VMS CAN Bus Sensor Array Capstone

Portland State University

Alice Ma
Braeden Hamson
Jade Nguyen
Pushpesh Sharma

VIKING MOTORSPORTS

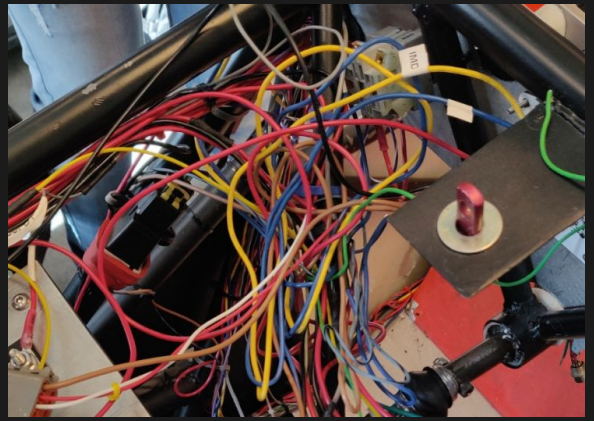
PORTLAND STATE UNIVERSITY
DOCTRINA URBI SERVIAT
1946

In This Presentation

- Project background
- CAN overview
- STM32 overview
- Configuring CAN on the STM
- CAN physical construction and integration

Problem

- Lack of system transparency leads to difficult debugging.
- Some testing has to be done near energized HV systems, by hand.
  (With HV gloves)
- Extremely short testing times and race times means no time can be wasted on debugging systems.
- A wealth of data are contained in our off the shelf system's CAN Bus's
  - System voltage, current, power, state of charge, battery temperature



Previous wiring harness

Project Goals

Increase the VMS team’s ability to locate electrical problems

Methods to address the main goal

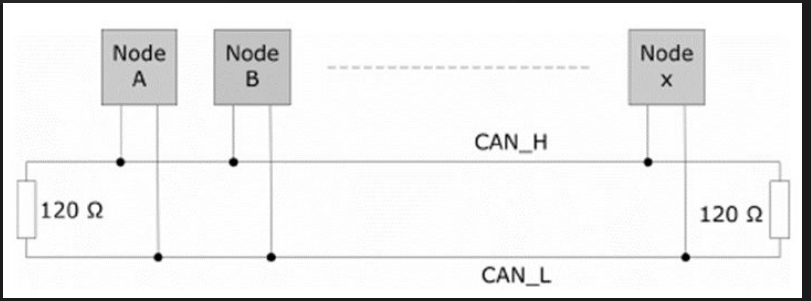
- Implement a CANBus
- Use STM32 microcontrollers
- Gather data and, important signals from key components of the car
  - Motor Controller
  - Battery Management System
  - Accumulator (Main HV Battery)
  - Pedal Monitoring Board
- Pass information from the CAN system to the driver’s display

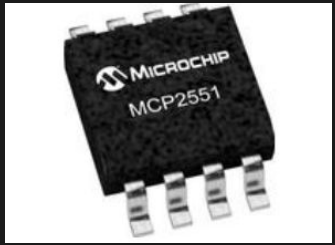
CAN Bus Overview

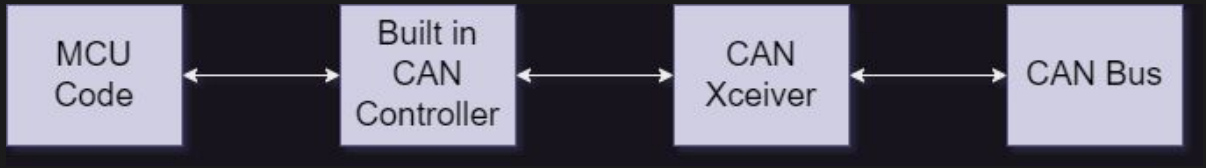
- **C**ontroller **A**rea **N**etwork bus
- CAN is a serial protocol, like morse code
- CAN does not have a master-slave configuration, there is no central controller
- High reliability, robust
- Simple but deterministic arbitration of bus
- Extended to applications beyond automotive (i.e. truck, tractor, …) and industrial automation
- Many protocols in use that tailor the network for specific needs

