FACULTY
OF ELECTRICAL
ENGINEERING
CTU IN PRAGUE

Open Rapid Control Prototyping, Education and Design Tools

Dion Beqiri
Contact: beqirdio@gmail.com

Supervisor: *Ing. Pavel Píša, Ph.D.*

Department of Control Engineering

Bachelor's Thesis

June 2022

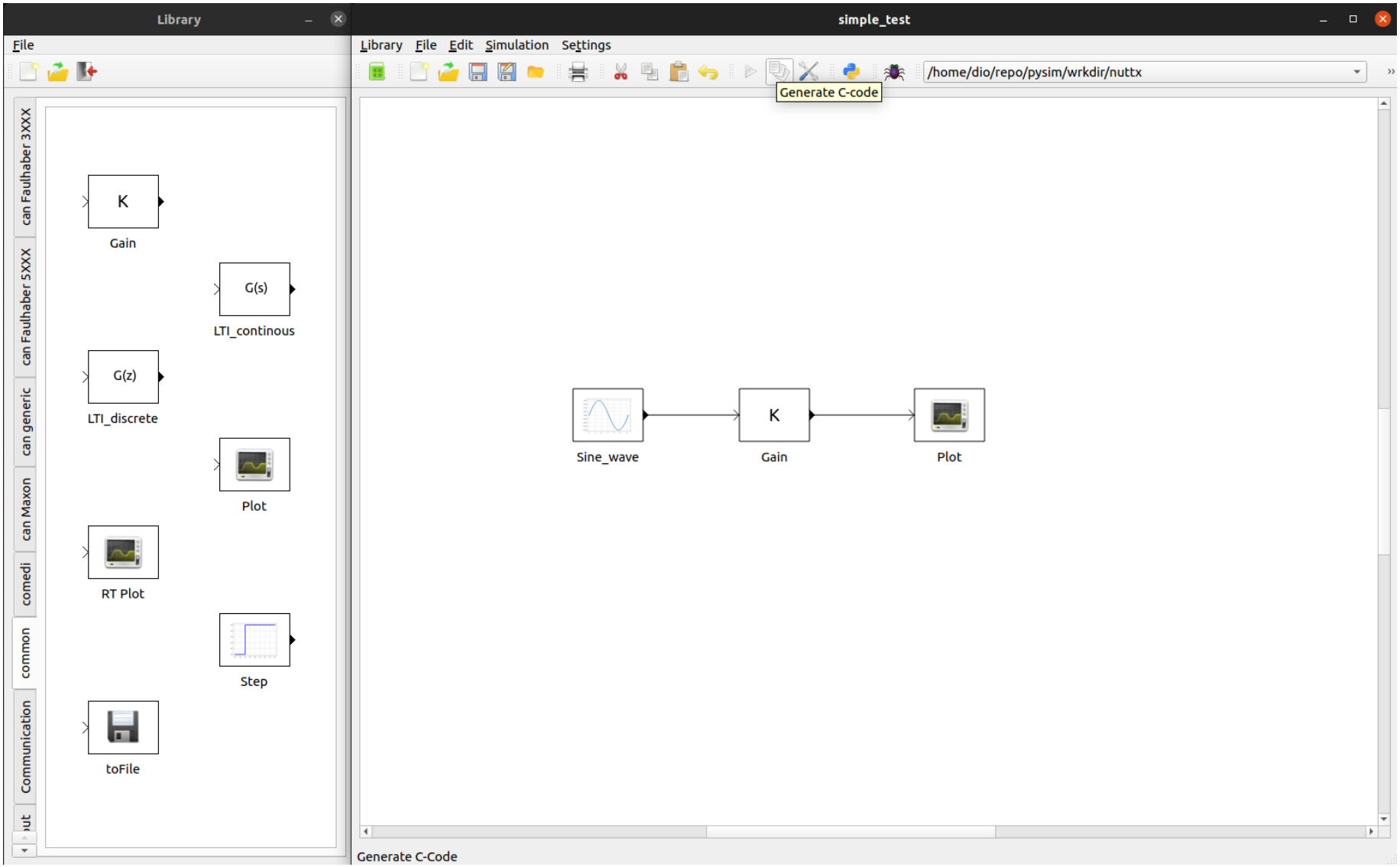
Goals

- Understanding and extending the pysimCoder project
- Vector signals feature
- Adding support for GNU/Linux targets:
  - MZ_APO Zynq based educational kit
  - RaspberryPi board (extension)
- Adding support for ESP32-C3 (RISC-V) board running NuttX RTOS
- Controlling some peripheral using combination of NuttX and pysimCoder

PysimCoder

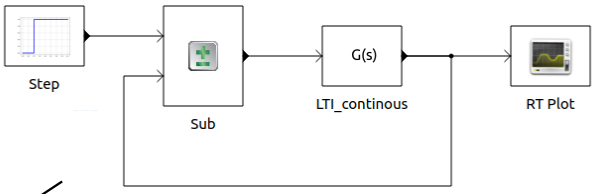
- Graphical editor for design of control systems
- Can generate real-time RPC code for various targets
- Similar in style to Simulink, Xcos, etc…
- Advantages:
  - Lighter, faster, compatible with any OS
  - Open source and freely available
  - Uses Python for most of its functions
- Disadvantages:
  - Less development, missing features, and poor (code) stability

