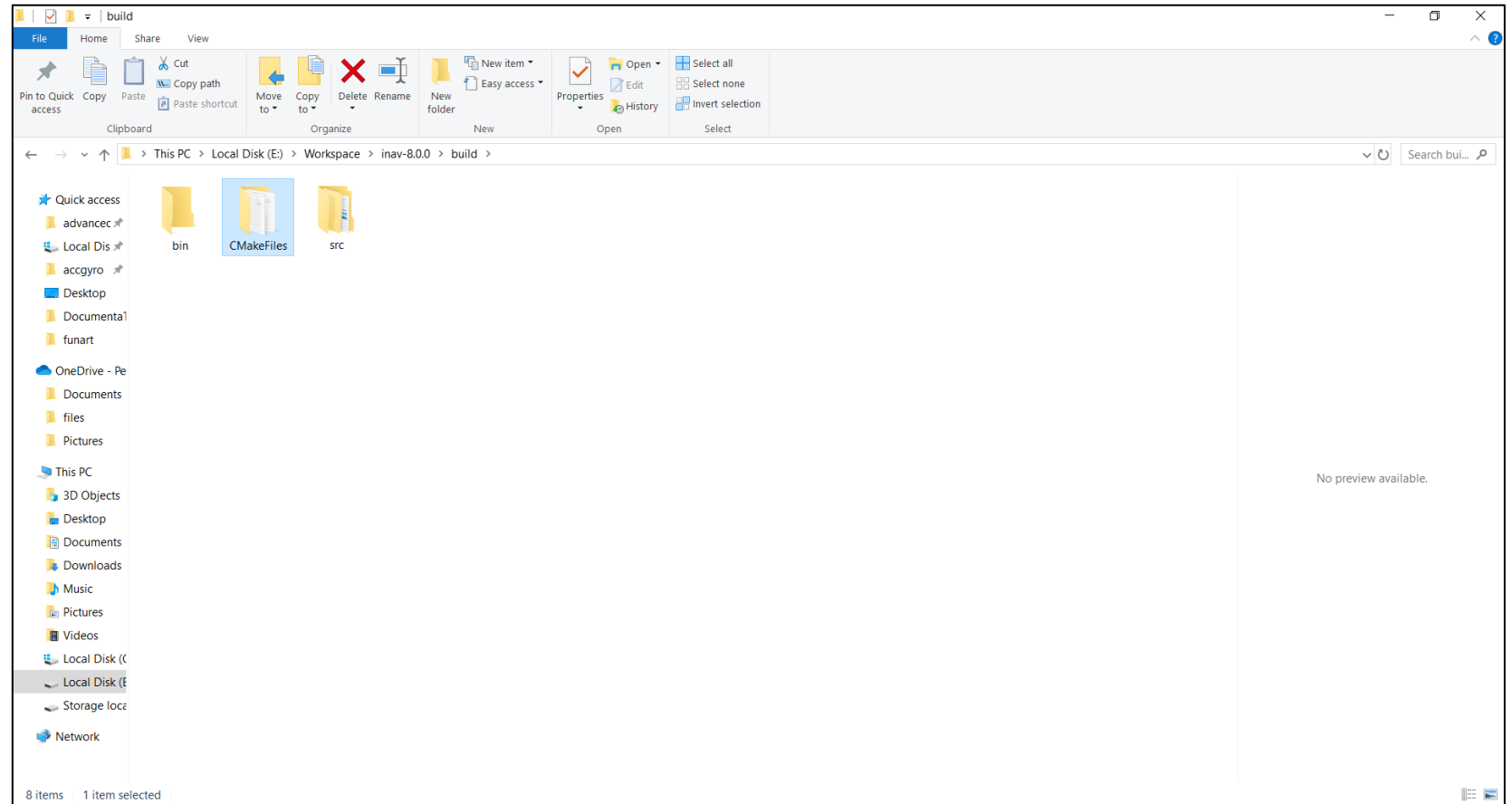
```
M /e/Workspace/xpack
CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-8.0.0/build
$ cd /e/Workspace/xpack
```

The image shows a Windows command prompt window with a title bar that includes a purple 'M' icon and the text '/e/Workspace/xpack'. The window has standard minimize, maximize, and close buttons. The command prompt shows the user 'CBKDELL@DESKTOP-FP78HAF' in a purple prompt, the current directory '/e/Workspace/inav-8.0.0/build' in yellow, and the command '\$ cd /e/Workspace/xpack' entered. The background of the command prompt is black, and the text is white. A vertical scrollbar is visible on the right side of the command prompt window.

