SDVUNx: Communications

There are several layers of communication between the various software and hardware components of the SDVUNx robots:

- Low-level communication protocols
- Internal software communication protocols
- Communication protocols with remote software

Low-level communication protocols

These protocols correspond to those used in PCBs, sensors and actuators. Knowledge of the operation at the hardware level is required: connectors, cables, voltages and pins. Being low level, the type of information, the format and the speed of transfer are key factors that must be considered when manipulating these protocols.

I2C protocol

This protocol consists of connecting a master device with a slave device, using two conductors: one for the clock signal and the other for sending data. Communication is unidirectional and based on requests and responses.

This protocol is used in the following PCBs

- Tiva VoltageCurrentMonitor
- Communication_PCB (SDVUN4)
- PCB Launchpad
- MSP Proximity Sensor
- FlexiforceLEDButton_PCB (SDVUN4)
- MotorDriver_PCB (SDVUN4)
- BatteryMonitor_PCB

USB2.0 and USB3.0

USB is a serial communication standard. In SDVUNx robots it is used for communication with peripherals that generate information at a high rate, such as the Tiva card or the ZED mini camera. USB 2.0 is the most used version and the simplest in terms of connectors: it uses two conductors for data transmission and reception. USB 3.0 uses more conductors to increase data transfer and requires devices to have USB 3.0-specific connectors mounted and the cable must also comply with the standard. In the case of the camera, the cable must be connected directly to the USB 3.0 port of the TX2 card in order to use the peripheral properly.

Ethernet

The Ethernet port is used in the robots to connect to the LiDar sensors (SDVUNx models 1, 2 and 3). This standard defines the type of cable and the speeds at which the information can be transmitted, along with the details of the datagrams.

Wifi

Wifi is a set of standards and protocols used to establish wireless communications between devices on the same network. In the SDVUNx robots, each on-board computer has a wireless network card that allows it to connect to the local network of the laboratory. Through this link, the robot can receive commands and obtain information from the Internet. The IP addresses are fixed for these robots and are configured on the lab access point:

- SDVUNx1: 192.168.1.11
- SDVUNx2: 192.168.1.12
- SDVUNx3: 192.168.1.13
- SDVUNx4: 192.168.1.14

Internal software communication protocols

The internal software of the SDVUNx robots complies with protocols that allow communication to be correct and punctual. The programs need to talk to each other: for this reason, the operating system is responsible for defining a PID identifier for each application that allows it to differentiate which programs generate data and which ones to redirect the information to. Since many programs use network protocols to send and receive information, network protocols such as logical ports or localhosts are also used.

Topics

ROS uses a communication standard between nodes that run simultaneously: topics. This standard defines who monitors the messages, that is, the rosmaster node, who sends the messages, known as the publisher node, and who listens to the messages, or listener node. Messages must have a clear data structure: ROS has a set of standard messages that is used in multiple parts of the Navigation Stack. There are also custom messages within the *sdv_un_ros* package for receiving custom sensor data or sending commands to the Tiva card.

Web sockets

In robots, Websockets are used to connect programs that are not part of ROS, such as PRIA, and thus be able to receive data within ROS from different sources, such as Web applications (SDV-Map-Viewer), mobile applications (UDP Android) and PRIA . Websockets define a bidirectional communication channel between two devices and use the HTML protocol for communication.

Communication protocols with remote software

HTTP

The HTTP protocol is used in the SDVUNx in the communication with PRIA and with SDV-Map-Viewer. It is a hidden protocol, there is not much interaction with it, since at the software level abstractions such as Websockets are used, which allow communication between remote devices without the need to deal with the particularities of the protocol.

JSON

Many of the messages that are sent by PRIA must use a plain text format that allows the information to be structured and that it reaches the client quickly. The most used is JSON: PRIA sends and receives informa-

tion in this format to and from Firebase. It is also used in communication between Websockets: topics are converted to JSON to be sent over RED when communication is established between the server and an SDV robot with the *sdv-nav-service* service.

Communications diagram