

# ShakerRacer

## Introduction

ShakerRacer allows you to control an RC car in a natural way – with movements of your mobile phone. A short movie showing the application in action is available online at:

<http://www.youtube.com/watch?v=EMjAYdF13cU>

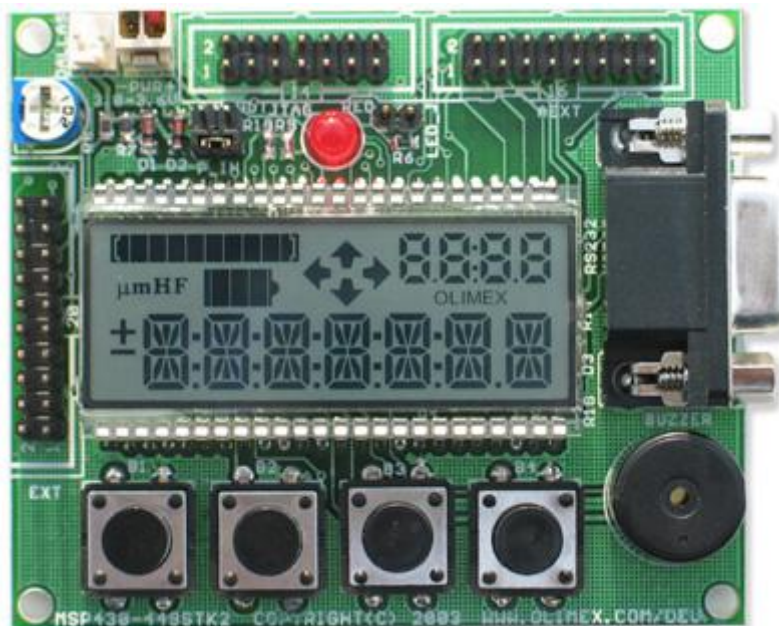


To accelerate, you just tilt your phone forward; for turning, you rotate it to the right or left. The movements are recorded with the internal acceleration sensor of current S60 phones.

Normal RC cars use their own proprietary wireless protocol and network transmission standard for sending commands from the remote to the car. Of course, a mobile phone can't emulate this. Therefore, you have to mod your car in order to integrate Bluetooth. See the next section for details.

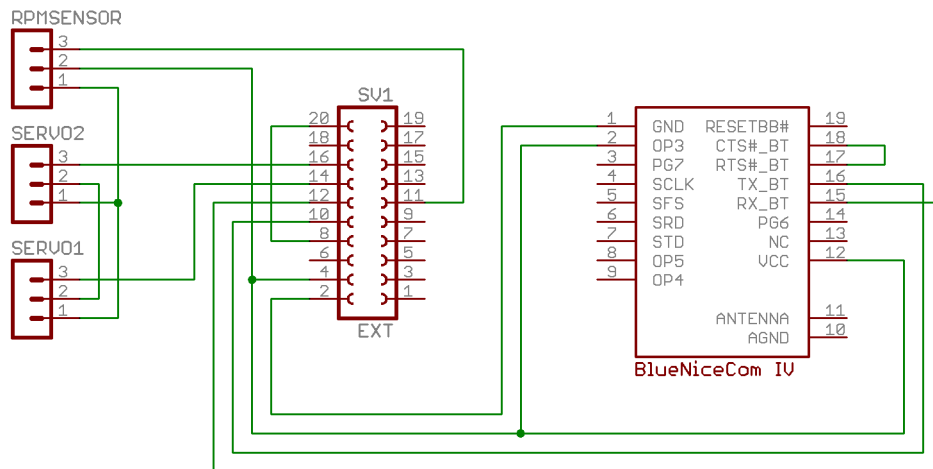
## Hardware Car Modding

The hardware is based on the MSP430-449STK2 development board by Olimex (<http://www.olimex.com>):

