AGH University of Science and Technology

Metodyki projektowania i modelowania systemów 1

Remotely controlled vehicle with Bluetooth control and obstacle avoidance - documentation

Author: Jakub Domin
lecturer: prof. dr hab. inż. Bogusław Cyganek

Monday 20th May, 2024

1 Introduction

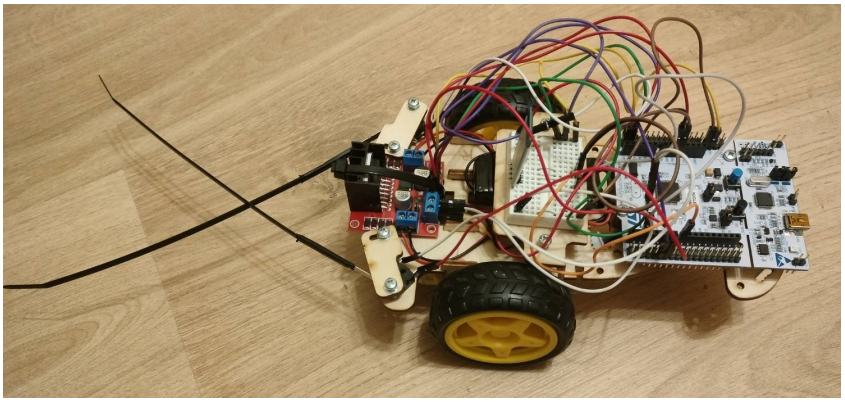

Figure 1: vehicle image

This project is a remotelly controlled vehicle with bluetooth connection and obstacle avoidance. It features two modes of operation toggled via buttons on the mictrocontroller board. Remote control is possible thanks to the commands sent through connected android application (not part of the project).

2 Main components

Vehicle consists of:

1. STM32 Nucleo L053R8 microcontroller
2. L298N 12V/2A motor driver
3. HC-06 ZS-040 Bluetooth module
4. 2 x 200 mA/3-6V MT78 motors
5. 2 x SPDT limit switch for collision detection
6. 7.4 V 2400 mAh LiPo battery
7. plastic wheels, wooden chassis

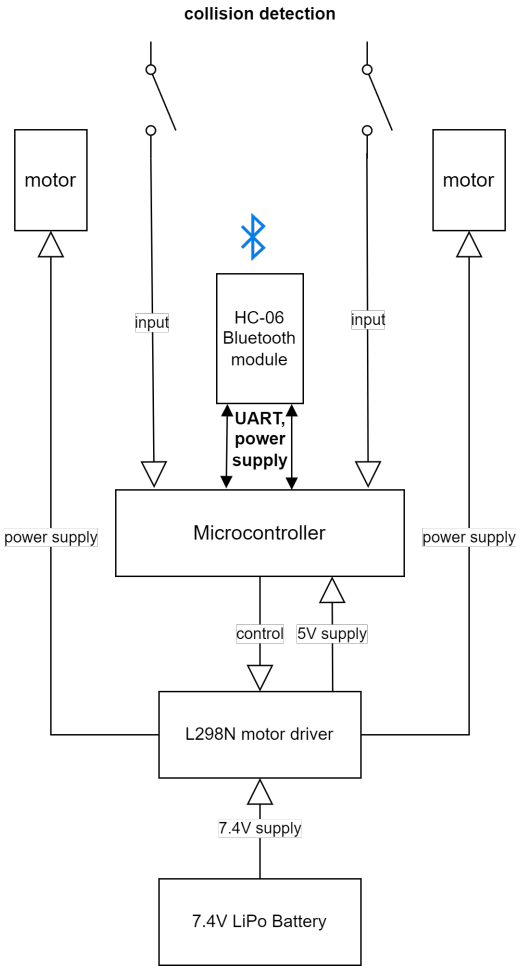

Figure 2: Vehicle schematic

3 Principle of operation

The vehicle offers two modes of operations:

1. autonomous driving with collision detection and obstacle avoidance
2. remote control via smartphone app(not part of the project) using bluetooth module

Autonomous driving mode lets vehicle drive on its own trying to find path without obstacles. If obstacle is detected via collision, the vehicle backs off and turns lightly to adjust its position, then resumes driving straight.
Remote control works through a free android app "RoboRemoDemo" wich offers programmable buttons that send commands to the bluetooth module connected to the microcontroller.

the modes of operation are toggled by pressing buttons placed on microcontroller:

1. blue button - activates autonomous driving mode
2. black button - activates remote control mode

At the start of operations (after reset) the vehicle is in remote control mode, which is based on sending commands via android application as seen below:

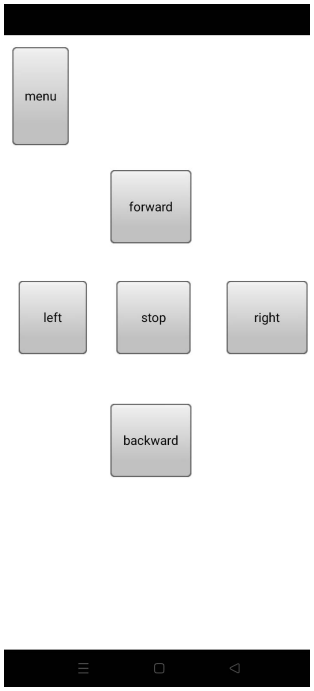

Figure 3: application view

4 Code structure

This section provides simplified code file structure of the project.
Entire code is written within STM32 IDE and uses its provided libraries, compiler, builder and debugger. the main function executes all periferial init. functions and procedes to enter an infinite while(1) loop. Collision detection and button pressing is handled via interrupt routines, UART reading is polled in while loop. Motor control is handled in L298N.c file, which uses provided tim.c file to generate PWM impulses through timer interrupts. apart from gpio.c and usart.c files the IDE provides system libraries listed below.

