RC Car Controlled by WiFi with an Android Smartphone

Antoine Monmarché

April 7, 2011

1 Objective

The goal of the project is to pilot a RC model via an Android smartphone. This document is an abstract of the one I wrote for a project for my last year at the Polytech'Tours engineering school in France.

Figure 1: NiKKO NE14

2 Presentation of the car

The car is a NE14 by NIKKO. The battery is 9,6V. With the original electronic board, it lasts 15 min. The board will be replaced by a prototype board with the Flyport. To controll the car, you have 2 motors to deal with:

- The first motor is used to change the direction of the car. The servo-motor I used is a HITEC HS-322HD which is a lot more easy to use than the servo in the car (6 wires).

- The second motor is used to change the speed of the car. It has to be driven by a Hbridge to enable the car to move forward and backward. This motor is powered directly by the car's battery.

3 Electronic

With the Flyport you will have to use 2 PWM, 2 digital output and the WiFi communication (UDP connection).

To controll the car with the Flyport I had to develop the following parts:

- voltage adaptation 9,6V to 5V

- Hbridge circuit

The voltage adaptation is done with a LM7805 by Fairchild. The command of the propulsion motor is done with a SN754410NE by Texas Instrument. You need 1 PWM and 2 digitals ouputs (out1 & out2) to controll the motor. To determined the duty cycle of the PWM I used an Arduino board.

- if out1=1 and out2=0, the motor goes clockwise at the speed set by the PWM

- if out1=0 and out2=1, the motor goes anti-clockwise at the speed set by the PWM

The servo for the direction is controlled by a PWM. The duty cycle of the PWM in function of the angle of the servo (0-180) has been determined with an Arduino board.

The figure 3 represents the electronic schematic. The figure 4 represents the electronic test board which is embedded in the car. The figure 2 represents the duty cycle of the PWM in function of the angle of rotation.

