

# pysimCoder

Step-by-step Installation in Ubuntu with Anaconda

by

Felipe Depine and Roberto Bucher

January 21<sup>st</sup> 2025

# Step 1: “Basic Installation”

- a) `uname -r`
- b) `sudo su`
- c) `apt-get install linux-headers-6.8.0-51-generic`
- d) `apt-get install make`

Step 1, part c:

```
emb@emb:~$ uname -r
6.8.0-51-generic
emb@emb:~$ sudo su
[sudo] password for emb:
root@emb:/home/emb# apt-get install linux-headers-6.8.0-51-generic
```

## Step 2: Install Anaconda

Link: <https://docs.anaconda.com/anaconda/install/>

## Step 3: Install Slycot and Python Control

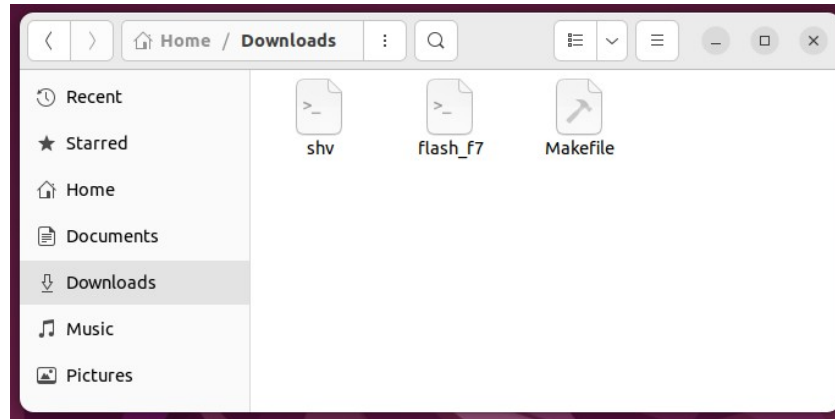
Link: <https://python-control.readthedocs.io/en/latest/intro.html>

# Step 4: Download files

Link: <https://www.robots5.com/>

Make sure  
“Allow executing files as program” is **checked**  
for “shv” and “flash\_f7”

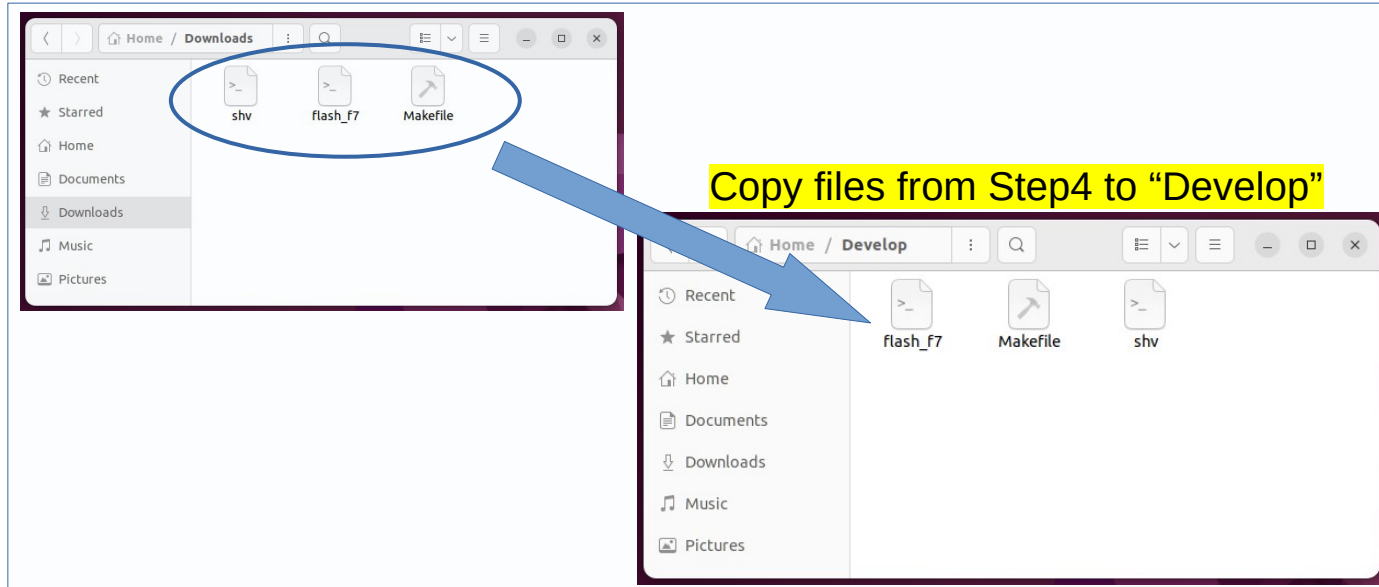
Execute ☒ Allow executing file as program