## Build Folder

E:/Workspace/INAV-8.0.0/build

If you have an old  
CMakeFiles folder of a  
different versions  
Delete it before Exporting a  
new path



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

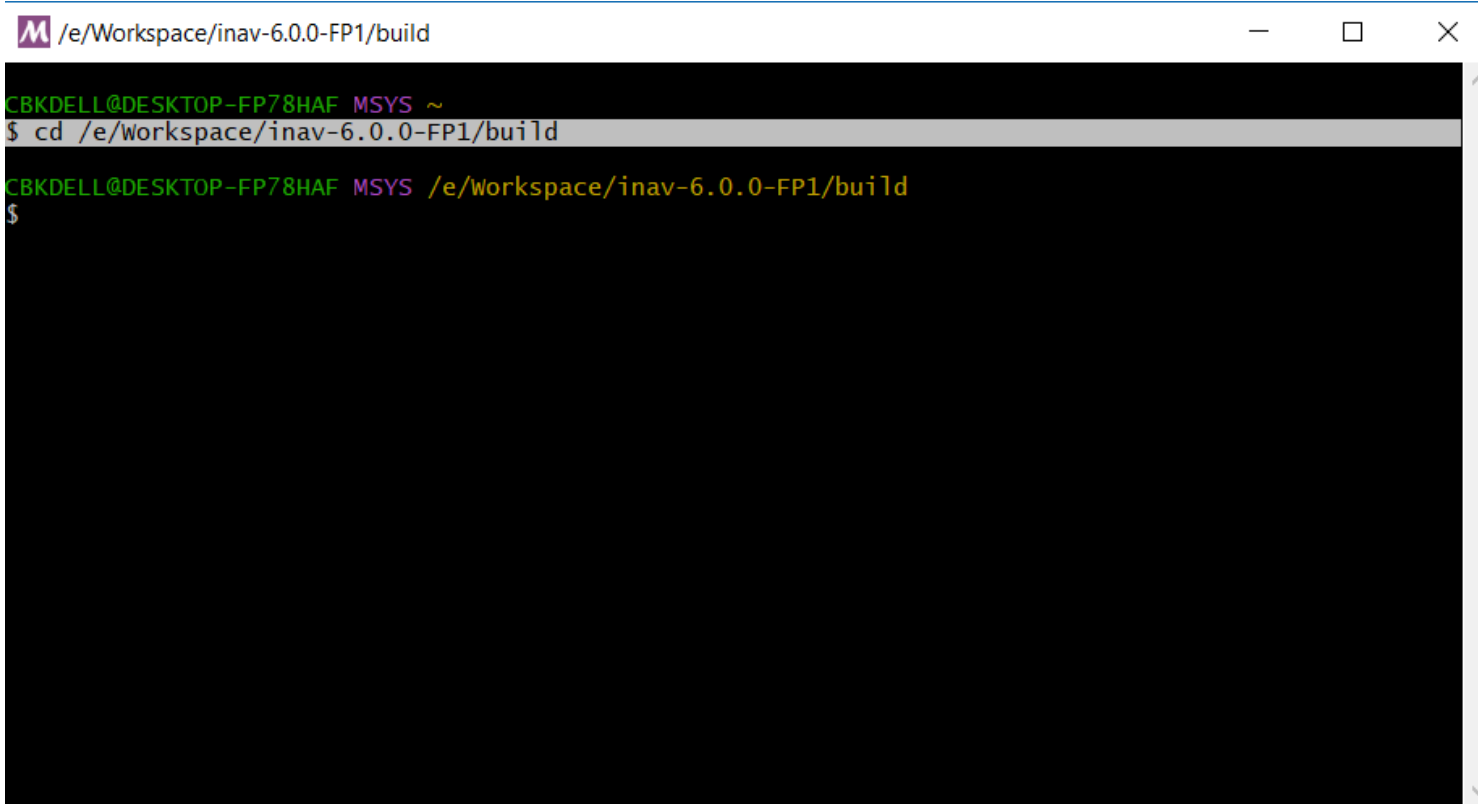
```

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
```

|                     |                                                                              |
|---------------------|------------------------------------------------------------------------------|
| INAV 5.1.0 – 6.1.0  | export PATH=/d/Workspace/xpack/xpack-arm-none-eabi-gcc-10.2.1-1.1/bin:\$PATH |
| INAV 7.0.0 – 7.1.2  | export PATH=/d/Workspace/xpack/xpack-arm-none-eabi-gcc-10.3.1-2.2/bin:\$PATH |
| INAV 8.0.0 - 10.0.0 | export PATH=/d/Workspace/xpack/xpack-arm-none-eabi-gcc-13.2.1-1.1/bin:\$PATH |

