

# MCMCAN Lab

# Stage or Master Thesis proposal

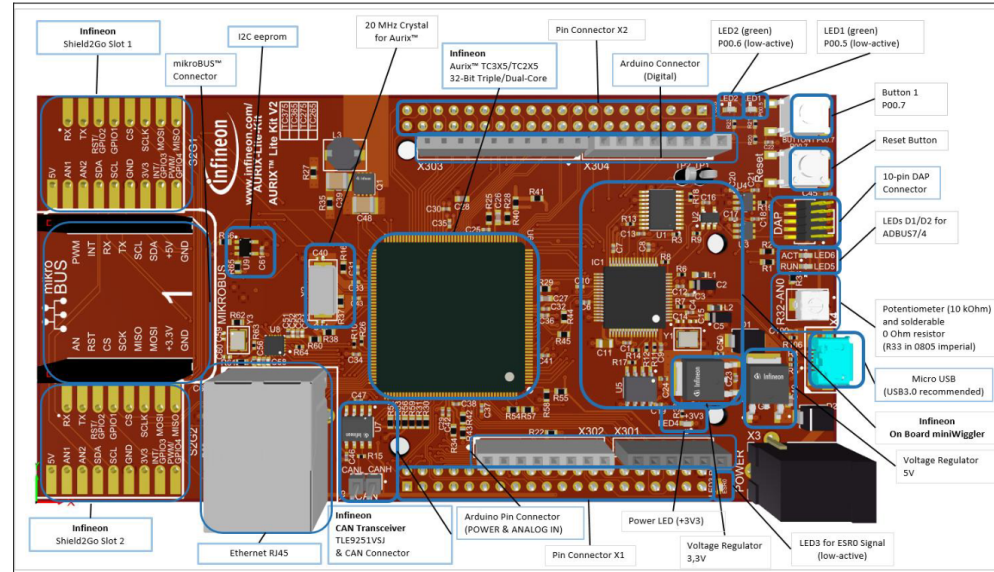
1. Embedded systems topics
  1. Rust vs C
  2. Developing driver in Rust for devices for cybersecurity (HSM of Infineon)
2. AI Topics
  1. LLM
  2. AI on embedded systems

If you are interested, we can talk about it after the lecture. Or contact me by e-mail.

# Today's steps

- MCMCAN Overview
- Launch MCAN\_1\_KIT\_TC375\_LK demo
- Modify the demo to start a communication between two boards
  - each board will be able to both send and receive messages.