# Step 5: “Additional files for Python and pysimCoder”

- a) mkdir Develop
- b) cd Develop
- c) Copy files to Develop
- d) make packages

Step 5, part c:

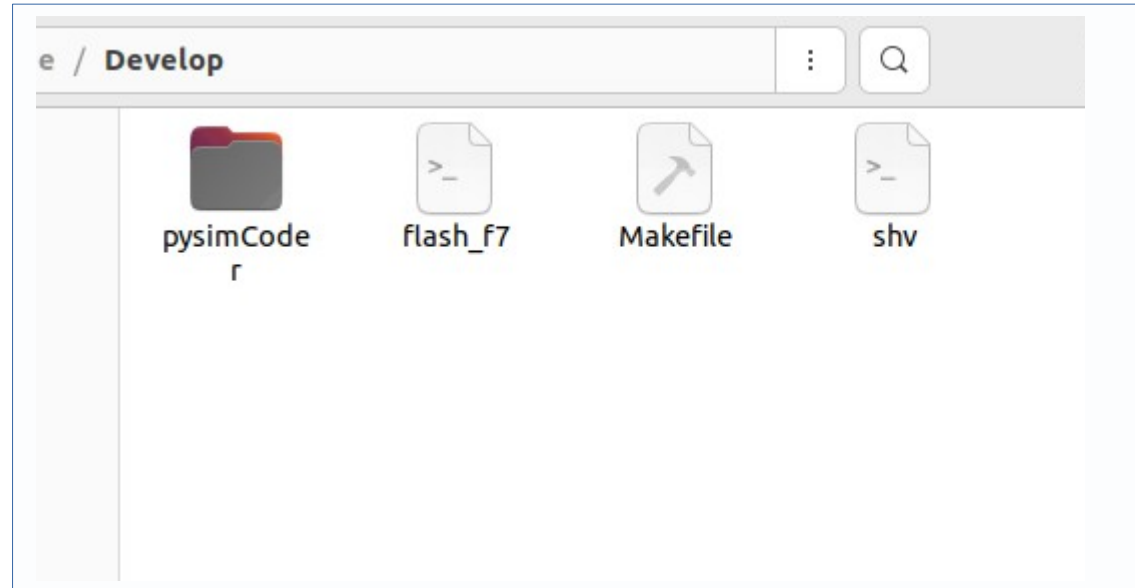


## Step 6: “pysimCoder”

a) make pysimcoder

Step 6, part a:

```
emb@emb:~/Develop$ make pysimcoder
```

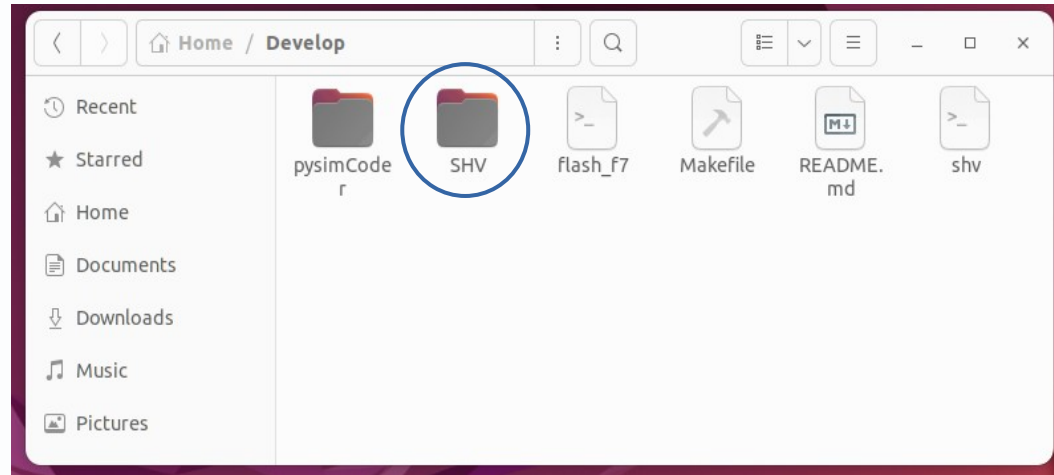


# Step 7: “Install SHV”

- a) make Shv
- b) Start a new terminal as “normal user”
- c) cd pysimCoder/CodeGen/LinuxRT/devices
- d) make SHV=1

