

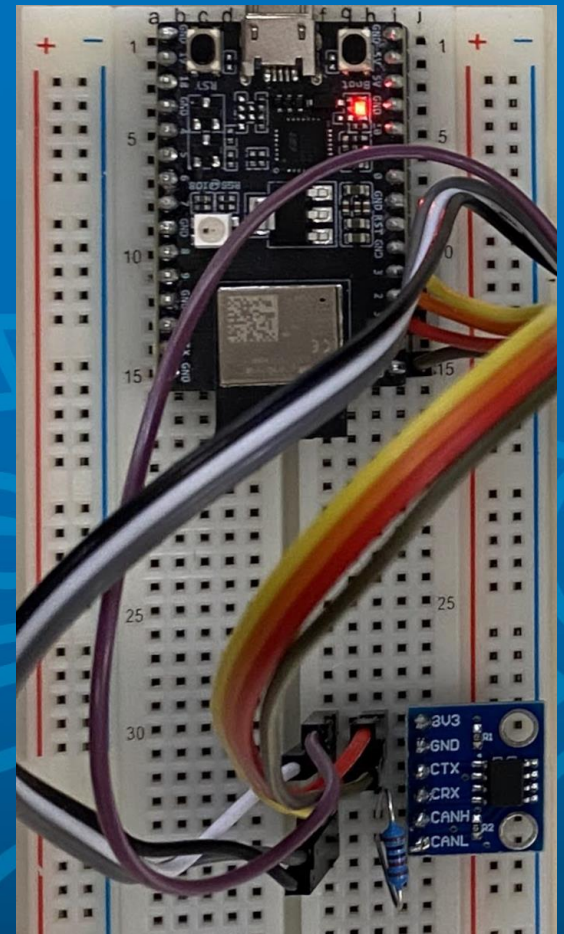


**ČVUT**

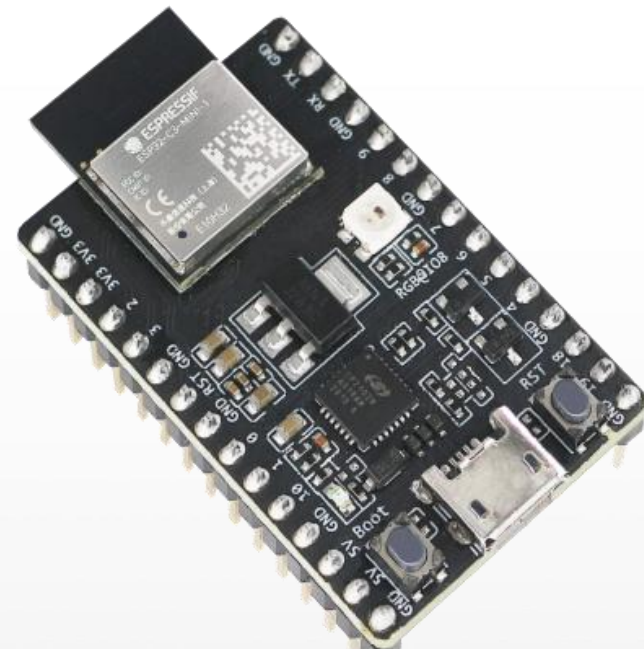
ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE