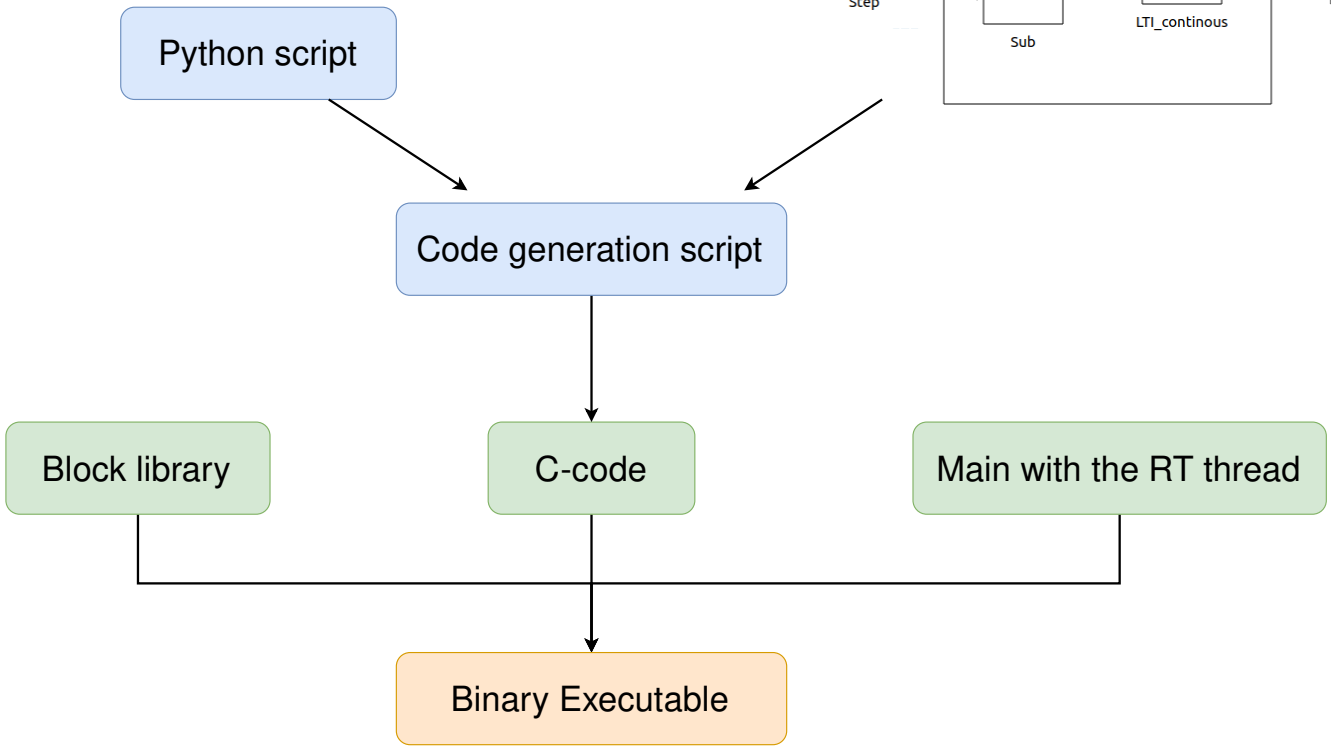
PysimCoder


Library
File
can Faulhaber 3XXX
can Faulhaber 5XXX
can generic
can Maxon
comedi
common
Communication
K
Gain
G(s)
LTI_continous
G(z)
LTI_discrete
Plot
RT Plot
Step
toFile
simple_test
Library File Edit Simulation Settings
/home/dio/repo/pysim/wrkdir/nuttx
Generate C-code
Sine_wave
K
Gain
Plot
Generate C-Code

