BTN415 CANBus Lab1

Introduction to CANBus Lab

In this lab, you will use the CanIF class to develop a piece of software that will transmit and receive standard Data Messages and Remote Terminal Requests.

LEARNING OUTCOMES

Upon successful completion of this lab, you will have demonstrated the ability to:

- Implement, compile and execute an Object Oriented C++ application
- Integrate and use a third party Software Development Kit vis Microsoft Visual Studio
- Generate and Transmit Data Messages and Remote Terminal Requests on a CAN Bus at 500kbps
- Receive and decode Data Messages on a CAN Bus at 500kbps

SPECIFICATIONS

Download the starting point from the course Github account, compile and verify the correct configuration for the ESD CANBus SDK.

- Instructions on how to accomplish this can be found in CanIF.h as a comment block.

void processSync(CanIF, bool&)*

The purpose of this function is to provide the logic for a thread that monitors the CANBus, reads messages and processes them. You need to implement the following capabilities:

a) Declare a CMSG Rx buffer that can receive CAN_BUFFER_SIZE messages
b) Loop until flag == false and perform the following tasks:
    a. Read the messages currently on the CANBus
    b. When a sync message from CANBus ID 0 is received, extract the time information and display it to the screen using std::cout

        NOTE: You should only print out 1 time stamp per second

    c. When a data message from CANBus ID 0 is received that is not a sync message, extract the information and display it to the screen using std::cout

        NOTE: You can assume the data will be larger than a sync message, the contents of the data frame contains character data and you should only display the message once

int main(int, char)*

The main function for this lab is incomplete. You job is to finish it by implementing the following requirements:

a) Using the CanNode object provided:
    a. Open a CAN Bus connection using the ID assigned to your physical ESD CANBus Controller
    b. Set the baud rate to 500bps
    c. Add the required IDs for monitoring the bus
b) Start a monitoring thread using processSync for the logic
c) Update the user input loop to transmit a Remote Terminal Request (RTR) message to CANBus ID 0

Running the Lab

In order to run this lab you require the following setup:

- A lab PC connected to an ESD USB CAN controller on a CANBus
- The MasterNode.exe running on a PC communicating on the CANBus using CAN ID 0
- Your software running on a PC communication on the CANBus using the CAN ID specified by the hardware connected to your computer

SUBMISSION REQUIREMENTS

Perform the following Demo to your instructor:

- Successful collection and display of the CANBus Sync Time Information (at once per second)
- Successful transmission of an RTR to CAN ID 0
- Successful reception of the CAN ID message in response to the RTR

Once you have completed your lab upload the following files:

- CANLab.cpp source file