# NuttX RTOS CAN Bus Driver for Espressif ESP32C3

**Faculty of Electrical Engineering**  
**Jan Charvát**  
**Supervisor – Ing. Pavel Píša, PhD.**  
**16.6.2022**



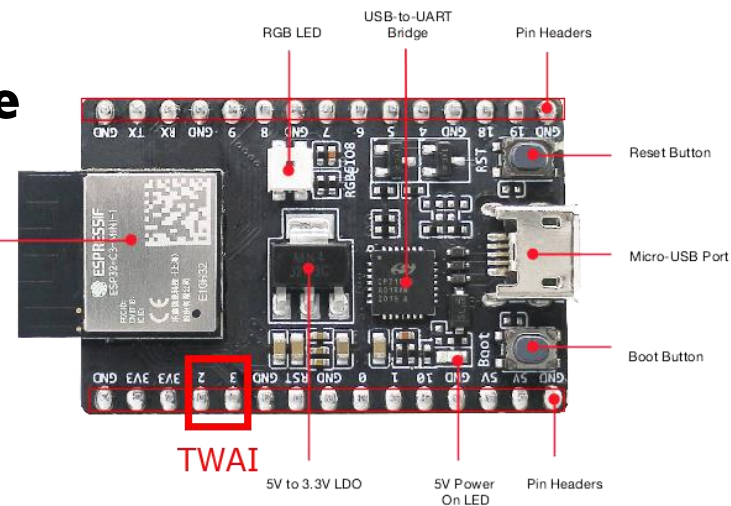
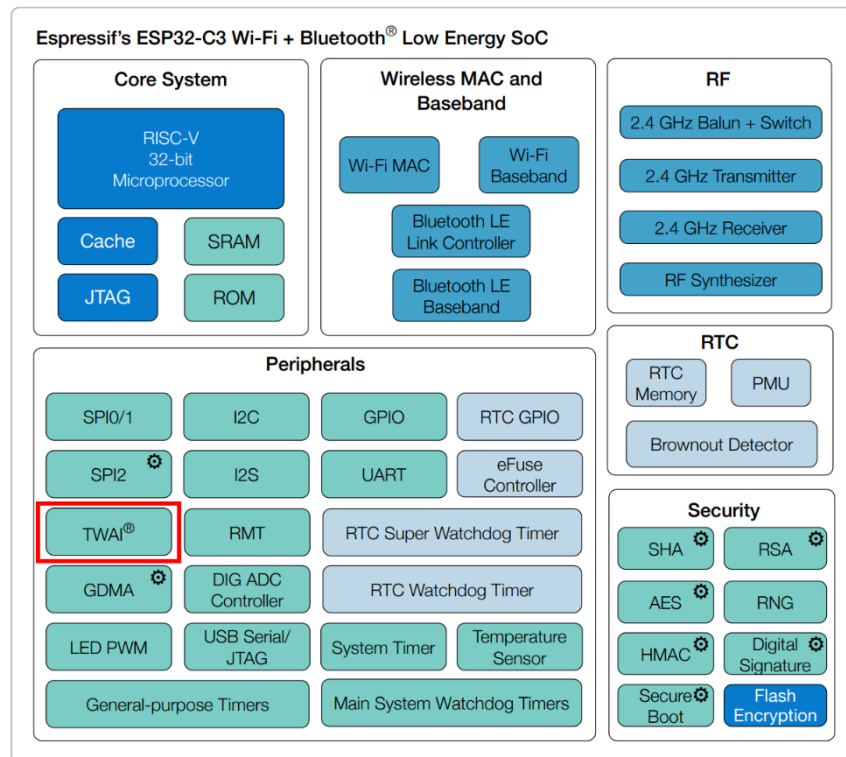
# Introduction



- **NuttX CAN subsystem analysis on ESP32C3**
