

Double CANbus Interface on a Raspberry PI Guide

Ivan Martini @ EagleTRT

February 2021

1 Explanation

During the development of the new project (i.e. Fenice) the *low voltage team* agreed to step up the whole intra-communication architecture, adopting a dual can setup. Obviously this shift meant a whole redesign (which were already scheduled) for all the boards on the hardware side, while on the software side, there were the need for an more or less simple integration and update of the old setups. In particular, for some boards this operation is quite simple, while on Raspberry there is the need for some more tinkering.

By default, the Raspberry PI can work using two different CAN adapter (MCP2515 in our case), namely CAN0 and CAN1, but they can not work at the same time, as the interface is just one. Recapping, two adapter can be mounted, but only one at the time can be used. Since the interface is basically a SPI interface devoted to the CANbus, this setup allows the programmer to select which adapter to use just via software, using a the **Chip Select (CS)**.

Due to this type of configuration, a fast and reliable solution (idk if it is also the optimal one, but I didn't found any better) is the one of using a generic SPI interface (the SPI1) of the raspberry as another dual CAN interface (with both can2 and can3 adapters) in order to have a total of 4 adapters, with 2 interfaces, able to work at the same time. If i don't remember wrong, the shield developed by *Damiano* work using the CAN2, on the standard SPI1 ports (ask him or double check the board schematic):

- GPIO19(35) → SPI1_MISO
- GPIO20(38) → SPI1_MOSI
- GPIO21(40) → SPI1_SCLK
- GPIO16(36) → SPI1_CE2

In this guide there is the actual compiled overlays and the source codes for both CAN2 and CAN3 adapters. The source code are the `.dts` files, while the

compiled are the `.dtbo`. These last ones are the one required on the actual raspberry. If you ever have to handle and recompile the sources, the procedure is a little bit tricky but straightforward, follow this guide:

<https://learn.adafruit.com/introduction-to-the-beaglebone-black-device-tree/compiling-an-overlay>

2 Configuration

1. Copy the file `XX` into the folder `/boot/overlays` of the raspberry PI, after burning the image. Double check that this folder containing also the file `XX`.
2. Add the following code in the usual config file `/boot/config.txt`:

```
#standard CANbus enabling
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=22

#Enable the SPI1 to work with 1 chip select. Using more
#chip select allows to use more can interfaces, but not
#at the same time. We need just one, as we use the other
#standard can0 interface
dtoverlay=spi1-lcs
dtoverlay=mcp2515-can2,oscillator=16000000,interrupt=23
```