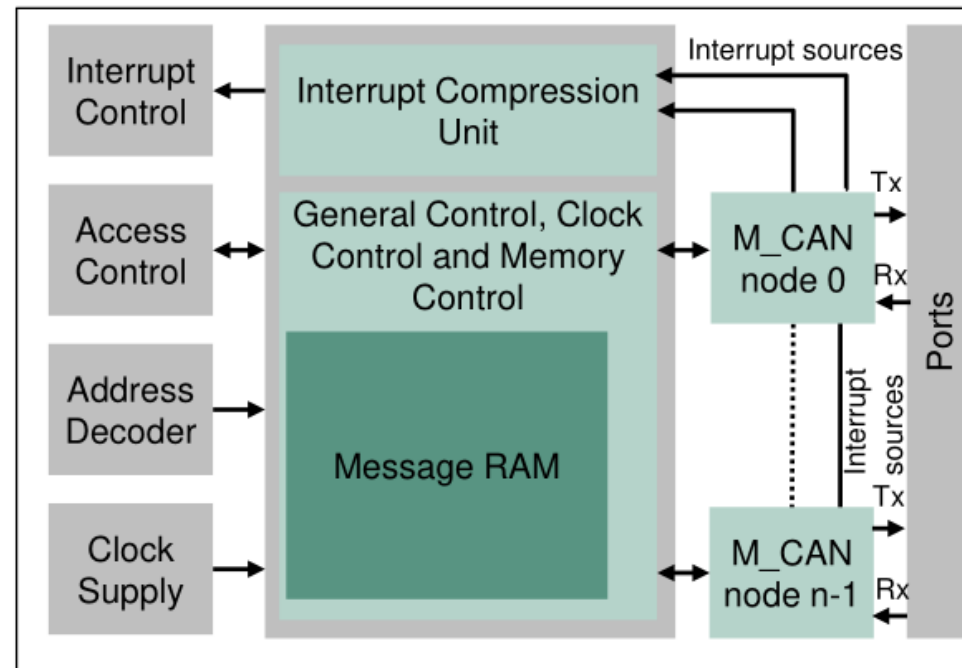
## What we are using



**Figure 2**      **AURIX™ lite Kit Board V2 View from the Top**

# Peripheral Overview

The MCMCAN peripheral on TC375 provides 2 CAN modules. Each of them provides 4 nodes.

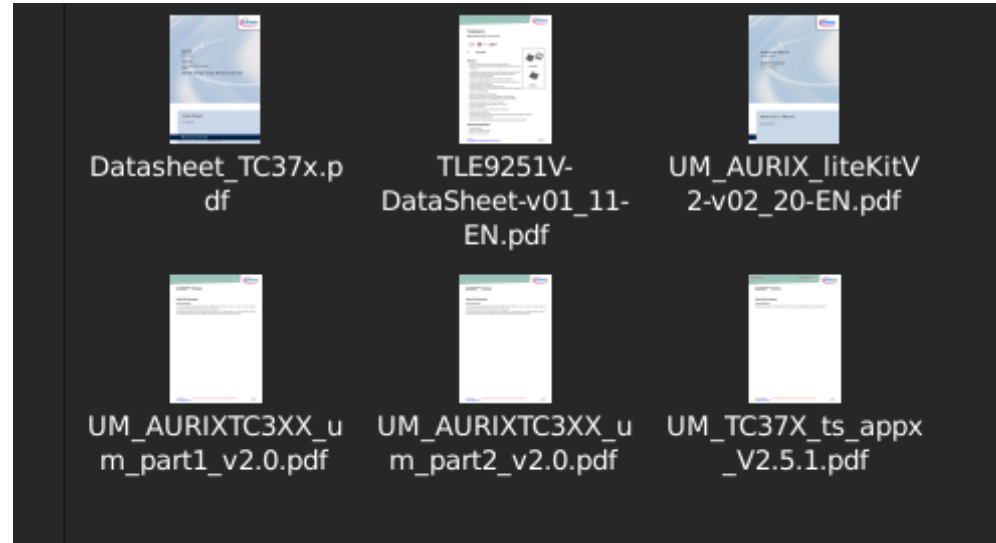


Infineon-AURIX\_TC3xx\_CAN\_Interface-Training

## Tons of information...

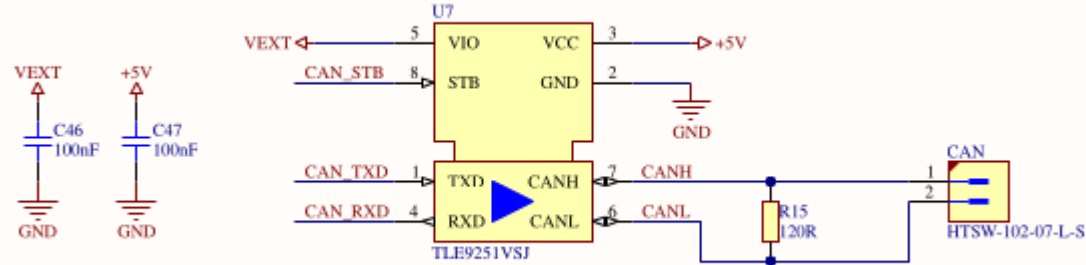
- Aurix TC3xx family user manuals
  - Part 1 and 2
- Aurix TC37x specific user manual
- Aurix TC37x datasheet
- Aurix lite Kit V2 user manual
- TLE9251V datasheet
  - the CAN transceiver we are using