Step 7, part a:

```
~/Develop$ make shv
```

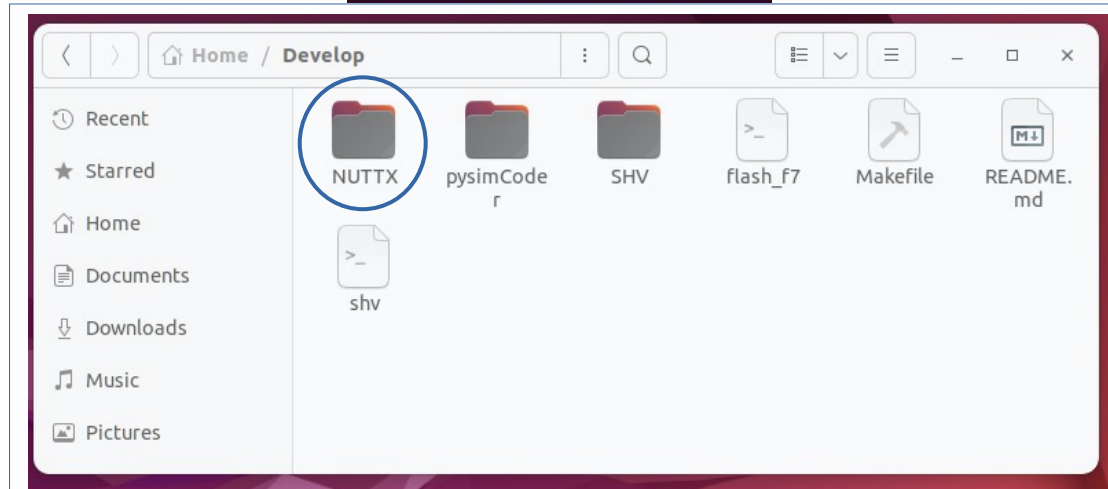




# Step 8: “Install NuttX”

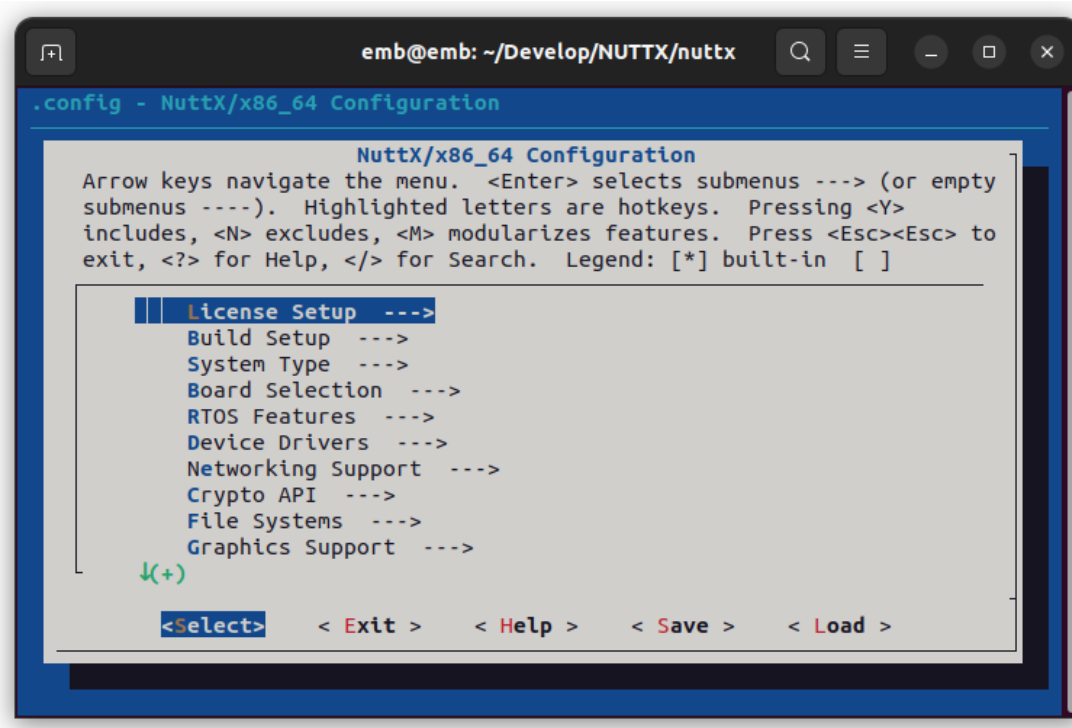
- a) make nuttx
- b) make f7
- c) Configure “make menuconfig” to match the target needs (next slides)

Step 8, part a: `~/Develop$ make nuttx`



# Step 9: Continuation “Install NuttX”

Optional  
Procedures  
depending on  
target needs.



The screenshot shows a terminal window titled "emb@emb: ~/Develop/NUTTX/nuttx". The main window is titled ".config - NuttX/x86\_64 Configuration". Inside, there's a sub-header "NuttX/x86\_64 Configuration" followed by instructions: "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 [ ]". Below this is a menu with the following options: "License Setup --->", "Build Setup --->", "System Type --->", "Board Selection --->", "RTOS Features --->", "Device