- **Implementation of CAN (TWAI) driver**
- **Testing and demonstration of driver functionality**

# ESP32C3 DevKit

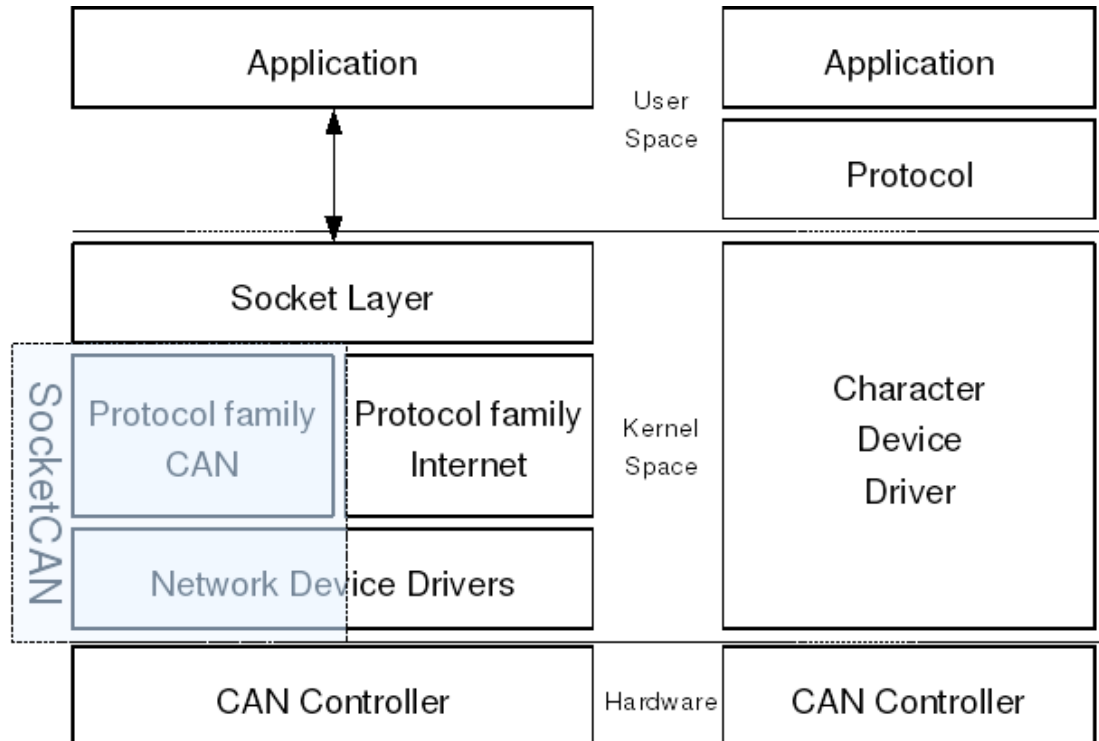
- **RISC-V single-core microcontroller**
- **Wi-Fi & Bluetooth 5 (LE) module**
- **SJA1000 CAN 2.0 controller**