PysimCoder

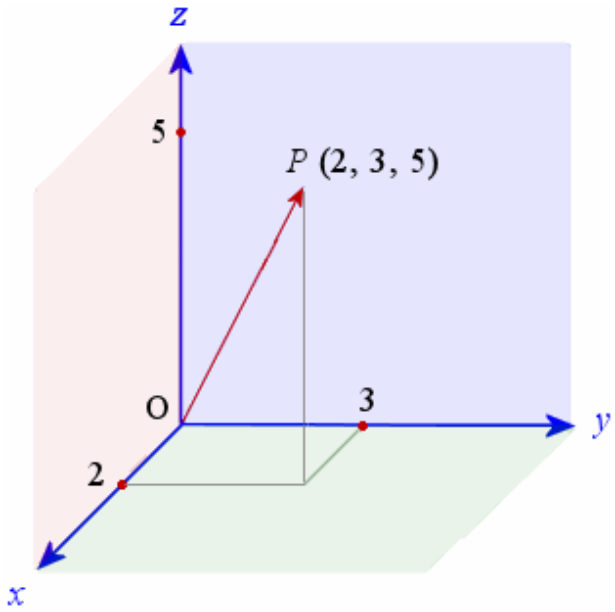
Basics of Code Generation



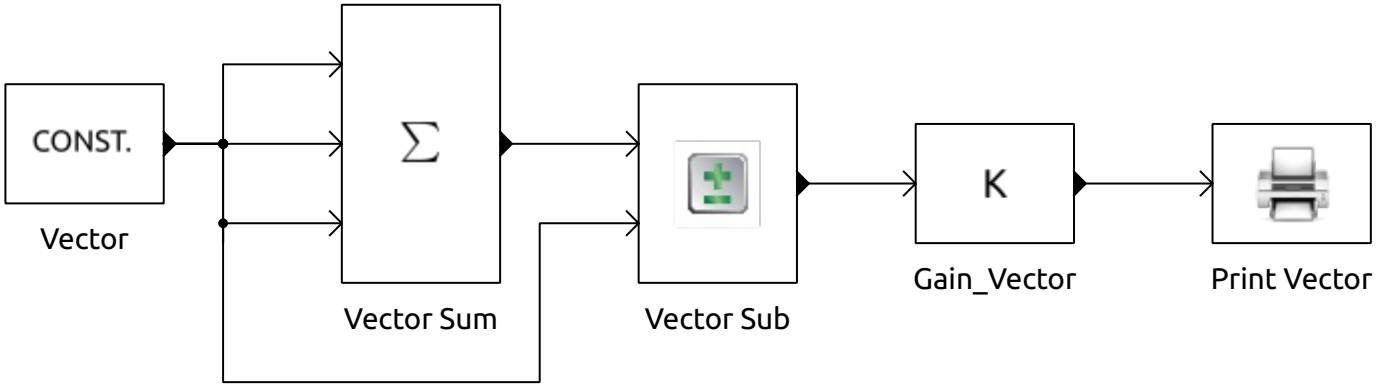


Vectors

- Importance of vectors in mathematics and control systems engineering
- Existing support for vector dimensions
- Enabling vector signals in pysimCoder
- Static setting of dimensions
- Automatic setting of dimensions

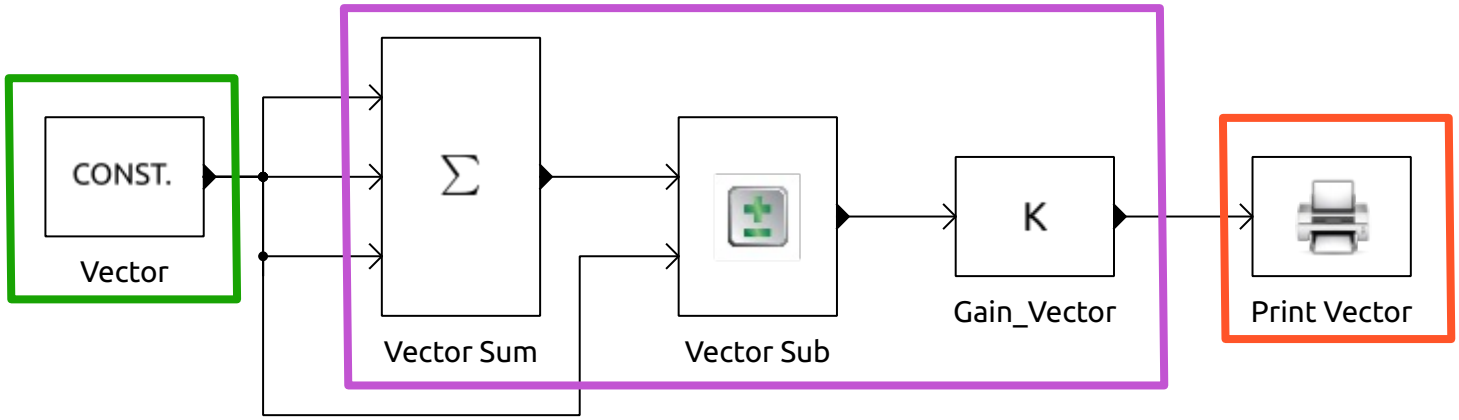


Vectors



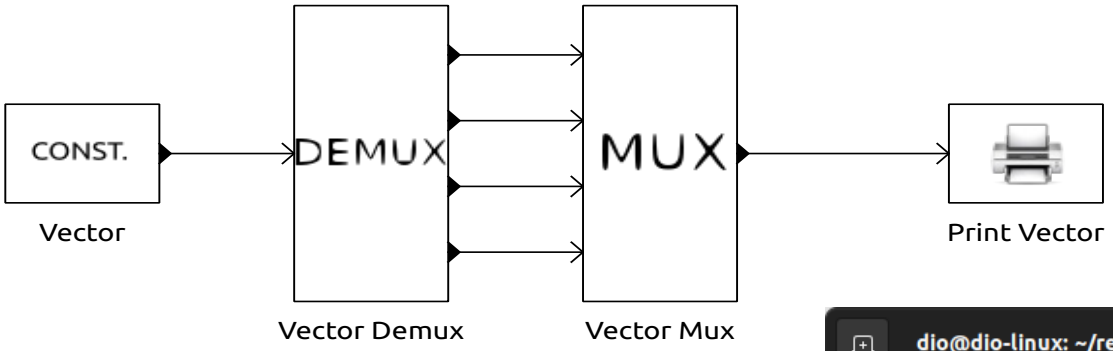
- New blocks for vector experimentation
- Mostly mathematical blocks
- Some blocks know their dimensions, others don’t
- Static dimensions → only human error is possible

