CONTROLLING SOFTWARE FOR RC CAR

Eng. Marek Tomas

Abstract of the project

The aim of the project was to develop an controlling software for previously built RC car. The application should work on Windows platform, should connect with RC car by Bluetooth and enable to control car's acceleration and steering. It also should show the car's battery voltage state. Made application fulfills all assumptions.

Intermediate Project
Witold Paluszynski, Ph.D.
Faculty Of Electronic
Wroclaw University of Science and Technology

18 January 2018

1. Description

The application was made for controlling previously built RC car. It enables to control acceleration and steering and also shows the battery voltage level in RC car. The application works on Windows platform and the environment used to built the application is Delphi 10.2 Tokyo by Embarcadero. The application contains buttons used for establishing connection, controlling the ride and showing current battery voltage. It also contains scroll bars. Scroll bars enables to regulate throttle or steering. To steer also keyboard keys can be used. Software is also equipped in additional buttons which are used for reversing the direction of accelerating and emergency stop. Application is also intuitive and good-looking.

2. RC Car

The RC car use Bluetooth communication to connect with controlling device. The heart of the car is Arduino module. To propel the car, the DC motor is used and to steer the angle of wheels inclination, a micro servomechanism is used. The car is shown on the picture below:

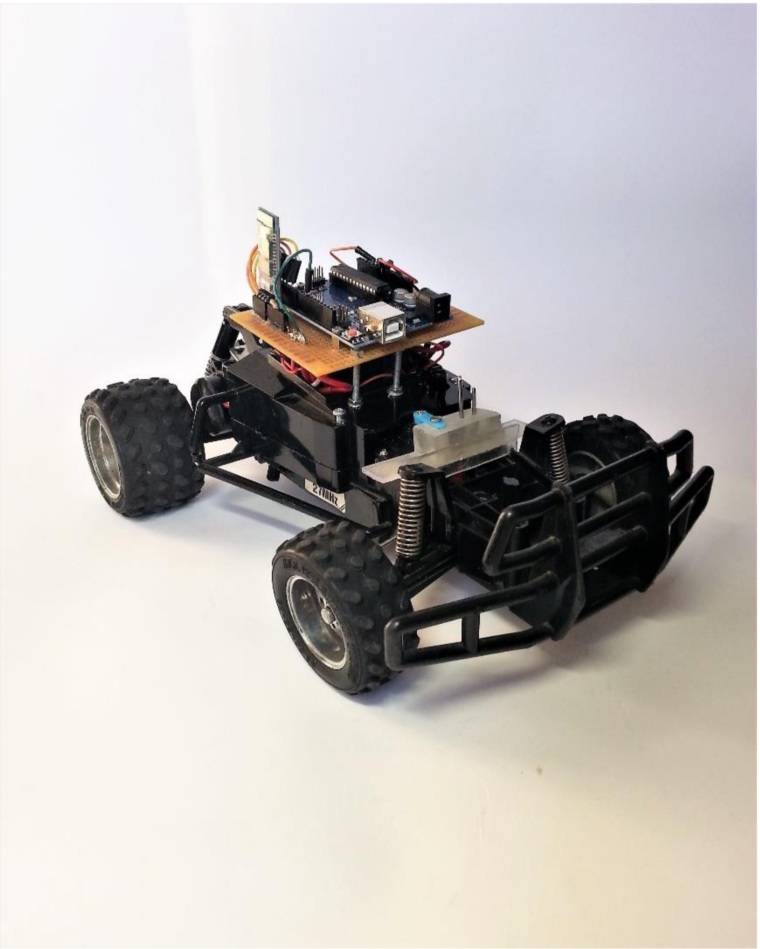


Fig. 1 RC Car

3. Integrated Software Environment

Delphi 10.2 Tokyo is IDE used for programing in Objective Pascal. This IDE was used because, this is a RAD (Rapid Application Development) type environment, which contains a lot of ready to use components, which shorten time of creating application and improves programming. For example, a Bluetooth component is included. This environment was recommended by an experienced programmer. Pascal language was chosen because I have some experience in creating application in it. It was good chance to check how the internal components for communication included in such IDE is working and how intuitive and easy to use is Delphi 10.2 Tokyo.

4. Application developing stages

4.1. Establishing Bluetooth connection

To establish Bluetooth connection, it is necessary to have a server and a client. The computer works as a client and a car's Bluetooth as a server. At first, it was required to pair with Bluetooth device in a car, then create a socket to connect with remote service. To specify the remote service, it was essential to provide a proper UUID number. After creating socket, the establishing connection was enabled.

