

**LOW-COST HARDWARE IN THE LOOP (HIL) TEST TOOL**

**Project Report**

Document ID: Low-cost Hardware in The Loop Test Tool  
Origin Date: Sep 27, 2022  
Applicable to: ECE-574 Adv SW Techniques for Eng Applications Fall 2022

Student Name: Luis Castaneda-Trejo

E-mail: luisct@umich.edu

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version No.** | **Date** | **Details of Change** | **Modified by** |
| 0.1 | 12 Sep 2022 | Project Proposal. | Luis Castaneda-Trejo |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Introduction 4](#_Toc113973170)

[1.1 Concept 4](#_Toc113973171)

[1.2 Scope 4](#_Toc113973172)

[1.3 Learning Objectives 4](#_Toc113973173)

[Requirements 5](#_Toc113973174)

[2.1 Hardware 5](#_Toc113973177)

[2.2 Software 5](#_Toc113973178)

[Project Elements 6](#_Toc113973179)

[3.1 Hardware 6](#_Toc113973184)

[3.2 Software 7](#_Toc113973185)

# Introduction

Test and Validation teams across several automotive companies use COTS (Commercial Off-the-Shelf) technology for the design and development of Automated Test Equipment (ATE). System development programs budget high costs for professional development tools like Vector CANoe/CANalyzer, Vehicle Spy from ICS, etc.

While these tools are excellent to design and develop large simulations and tests scenarios, sometimes startup companies do not require or cannot afford these tools with high-cost licenses. On the other hand, when the test development team has finished the test automation and the rest of the R&D teams just need a simple tool to execute predefined test cases, they need to borrow at least a Runtime license of one of these expensive tools in order to run the test scenarios again to keep with OEM requirements for product maintenance.

## Concept

The purpose of this project is to emulate the functionality of an Automated Test Equipment (ATE) capable of running pre-defined test scenarios via CAN communication to special types of ECUs. For this academic project, a basic simulated Body Control Module will be simulated using Vector CANoe. The simulation will include vehicle engine status, Door locks, Lights and a temperature sensor.

The HIL Test Tool will consist of an STM32 MCU (NUCLEO-H723ZG). The microcontroller will be running FreeRTOS. It will contain the required tasks to interact with the Device Under Test (DUT) simulated in CANoe via CAN. It will receive instructions to run test and configuration modes from a simple TCP client.

## Scope

The HIL Test Tool will provide the ability to run predefined test scenarios to any ECU that has a CAN communication layer.

## Learning Objectives

* Apply the best software design practices for Software and Test Engineering.
* Have a state-of-the-art project template for the development of future engineering tools.
* Strengthen my ANSI C coding skills and use proper data structures.
* Strengthen my skills in CAN network simulation and knowledge with Vector tools (CANoe).
* Learn how to use the LWIP layer (Ethernet) for STM32 MCU.

.

# Requirements

This section describes the project requirements for software and hardware.



## Hardware

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Comment | Status |
| HW-001 | STM32 Development Board | Development board needs to have at least one CAN HS controller. | The STM32H7 MCU has 3 CAN FD controllers. The nucleo board has access to CAN1 (PD0 and PD1 pins). | TBD |
| HW-002 | CAN Transceiver | To be able to connect to a CAN network the development board needs to have a CAN transceiver. | The NXP TJA1441AT will be used to connect to a CAN network. The transceiver supports both CAN HS and FD modes. | TBD |
| HW-003 | Ethernet cable | CAT 5e or greater need to be connected to the ethernet port of the development board. |  | TBD |
| HW-004 | Bread board | Electrical breadboard needed to interconnect signals |  | TBD |
| HW-005 | CAN termination | Resistor of 120 ohm need to be added to pin 2 and 7 of the DB9 connector. |  | TBD |
| HW-006 | TFT Screen | A TFT screen should indicate the status of the test tool | A 2.6” TFT screen will be used. | TBD |
| HW-007 | USB CAN interface | A USB CAN interface is needed to simulate an independent ECU. | A Vector VN1640 will be used to provide CAN communication to the HIL Test Tool. | TBD |

## Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Description | Comment | Status |
| SW-001 | RTOS | An RTOS should be used as main OS for the development board | FreeRTOS will be used for the development board. | TBD |
| SW-002 | CAN bus speed | Simulated CAN network will run at 500 kbps | Speed will be calculated to achieve the required speed. | TBD |
| SW-003 | Serial COM | RT System must be able to provide logs for debugging | UART communication will be enabled. | TBD |
| SW-004 | Real Time response | RT System must start execution of the test scenarios in RealTime. |  | TBD |
| SW-005 | TCP communication | RT System must serve as a TCP server allowing other TCP clients to connect to it to send test scenarios. |  | TBD |
| SW-006 | ECU simulation | CANoe simulation serving as Device Under Test |  | TBD |
| SW-007 | TCP Client | A TCP client to interact with the HIL Test Tool |  | TBD |
| SW-008 | Test Mode 1 | Set of tests dedicated to test the lights via CAN. |  | TBD |
| SW-009 | Test Mode 2 | Set of tests dedicated to read temperature via CAN. |  | TBD |
| SW-010 | Test Mode 3 | Set of tests dedicated to read the locks. |  | TBD |
| SW-010 | Test Mode 4 | Set of tests dedicated to read engine status. |  | TBD |

# Project Elements

This section describes the parts of the project that will be used both in hardware and software.

Chart

Description automatically generated with low confidence

**Fig 1. Main Project Elements**



## Hardware – Mini HIL Tester Tool

The mini HIL test tool consists of an STM32 (Nucleo-H723ZG) microcontroller. The selected board has an Arm 32-bit Cortex M7 with 1M of Flash and 32Kb of RAM. The microcontroller has 3 CAN controllers supporting flexible data rate. The selected CAN interface is configured as CAN High speed (HS) only because the data that is going to be monitored does not require more than 8 bytes for payload.

To communicate with a CAN network, the TJA1441AT CAN transceiver from NXP was used. This transceiver supports up to 5 Mbit/s in FD mode. The configured speed for the CAN controller is 500 Kbit/s.

* MCU: [STM32H7](https://www.st.com/en/evaluation-tools/nucleo-h723zg.html#overview&secondary=st_all-features_sec-nav-tab) (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.

## 

## Software

* Vector CANoe 12.0
* National Instruments LabVIEW 2020.
* STMCubeIDE 1.10.1