# NuttX CAN subsystem

- **Open-source RTOS**
- **Character Device Driver**
- **"lower" half driver**



## Related work

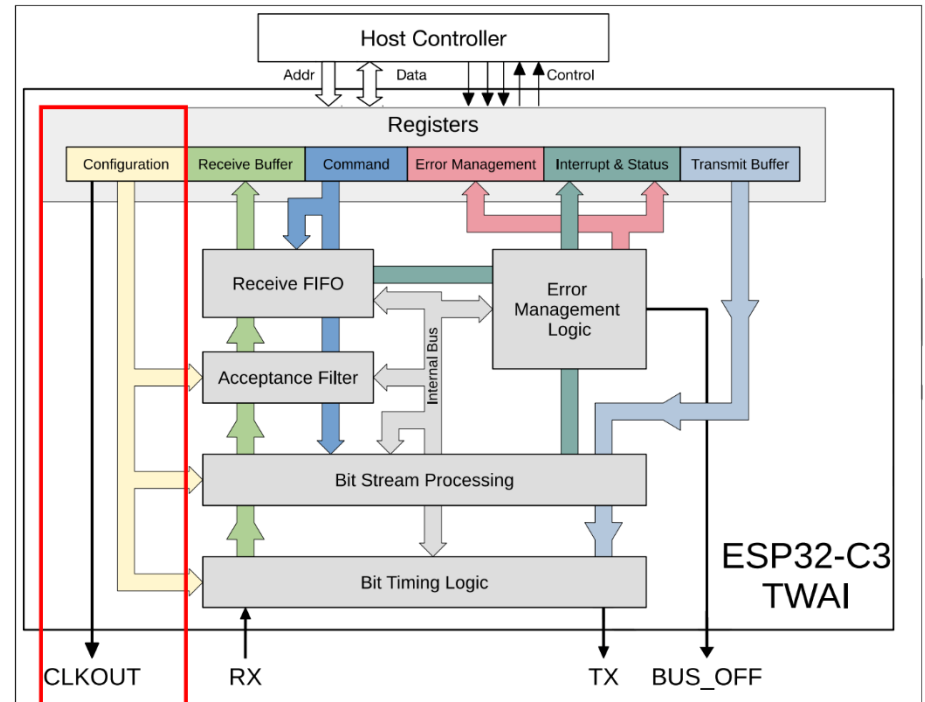
- **ESP-IDF TWAI**
  - **Official Espressif framework**
- **LinCAN**
  - **SJA1000 support**
  - **Thorough bit timing solutions**
- **NXP Ipc17-40 CAN**
  - **NuttX CAN implementation**





