**Communication Challenge 3: CAN Bus**



*By: Johnson Domacasse (#4471709)*

*Oktober 2023*

# Abstract:

Contents

[Abstract: 1](#_Toc149550676)

[Introduction: 3](#_Toc149550677)

[Procedure: 3](#_Toc149550678)

[Results: 3](#_Toc149550679)

[Conclusions: 3](#_Toc149550680)

[References: 3](#_Toc149550681)

# Introduction:

The focus of this document is to hold all of the information that was gathered and processed regarding the CAN bus. CAN stands for Controller Area Network. common digital data network used in automotive, industrial, medical and scientific systems. The CAN bus is used for routing sensor data between pieces of equipment. The main advantages are high resilience to noise, reliability, low cost, simple wiring and ease of use. The disadvantages are that the data packet lengths are small, transmission rates are low and the message transmission cycle time can vary. **[1]**

*NOTE: For the sake of this assignment, I will not be using the provided link as this is outdated. The only thing I will keep is the circuit that was used. Finally,* Due to us not having a soldered joystick, we will be implementing this section using a button and statmachines. They will simulate what needs to be done if we were to have a normal joystick.

## Procedure: Part 1:

In this section our primary goal is to make 2 CAN bus devices communicate with each other. From here we need to decide if we are going to use jumper J1. Once we have the communication ready, all we have to do is make the programs so that the sender can switch an LED on and off on the receiver side. This is done simply by typing in the serial monitor. We will also not be using J1 since we are working with a relatively small system with little to no noise.

For this we will not be needing 8 data bytes so 1 is sufficient. See figure 1 for the circuit.

A diagram of a circuit board

Description automatically generated

*Image 1: Can circuit.*

For this part, a simple state machine was developed to toggle an LED on and off based on the message that is received on the receiver side. See figures 2 and 3 for both state machines.

A diagram of a data flow

Description automatically generated  
*Figure 2: transmitter state machine.*

A diagram of a computer

Description automatically generated

*Figure 2: receiver state machine*

## Part 2:

Now that we have the basics of can down, we focus on how we can make use of this for a bigger system. In this case a system that resembles a car.

# Results:

# Conclusions:

# References:

**[1]** - *User requirement*. User Requirement - an overview | ScienceDirect Topics. (n.d.). <https://www.sciencedirect.com/topics/engineering/user-requirement>