# UART Problem with multiple drivers

When connection multiple drivers you are able to read/write commands to the drivers individually using the same bidirectional line.

Strangely enough I could not figure out why the above configuration did not work.

Afbeelding met tekst

Automatisch gegenereerde beschrijvingI configured one driver to be 01 and the other 00 by connection the ms1 and ms2 to GND and VCC the way it was shown in figure 4.1 above. Somehow I always get this message: Afbeelding met tafel

Automatisch gegenereerde beschrijving

These are the configurations I tested with:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PDN UART | TMC software address settings | Result | Additional notes |
| TMC driver A | Connected to RX | 1 | Functional | Changing one address to 0 also makes it NOT functional |
| TMC driver B | Floating | 1 |  |  |
|  |  |  |  |  |
| TMC driver A | Floating | 0 | Functional |  |
| TMC driver B | Connected to RX | 0 |  |  |
|  |  |  |  |  |
| TMC driver A | Connected to RX | 0 | NOT Functional |  |
| TMC driver B | Floating | 1 |  |  |
|  |  |  |  |  |
| TMC driver A | Floating | 0 | NOT Functional |  |
| TMC driver B | Connected to RX | 1 |  |  |

First Conclusion/solution

The driver UART control only works if there is one driver connected to RX and assigned address to this driver is selected in software (according to the ms1 + ms2 settings you chose). You can however use 2 drivers if you go with the “more than 4 drivers solution” which Trinamic provided. This could be achieved with transistors as well I suppose.

If reading from the driver is not required, using the write only function as seen in figure 4.1 of the datasheet is a viable option. This way we only need to use n+1 pins of the raspberry pi to control the motors, where n increases with every driver added (enable pin). This would treat all steppers as the same and would control them using the enable pin and sending data over the RX pin.

# Further testing UART communication TMC2209

Configuration (test script 05: vactual.py) **RX and TX are flipped** **in the pictures:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Afbeelding met toetsenbord

Automatisch gegenereerde beschrijvingThis is what a successful transmission, with a few rotations and direction changes should look like.

Afbeelding met tekst, computer

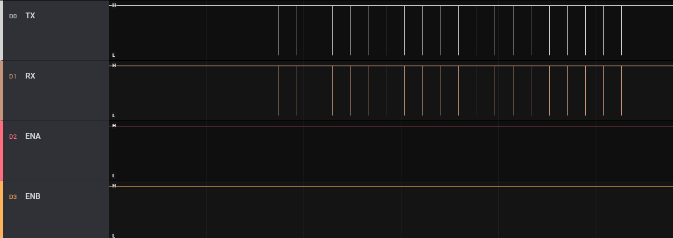
Automatisch gegenereerde beschrijvingWhen zoomed in it is clear that the TX deviates from the RX line data.

Further inspection shows that it does sometimes match the RX line

It is however always the case that TX continues sending after RX is done with a package.

It is however always the case that TX continues sending after RX is done with a package.

Using these settings (test script 06: multiple drivers.py) **RX and TX are flipped** **in the picture:**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

This is what a unsuccessful transmission looks like:

Afbeelding met tekst

Automatisch gegenereerde beschrijvingWhen zoomed in it seems that TX receives exactly the same information as it sends out. While, as stated earlier, it is expected that TX sends more data after RX is finished.

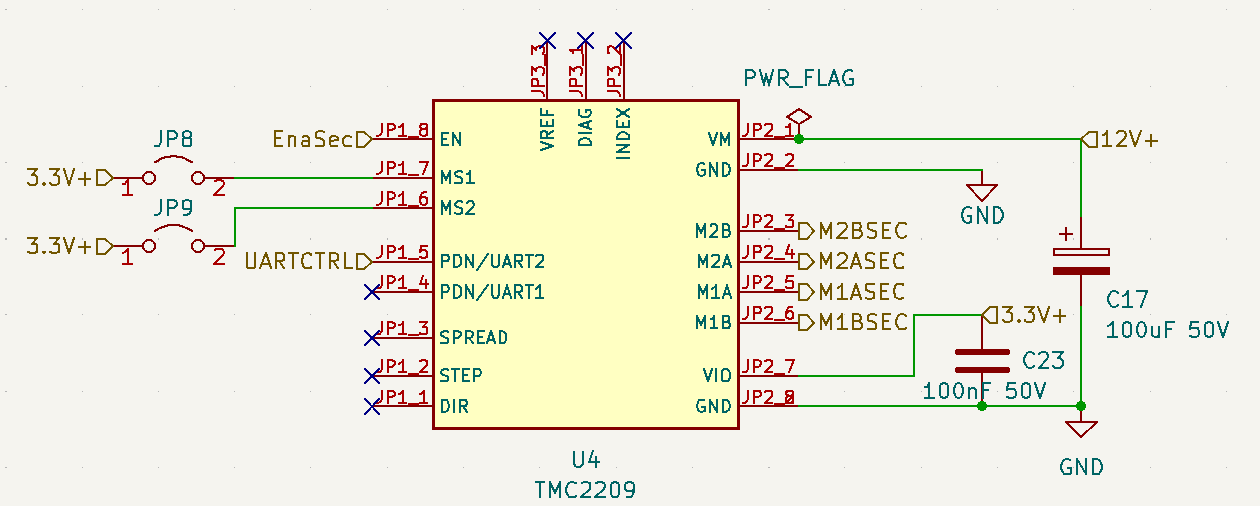
### 

### Second conclusion and solution for the UART error

I can conclude that somehow TX is getting the same information as it sends out on the RX line.

With the help of Thomas Ijsseldijk we found the cause and the fix for this problem. Lowering the TX resistor to 500ohms (from 1k) increases the voltage available for the IO of the TMC driver which results in the TMC being able to read the signal. It appears to be very important to choose the right resistor value. It is now possible to control the stepper motors via UART. This reduces the pins required on the raspberry pi with 7 pins (removing all step and dir pins)

The final configuration for the stepper motor drivers is as following:



We decided to go with the TMC2209 and we operate it in UART mode only. This means we need to connect to PDN UART and use one enable pin per driver to select it. The jumpers are there to select the driver address (up to 4 drivers in total).

### Final conclusions

After having spend days debugging the hardware; checking continuity, isolating components, swapping resistors values, and rebuilding my original test setup I can conclude that I never had a UART connection to begin with. What exactly happened when the motors started driving is not clear to me. This answer lays in the library.

What I do know is that the TMC2208 driver that we used (the old driver) had a solder jumper that was not soldered. A really strange decision by the manufacturer. When the solder connection was made, UART was functional. That was the only hardware problem relating UART. Now the software was where the real problem was.

We needed to disable the system service that initializes the modem, so it does not connect to the first PL011 (UART0; /dev/ttyAMA0). In the command line:

*$sudo systemctl disable hciuart*

And further we (only!) disabled the login shell on the serial port in the interfacing options of "sudo raspi-config", and rebooted the system.

When we started the UART test script for the tmc2209, we immediately got everything working. We than tested 2 drivers and got the same result:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

The extra resistors added to UART per driver was also proven to be a myth. 2 drivers can communicate perfectly fine with a 1K resistor only.