SI4

CANopen vs Ethernet

By Dimitar Dyulgerov

Date: 12.05.2023

Abstract

The purpose of this report is to highlight the advantages and drawbacks of both protocols for communicating with the robot arm and defend the decision of choosing one over the other for the current project.

Table of contents

Introduction **3**
Procedure **4**
CANopen 4
Problems 4
Breakthroughs 4
Ethernet 5
Conclusion **6**

Introduction

The Igus robot arm has two ways of interfacing with the outside world - CAN and TCP/IP (Ethernet). The main goal of the assignment was to establish CANopen communication between the robot arm and a Beckhoff PLC, but there have been difficulties in getting the CANopen module working with the Beckhoff PLC, and getting any CAN commands working in general with the robot arm.

Procedure

CANopen

CAN is a communication protocol that is widely used in the industry. CANopen handles the layers from Application to Network, whereas CAN occupies the physical and datalink layers. CANopen research was started at the beginning of the project and was planned to be the protocol used for communication with the Igus robot arm (RL-D-RBT-5532S-BF). Three main problems occurred, however.

Problems

1. EL6751 setup

The initial issue I had with CANopen was setting up the EL6751 module as a CANopen device. Only debugging information is shown in the TwinCAT 3 IDE, without a way to map the inputs/outputs to variables in the program. No viable information was found regarding the issue.

2. Lack of up-to-date documentation regarding CANopen communication via EL6751

A lot of the documentation regarding the EL6751 found on Beckhoff's website was either irrelevant or out of date - using TwinCAT 2 examples. There is also a lack of actual examples. Most of the documentation is regarding the technical specifications of the module, not its use. As it is an industrial module, it is not used by the average person, therefore no third party examples could be found on the Internet.

3. Robot arm CAN responses

In order to test sending CAN commands to the robot arm, a C# application developed by the robot arm creators was used, along with a USB-To-CAN adapter (PCAN). Upon analysing the messages and responses, I came to a conclusion that the robot arm does not respond as it is meant to, according to its own documentation. Therefore, CAN is not a viable option for this project, given the timeframe available. I suspect there may be something wrong with the robot arm's controller, though I cannot be completely sure, as we do not have time to research the issue.

Breakthroughs

In spite of not being able to set up the EL6751 module for use with CANopen, and the robot arm not responding properly to CAN messages, we were able to set up the EL6751 module for CAN communication. The Received and Transmitted CAN message lists were successfully mapped in TwinCAT 3.
A new side project, as discussed with the client, is to try to send CAN messages to another CAN node (an Arduino with a CAN bus module, for example) and see if the proper messages are sent from the PLC. The other CAN node is to be used for debugging/displaying the messages in the future.

Ethernet

The Igus robot arm also supports TCP/IP communication via Ethernet. It uses a protocol called CRI, developed by the robot arm creators. We were able to successfully connect to the arm and control via a C# application, serving as a CRI client, provided by the creators of the arm.

The main challenge is finding out how to use the etherCAT ports on the PLC to send/receive data over ethernet. That is to be researched in Sprint 3.

One of the benefits of using CRI/Ethernet communication is the possibility of using the digital twin available which can communicate via CRI and can be used for testing without needing the physical hardware.

Conclusion

In conclusion, due to the lack of concise and proper documentation and the use of an unfamiliar IDE (TwinCAT 3), we came across a lot of unexpected problems that we had not accounted for. Therefore, we have decided to use Ethernet (TCP/IP) to communicate with the robot arm.