4.2. Sending data

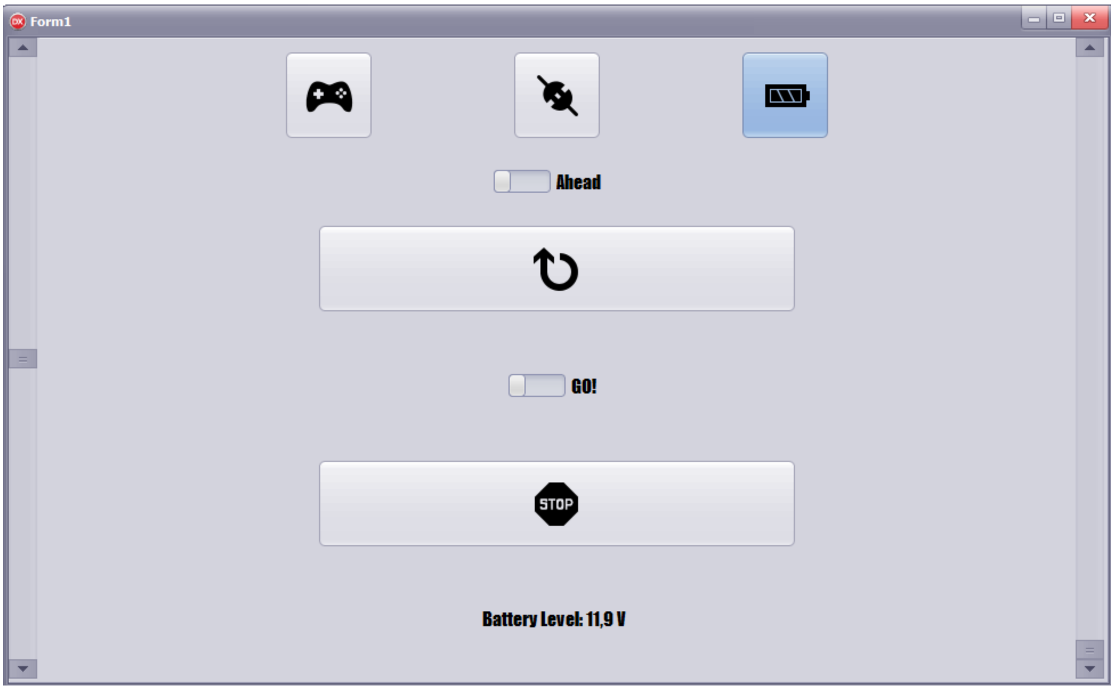
The car control is conducted by sending proper values. Special ranges of values are responsible for driving forward, driving backward, regulating the angle of steering. Proper values are sent depends on the scroll bars positions. To send a value, it is necessary to convert it to proper type. The numbers are sending in intervals, using timer. It is necessary to set not too small interval because the time of receiving and processing data in Arduino is limited, so the reaction's delay can be noticed if the interval is set to small (e.g. 50 ms is to small but 100 ms is proper).

4.3. Receiving data

The car is sending values about the battery voltage state. The Bluetooth component in RAD Studio contains an internal function responsible for receiving data. After receiving, it is necessary to process it, to get the battery voltage state.

4.4. GUI

The requirements of GUI is to make intuitive, good-looking interface. The interface contains a buttons responsible for connecting, starting controlling, receiving battery voltage state, reversing direction of riding and emergency stopping. It also contains a scroll bars for regulating speed and angle of wheel's inclination and the area to show the battery voltage state. The GUI is shown on picture below:



Battery Level: 11,9 V

Fig. 2 GUI

4.5. Mock-up for test

To execute fast test and improve work, special mock-up was constructed. It contains Bluetooth connected to Arduino and special software which send values and shows received values on serial monitor. The picture below shows the mock-up:

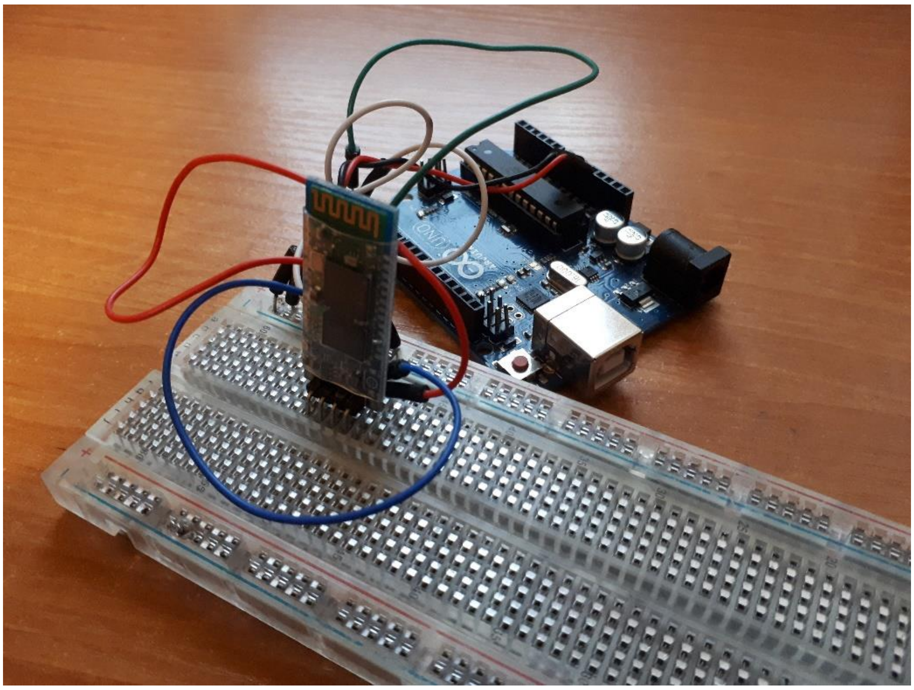


Fig. 3 Mock-up

5. Conclusion

After executing several test, the application works perfectly. There are no delays, frames are sent completely, as well as the frames are received completely. There are no problem with establishing connection with car (from software, not car, side). The main problems were to establish stable Bluetooth connection and realize that strange behavior of car is caused by setting wrong timer interval of sending values.

Delphi 10.2 Tokyo is really easy to use. Most of components, which are needed, are built-in. Bluetooth component is also easy to use and decreases the time of creating part of program to establish connection. The IDE is intuitive and there were no problems with usage or configuration.

6. Bibliography

- http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Using_Classic_Bluetooth
- http://docwiki.embarcadero.com/Libraries/Tokyo/en/System.Bluetooth.Components.TBluetooth