MasterSECIL_CANOPEN-ROS2-Jetson_2024-2025

NEXEYA
A Hensoldt Company.
Université Paul Sabatier - Toulouse III
Laboratoire IRIT
118 route de Narbonne
31062 Toulouse cedex 9

OFFRE DE PROJET M1/M2

[neOCampus] CANopen@stm32duino + ROS2@Nvidia

Contexte

Ce projet se déroulera dans le contexte des systèmes ambiants appliqués à l'opération neOCampus (http://neocampus.univ-tlse3.fr/wiki). Cette opération vise à doter le campus de l'Université Paul Sabatier d'une intelligence pervasive au service des utilisateurs. Pour cela, elle s'appuie sur un grand nombre de capteurs sans fil disséminés dans les bâtiments et sur des effecteurs pour piloter des équipements tels que volets roulants, ventouses magnétiques, luminaires etc.

Description

Lancée en 2013, l'opération neOCampus collecte de la donnée issue de capteurs disséminés sur le campus. Les points de collectes se présentent soit sous la forme d'un automate programmable[1] ou bien d'un *end-device* (e.g neOSensor[2]): dans les deux cas, les capteurs et/ou effecteurs y sont directement attachés. Cependant, lorsque ces capteurs / effecteurs se retrouvent disséminés à l'échelle d'un immeuble, la solution passe alors par la mise en œuvre d'un bus de terrain.

Nous vous proposons ici la mise en œuvre du **protocole CANopen** pour des communications entre un master (Jetson Nano) et des systèmes embarqués (stm32)

sur **bus CAN**. Ce type de bus est également présent dans le milieu des **V**éhicules **A**utonomes **C**onnectés (VACOP ---projet aut**OC**ampus) et de **trains** (Nexeya).

Pour aller un cran plus loin dans l'innovation, ce projet ambitionne de fusionner alimentation DC et transfert de données: ainsi, le câble DC (e.g 24v) va également servir au transport du protocole CAN via une modulation à définir.

Mise en oeuvre

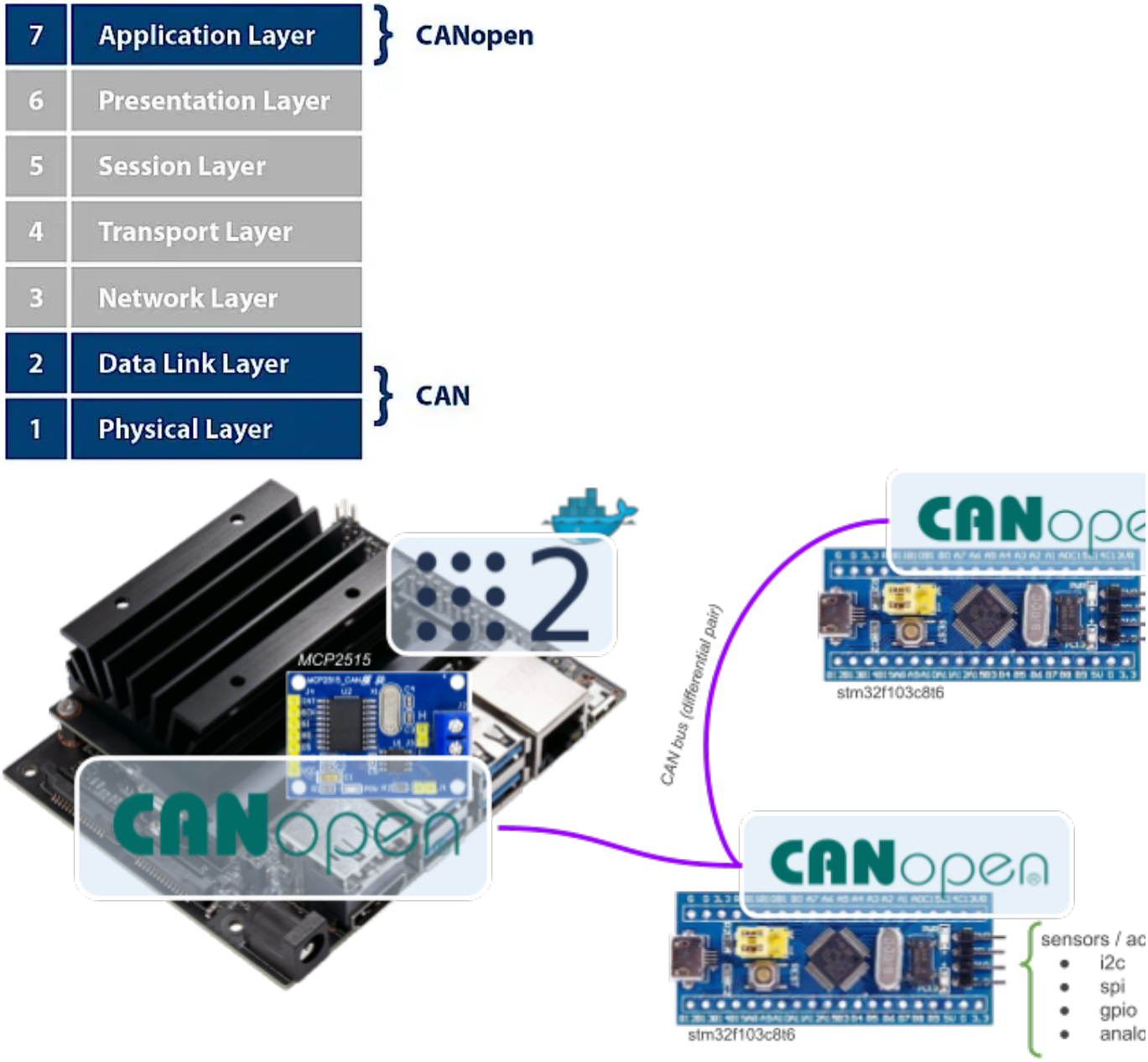
Le bus CAN utilise une paire torsadée comme médium de communication sur lequel transite des signaux différentiels. Sa vitesse de transmission va de 1Mbits/s sur qq dizaines de mètres à plusieurs kilomètres pour un débit de 10kbits/s.