sudo apt remove gcc-arm-none-eabi

Go to the build directory by entering this command



A terminal window with a title bar showing the path `/e/Workspace/inav-6.0.0-FP1/build`. The terminal content shows the user `CBKDELL@DESKTOP-FP78HAF` in the `MSYS` environment. The command `cd /e/Workspace/inav-6.0.0-FP1/build` is entered and executed, resulting in the prompt `CBKDELL@DESKTOP-FP78HAF MSYS /e/Workspace/inav-6.0.0-FP1/build`.

```
/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 /d/Workspace/inav-5.1.0/build`

`cd /d/Workspace/inav-6.1.0/build`

`cd /d/Workspace/inav-7.0.0/build`

`cd /d/Workspace/inav-7.1.0/build`

`cd /d/Workspace/inav-7.1.1/build`

`cd /d/Workspace/inav-7.1.2/build`

`cd /d/Workspace/inav-7.1.2-1/build`

`cd /d/Workspace/inav-8.0.0/build`

`cd /d/Workspace/inav-8.0.1/build`

`cd /d/Workspace/inav-8.0.1-1/build`

`cd /d/Workspace/inav-9.1.1/build`

`cd /d/Workspace/inav-10.0.0-rc2/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/b
us_spi.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/s
erial_uart.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/system_stm32
f4xx.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/config/co
nfig_streamer_stm32f4.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/config/co
nfig_streamer_ram.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/config/co
nfig_streamer_extflash.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/a
dc_stm32f4xx.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/b
us_i2c_stm32f40x.c.obj
Building C object src/main/target/SYNERDUINOSTMSV/CMakeFiles/SYNERDUINOSTMSV.elf.dir/__/__/drivers/s
erial_uart_stm32f4xx.c.obj

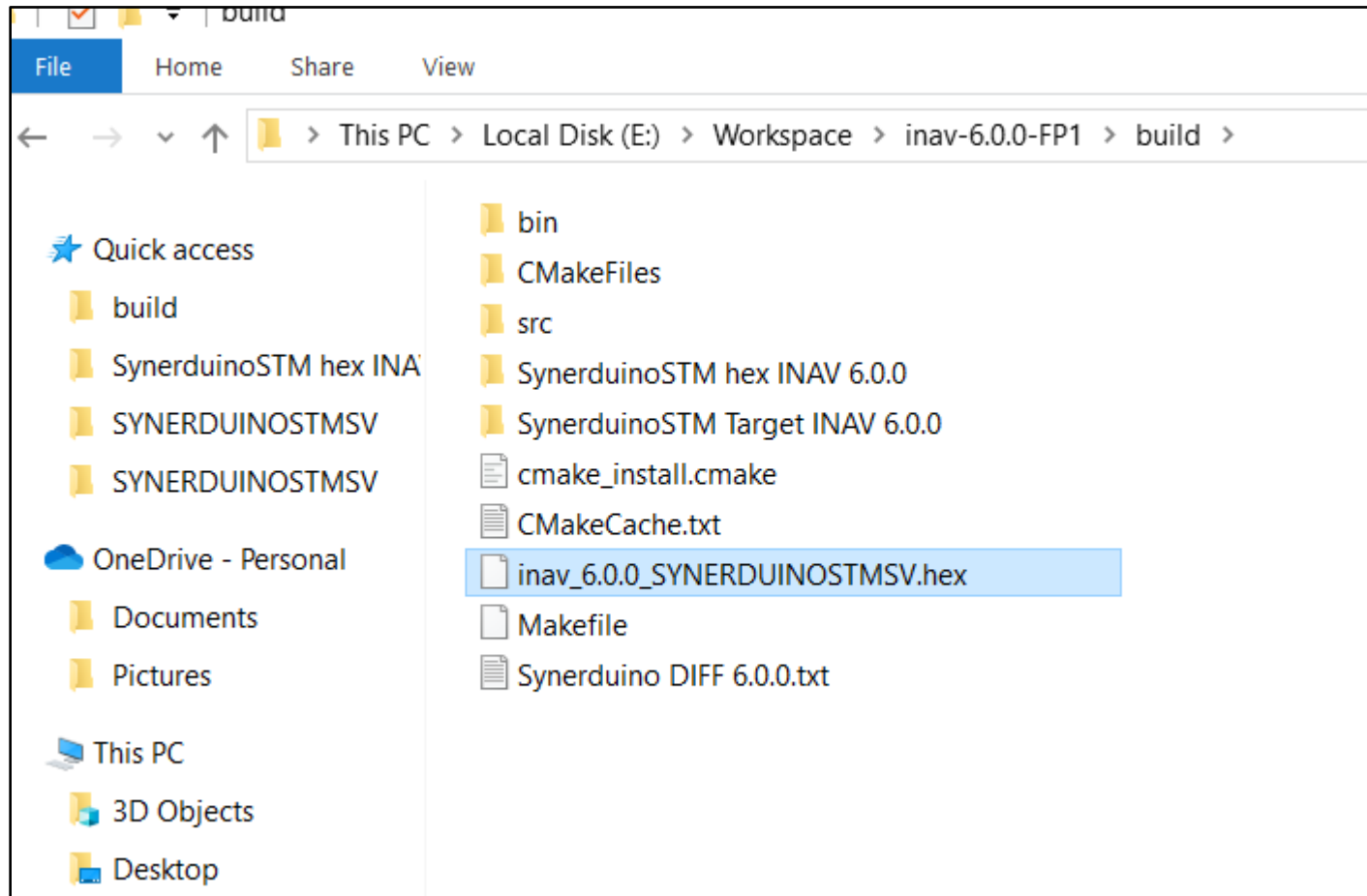
```