Vectors



- Algorithm for setting dimensions without user input
- Assumption that blocks are ordered sequentially
- Propagation path of dimensions:
  - Source blocks (know dimensions internally)
  - Middle blocks (get dimensions externally)
  - Destination blocks (externally also)
- Middle blocks contain variable for I/O dimension ratio

Vectors

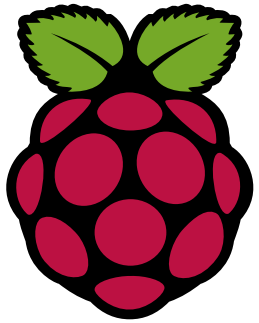


- Multiplexer combines scalar values into a vector signal
- Demultiplexer is inverse
- Link between vector-able and normal blocks
- Testing on each other for verification

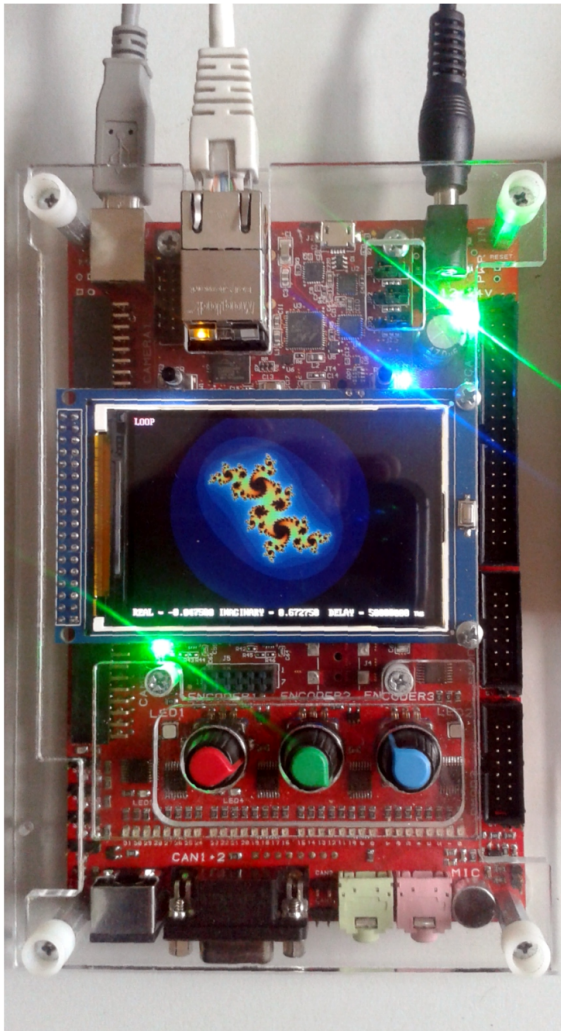
```
dio@dio-linux: ~/repo/pysim/wrk...

9.900000        [1.000000, 2.000000, 3.000000, 4.000000]
9.910000        [1.000000, 2.000000, 3.000000, 4.000000]
9.920000        [1.000000, 2.000000, 3.000000, 4.000000]
9.930000        [1.000000, 2.000000, 3.000000, 4.000000]
9.940000        [1.000000, 2.000000, 3.000000, 4.000000]
9.950000        [1.000000, 2.000000, 3.000000, 4.000000]
9.960000        [1.000000, 2.000000, 3.000000, 4.000000]
9.970000        [1.000000, 2.000000, 3.000000, 4.000000]
9.980000        [1.000000, 2.000000, 3.000000, 4.000000]
9.990000        [1.000000, 2.000000, 3.000000, 4.000000]
10.000000       [1.000000, 2.000000, 3.000000, 4.000000]
```

GNU / Linux

- Popular operating system
- Supports real-time extensions
- Open source and freely available
- Increased usage in embedded systems
- pysimCoder supports various Linux targets
- MZ_APO educational kit runs on real-time Linux (Debian distribution)
- RaspberryPi board (Raspbian distr.)