Drivers --->", "Networking Support --->", "Crypto API --->", "File Systems --->", and "Graphics Support --->". A green arrow with a plus sign points to "System Type". At the bottom, there are navigation options: "<Select>", "<Exit>", "<Help>", "<Save>", and "<Load>".

```
emb@emb: ~/Develop/NUTTX/nuttx
.config - NuttX/x86_64 Configuration

NuttX/x86_64 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 [ ]

License Setup --->
Build Setup --->
System Type --->
Board Selection --->
RTOS Features --->
Device Drivers --->
Networking Support --->
Crypto API --->
File Systems --->
Graphics Support --->

↓(+)

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

Go to “System Type”, then “Timer Configuration”

# Step 9: Continuation “Install NuttX”

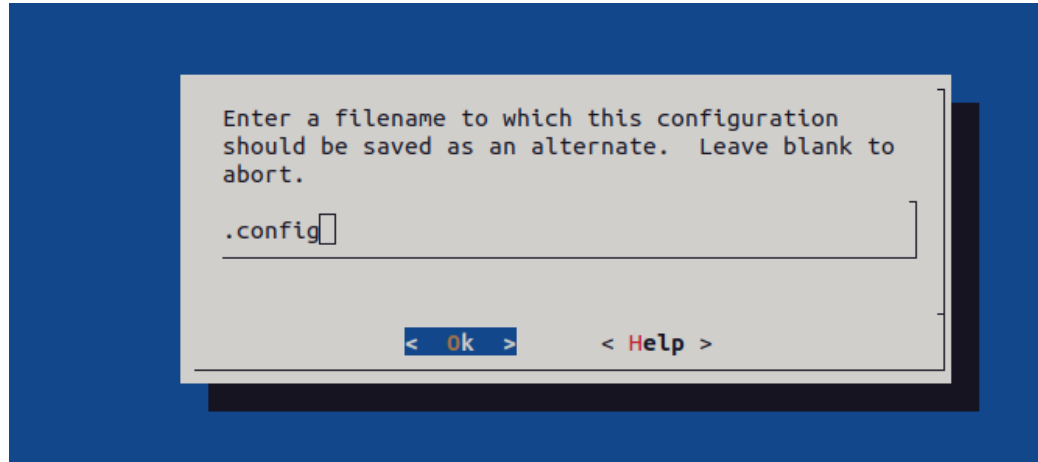
Optional  
Procedures  
depending on  
target needs.