|       |                       |                     |                          |
|-------|-----------------------|---------------------|--------------------------|
| INAV6 | make SYNERDUINOSTM    | INAV7<br>INAV8 - 10 | make SYNERDUINOSTM_F411  |
|       | make SYNERDUINOSTM2   |                     | make SYNERDUINOSTM_F405  |
|       | make SYNERDUINOSTMSV  |                     | make SYNERDUINOH7        |
|       | make SYNERDUINOSTMSV2 |                     | make SYNERDUINOSTM_H743A |

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 .

boot the board in DFU mode and  
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-6.0.0/build
```

```
cd /e/Workspace/inav-7.0.0/build
```

```
cd /e/Workspace/inav-7.1.0/build
```

```
cd /e/Workspace/inav-7.1.1/build
```

```
cd /e/Workspace/inav-8.0.0/build
```

```
cd /e/Workspace/inav-9.0.0/build
```

```
cd /e/Workspace/inav-10.0.0/build
```

**This article endeavours to provide a generic guide for compiling INAV on Linux for INAV 2.6 and later.**

INAV requires a reasonably modern gcc-arm-none-eabi cross-compiler.

Different Linux distros will provide different versions of the cross-compiler. This will range from obsolete versions (e.g. Debian, Ubuntu LTS) to the latest stable release (Arch Linux).

In order to provide a uniform and reasonably modern cross compiler, INAV provides for the installation of a "known good / working" cross compiler, as well as a mechanism to override this if your distro provides a more modern option (e.g Arch Linux).

In general, from a security perspective, Linux distros discourage the installation of software from sources other than the official distribution repositories and 'approved' sources (Ubuntu PPA, Arch AUR). The INAV approach of providing a recommended compiler is however both sound and justified:

- The cross-compiler is installed from a reputable source (ARM, the company that makes the CPUs used in our flight controllers)
- Distro cross-compilers are often older than the recommended INAV compiler
- The installed cross-compiler is only used to build INAV and it not obviously / generally available outside of the INAV build environment.

There are however some specific cases for using the distro cross-compiler in preference to that installed by INAV:

- You are using a distro that installs a more modern compiler (Arch)
- You are using a host platform for which ARM does not provide a compiler (e.g. Linux ia32).



## Prerequisites

In addition to a cross-compiler, it is necessary to install some other tools:

- git : clone and manage the INAV code repository
- cmake : generate the build environment
- make : run the firmware compilation
- ruby : build some generated source files from JSON definitions
- gcc : native compiler used to generate settings and run tests

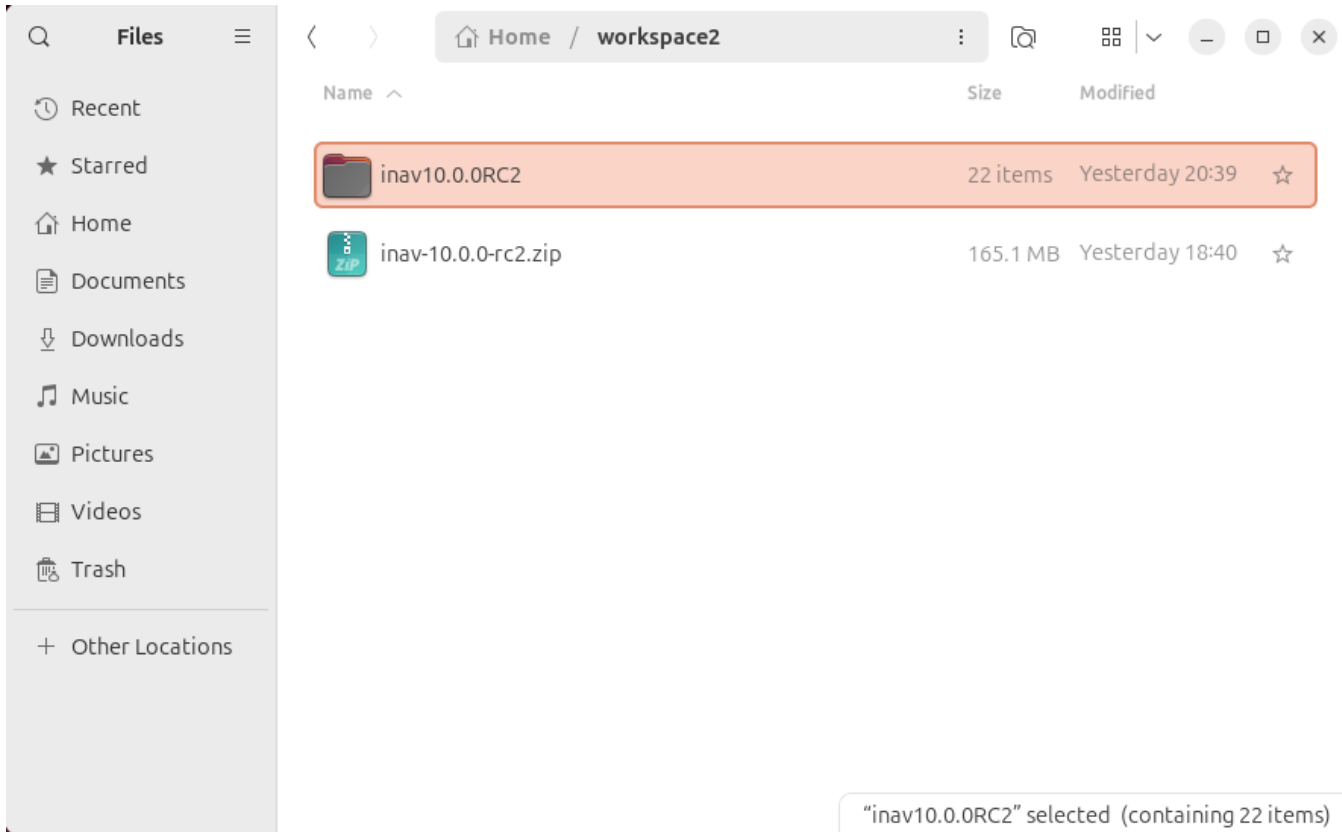
Note that INAV requires cmake version 3.13 or later; any distro that provides cmake 3.13 will also provide adequate versions of the other tools. Note also that Ubuntu 18.04 LTS does NOT provide a modern enough cmake; it is recommended that you upgrade to Ubuntu 20.04 LTS which does.

## Ubuntu / Debian

# make sure the system is updated first

