ECE 554-002 Embedded Systems

Luis Castaneda-Trejo
University of Michigan-Dearborn
UM ID: 40872210
Date: 1/12/2022

LOW COST MINI HIL TEST TOOL

I. BACKGROUND

The Test and Validation platform team at *XYZ Automotive Systems* designs and develop automated test equipment for Engineering, Design and Product validation of several automotive ECUs. Most of the equipment consist in COTS (Commercial Off-the-Shelf) technology. Programs have budget to build 1 or 2 test systems at the most due to the high cost of the testers. One of the elements that significantly impact the overall cost of the tester is Vector CANoe and Vector's hardware. CANoe is used to communicate with the DUT (Device Under Test) and is the one in charge of executing the full test scenario. During product development, the DUT development teams need a tool to execute the same or similar tests that the tester executes during validation phases but due to the high cost of Vector CANoe licenses early engineering and robustness testing is not always executed, the teams are limited only to unit testing, creating bottlenecks due to software issues during development and most of the time they must wait until the final tester is available and ready to verify product issues.

II. GOAL

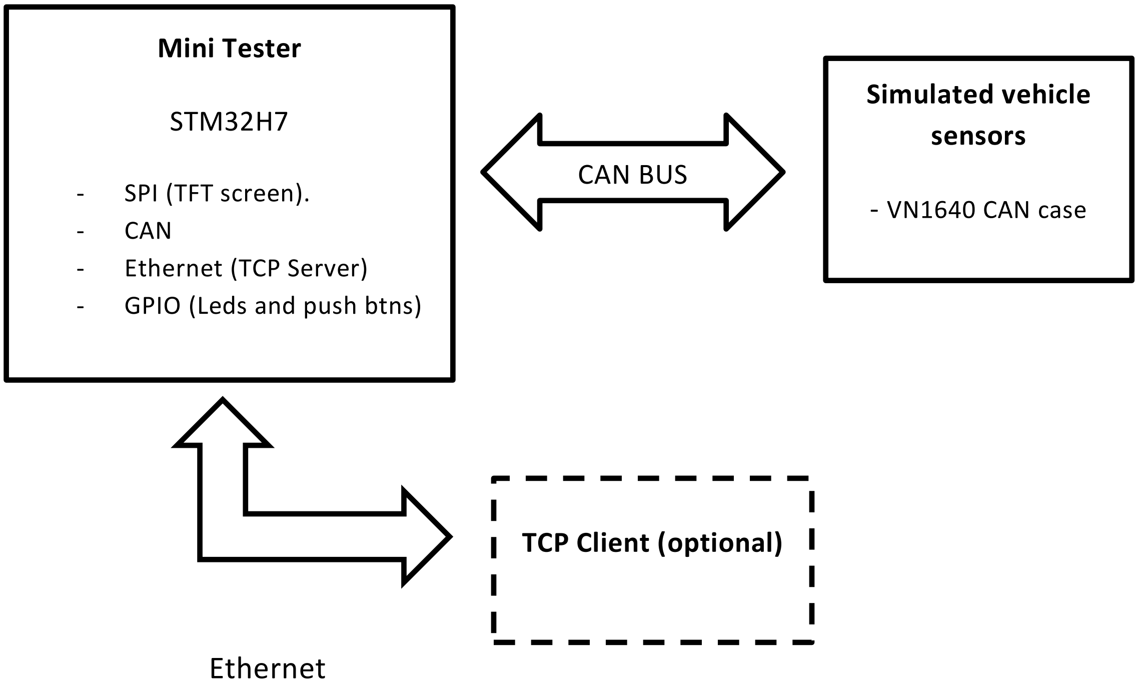
Design and development of a low-cost Real-Time embedded controller that has the capability of running essential/core tests via CAN to different vehicle ECUs. The mini HIL test tool will be able to run scheduled core tests periodically without the need of any special licenses or additional hardware once configured by the user via TCP/IP. A vehicle with a couple of sensors (Engine, Doors, Lights, Speed and RUN status) will be simulated using CANoe and CAPL scripting. Information and status of simulated sensors will be sent via CAN to the RT controller/tester when the tester executes the different test scenarios (scenarios are TBD). If time permits a simple TCP client will be developed using NI LabVIEW to send the configuration information to the mini tester, if time is not sufficient the configuration information will be hardcoded to the controller using a global library.

III. HARDWARE

- Microcontroller: STM32H7 (Cortex-M7) running FreeRTOS 10.3.1 with the following modules:
  - SPI module for TFT 1.8” screen.
  - CAN FD module configured for CAN HS.
  - Ethernet for TCP communication.
  - GPIO for user buttons and LEDs.

- CAN transceiver NXP TJA1441AT.
- Vector VN1640 CAN case to receive data from simulated vehicle sensors.

IV. PROJECT ELEMENTS


Mini Tester
STM32H7
- SPI (TFT screen).
- CAN
- Ethernet (TCP Server)
- GPIO (Leds and push btns)
CAN BUS
Simulated vehicle sensors
- VN1640 CAN case
TCP Client (optional)
Ethernet