# TWAI initialization

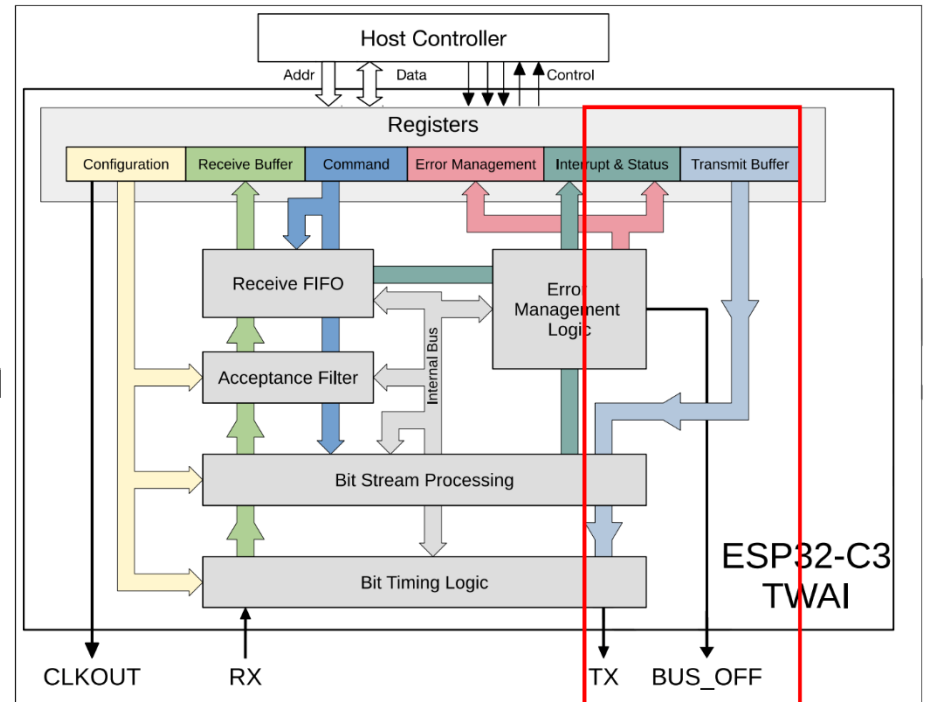
- **Power and clock enabled,  
pins configuration**
- **Reset TWAI device – different register layout**
  - **Error counters, Bit timing**
  - **Acceptance filters**
- **Setup interrupts**





# TWAI Transmission

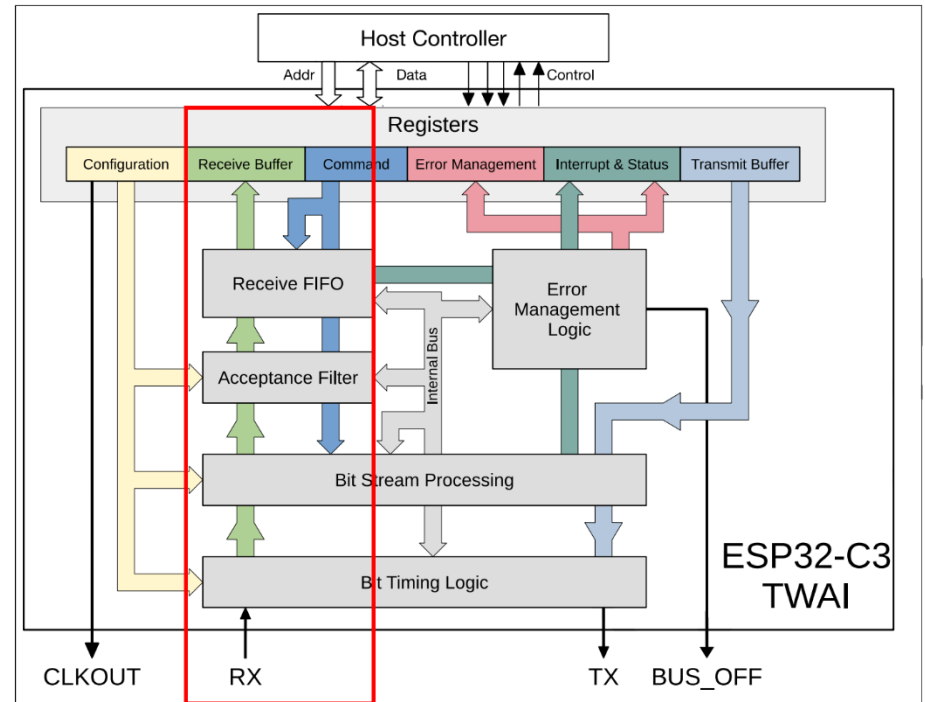
- **Single TX buffer**
- **Upper driver holds SW TX FIFO**
- **Modes for transmission**
  - **Loopback mode**
  - **Standard mode**
- **Support for extended IDs**