Let's open all of them

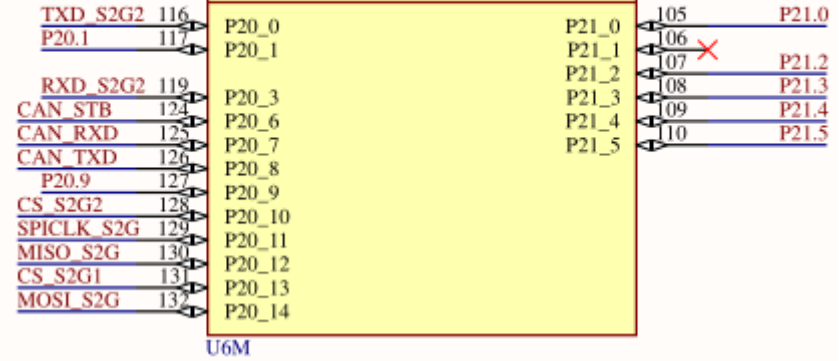


## CAN Transceiver (TLE9251V) and CAN pins

### CAN Transceiver



### Port 20 & 21 GTM/ASC3/HSCT/QSPI



# TLE states

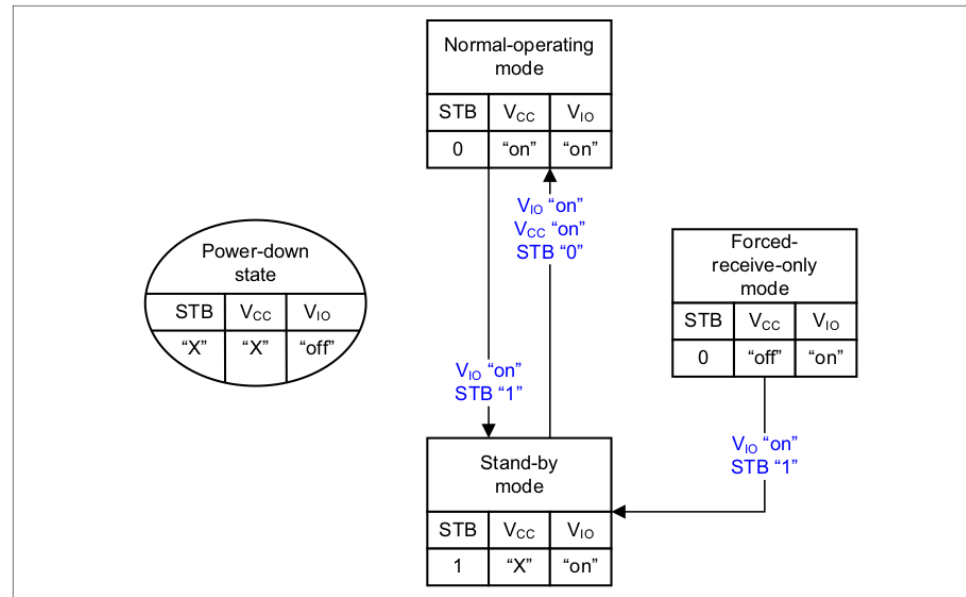


Figure 7 Mode selection by the STB pin



# We need to setup the STB PIN

1. The *Infineon Low Level Drivers* (iLLD) help us
2. Without them we would:
  1. identify the registers required to perform the necessary task.
  2. identify their memory position ("where" they are)
  3. understand which bits to manipulate and when.
    1. sometimes we may need to update more than one bit simultaneously
    2. or we might need to be sure that a bit has been correctly updated after a certain period of time.
  4. So we would need to study the needed sections of microcontroller's manual (Aurix TC3xx UM, Aurix TC37x UM)
3. iLLD do this job for us.
  1. Look at *IfxCan\_Can\_initModule* (and its inner functions *IfxCan\_isModuleEnabled*, *IfxCan\_enableModule*)