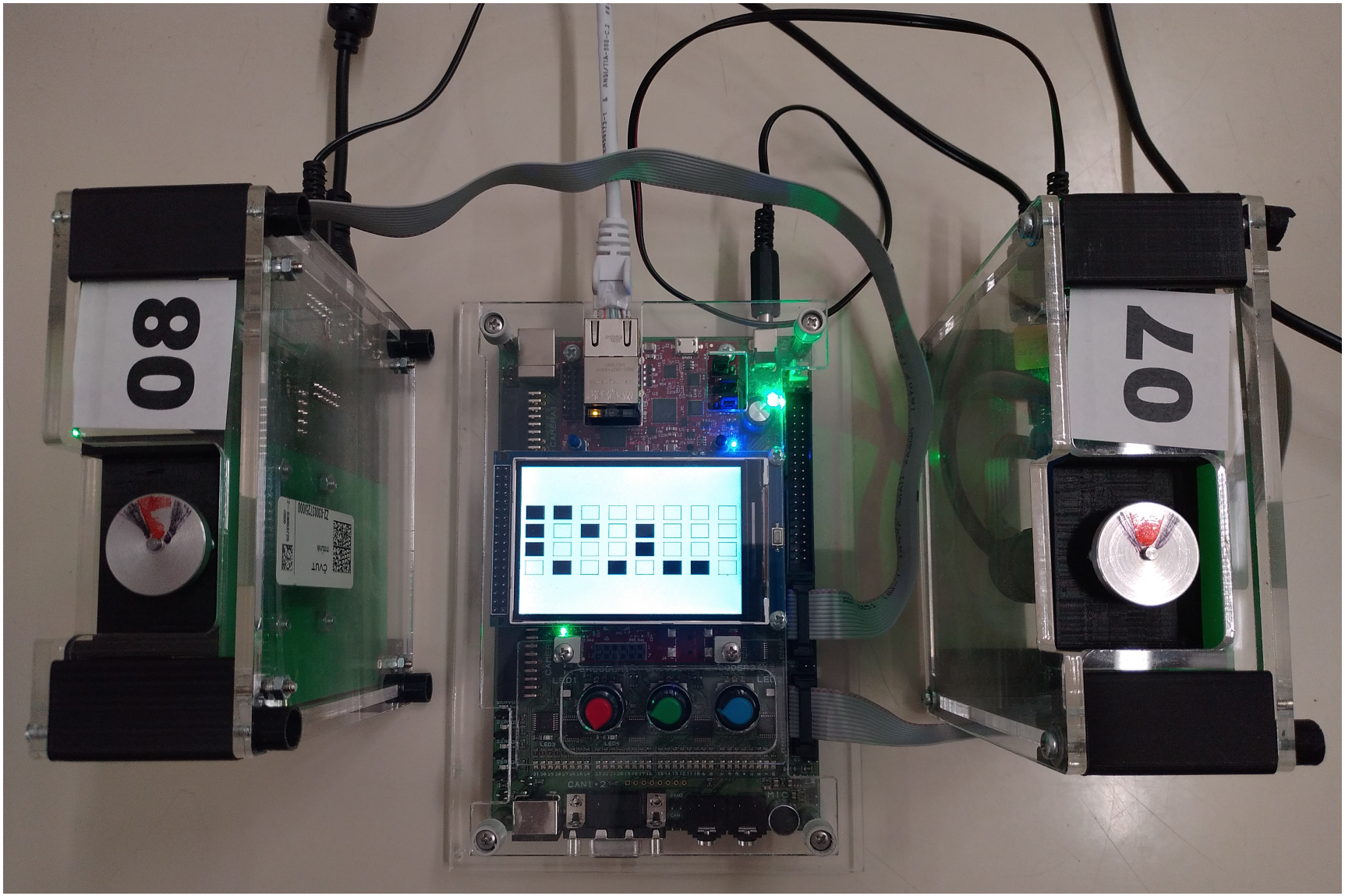


MZ_APO educational kit



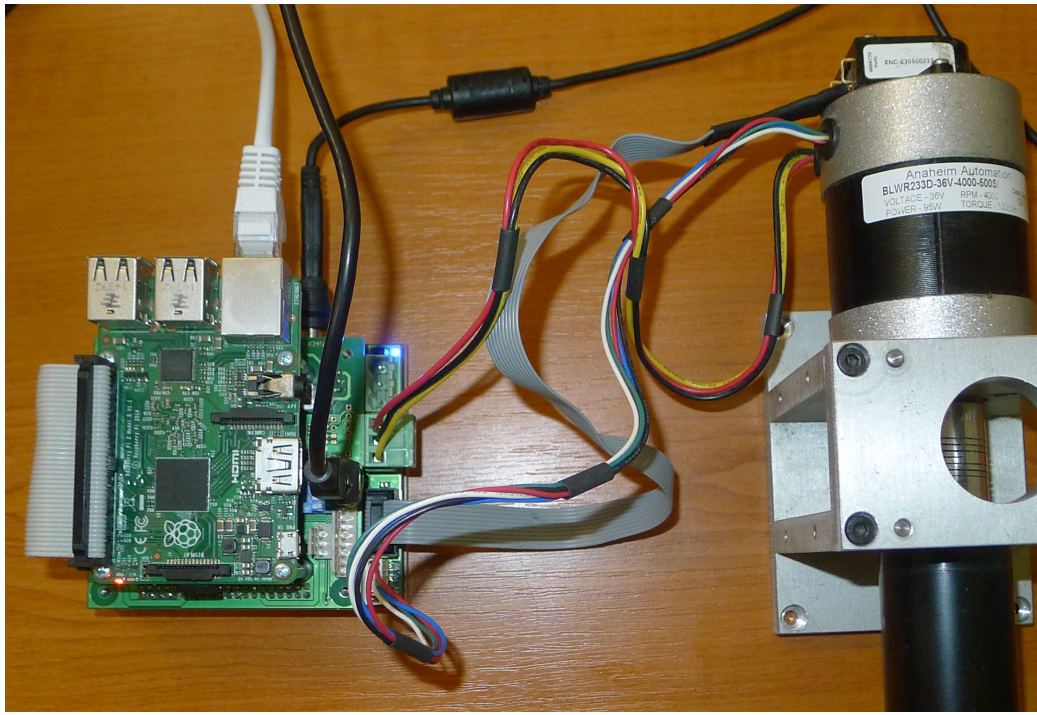
- Existing implementation of MZ_APO Simulink blocks:
  - Rotary knob incremental encoders
  - DC motor
- Adaptation into pysimCoder blocks
  - *.xblk*, *.py*, and *.c* files
- Construction of *Makefile* specific to Zynq based Linux targets
- DC motor follower with real-time PID tuning (using incremental encoders) demonstration

