

Software Documentation

Inverter Start-up Control

MAC EV MK IV

MAC Formula Electric

Software Team

Lin Fu

Last Revised: March 19, 2023

Table of Contents

[Key Contacts 3](#_Toc130135925)

[Purpose 3](#_Toc130135926)

[Design Criteria 3](#_Toc130135927)

[Design Overview 3](#_Toc130135928)

[High-Level Overview 3](#_Toc130135929)

[Hardware Requirements 4](#_Toc130135930)

[Related Resources 5](#_Toc130135931)

[Architecture Constraints 5](#_Toc130135932)

[Low-Level Design 5](#_Toc130135933)

[Building Block View 9](#_Toc130135934)

[User Interfaces 11](#_Toc130135935)

[Test Strategy 12](#_Toc130135936)

# 

# Key Contacts

**Project Coordinator:** Dharak Verma

**Programming Lead:** Lin Fu

**Testing Lead:** Lin Fu

# Purpose

The purpose of the Inverter Start-up Control is to allow testing and real-time debugging of the motor interface subsystem of the vehicle control system in Simulink, including the start-up and shutdown sequence of the AMK motor kit. This model is critical to the proper operation of the vehicle's powertrain.

# Design Criteria

The Simulink model should be:

* Simple
* Efficient
* Lightweight
* Modular

# Design Overview

The model utilizes the vehicle network toolbox in MATLAB, which receives output CAN messages from the AMK motor kit and passes them to the motor interface subsystem, it then sends CAN messages to the motor according to user command and the on-off diagram in the AMK documentation.

## High-Level Overview

Entire system:

Graphical user interface, diagram

Description automatically generated with medium confidence

Detailed view of the Control block:

Diagram, schematic

Description automatically generated

Motor Interface subsystem:

Diagram

Description automatically generated

## Hardware Requirements

CAN communication:

Kvaser USBcan Light 2xHS

Motor kit:

AMK RACING KIT 4-wheel drive "Formula Student Electric"

## Related Resources

There is a folder under **Software & Vehicle Controls > Files > AMK\_RACING\_KIT\_TRAINING\_EN\_V01\_2022**. Here you will find links and resources for things like AMK motor datasheets, as well as detailed overviews of the motor kit function.

# Architecture Constraints

Describe what constraints someone further developing this software should adhere to, and why. Should they not use tool x or operating system y. You can use the table as below, or just put a list.

|  |  |
| --- | --- |
| Constraint | Description |
| Kvaser USBcan Light 2xHS | The Kvaser CAN interface is essential for using the vehicle network toolbox in Simulink to send and receive CAN messages. Once all relevant blocks in Simulink have been configured correctly, the device will be plug-and-play |
| AMK RACING KIT 4-wheel drive "Formula Student Electric" | The AMK motor kit is what's being controlled in the model. The powertrain CAN needs to be connected to the Kvaser tool with proper termination for communication with the Simulink host computer |

Table 1: Hardware Constraints

|  |  |
| --- | --- |
| Constraint | Description |
| MATLAB/Simulink | The model was developed in Simulink. The following add-on packages in MATLAB needs to be installed for this model:   * Simulink * Stateflow * Vehicle Network Toolbox * Vehicle Network Toolbox Support Package for Kvaser CAN Devices |
| Windows | Kvaser tools does not currently support macOS, a Windows operating system is required for this application |

Table 2: Software Constraints

## Low-Level Design

AMK start-up and shutdown sequence:

Diagram

Description automatically generated

The sequence is represented by the Stateflow blocks in the motor interface subsystem:

Chart

Description automatically generated with medium confidenceDiagram

Description automatically generatedDiagram

Description automatically generated Diagram

Description automatically generated Diagram

Description automatically generated

## Building Block View

The AMK\_Input subsystem along with CAN Receive block takes the incoming CAN messages, and break them down into individual CAN signals according to the AMK DBC file using the CAN Unpack block. The signals are then fed into the Control subsystem for further processing.

Diagram

Description automatically generated

Diagram, schematic

Description automatically generated

The AMK\_Output subsystem takes the output from the Control subsystem and packs them into CAN messages according to the DBC file using the CAN Pack block, then send the CAN messages to the Inverter powertrain CAN using CAN Transmit block.

Diagram

Description automatically generated

Diagram

Description automatically generated with medium confidence

# User Interfaces

The shutdown command for the motors can be toggled by a manual switch in the Control subsystem. 1 for sending shutdown request, 0 for not sending shutdown request.Diagram

Description automatically generated

Diagram

Description automatically generated

The motor torque limit and speed request can be send when the motor is in AMK\_running state by toggling the manual switch in the Control subsystem.Diagram

Description automatically generated with low confidence

# Test Strategy

1. Connect Kvaser channel 1 to the powertrain CAN on the AMK motor kit
2. Plug in the Kvaser tool into computer before starting MATLAB, otherwise MATLAB may not recognize the device
3. Start MATLAB and open the model in Simulink

Double click the CAN Configuration block, in the Device drop-down menu, change the device to Kvaser USBcan Light channel 0, keep other parameters as defaultGraphical user interface, text, application

Description automatically generated

1. Perform the same action on CAN Receive and CAN Transmit block
2. Run the model, then turn on the AMK motor kit and wait the model to enter the AMK\_running state
3. The torque limit and speed request can then be changed by the user