

Eurobot 2018 manual for EE team

Sergei Vostrikov

October 5, 2018

Introduction

Eurobot is an international robotics competition for students. The team RE-set of Skoltech regularly participates in the contest since 2014. Each year on a basis of competition ISP lab selects new team members from the 1st year MSc students to form new team. In such conditions the key component of team's development is an effective transfer of knowledge between the generations. This document is a short guide for a STM 32 firmware that was developed by the Eurobot 2018 team. The paper is organised in a following way. In a first section the author makes an overview of used architecture to presenting breakdown structure of the firmware. In the following sections subsystems are described in a more detailed way.

1 Breakdown structure of the firmware

The breakdown structure of the firmware for STM32f4 microcontroller is represented in the figure 1.

The pattern of a superloop with interrupts is used to implement all necessary functionality. After power on all the peripheral modules are initialized at beginning of `(int main())`. Then inside a "while" loop (`main.c`) we can observe only functions that are responsible for two tasks:

- Communication with high level.
- Global shutdown.

All other tasks are implemented inside interrupt handlers of timer modules. They are:

- Movement control.
- Collision avoidance control.
- Task execution of servo manipulators.
- Global timer update.

In the next subsections each subsystem will be shortly described and mapped to concrete files where the functionality is implemented.

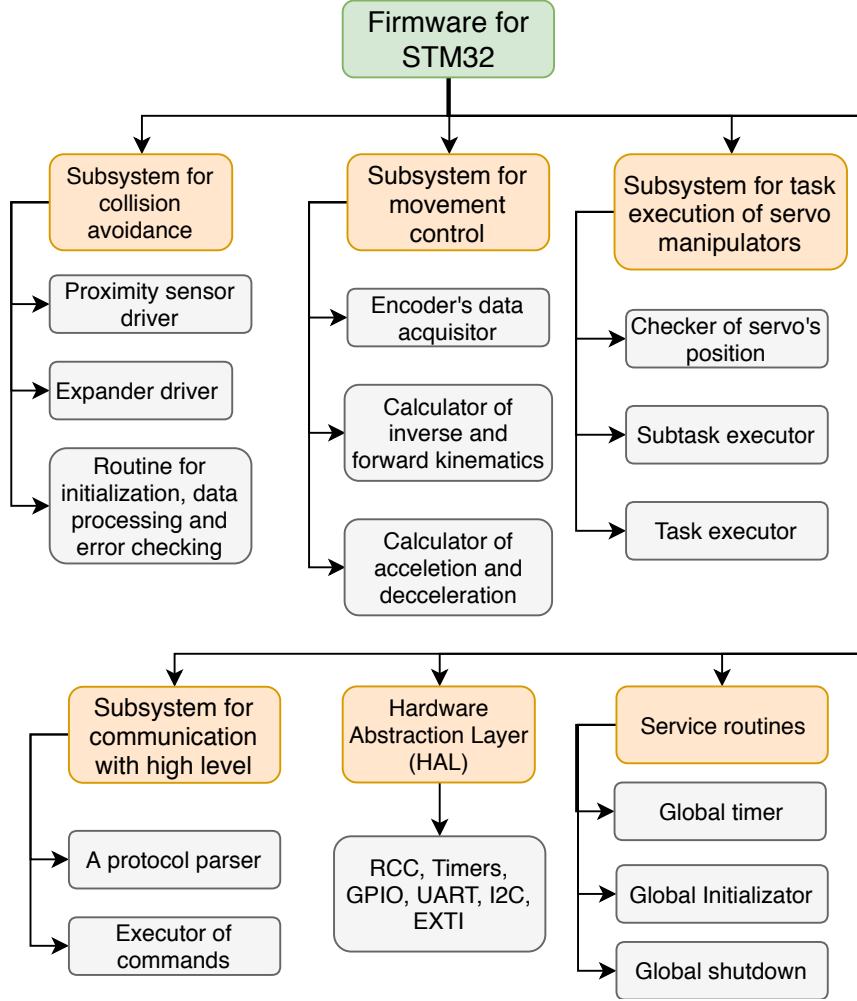


Figure 1: The breakdown structure.

1.1 Subsystem for collision avoidance

The collision avoidance system of the robot is based on a set of proximity sensors *VL6180X* from *ST* company. It is a time of flight (TOF) sensor that has an approximate precision (noise) of 2 mm and typical accuracy (range offset for surfaces with different reflectance) of about 13 mm . The sensor has digital I2C interface with a fixed I2C address that can be changed after power on. As we are using a set of sensor with the same initial address the subsystem also contains GPIO I2C expander *MCP23017*. The expander allows to perform subsequent initialization of sensor's array and local reinitialization of particular sensor in case of an error. The maximum frequency of sensor's data acquisition for one sensor depends on the reflectance of the target, but

it could be configured with constrained convergence time (see the datasheet for *VL6180X*). In the firmware a custom method of sensor's initialization is used that allows to achieve $50Hz$ data acquisition frequency and reduce the sensitivity cone for the proximity sensors.

The source code files for *VL6180X* driver:

- *VL6180x.h*
- *VL6180x.c*

The source code files for *MCP23017* driver and other subroutines:

- *Collision_avoidance.h*
- *Collision_avoidance.c*

Periodical data acquisition of ranges is implemented in the interrupt handler of a timer called *COLL_AVOID_TIM_MODULE*. The code of the handler is located in *Interrupts.c*.

1.2 Subsystem for movement control

1.3 Subsystem for task execution of servo manipulators

1.4 Subsystem for communication with high level

1.5 Service Routines

1.6 Hardware abstraction layer (HAL)