```
[ ] PWM low-level operations
[*] TIM1 PWM
(0) TIM1 Mode
(0) TIM1 Lock Level Configuration
(0) TIM1 t_DTS Division
(10) TIM1 Initial Dead-time
[*] TIM1 Channel 1
(6) TIM1 Channel 1 Mode
[*] TIM1 Channel 1 Output
[*] TIM1 Channel 1 Complementary Output
[*] TIM1 Channel 2
(6) TIM1 Channel 2 Mode
[*] TIM1 Channel 2 Output
[*] TIM1 Channel 2 Complementary Output
[ ] TIM1 Channel 3
[ ] TIM1 Channel 4
[ ] TIM1 Channel 5 (internal)
[ ] TIM1 Channel 6 (internal)
[ ] TIM2 PWM
[ ] TIM3 PWM
[ ] TIM4 PWM
[*] PWM Multiple Output Channels
[ ] TIM PWM TRGO support
[ ] TIM1 ADC
[*] TIM2 ADC
    Select TIM2 ADC channel (TIM2 ADC channel 1) --->
[ ] TIM3 ADC
[ ] TIM4 ADC
(5000) ADC1 Sampling Frequency
(1) ADC1 Timer Trigger
[ ] TIM1 Capture
[ ] TIM2 Capture
[ ] TIM3 Capture
[ ] TIM4 Capture
STM32 TIMx Outputs Configuration --->
```

```
[ ] PWM low-level operations
[*] TIM1 PWM
(0) TIM1 Mode
(0) TIM1 Lock Level Configuration
(0) TIM1 t_DTS Division
(10) TIM1 Initial Dead-time
[*] TIM1 Channel 1
(6) TIM1 Channel 1 Mode
[*] TIM1 Channel 1 Output
[ ] TIM1 Channel 1 Complementary Output
[*] TIM1 Channel 2
(6) TIM1 Channel 2 Mode
[*] TIM1 Channel 2 Output
[ ] TIM1 Channel 2 Complementary Output
[ ] TIM1 Channel 3
[ ] TIM1 Channel 4
[ ] TIM1 Channel 5 (internal)
[ ] TIM1 Channel 6 (internal)
[ ] TIM2 PWM
[ ] TIM3 PWM
[ ] TIM4 PWM
[*] PWM Multiple Output Channels
[ ] TIM PWM TRGO support
[ ] TIM1 ADC
[*] TIM2 ADC
    Select TIM2 ADC channel (TIM2 ADC channel 1) --->
[ ] TIM3 ADC
[ ] TIM4 ADC
(5000) ADC1 Sampling Frequency
(1) ADC1 Timer Trigger
[ ] TIM1 Capture
[ ] TIM2 Capture
[ ] TIM3 Capture
[ ] TIM4 Capture
STM32 TIMx Outputs Configuration --->
```

Uncheck complementary outputs for TIM1

## Step 9: Continuation “Install NuttX”



Save as .config and exit

## Step 10: Continuation “Install NuttX”

- d) Start a new terminal as “normal user”
- e) `cd pysimCoder/CodeGen/nuttx/devices`
- f) `make SHV=1`

```
:~/Develop/pysimCoder/CodeGen/nuttx/devices$ make SHV=1
```

# Step 11: “Configure SHV Application”

- a) Set user and password in the shvspy window
- b) User: admin
- c) Password: admin!123

## Step 12: “pyshv module”

a) `sudo pip install pyshv`

Done, installation is complete!