Physical Layer

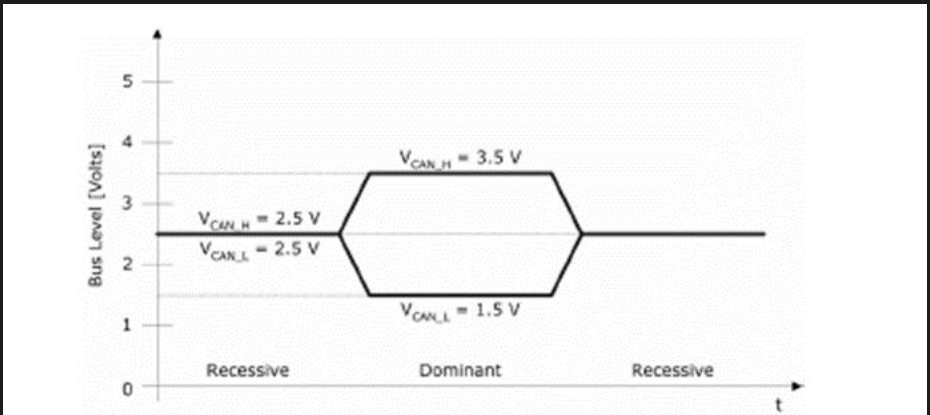
- 2 wire, differential signaling
- Speeds up to 1 Mb (length dependent)
- Nodes, stubs called out in various specs
- Transmission line, so requires termination at ends
- “Built in” prioritization, arbitration and error detection







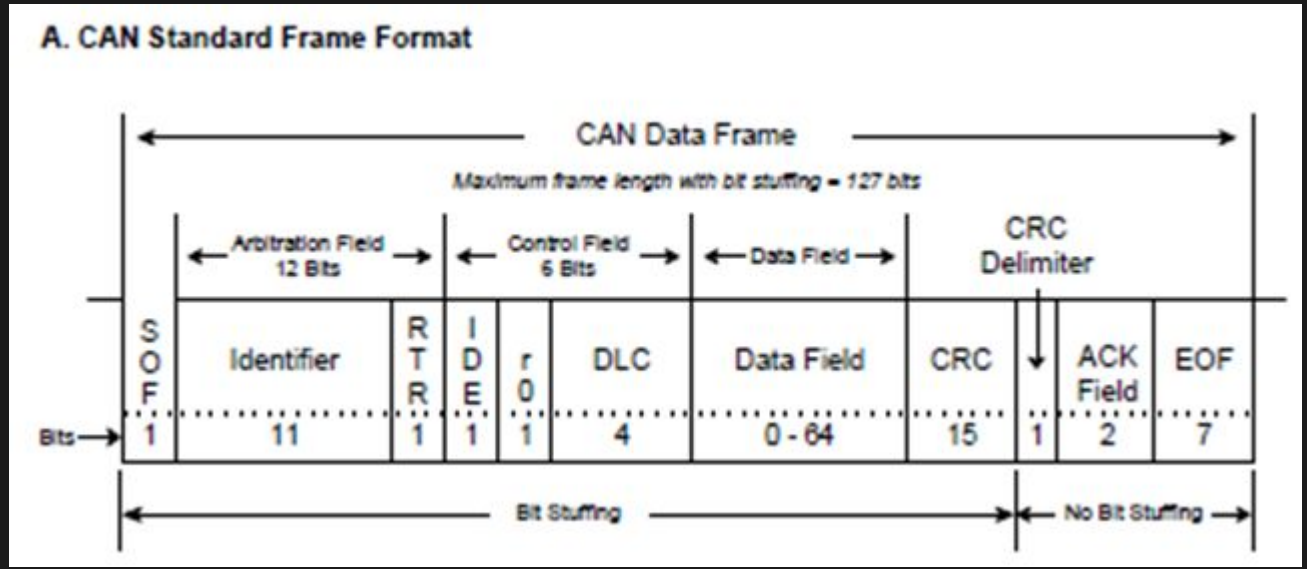
CAN - Physical



- Dominant – actively driven; Recessive – not driven

Identifiers and message frames

Original (standard): 11 bit ID and 8 byte message (data) frame, with additional bits (ref 1)