```
sudo apt update && sudo apt upgrade  
sudo apt install git make ruby cmake gcc
```





## Cloning the repository

git clone <https://github.com/iNavFlight/inav.git>

Note: If you have a Github account with registered ssh key you can replace the git clone command with git clone

<git@github.com:iNavFlight/inav.git>

instead of the https link.

The git clone creates an inav directory; we can enter this directory, configure the build environment and build firmware.

**Here Ive just**

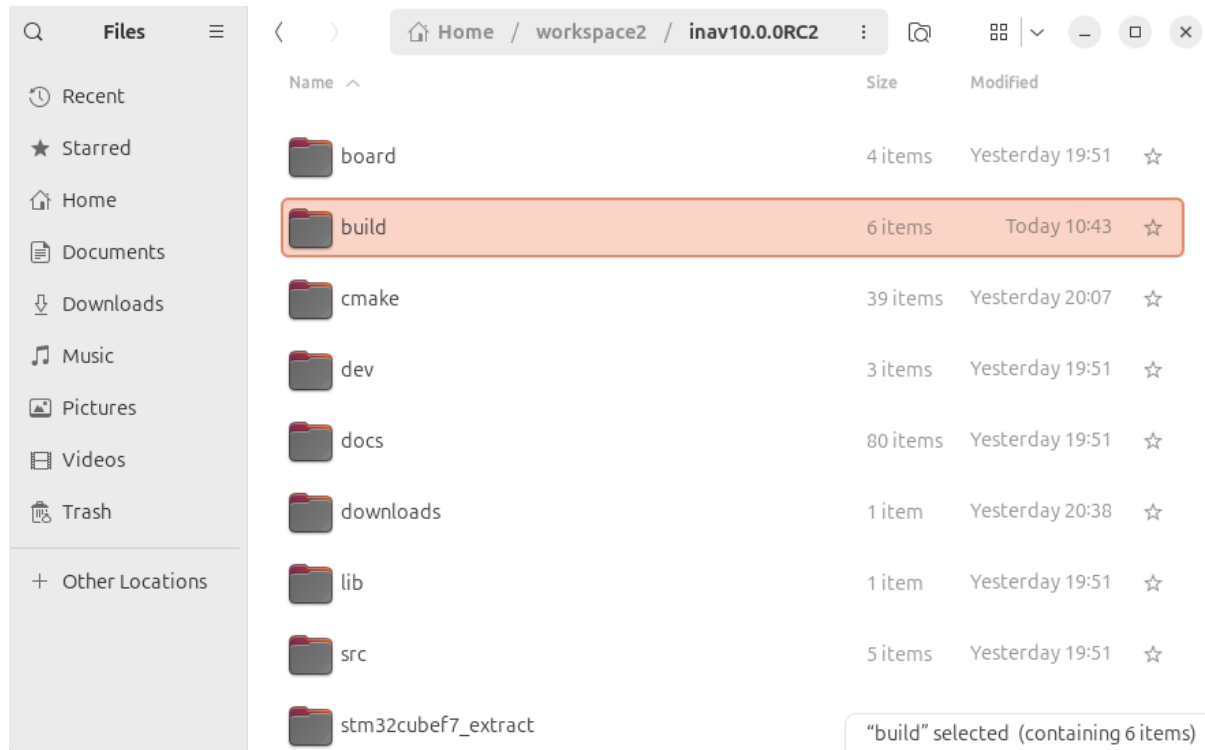
**Create New folder name it workspace2**

**Downloaded the Zip file from Github with the inav 10.0.0-rc2 tag**

**Unzip it in workspace 2**

## Build tooling

For 2.6 and later, INAV uses cmake as its primary build tool. cmake simplifies various platform and hardware dependencies required to cross compile multiple targets. cmake still uses GNU make to invoke the actual compiler. It is necessary to configure the build environment with cmake before we can build any firmware.



## Using cmake

The canonical method of using cmake is to create a build directory and run the cmake and make commands from within the build directory.

So, assuming we've cloned the firmware repository into an inav directory, we can issue the following commands to set up the build environment.