MZ_APO educational kit

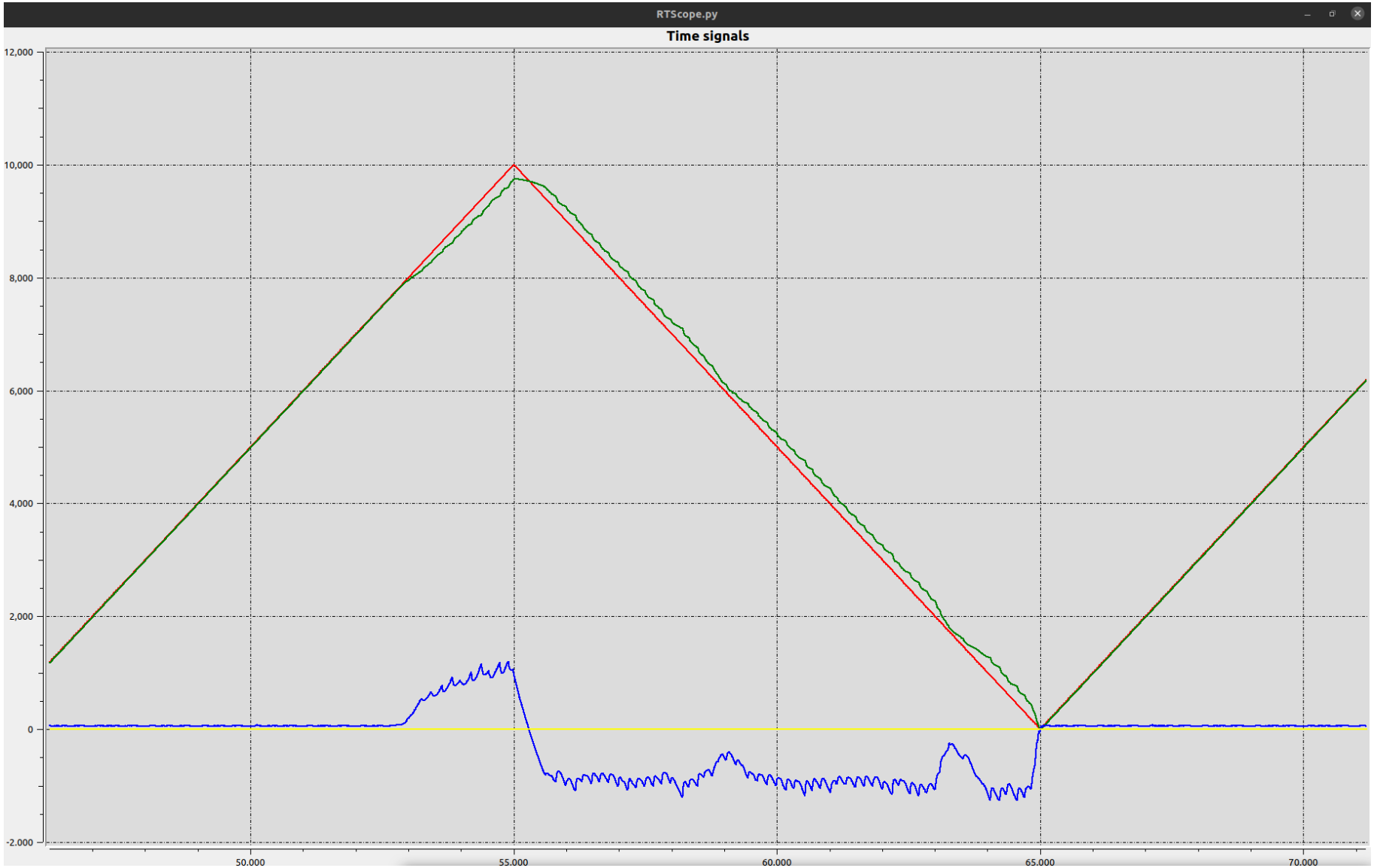


Raspberry Pi (PMSM control)

