CAN Kingdom on STM32F407

The document provides instructions to execute the ‘CAN Kingdom Communication on STM32F407’ project. The project is an implementation of Action/ Reaction pairs in KVaser CAN Kingdom on STM32F407 Discovery boards.

Requirements

The execution of the project has specific hardware and software requirements which are as follows.

Hardware Requirements

- STM32F4 Discovery Boards (2)
- KVaser LeafLight v2.0 and CAN Bus cable (connector) terminated with a resistor
- CAN Transceivers (2)
- Jumper wires

Software Requirements

- Visual Studio Code IDE with PlatformIO plugin installed
- KVaser CAN King

The reason for VIsual Studio Code IDE with PlatformIO plugin is that the project uses LibOpenCM3 libraries to transmit and receive CAN messages. The IDE is required for the support of the libraries.

Hardware Setup

The peripherals PB8 and PB9 on the STM32F4 Discovery board function as CAN peripherals for reception and transmission respectively. So, the CAN transceiver’s Rx pin needs to be connected to PB8 pin and the Tx pin needs to be connected to PB9 pin on the STM32F4 Discovery board.

While any CAN transceiver can be used, Waveshare CAN transceivers were used to perform the experiment. The schematic shown below can be followed to connect the Waveshare transceivers to the board and the boards to the KVaser LeafLight v2.0 device.

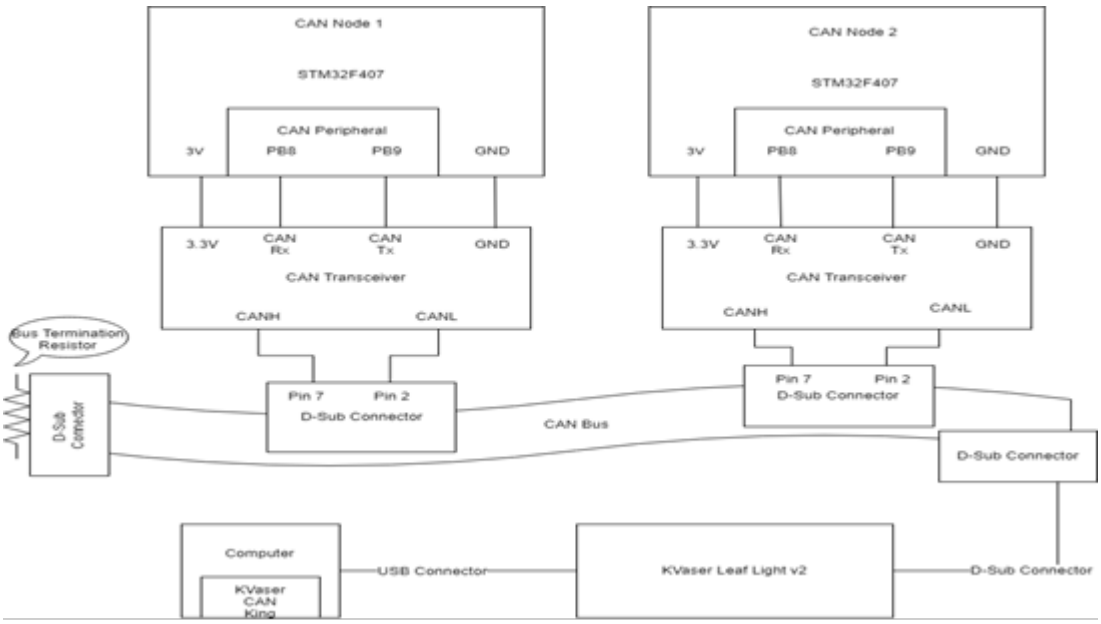


Figure 1: STM32F407 Discovery Board Connected to KVaser Leaf Light v2 for CAN Data Exchange

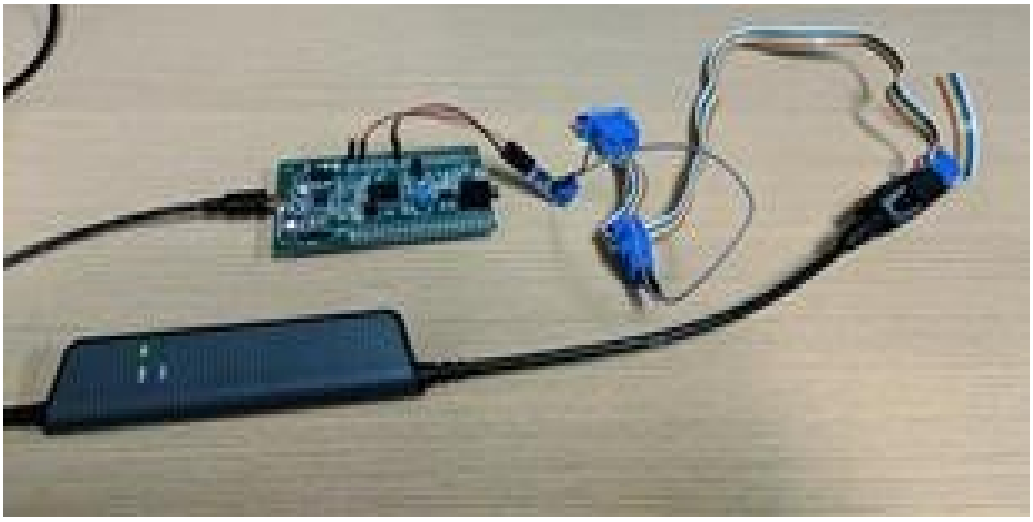


Figure 2: STM32F407 Discovery Board Connected to KVaser Leaf Light v2 for CAN Data Exchange

