

# VarastoRobo – GoPiGo3 Communication

Simo Laakkonen, Joonas Mustonen TVT17SPL  
Oulu University of Applied Sciences  
Information Technology, Option of Device and Product Design

## Introduction

The aim of this project was to make an autonomous storage management system, in which multiple devices work in co-operation to transport ordered packages to a drop-off point. In addition a drone was to function as surveillance system for the premises.

## Objectives

We were responsible for the GoPiGo robot's two-way communication with master server in the VarastoRobo -project. GoPiGo robot (FIGURE 1.) was required to run python server in order to receive commands from the master server and act on them accordingly. Robot was also required to be able to send replies to the master server depending on the commands received from the master server.



FIGURE 1. GoPiGo3 -robot

## Methods

To form TCP connection between master server and GoPiGo, master is broadcasting itself with UDP-message (System Broadcast Message). After receiving the System Broadcast Message GoPiGo should be able to get IP-address from the received message. With the received IP-address robot can form a TCP-connection to the master server. After predetermined hand-shake (FIGURE 2.) with master, GoPiGo should be ready to receive commands and reply to them. Additionally after the forming of TCP-connection, GoPiGo still needs to listen and be able to receive UDP-messages in case of emergency stop message.

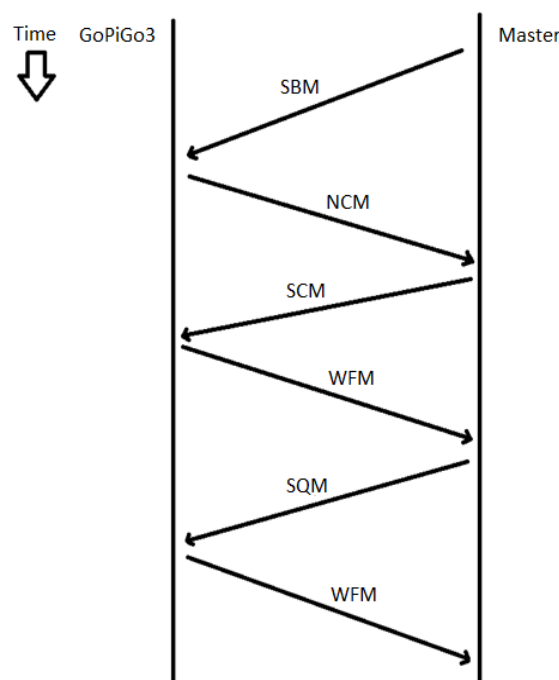


FIGURE 2. Handshake-protocol

## Results

Finished python client acts as a broker for GoPiGo robot and master server. Commands are received successfully from the master server and passed on (FIGURE 3). The client also gets replies from the robot when the commands given for it are fulfilled and then in turn sends reply messages to the master server.

```
pi@dex3: ~/Desktop/GoPiGo3_kootut
System Broadcast Message received
Master's ip: 192.168.1.33
New Connection Message replied
Setup Connection Message received: b'\x03\x02\x00\x00\x00\x04'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
Status Query Message received: b'\x06\x00\x00\x00\x00'
```

FIGURE 3. Terminal view of script running

## Conclusions

It was interesting to work with the project. Especially because it allowed us to gain experience at working with large project group and it brought out the importance of communication and sharing the information.

## References

1. Python 3.8.0 documentation. Python Software Foundation. Available: <https://docs.python.org/3/>
2. GitHub repository: <https://github.com/Jarno-Poikonen/VarastoRobo>