Claims Investigation Committee Multi-Testing Input Device

ECE-4820: Electrical and Computer Engineering Design II

Dylan-Matthew Garza Daniel Baker Rohullah Sah

Department of Electrical and Computer Engineering
Western Michigan University

ZF Group
Auburn Hills, MI

Fall 2024

Faculty Advisor:
Dr. Janos Grantner

Sponsor Manager:
Patrick McNally

Table of Contents

ZF

What is ZF?

- Global technology company and Tier 1 automotive supplier
- Provides advanced safety systems and vehicle control solutions
- Partners with major OEMs: Daimler, Chrysler, Tesla, Waymo(Google), etc.
- A leading innovator in commercial vehicle technology

Figure 1: Source: google.com
ZF Group Office in Auburn Hills, MI

Project Background

- Claims Investigation Committee (CIC) required enhanced testing capabilities
- Focus on key component: Brake Signal Transmitter (BST)
- BST critical for highest volume commercial vehicle platform in North America (Daimler)
  - Daimler Truck AG - World's largest commercial vehicle manufacturer
  - Previous parent company of Mercedez Benz before splitting in 2021
- Need for rapid, accurate analysis of field returns

Need for Multi-Testing Input Device

Project Drivers

- BST implementation in Daimler's new platform drive urgent need
- Current testing methods are too time-consuming for production volumes
- Need for quick validation of warranty claims
- Opportunity to expand test capabilities to other electronic components

Key Devices Under Test (DUTs)

1. **Brake Signal Transmitter (BST)**
   - Primary focus - critical new component for 2025 production
   - Acts as the brain that reads how hard a driver presses the brake.
2. **Continuous Wear Sensor (CWS)**
   - Works like a monitor for your brake pads and discs
   - Warns when brakes are wearing down using voltage
3. **Pressure Sensor**
   - Continuously measures relative pressure in vehicle control systems
4. **Electronic Stability Control Module (ESCM)**
   - Acts as a safety system that helps prevent skidding and rollovers
   - Monitors the vehicle's movement and intervenes to keep it stable

Table of Contents

Project Specifications

What this project aims to accomplish:

1. **Device Interfacing**

   1.1 Properly read Device Signals using the ARM Cortex-M4 on the onboard microcontroller on the **STM32MP157F-DK2**:

   - PWM Duty Cycle
   - Frequency
   - Voltages through an analog-to-digital converter (ADC)
   - CAN frames

Project Specifications (cont.)

Project Specifications

2. **Physical Components and Hardware**
   - 2.1 Printed Circuit Board (PCB) for interfacing with DUT
   - 2.2 PCB for scaling and managing power for the DUT and to the microcontroller
   - 2.3 Enclosure for PCBs and **STM32MP157F-DK2** board

Project Specifications (cont.)

What this project aims to accomplish:

3. **Software**
   3.1 Custom embedded **Linux** distribution that will run on the onboard ARM Cortex-A7 microprocessor on the **STM32MP157F-DK2**
   3.2 Simple user interface web-based application
   3.3 Custom Webserver to process information from web application to microcontroller
   3.4 Communicate collected information from ARM Cortex-M4 to ARM Cortex-A7
   3.5 Ability to download measured data, formatted as a CSV, through the web application

Project Specifications and Overview

Gantt Chart

Budget Projection

Table of Contents

Custom Hardware Design

Table of Contents

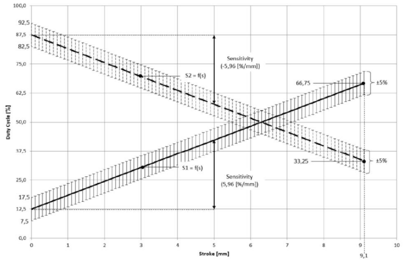
Firmware to Test Brake Signal Transmitter (BST)

Purpose

- Developed firmware on the onboard Cortex-M4 microcontroller to validate BST
- Ensures accurate detection of how hard a driver presses the brake pedal

Method

- **Input Capture:** Timers captures read two PWM signals from the BST
- **ADC Reading:**Optional string potentiometer for direct analog voltage measurements via ADC
- **Processing:** Calculates duty cycles, frequencies, and estimated stroke via timer interrupts
- **Validation:** Compare measurements against expected values according to product specifications to verify BST accuracy
- **Results:** Sends test results to the main processor for logging and user display



The nominal requirements, are fulfilled:

	S1 (PWM signal 1)	S2 (PWM signal 2)
Offset	12,5 % DC	87,5 % DC
Tolerance	± 5 % DC	
Sensitivity	5,96 % DC/mm	-5,96 % DC/mm
Frequency	200 ± 10 Hz	

Figure 2: *Product Specifications for BST*

Firmware to Test Continuous Wear Sensor (CWS)

Firmware to Test Pressure Sensor

Table of Contents

Embedded Linux

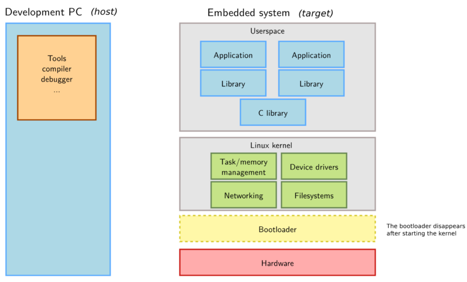


Figure 3: Source:https://bootlin.com/
Embedded Linux system architecture

Why use embedded Linux?

- Industry standard for any embedded operating system
- Access to open-source software (OSS) and tools
- Networking and connectivity made easy
- Easily save/access data with filesystem

Using The Yocto Project to Build a Custom Distribution

What is the Yocto Project and why?

- Most popular set of tools for embedded Linux Development
- Collection of OSS tools to make a custom Linux distribution
- Independent of target architecture
- **bitbake** build tool handles **metadata**
- **MetaData** can be in the form of
  - software build/patch instructions
  - configuration files for software
- **MetaData** organized in its **Layer Model**

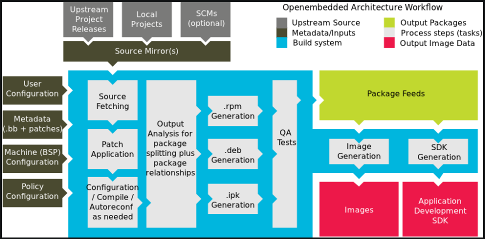


Figure 4: Source: https://docs.yoctoproject.org
High-level diagram representing how builds work using The Yocto Project

Custom Linux Image for the STM32MP1-DK2

What is used in the deployed image?

- ST's BSP (board support package) layer provides metadata
  - Hardware drivers
  - Kernel Configurations
  - Devicetree
- Custom layer **meta-zf-project**
  - **nginx** (webserver), **wpa_supplicant** (Wi-Fi access client/ IEEE 802.1X supplicant)
  - recipes for custom applications (Web application, Server, Cortex-M4 Firmware)
  - Kernel configurations and custom Devicetree

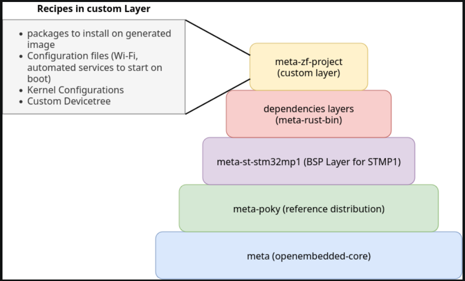


Figure 5: *Layer Model representation of this project for deploying onto a STM32MP1-DK2*

Table of Contents

Inter-Process Communication on a Heterogenous Architecture

With a heterogenous architecture (ARM Cortex-A7 and ARM Cortex-M4) how can information be shared?

Hetergenous multiprocessor SoCs cannot directly communicate

OpenAMP (Asymmetric Multi-Processing) Project

- Software framework that places standard protocol for shared memory
- Implemented on top of **virtio** framework
- STM provides **virt_uart** driver for recieving/transmitting messages over **RPMsg protocol**
- STMP1 layer automatically enables the **RPMSG tty driver** kernel module
  - creates file in Linux filesystem: **/dev/ttyRPMSG<X>**
  - can read and write to like a normal file
- **remoteproc** framework allows dynamic and remote loading of Cortex-M4 firmware
- **Resource Table** defined in firmware opens a trace in **/sys/kernel/debug/remoteproc/remoteproc0/trace0**
  - Used for logging measured data in CSV format

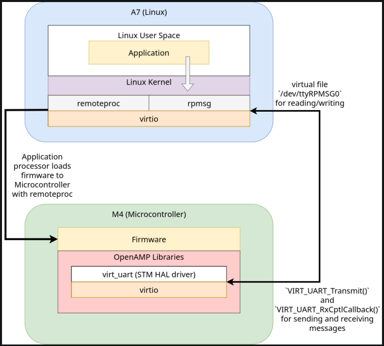


Figure 6: IPC between Linux (A7) and Microcontroller (M4)