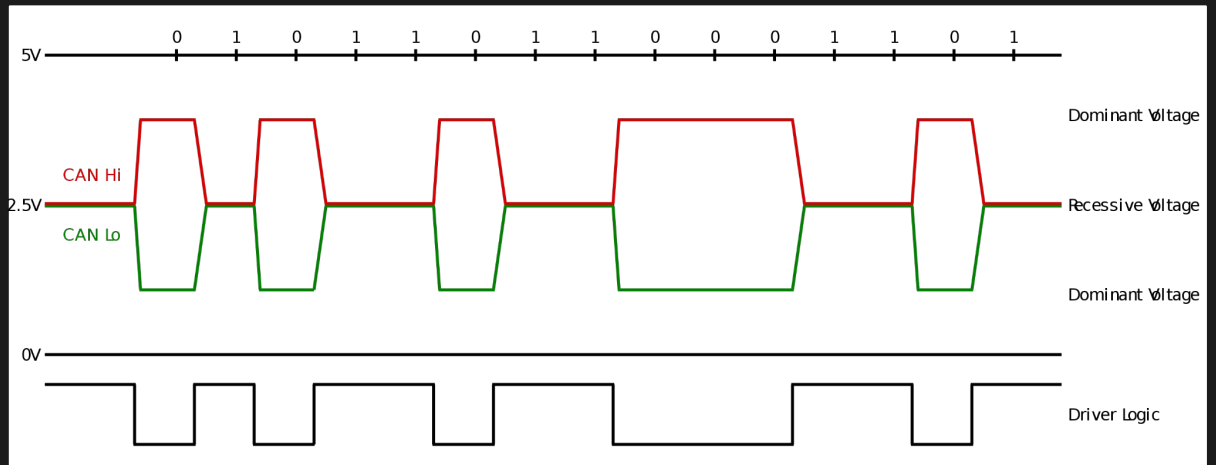


Address Priority

The address of each node denotes it’s priority.

Lower addresses have a higher priority

Lower addresses have more dominant bits, a dominant bit will overrule a recessive bit



CAN implementation on the car

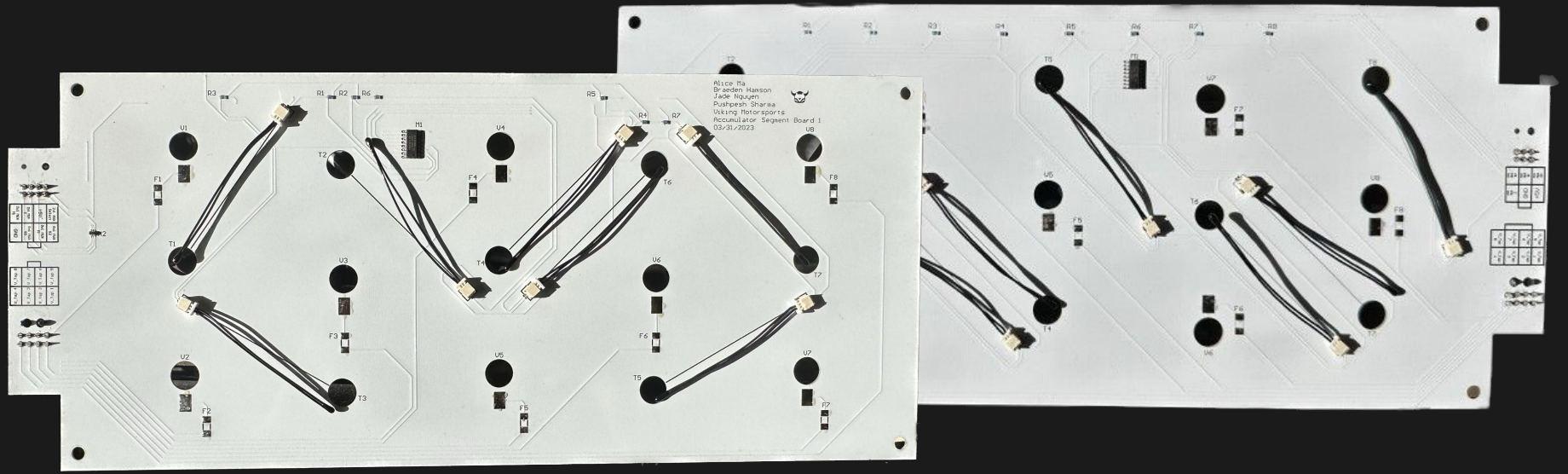
- Data collected from each node on the car and distributed via CAN Bus.
- Specific PCBs created to collect information send to the next node.
- Altium Designer 22.10.1 used to create schematics and PCB layouts as well as manufacturing gerber files.

CAN implementation on the car

- PCBs manufactured via JLC PCB and Oshpark
- PCBs then assembled using stencils when available
- Wiring harnesses created for all pinouts from schematics
- For testing purposes, the existing chassis was used to connect the boards and implement the CAN network