# TWAI Reception

- **RX interrupts**



- **RX buffer – first received frame mapped on DATA registers**
- **Support for extended IDs**



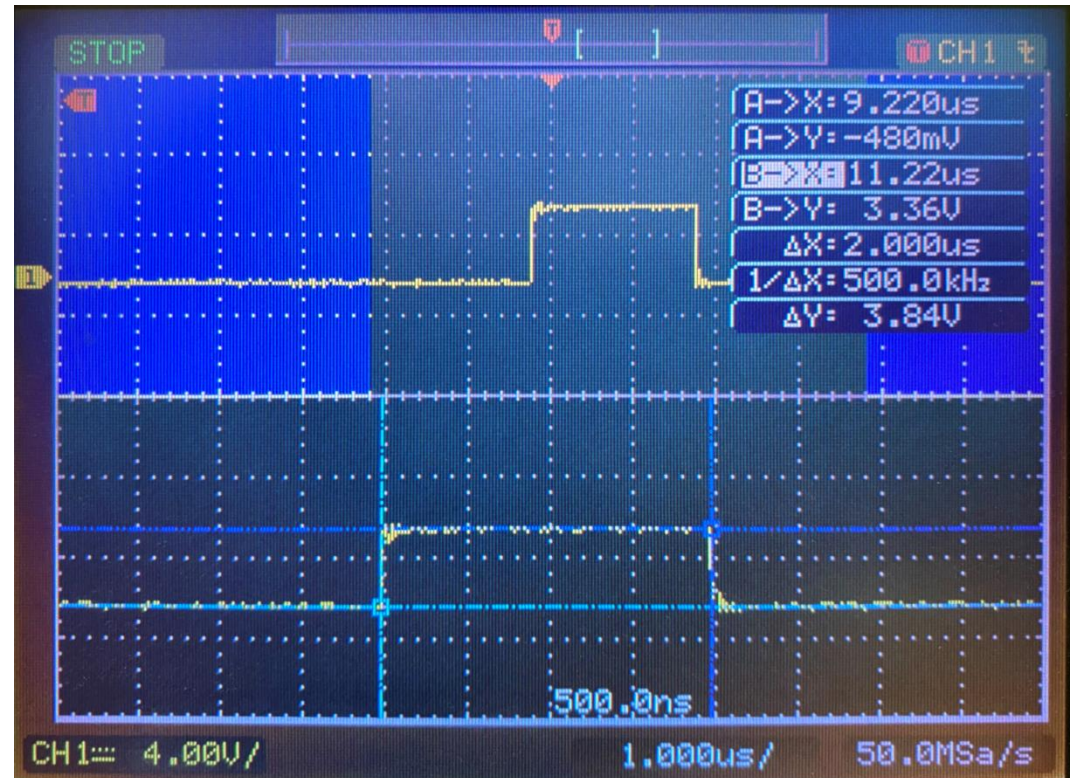


ČVUT

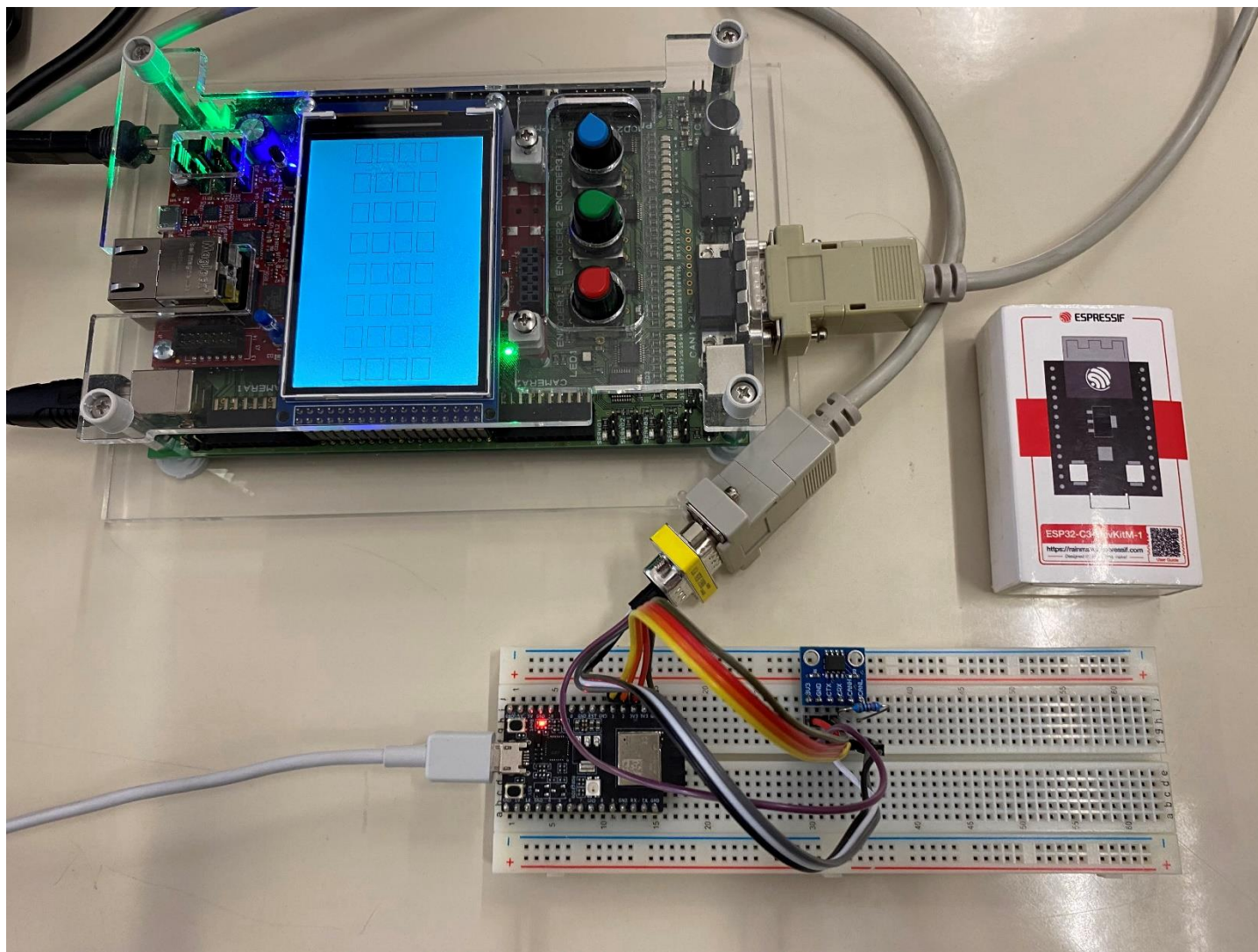
ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE

# Testing

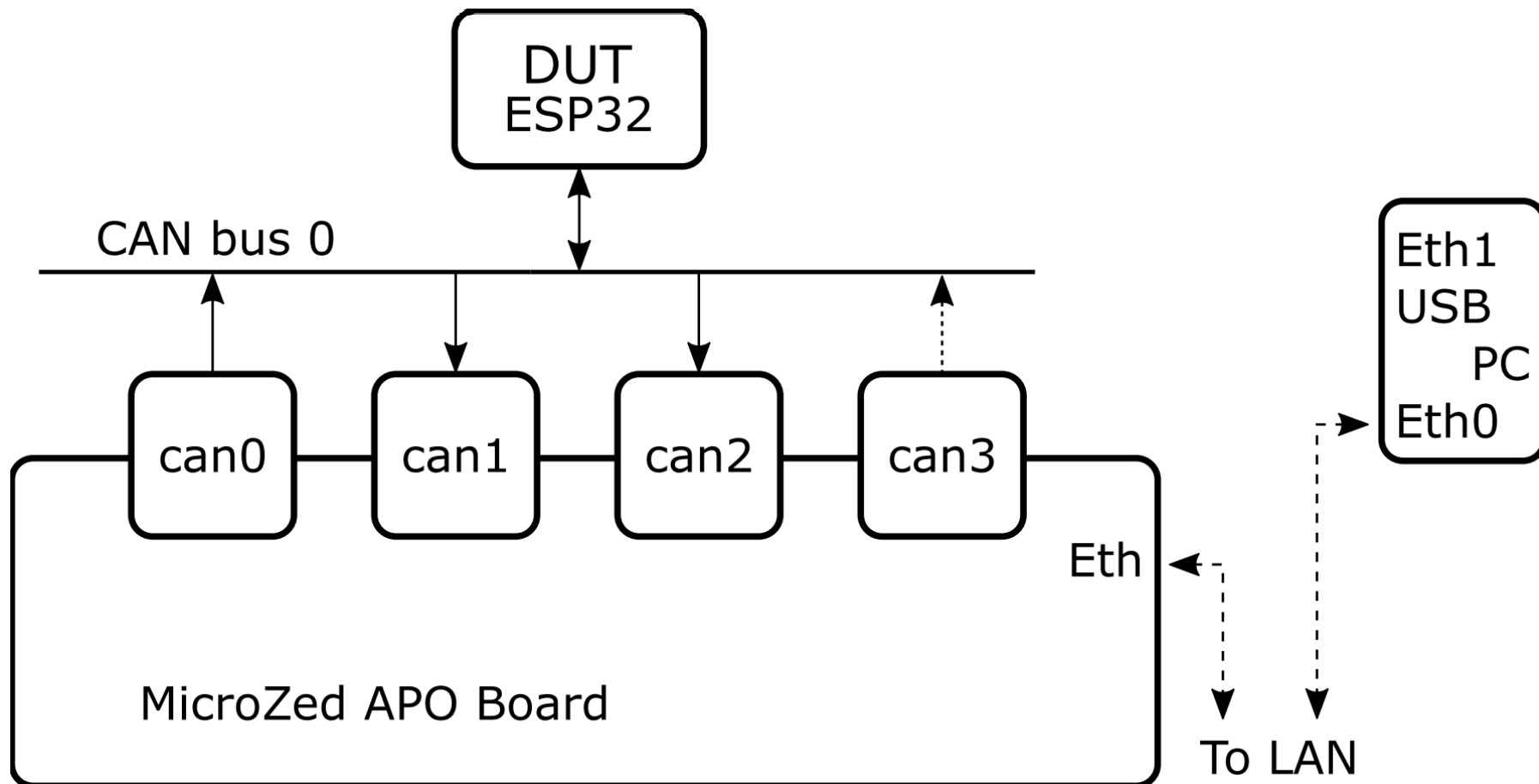
- **Bit timing**
- **Linux CAN utils**
- **Latency Tester**
  - **Timing Analysis of a Linux-Based CAN-to-CAN Gateway**
  - **Matěj Vasilevski contribution**
- **Motor control application in pysimCoder**



