



Université Paul Sabatier - Toulouse III
Laboratoire IRIT
118 route de Narbonne
31062 Toulouse cedex 9

OFFRE DE PROJET M1/M2

neOCampus | CAN bus over DC network for Raspberry Pi

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

Depuis 2015, 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 programmable¹ ou bien d'un *end-device* (e.g neOSensor²): dans les deux cas, les capteurs et/ou effecteurs y sont directement attachés. Cependant lorsque ses capteurs / effecteurs sont disséminés à l'échelle d'un immeuble, la solution passe par la mise en œuvre d'un bus de terrain.

Nous vous proposons ici la mise en place d'un **bus CAN** sur **Raspberry Pi** auquel va être raccordé des modules d'entrées / sorties CAN. Ce type de bus est également présent dans le milieu des **Véhicules Autonomes Connectés** (VAC ---projet autOCampus).

Pour aller un cran plus loin dans l'innovation, ce projet ambitionne de fusionner la dimension alimentation DC et transfert de données: ainsi, le câble DC (e.g 24v) va également servir au transport du protocole CAN jusqu'au Raspberry Pi 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 mise en œuvre d'un banc de test. Ce dernier sera composé d'un module maître et d'un ou plusieurs module(s) esclave(s):

¹ Programmable Logic Controller (PLC); e.g <https://neocampus.univ-tlse3.fr/projects/concentrator>

² neOSensor <https://neocampus.univ-tlse3.fr/projects/neosensor>

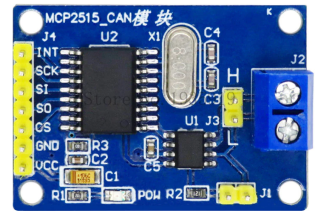
- [master] Raspberry Pi couplé à un contrôleur CAN **MCP2515** via le bus **SPI**,
- [slave] DingTian Ethernet Relay module featuring a CAN bus.

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

[2] system

Le noyau Linux dispose du support pour les contrôleurs CAN MCP2515. Notre module n'étant pas un HAT (i.e nous ne disposons pas de la eeprom sur le bus i2c system), il faudra configurer à la main le fichier `/boot/config.txt` (add `dtoverlay` ---voir références).

Vous écrirez ensuite une application python qui fera montre d'interactions entre le master du bus CAN et la carte DingTian.



[3] stm32 embedded system + CAN transceiver

Vous allez réaliser un PCB (KiCad) d'une carte basée sur un STM32F103c8t6 auquel vous ajouterez la conversion d'énergie (24VDC → 5v) et un transceiver CAN. Vous anticiperez également la mise en œuvre prochaine du protocole SWcan.

Sur le plan logiciel, vous concevrez une/des librairie(s) qui permette une découverte automatique des end-devices et qui puisse réaliser des opérations d'acquisition de données numériques ou analogiques.



[4] HomeAssistant integration

La carte DingTian fait état de deux sorties relais et de deux entrées. Vous allez configurer (ou créer les *bindings* selon) pour permettre à **Home Assistant** d'interagir avec cette carte. Vous ferez de même pour votre carte basé sur le STM32.

[5] 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 Raspberry Pi et votre carte STM32 faisant tous deux état de transceivers SWcan. Vous concevrez ensuite une modulation (e.g frequence avec deux porteuses pour les 1 et les 0) pour assurer les mêmes communications que précédemment.

Contact

Dr. François Thiebolt thiebolt@irit.fr
 Pr. Hugues Cassé casse@irit.fr
 Pr. Marie-Pierre Gleizes gleizes@irit.fr

Références

Raspberry Pi CAN bus testbed

<https://www.beyondlogic.org/adding-can-controller-area-network-to-the-raspberry-pi/>

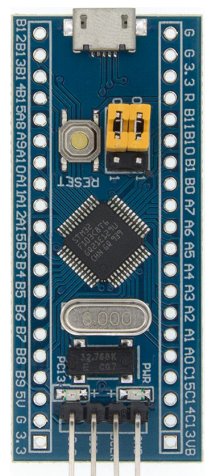
MCP2515 CANbus module

<https://www.electronicshub.org/arduino-mcp2515-can-bus-tutorial/>

DingTian Ethernet Relay module with CAN bus:

- <https://www.aliexpress.com/item/4000999069820.html>
- User manual and SDK ftp://ftp.dingtian-tech.com/relay_sdk.zip

The [neOCampus] canOCampus project <https://neocampus.univ-tlse3.fr/projects/canocampus>



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 : Dr Thiebolt François thiebolt@irit.fr

Contexte : Campus Ambient

Niveau : M1secil / M2siame

Dates : 2021-2022

Rémunération : *non applicable*

Keywords : Raspberry Pi, Python, Arduino IDE, CAN bus, SPI bus, remote I/Os, Data over DC networks, KiCad PCB.