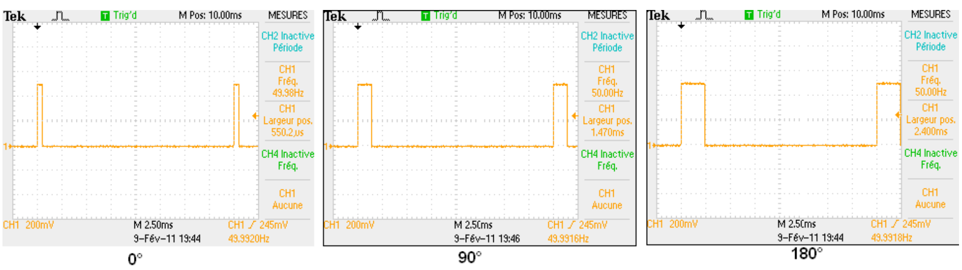

Figure 2: Electronic schematic

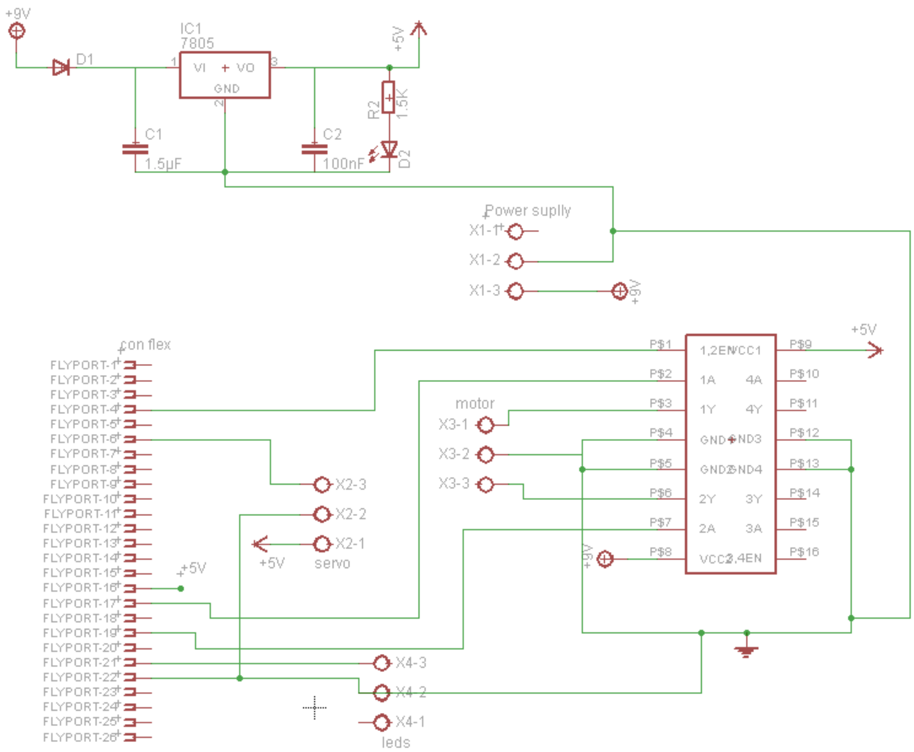

Figure 3: Electronic schematic

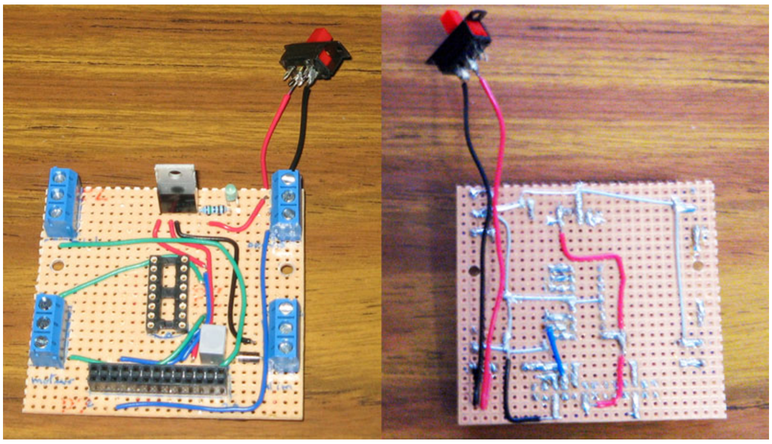

Figure 4: Test board

4 Software

Communication protocol: The UDP packet sent to the car contains the following information (port 25000):

- 1 char for the direction: between '0' and '9'
- 1 char for the speed: 'H', 'h', '0', 'l', 'L'
- 1 char for the led: 'y' or 'n'

An evolution of this protocole would be to use the first char to send the angle (between 0 and 180).

The network parameters of the car are:

- IP: 192.168.1.115
- adhoc network SSID: flyport
- security: none
- DHCP: disabled

All the other parameters are as default.

The Android application uses the Wifi connection and the accelerometers. The version of Android used is the 2.1 update 1 but it should work on other versions (tested on 1.5). The figure 5 represents the GUI of the Android application.

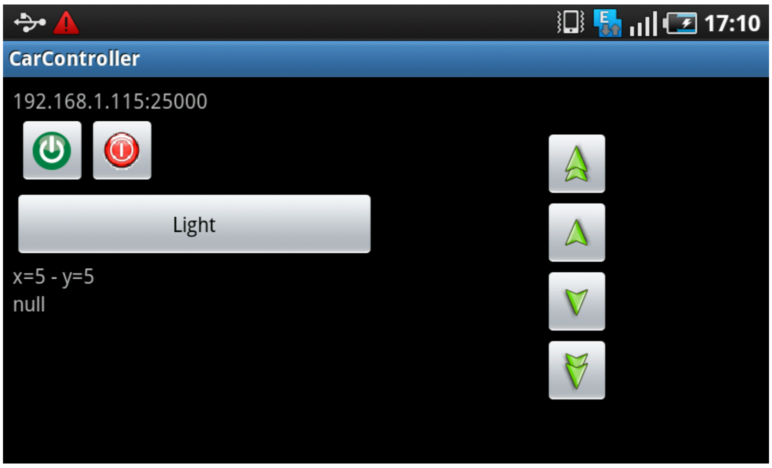

Figure 5: Android GUI

With an Arduino board I have determined the frequency and the duty cyle I had to use to controll the servo for the direction of the car. I obtain a linear equation. The frequency delivered by the Arduino was 50Hz so, to make it more simple, I decided to use the same frequency. The figure 6 represents the values of the dutycycle in function of the rotation angle of the servo (based on figure 2).

I tried to implement this function in the flyport but the result was not efficient. Maybe for a new version.