```
cd inav # first time only, create the build directory
mkdir build
cd build
cmake .. # note the "..", this is required as it tells cmake where to find its ruleset
```

**Create a new folder inside your INAV directory ,  
Name it build**

# Firmware Patch

Note you need to download and fix the H7xx BMI160 Diver by replacing it . Its available from synerflight website

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

accgyro\_BMI160.c

(E/Workspace/inav-10.0.0/scr/main/drivers/accgyro)  
Readme file is also available in the download  
is also modify to use with the SynerduinoSTM shield

system\_stm32h7xx.c  
(E/Workspace/inav-10.0.0/scr/main/target)

## Xpack requirment

(INAV 5 – 6) [xpack-arm-none-eabi-gcc-10.2.1-1.1-win32-x64](#)

(INAV 7 ) [xpack-arm-none-eabi-gcc-10.3.1-2.2-win32-x64](#)

(INAV 8-10) [xpack-arm-none-eabi-gcc-13.2.1-1.1-win32-x64](#)

## Firmware Patch

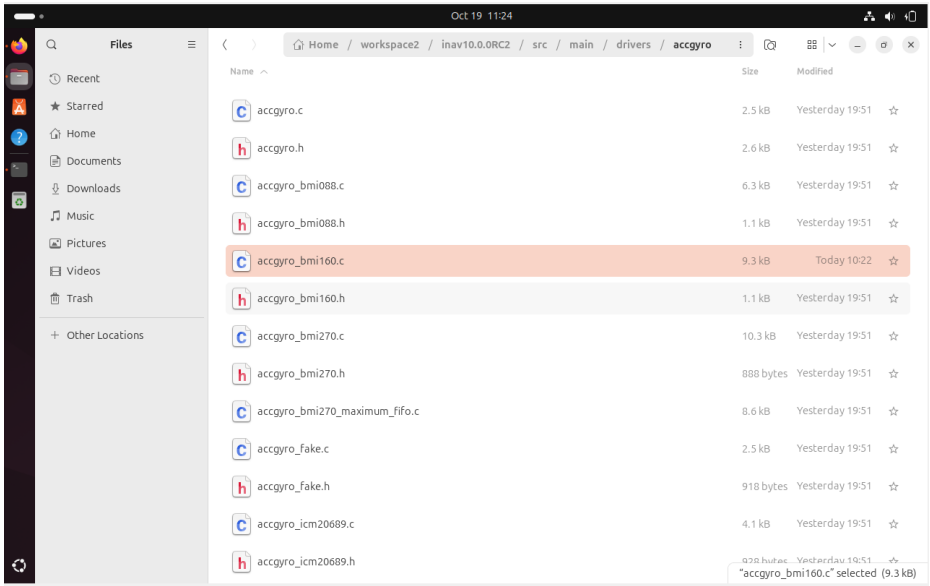
this contains the GYROACC driver fix and the H7xx Fix

[firmware patch INAV9.0.0](#)

[firmware patch INAV10.0.0](#)

## VSCode-for scripting

[VSCodeUserSetup-x64-1.72.2](#)



## Step 1: Force APT to Repair Broken Dependencies

Open your terminal and force Ubuntu to resolve the missing libraries and clear the stuck kernel packages by running:

**bash**

