Modular and Self-Aware Bimodal Wearable Camera Platform

Student

Yanik Kuster

Initial Situation: In a past semester project, a portable camera platform has been built (Fig. 1). This platform consists of one or more cameras and a single board computer connected to two separate electronic boards. The battery board manages charging and discharging of two Li-ion cells to supply power to the platform. The interface board collects data from a ToF camera and an IMU, which is then forwarded to a suitable target destination. A 3D point cloud of known origin and orientation can be computed with these sensors. Based on the point cloud, obstacle detection could be implemented, which is the original motivation for this platform. The basic functionality of the boards was proven by a simple application within the initial semester project. However, there still have been only partially tested or untested features left.

Approach: The first goal of this project was to analyse the platform's capabilities and functionally verify them. The second goal was to design a software architecture. Finally, the third goal was to implement and verify said architecture on both boards. The features were tested while familiarising with the boards. Found errors were either fixed or bypassed. After that, the software architecture was defined with a particular focus on the platform's high modularity. For example, the boards do not have to be used together, each of them could also be controlled from a third-party device (e.g., a single board computer) instead. To facilitate the implementation for the third-party device, an instruction-based control scheme which uses Protobuf was defined. Protobuf is a tool that generates classes in many different target languages which serialise and deserialise data with a small metadata overhead. The system has been designed to be self-aware of its configurations for handling the hardware's modularity. Specifically, the boards monitor the state of their connectors and react if a state change (e.g., a disconnection) is detected. A typical reaction to state change is redirecting the dataflow to a new destination. The battery board has an Arm Cortex-M0 with a limited program memory, such that a bare-metal architecture is chosen, which is event-driven. The interface board on the other hand primarily controls dataflow. It uses FreeRTOS on an Arm Cortex-M4 due to its multiple communication interfaces being served simultaneously.

Result: Some minor routing errors have been found, and have been bypassed on the existing boards, but will be corrected in future platform versions. The startup of the ToF camera results in a high inrush current that leads to a considerable voltage drop on the supply rail, which might cause the microcontroller to reset. A dataflow model which controls the flow from reception to transmission was designed and verified in C++ (Fig. 3). The basic concept of this model is compatible with both boards and uses input and output buffers to decouple the serial interfaces from the CPU (Fig. 2). The core structure of the event-driven architecture of the battery board has been implemented and verified using Google Test. The dataflow model used by the battery board is from an early iteration and must be updated to enable external control. Configuration of the battery management chip still needs to be implemented. The interface however implements the current version of the dataflow model and its core functions have been verified. The configuration transition detection for enabling full self-awareness is still left out for future implementation.

Fig. 1: Camera platform with battery board supplying power and interface board collecting data from the ToF camera.
Own presentment

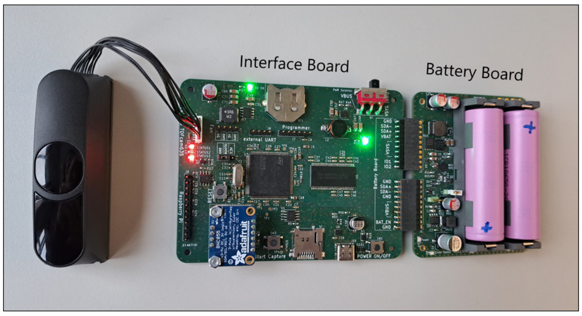


Fig. 2: Basic communication concept with input and output buffers. The messages are serialised using Protobuf.
Own presentment

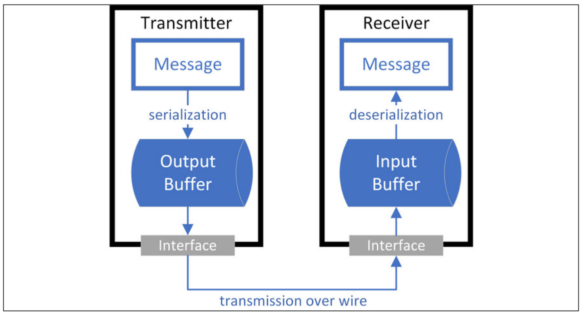
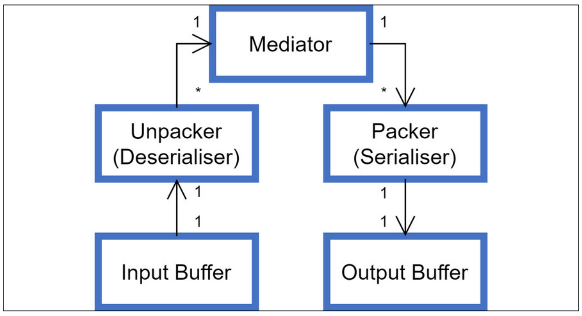


Fig. 3: Software dataflow model, which is used by both boards to transmit or receive messages.
Own presentment



Advisor
Prof. Dr. Andreas Breitenmoser

Subject Area
Electrical Engineering, Software and Systems, Sensor, Actuator and Communication Systems