## Setup CAN STB pin to low for enable CAN transceiver

- define the GPIO port (MODULE\_P20)
- define the GPIO pin that is connected to the CAN\_STB (PIN6)
- define the general GPIO pin usage (no alternate function used)
- define the pad driver strength
- initialize the pin connected to CAN\_STB to level "LOW"
- set the pin output mode for pin connected to the CAN\_STB
- set the pad driver mode for pin connected to the CAN\_STB

### Hints

- Open *IfxPort.h* and try to understand which functions you need for:
  - Put the pin low
  - Configure P20.6 as output mode
  - Set the pad driver mode for pin connected to the CAN\_STB
- Then, open *IfxPort\_reg.h* and try to understand which parameter types are needed by the identified functions

## Configuration of the CAN node

In the actual example are configured two nodes one for the transmission and the other for the reception and connected together using the Loop-back mode.

Now we want to connect together two Lite Kit using a CAN-bus and for this one node is enough.

The next steps are:

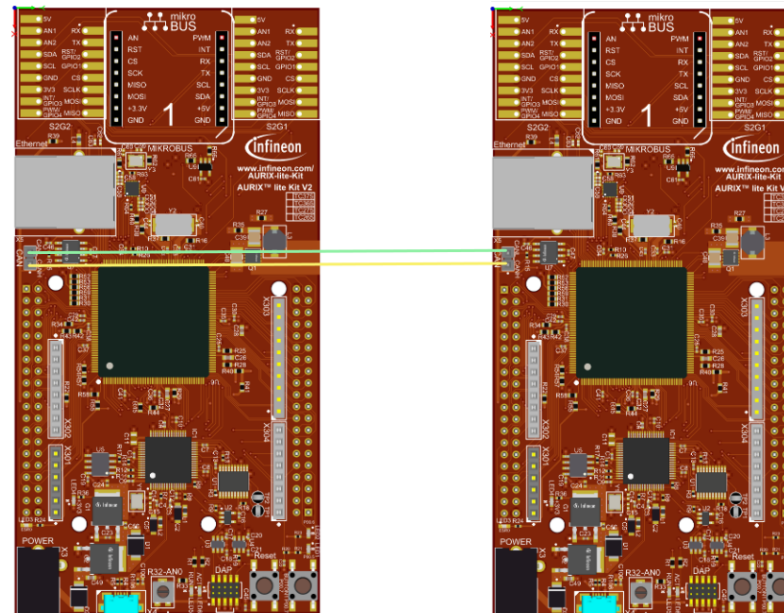
- remove the configuration of one node
  - fix the problems
- disable Loopback configuration
- configure baud-rate
- configure the node for transmit and receive

## Configuration of TX and RX pins

- enable the interrupts for transmission and reception on the same node
- setup TX and RX CAN pins
  - we have to deal with the *alternate PIN function*
- Hints:
  - Go back a little bit into the litekit UM
  - Go into the *datasheet* of the TC37x
  - ... and finally into the TC3xx UM and TC37x UM
  - look at the section *Node Port Control Register* and *Input/Output Function Selection in Ports* of TC3xx UM (MCMCAN chapter)
  - look at the TC37x UM (CAN Interface - connectivity section)

## Final steps

- add a periodical transmission
- connect the boards together
  - Before connecting the two boards, make sure they are switched OFF.
  - Before powering up the boards, CALL US to check that the connection has been made correctly.



CAN-H: green wire

CAN-L: yellow wire

**Thanks for listening!**

**Mattia Bottaro**

mattia.bottaro@bluewind.it

**Davide Ceccon**

davide.ceccon@bluewind.it

**<https://www.bluewind.it>**