[1] hardware

La première étape sera la remise en service du banc de test précédemment mis en œuvre lors d'une précédente itération du projet. Celui-ci se compose d'un module maître et d'un ou plusieurs module(s) esclave(s):

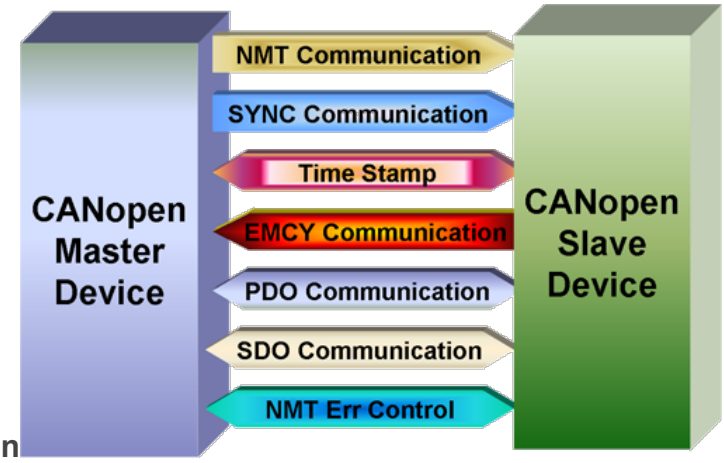
- [master] **Jetson Nano** couplée à un contrôleur CAN **MCP2515** via le bus **SPI**,
- [slaves] **stm32f103c8t6** + CAN transceiver.

Note: MCP2515 needs HW mods due to the SBC's GPIO not being 5v tolerant !



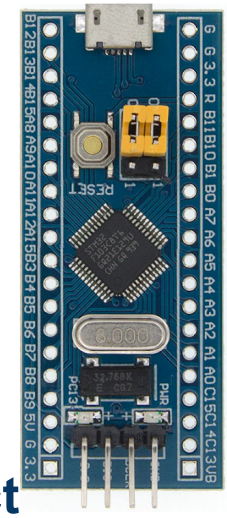
[2] stm32 embedded system + CAN transceiver
Les stm32f103c8t6 (*BluePill*) disposent nativement d'un contrôleur CAN auquel nous connectons un transceiver.
Vous développerez une application via l'IDE Arduino + stm32duino qui satisfasse aux exigences CANopen et qui permette de piloter les moteurs (intégration en cours).

[3] system
Afin de simplifier les choses, vous pourrez choisir de mettre en oeuvre une Jetson Nano ou un Raspberry Pi 4 du moment que ce dernier dispose de ROS2
A noter que le noyau Linux dispose nativement du support pour les contrôleurs CAN MCP2515.



[4] intégration
Vous écrirez ensuite une application python qui fera montre d'interactions entre le master du bus CAN et les esclaves.

[5][advanced] SWcan modulation over DC network
Le standard SWcan permet de connecter jusqu'à 32 end-devices à une vitesse entre 33 et 83kbits/s … amplement suffisant pour des objets connectés.
Dans un premier temps vous monterez un POC sur rail DIN avec un Jetson Nano un esclave stm32
Vous concevrez ensuite une modulation apte à transporter les signaux CAN sur une ligne DC d'un bâtiment