```
# 1. Clear any lock files and partial downloads
```

```
sudo apt clean
```

```
# 2. Resynchronize your system package index lists
```

```
sudo apt update
```

```
# 3. Force APT to evaluate and fix the broken packages automatically
```

```
sudo apt --fix-broken install -y
```

(This command will actively search for the missing `libjsoncpp25`, `librhash0`, and kernel header versions to align your cache).  Linux Journal +1

Once it completes, try your installation command again:

**bash**

```
sudo apt install -y git cmake build-essential gcc-arm-none-eabi libnewlib.
```

Use code with caution.



Due to the Size of INAV10 and later we use ubuntu terminal for this

You can start by clearing any old or broken installations

Open your ubuntu terminal

Sudo apt clean

Sudo apt update

Sudo apt --fix broken install -y

Sudo apt install -y git cmake build-essential gcc-arm-none-eabi libnewlib

## Step 2: The Best Alternative (Bypass System Dependency Errors entirely)

If your native system packages remain stubborn due to local mirror mismatches, you do not need to waste time fixing local kernel errors just to build your drone code.

Instead, switch to installing **Snap CMake** and compiling INAV inside **Docker**, which completely isolates the application from your broken local environment variables:

**bash**

```
# 1. Bypass your system's native apt error by installing CMake via Snap
sudo snap install cmake --classic

# 2. Install Docker to handle the ARM compiler and standard C library layers automatically
sudo apt install -y docker.io

# 3. Run the compiler toolchain safely inside the official isolated container
cd inav
docker run --rm -v "${pwd}:/src" inavflight/inav-build make TARGET=YOUR_BOARD_TARGET
```



```
syner@syner-VirtualBox:~/workspace$ cd  
syner@syner-VirtualBox:~$ cd workspace2  
syner@syner-VirtualBox:~/workspace2$ cd inav10.0.0RC2  
syner@syner-VirtualBox:~/workspace2/inav10.0.0RC2$ cd build  
syner@syner-VirtualBox:~/workspace2/inav10.0.0RC2/build$ cmake ..
```



Once done you can cd workspace2/inav10.0.0rc2/build

Cmake..