Software Setup

The code for the application is on the 'Master' branch in the project '**CAN Kingdom Communication on STM32F407** ' on the git repo: https://github.com/VenkatMargapuri/STM32F407-Code- Projects/tree/master/CAN%20Kingdom%20Communication%20on%20STM32F407

The git repo can be cloned and the code for the application can be opened on the Visual Studio Code IDE. The addition of PlatformIO plugin should give the ability to build the project and

deploy it onto the STM32F4 Discovery boards. The setup definitions for PlatformIO are built into the code already.

Since the code for both STM32F4 Discovery boards being used is the same, there is a minor commenting/ uncommenting that needs to be done before the code is executed on each of the devices (the code is documented to help understand the functionality it achieves). Follow the steps below before executing the code on each of the devices.

Step 1: Observe that the lines 61, 64 and 161 are commented out in code.

Step 2: Give the STM32F4 Discovery boards a label or an identifier to be able to recognize and distinguish between the boards since one of the boards will act as City 1 and the other board will act as City 2.

Step 3: On the board identified as City 1, uncomment line 61 (lines 64 and 167 are to remain commented) and deploy the code on the board using Visual Studio Code IDE.

Step 4: Upon deploying the code, comment line 61 again.

Step 5: On the board identified as City 2, uncomment lines 64 and 161 (line 61 is to remain commented) and deploy the code on the board using Visual Studio Code IDE.

Step 6: Upon deploying the code, comment lines 64 and 161 again.

At this point, the STM32F4 Discovery boards have been deployed with the code required for Action/ Reaction pair functionality using KVaser CAN Kingdom.

The next step is to launch KVaser CAN King on the computer and send a series of messages to the Discovery boards as if a King were sending them to the mayors. For the purposes of setting up Action/ Reaction pairs, King's pages 2, 5 and 16 are used where King's page 2 helps to assign an envelope to a folder, King's page 16 is used to place a document into a folder and King's page 5 is used to instruct a mayor to respond to a page in one document with a page in another document upon a receiving a CAN message.

For the purposes of this experiment, the following messages are sent from the KVaser CAN King to setup the Action/ Reaction pair.

Chn	Identifier	Flg	DLC	D0	1	2	3	4	5	6	D7	Time
0	0		8	0	0	0	0	0	0	0	0	11255.63
0	0		8	0	1	0	0	0	0	0	0	11260.24
0	0		8	0	2	0	0	0	0	0	3	11279.67
0	0		8	0	2	1	0	0	0	3	3	11288.30
0	0		8	0	16	3	136	240	0	1	0	11307.39
0	0		8	0	2	2	0	0	0	4	3	11331.50
0	0		8	0	16	4	136	240	0	1	0	11379.81
0	0		8	0	5	3	1	0	4	1	0	11395.67

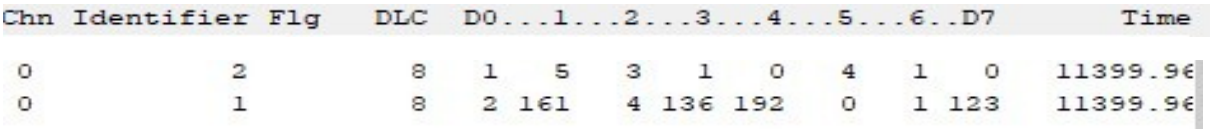
Figure 3: CAN Messages from KVaser CAN King to Cities

All the messages have the Identifier flag set to 0 to indicate to the mayors that the document being sent is that of the King's.

The last message which is the King's page 5 message indicates that upon receiving message with folder 3 and document 1, a response with folder 4 and document 1 needs to be sent.

Upon sending all the messages from the KVaser CAN King software which setup an action/ reaction pair, press the PA0 button (Blue button) on the STM32F4 Discovery board identified as City 2. Doing so sends a message to City 1 with folder 3 and document 1 in it. As indicated by the King to City 1, the mayor of the city sends a response with folder 4 and document 1 in it.

In order to realize that City 2 indeed receives the correct response, the 8th bit of the response is set to 123 by the mayor of City 1. Upon receiving a 123 in the 8th bit of the response, the board identified as City 2 toggles an LED as a visual indication. Messages being transmitted over the CAN bus can also be observed on the CAN King software's output window as proof that City 1 indeed transmits a response.



Chn	Identifier	Flg	DLC	D0	1	2	3	4	5	6	D7	Time
0	2		8	1	5	3	1	0	4	1	0	11399.96
0	1		8	2	161	4	136	192	0	1	123	11399.96

Figure 4: Action/ Reaction Documents exchanged between Cities