Contact

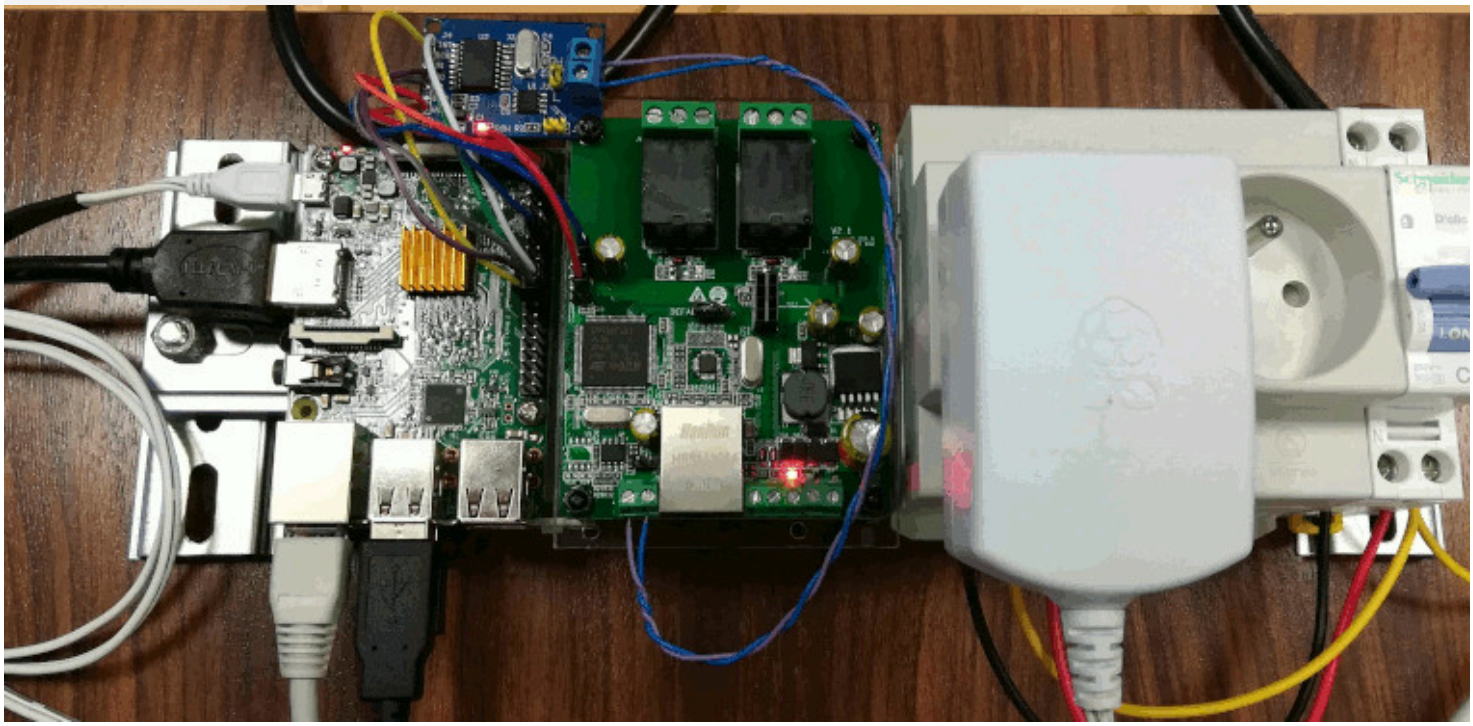
Dr. François Thiebolt thiebolt@irit.fr
IR Marie Bureau marie.bureau@irit.fr
[Nexeya / Hensoldt] M. Anthony Leib anthony.leib@hensoldt.fr

Références

Bluepill https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html
Raspberry Pi CAN bus testbed
https://www.beyondlogic.org/adding-can-controller-area-network-to-the-raspberry-pi/
MCP2515 CANbus module
https://www.electronicshub.org/arduino-mcp2515-can-bus-tutorial/
[neOCampus] VACOP-stm32-embedded project
https://gitlab.irit.fr/gis-neocampus/vacop/VACOP-stm32-embedded
CanOpenNode https://github.com/CANopenNode/CANopenNode
CAN library for stm32duino https://github.com/pazi88/STM32_CAN
Single Wire CAN bus
https://www.can-cia.org/can-knowledge/can/sae-j2411-single-wire/
https://www.beyondlogic.org/swcan-single-wire-can-transceiver-breakout-board/

Summary

Responsable : Marie Bureau marie.bureau@irit.fr & F.Thiebolt thiebolt@irit.fr
Contexte : PIA-MTI
Niveau : Master SECIL / Master SME
Dates : 2024-2025
Rémunération : *non applicable*
Keywords : ROS2, CANopen, Yocto Linux, Nvidia Jetson Nano, Python, docker, Arduino IDE, stm32duino, CAN bus, SPI bus, remote I/Os, Data over DC networks, KiCad PCB.



[1] **P**rogrammable **L**ogic **C**ontroller (PLC); e.g https://neocampus.univ-tlse3.fr/projects/concentrator
[2] ne**OS**ensor https://neocampus.univ-tlse3.fr/projects/neosensor