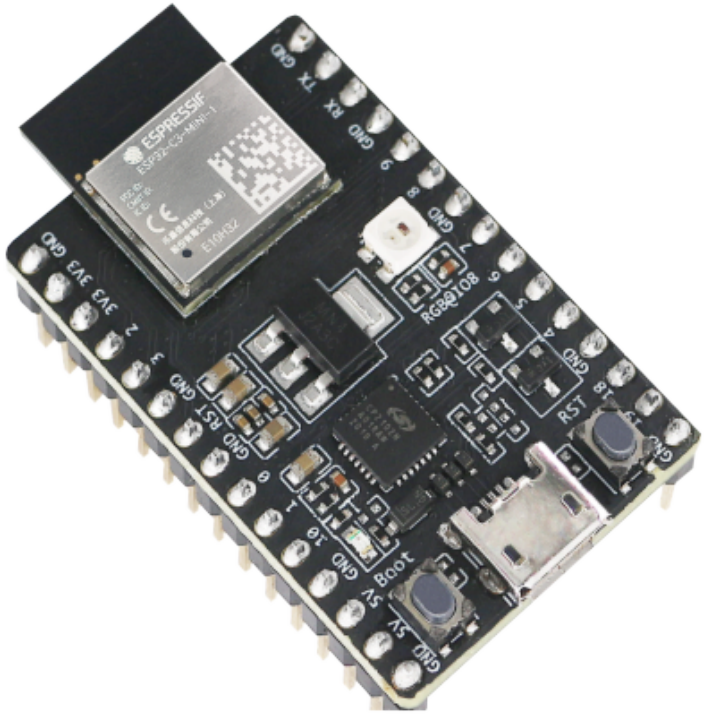
- RaspberryPi Model 2B (v1.1)
- Already supported in pysimCoder
- Existing Simulink project for controlling 3-Phase *Permanent Magnet Synchronous Motor*
- Possible adaptation to pysimCoder block and diagram



Raspberry Pi (PMSM control)