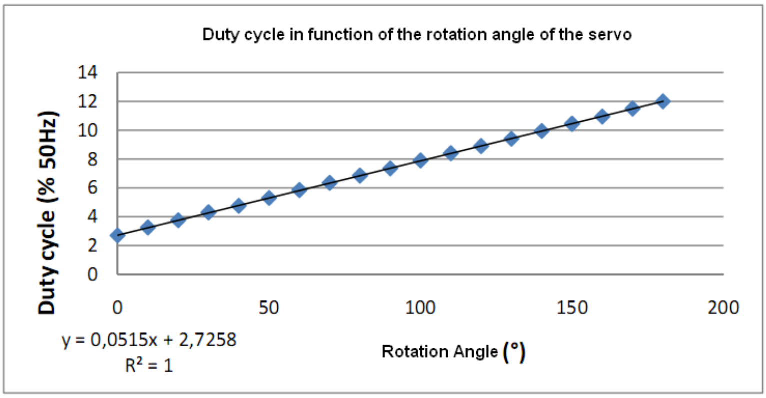

Figure 6: PWM output in function of the Dutycycle

5 Test

The Hbridge used is limited to 1A and the propulsion motor requires lot of power to run at full speed. There is no problem when the tyres are in the air but when the car is put on the floor, the motor will drawn to much power and will not move as fast as it should. A friend of mine advised me to use a LMD18200T (Hbridge) but I didn't try it yet. With the original battery, the car works for about 10min. It exists battery of the same size which can last 40min.

6 Integration

The servo I choose perfectly fits in the frame of the car. I had to cut the plastic to allow it to turn. The upper part of the frame has been removed because it was difficult to receive the wifi from the car. The figures 7 and 8 represent the integration of the board in the car.

7 Cost

Cost of the project:

- the car: 50€
- the Flyport + nest: 49+19€
- the electronic: less than 15€

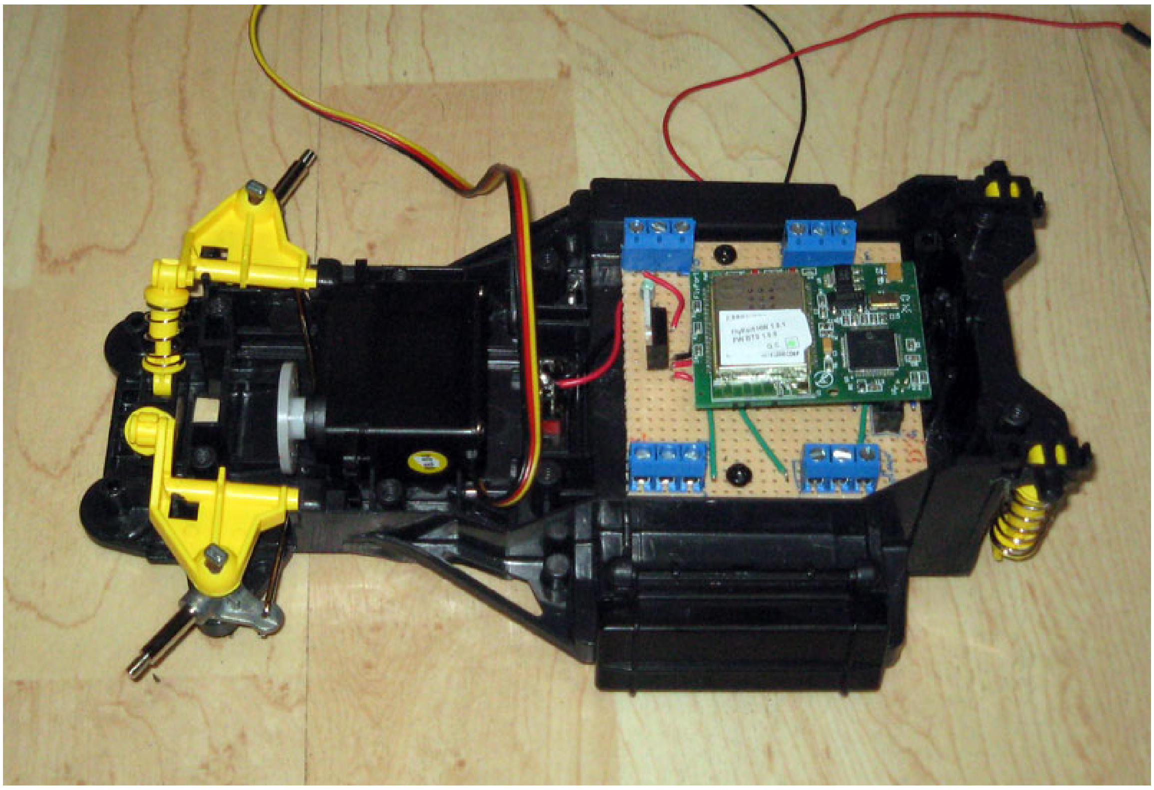

Figure 7: Integration

8 Bibliography

www.openpicus.com
www.arduino.cc
www.firtzing.org
http://nikkoevolution.fr/
http://fribotte.free.fr/bdtech/pic/pic_et_servo.html

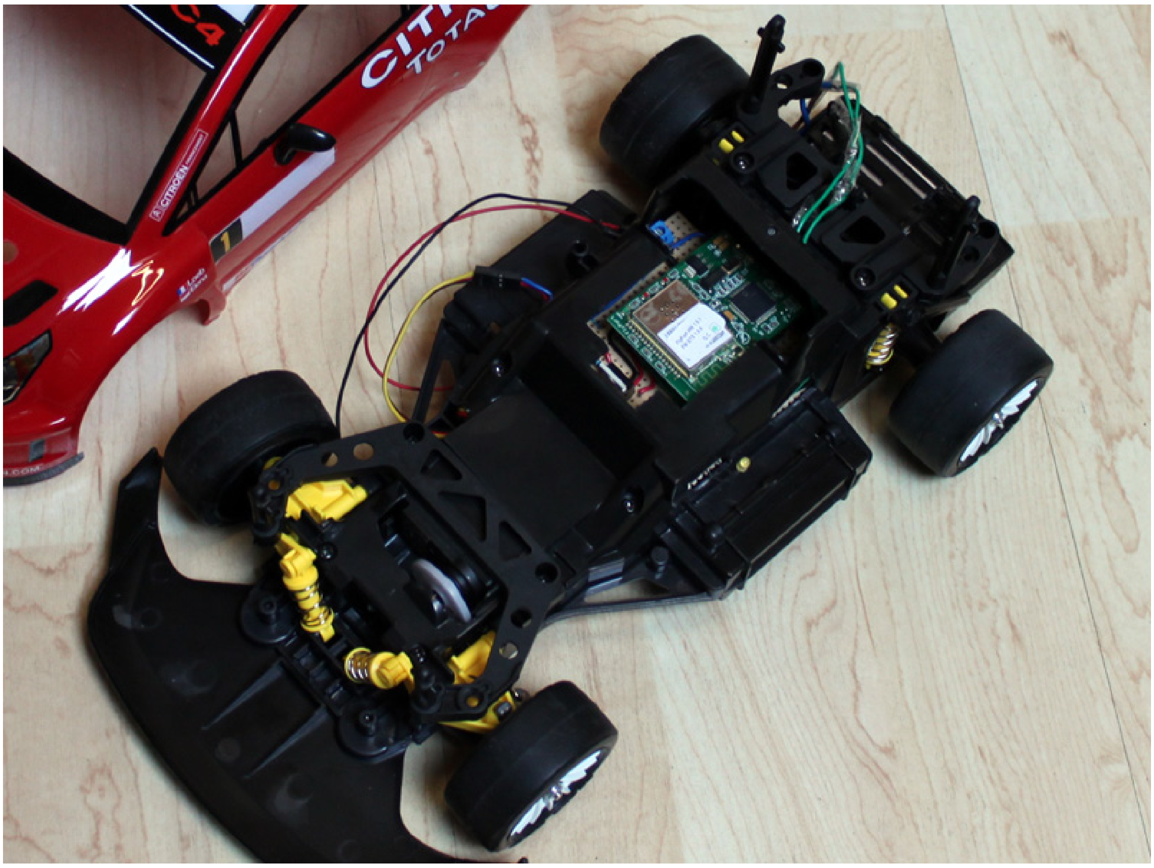

Figure 8: Integration 2