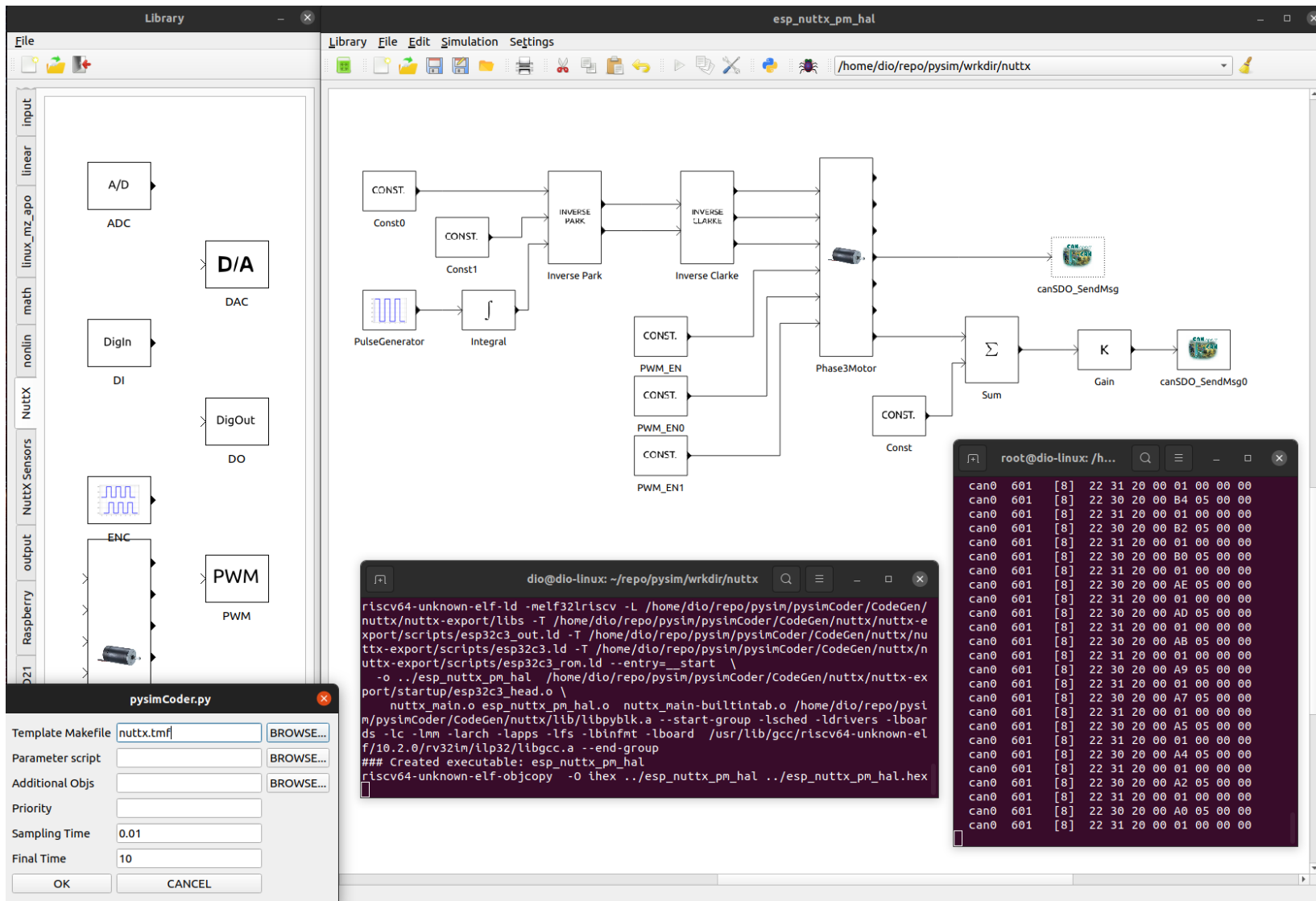
# Linux CAN utils



# Latency Tester



# Motor control in pysimCoder



The screenshot displays the pysimCoder environment for motor control simulation. The main workspace shows a block diagram of a motor control system. The diagram includes a Phase3Motor block, an Inverse Park block, an Inverse Clarke block, a Sum block, a Gain block, and a canSDO\_SendMsg block. The system is configured with various constants and a PulseGenerator.

A terminal window in the bottom left shows the compilation process for the motor control system:

```
riscv64-unknown-elf-ld -melf32lriscv -L /home/dio/repo/pysim/pysimCoder/CodeGen/nuttx/nuttx-export/libs -T /home/dio/repo/pysim/pysimCoder/CodeGen/nuttx/nuttx-export/scripts/esp32c3_out.ld -T /home/dio/repo/pysim/pysimCoder/CodeGen/nuttx/nuttx-export/scripts/esp32c3.ld -T /home/dio/repo/pysim/pysimCoder/CodeGen/nuttx/nuttx-export/scripts/esp32c3_rom.ld --entry=__start \
-o ../esp_nuttx_pm_hal /home/dio/repo/pysim/pysimCoder/CodeGen/nuttx/nuttx-export/startup/esp32c3_head.o \
nuttx_main.o esp_nuttx_pm_hal.o nuttx_main-builtintab.o /home/dio/repo/pysim/pysimCoder/CodeGen/nuttx/lib/libpyblk.a --start-group -lsched -ldrivers -lboard -lc -lm -lmm -larch -lapps -lfs -lbinfmt -lboard /usr/lib/gcc/riscv64-unknown-elf/10.2.0/rv32im/ilp32/libgcc.a --end-group
### Created executable: esp_nuttx_pm_hal
riscv64-unknown-elf-objcopy -O ihex ../esp_nuttx_pm_hal ../esp_nuttx_pm_hal.hex
```

A CAN bus data monitor window in the bottom right shows the following data:

can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	B4	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	B2	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	B0	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	AE	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	AD	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	AB	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	A9	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	A7	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	A5	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	A4	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	A2	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00
can0	601	[8]	22	30	20	00	A0	05	00	00
can0	601	[8]	22	31	20	00	01	00	00	00



# Conclusion

- **Driver integration completed**
- **TWAI (CAN) support has been added to NuttX mainline**
  - **<https://github.com/apache/incubator-nuttx/pull/6005>**
  - **At Espressif Alan C. Assis ported to Xtensa ESP32**
- **Successful demonstration of functionality**
- **Future work**
  - **Implement rest of IOCTL calls**
  - **Restore from overflow**



**ČVUT**

ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE

# **Thank you for your attention**

**Faculty of Electrical Engineering**

**Jan Charvát**

**16.6.2022**