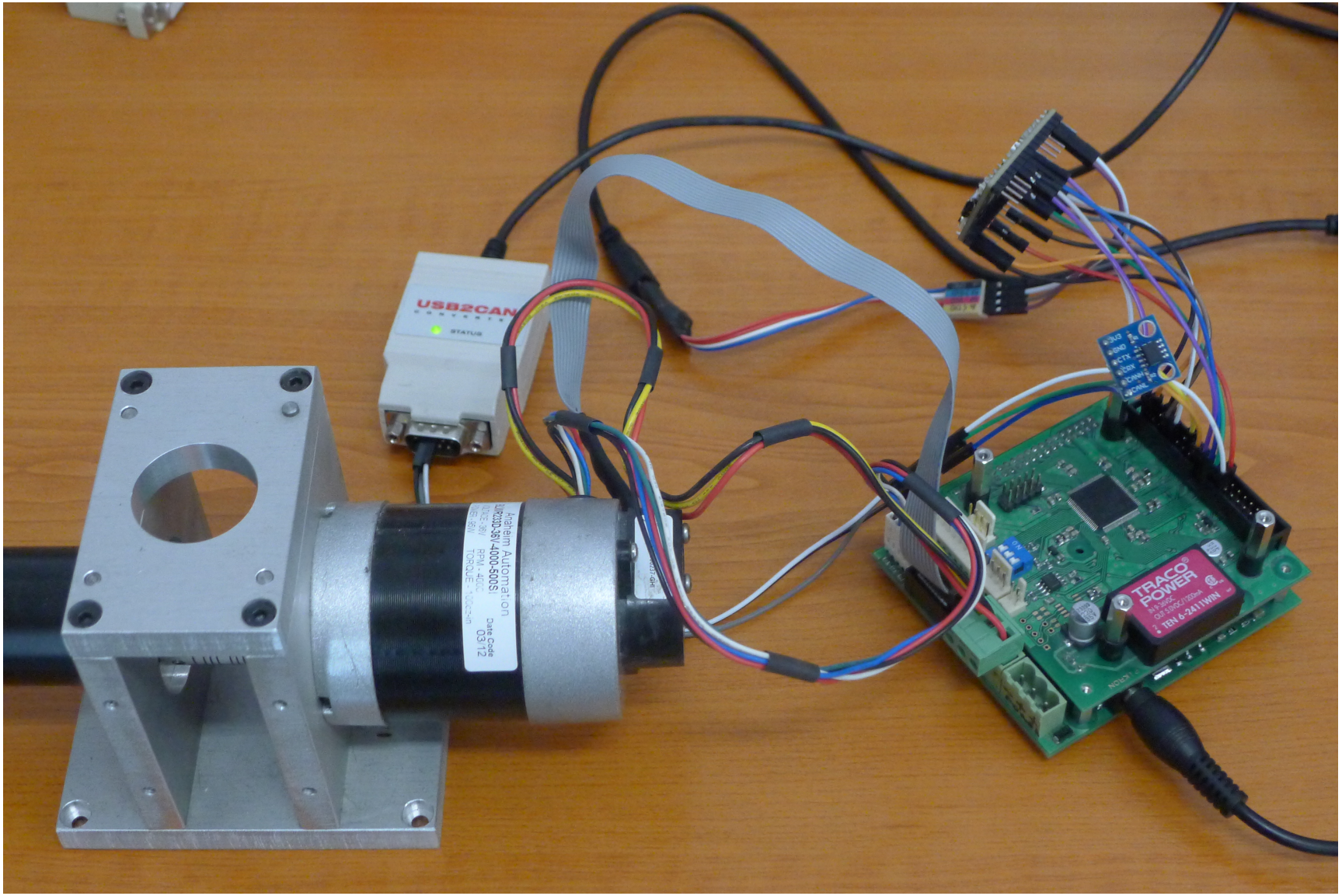


NuttX and ESP32C3

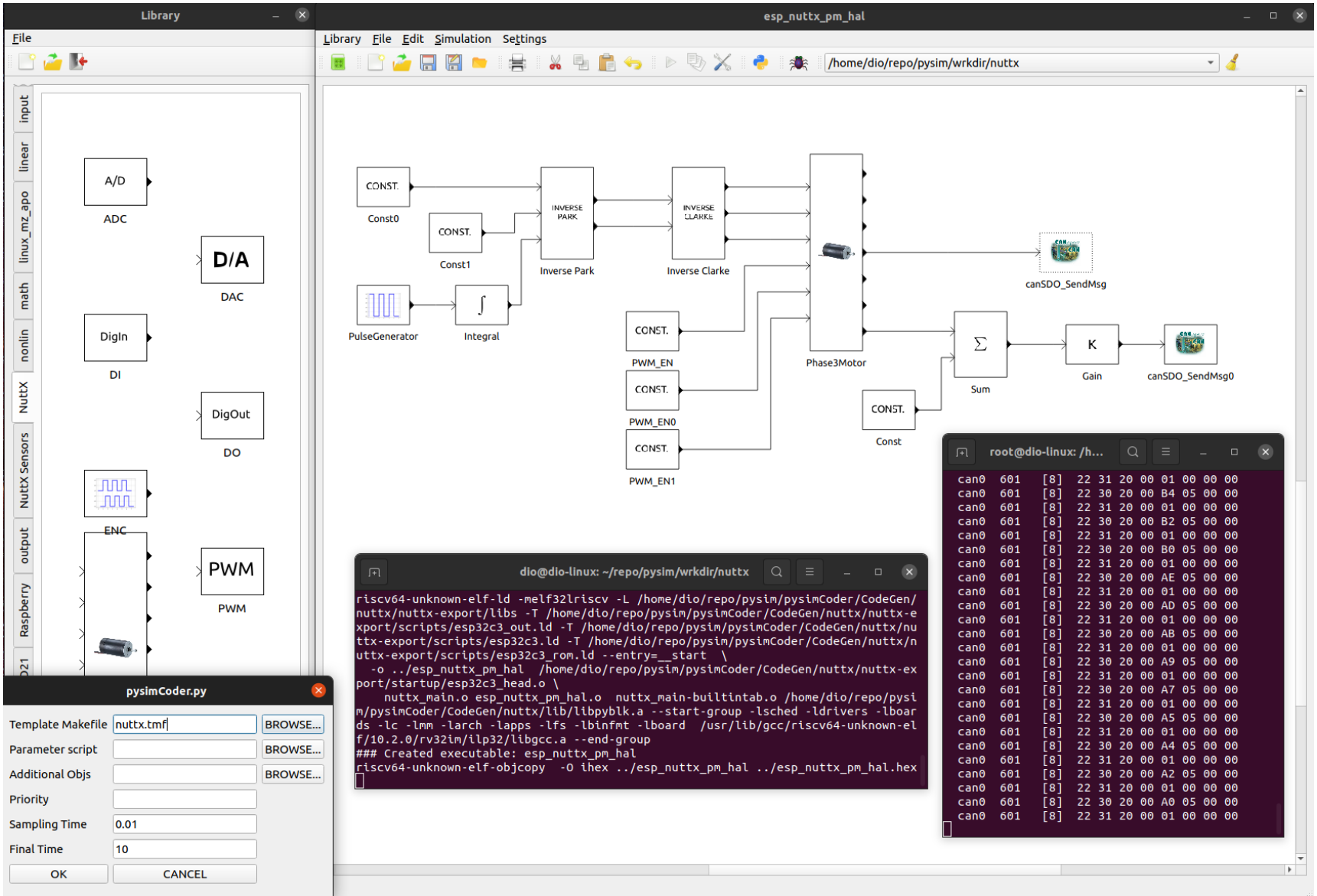


- NuttX: Real Time Operating System
- ESP32C3-DevKitM-1
- Testing NuttX blocks on ESP32C3 board
- Replacing RaspberryPi with ESP32C3 for 3-Phase PMSM motion control
- Main changes for NuttX implementation

NuttX and ESP32C3



NuttX and ESP32C3



Conclusions

- Vectors:
  - Testing on more complex diagrams
  - Optimization of dimension setting algorithm
  - Placement of algorithm on editor level
- GNU/Linux:
  - More MZ_APO hardware blocks
  - New RPi block only for specific hardware
- NuttX + ESP32C3:
  - PMSM motion control without PID due to unattainable sampling frequencies

Final Remarks

- Learned a lot both in hardware and software
- Got to use various types of controllers and peripherals
- Meaningful contributions to an open-source project
- Learned to appreciate Linux OS (for development and real-time purposes)