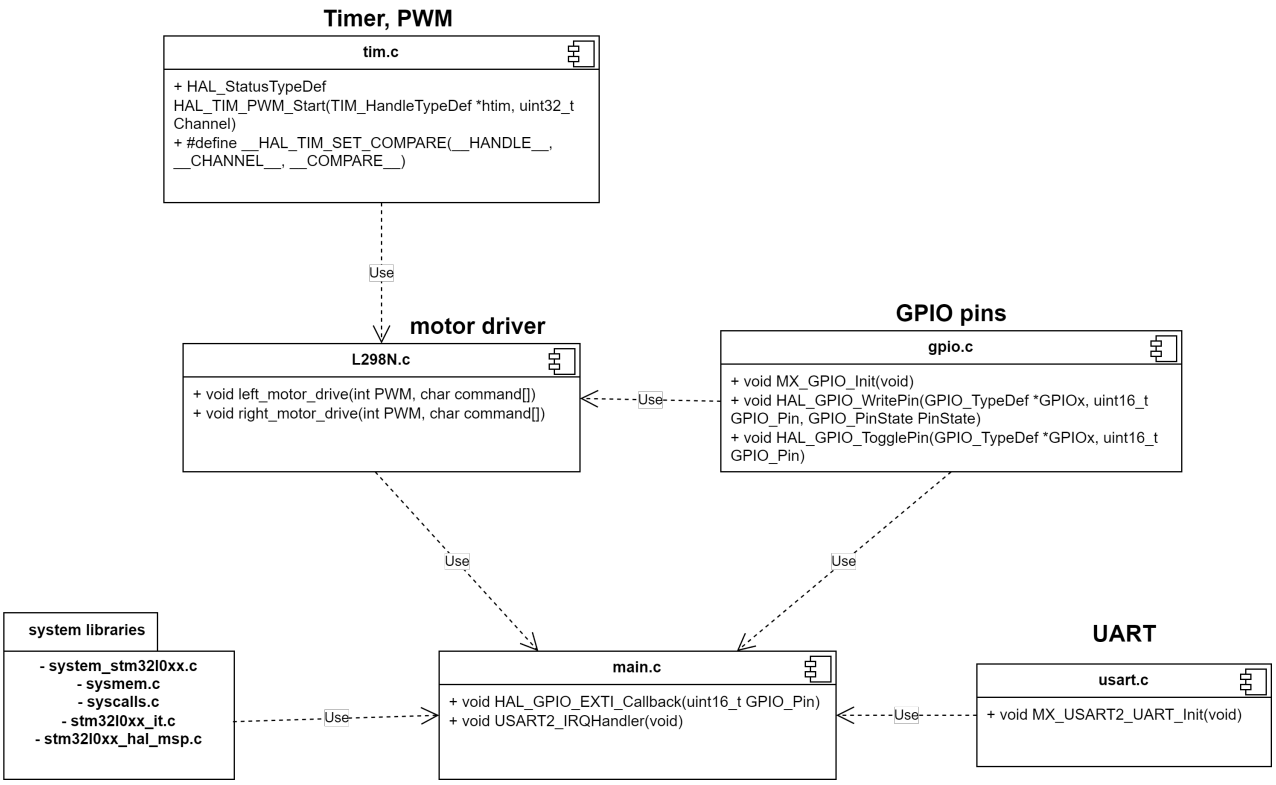

Figure 4: Code structure

5 Algorithms

This section illustrates algorithms used in designing microcontroller code.

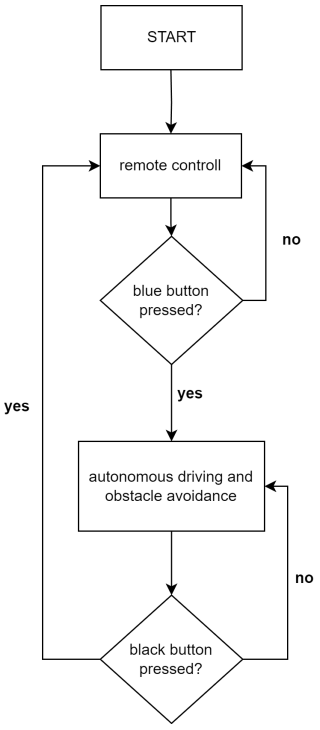

Figure 5: general behaviour algorithm

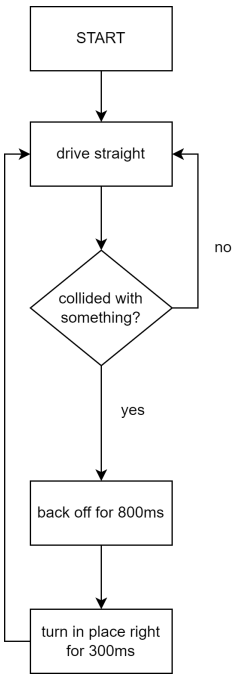

Figure 6: autonomous driving algorithm

Project's github: `https://github.com/JakubDomin/Metodyki-projektowania-i-modelowania-system-w-1`