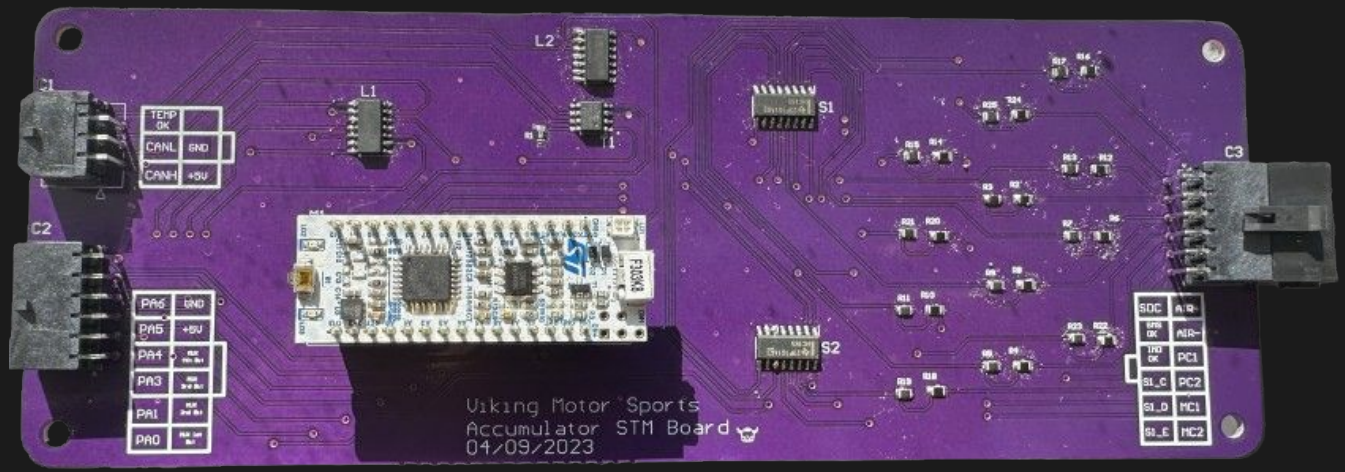
Battery Segment Boards

- Collect temperature data from batteries and send to Accumulator Main Board via GPIO pins.
- Collect Voltage data and send to Battery Management Systems



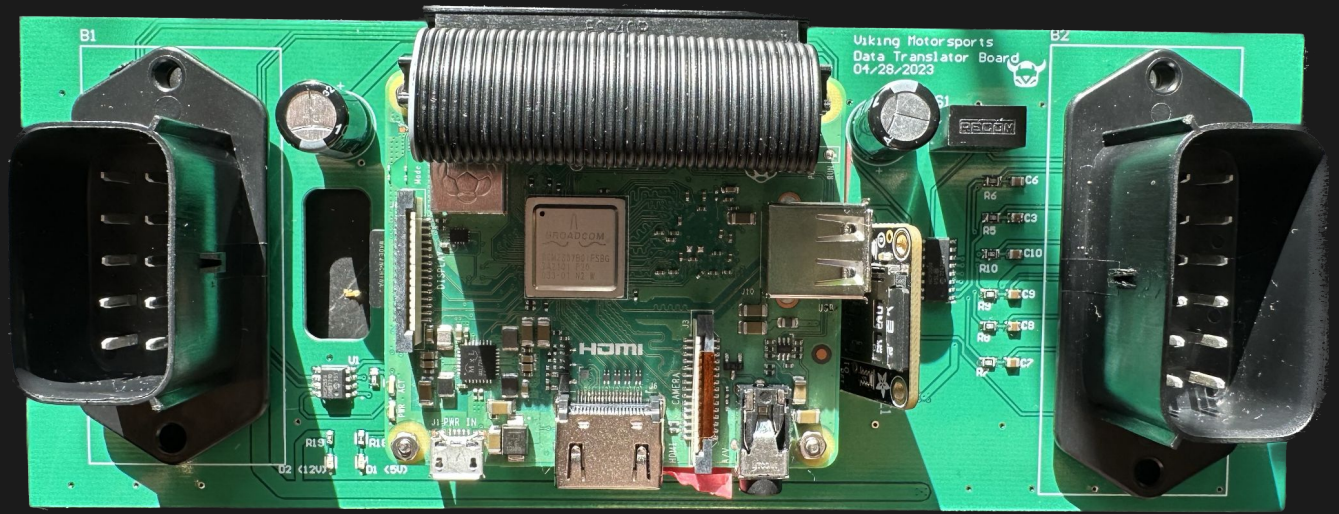
Accumulator Main Board

- Receives temperature data from the Battery Segment boards through GPIO pins
- Receives and decodes CAN messages from Battery Management Systems
- Formats and sends the data received through CAN bus to the Data Translator Board.



Data Translator Board

- Receives CAN messages from accumulator board, pedal controller board, motor controller, and the BMS.
- Formats collected data from CAN and send it to Raspberry Pi through UART protocol



Things that didn't go well with the boards

- Should have spent a longer amount of time double checking components, their placement, their footprints, and 3D layouts.
  - Inaccurate components caused missing features and necessary bodging with jumper wires.
- Should have spent a longer amount of time organizing parts ordering to ensure availability of all components for each board.
- Should have been prepared to have at least one board revision to detect and fix any mistakes.

Conclusion

CAN was successfully integrated into the existing chassis.

Data communication between each board was accurate.

Data is displayed onto screen successfully.

Our system is ready for any revisions and updates to be integrated into the future EV projects for the VMS team.

Questions?