```
Oct 19 10:36
syner@syner-VirtualBox: ~/workspace2/inav10.0.0RC2/build
syner@syner-VirtualBox:~/workspace2/inav10.0.0RC2/build$ cmake ..
-- found arm-none-eabi-gcc 13.2.1 at /home/syner/workspace2/inav10.0.0RC2/tools/arm-gnu-toolchain-13.2.rel1/bin/arm-none-eabi-gcc
-- expecting arm-none-eabi-gcc version 4:13.2.0, but got version 13.2.1 instead
-- downloading arm-none-eabi-gcc 4:13.2.0 from https://developer.arm.com/-/media/Files/downloads/gnu/13.2.rel1/binrel/arm-gnu-toolchain-13.2.rel1-x86_64-arm-none-eabi.tar.xz
-- extracting arm-gnu-toolchain-13.2.rel1-x86_64-arm-none-eabi.tar.xz
-- The C compiler identification is GNU 13.2.1
-- The CXX compiler identification is GNU 13.2.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /home/syner/workspace2/inav10.0.0RC2/tools/arm-gnu-toolchain-13.2.rel1/bin/arm-none-eabi-gcc - 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: /home/syner/workspace2/inav10.0.0RC2/tools/arm-gnu-toolchain-13.2.rel1/bin/arm-none-eabi-g++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- The ASM compiler identification is GNU
-- Found assembler: /home/syner/workspace2/inav10.0.0RC2/tools/arm-gnu-toolchain-13.2.rel1/bin/arm-none-eabi-gcc
-- toolchain: arm-none-eabi, WARNINGS_AS_ERRORS: OFF
-- Could not find openocd, debugging won't be available
-- DEBUG_HARDFULTS: OFF, SEMIHOSTING: OFF
-- DEBUG_HARDFULTS: OFF, SEMIHOSTING: OFF
-- Performing Test LINKER_SUPPORTS_NO_RWX_WARNING
-- Performing Test LINKER_SUPPORTS_NO_RWX_WARNING - Failed
-- 81 targets (63 for release) found for toolchain arm-none-eabi
-- Build type: RelWithDebInfo
-- Configuring done (17.2s)
-- Generating done (18.1s)
```

Cmake..

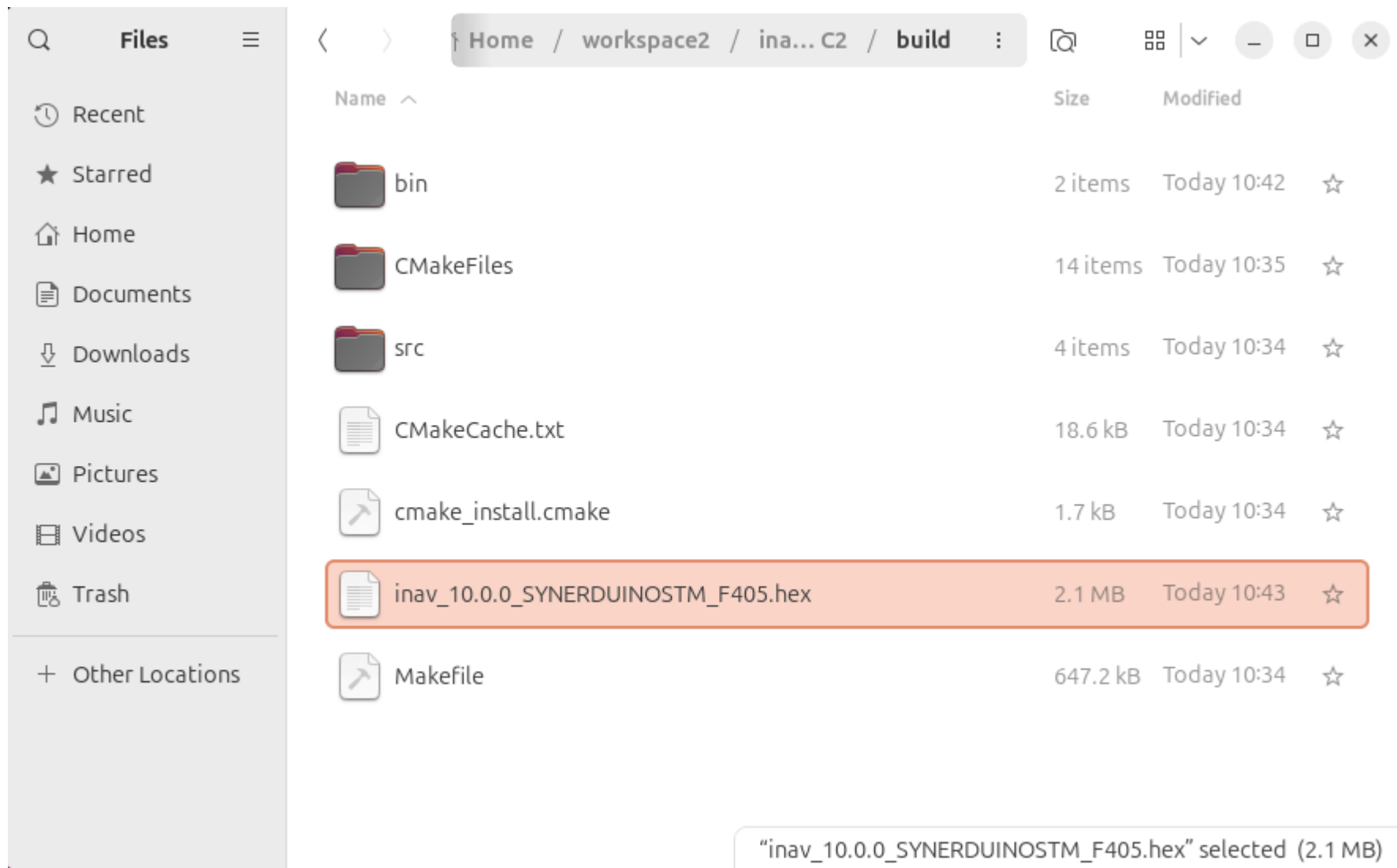
After Cmake complete  
build

Then

make  
SYNERDUINOSTM\_F405

```
syner@syner-VirtualBox:~/workspace2/inav10.0.0RC2/build$ make SYNERDUINOSTM_F405
```

```
sdhc_generated/src/uavcan.protocol.param.Value.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.protocol.RestartNode_req.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.protocol.RestartNode_res.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.protocol.SoftwareVersion.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.Timestamp.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.tunnel.Broadcast.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.tunnel.Call_req.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.tunnel.Call_res.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.tunnel.Protocol.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.tunnel.SerialConfig.c.obj
Building C object src/main/target/SYNERDUINOSTM_F405/CMakeFiles/SYNERDUINOSTM_F405.elf.dir/__/__/__/lib/main/Dronecan/d
sdhc_generated/src/uavcan.tunnel.Targetted.c.obj
Linking C executable ../../../../bin/SYNERDUINOSTM_F405.elf
Memory region      Used Size  Region Size  %age Used
      FLASH:        731436 B      896 KB      79.72%
  FLASH_CONFIG:         0 GB      128 KB       0.00%
         RAM:        110276 B      128 KB      84.13%
         CCM:         28456 B        64 KB      43.42%
  BACKUP_SRAM:         0 GB         4 KB       0.00%
    MEMORY_B1:         0 GB         0 GB
Built target SYNERDUINOSTM_F405.elf
Built target SYNERDUINOSTM_F405
syner@syner-VirtualBox:~/workspace2/inav10.0.0RC2/build$
```



SYNERDUINOSTM\_F405.h  
ex would show up on your  
build folder