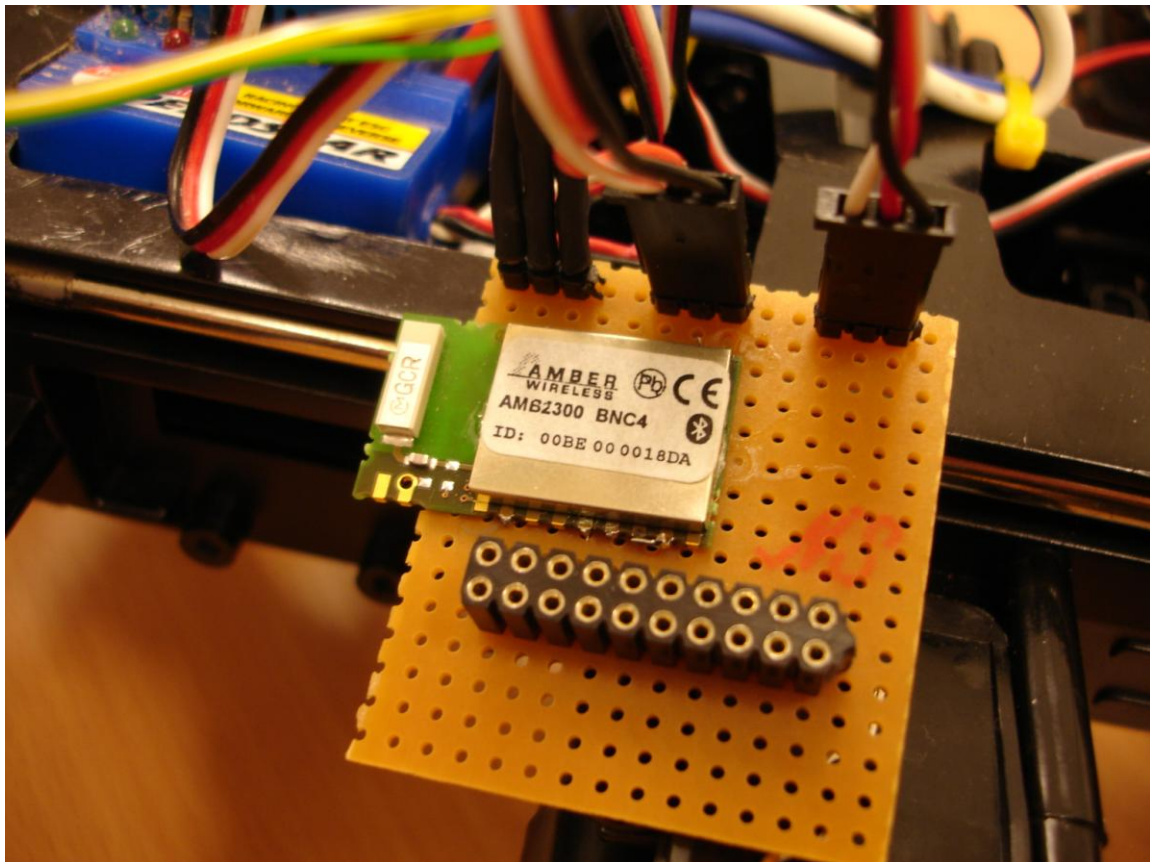


The microcontroller on this board (MSP430 by Texas Instruments, <http://www.ti.com>) receives commands from the mobile phone via Bluetooth and generates the PWM signals needed to control

the RC servos. The connection between the Bluetooth module (BlueNiceCom IV, [http://www.amber-wireless.de/pdf/AMB2300\\_MA\\_V1.9.pdf](http://www.amber-wireless.de/pdf/AMB2300_MA_V1.9.pdf)) and the controller as well as the interface to the servos is as follows:

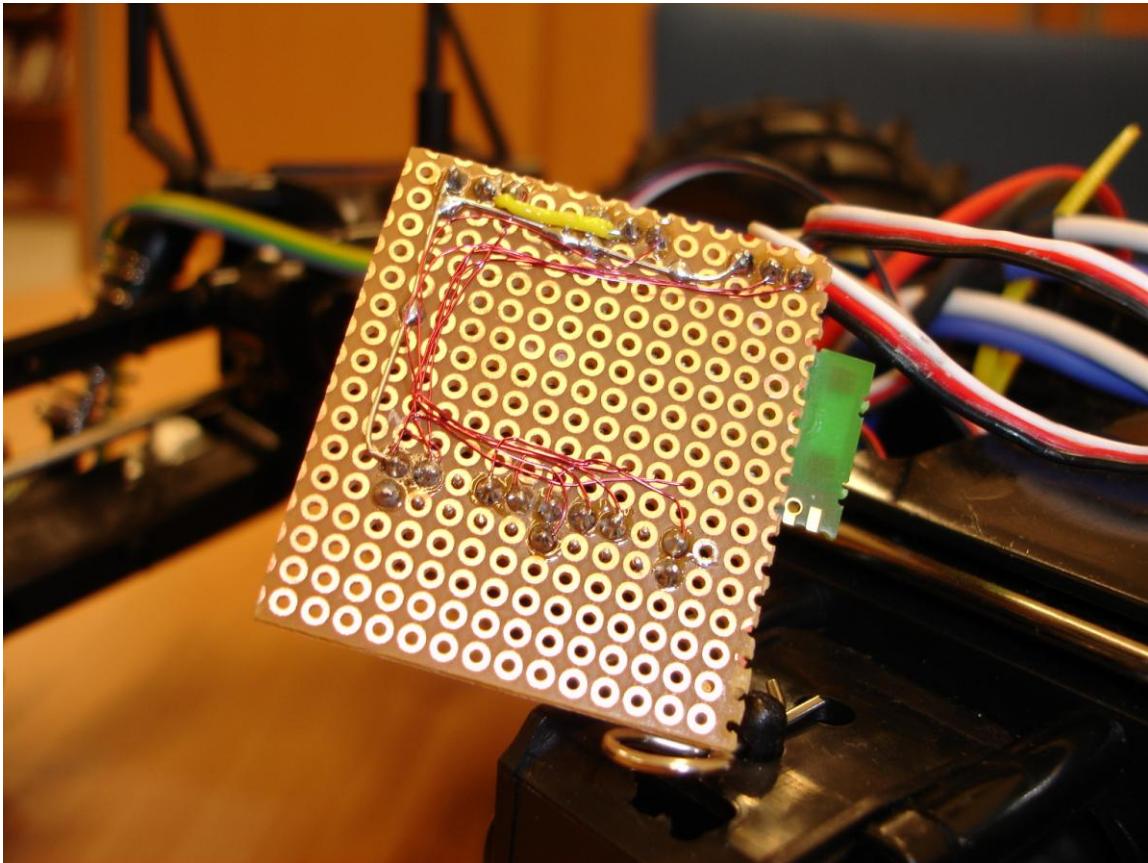


We used a small perf board to mount the Bluetooth module and the connector. The connections are made using verowire (manual dexterity and good eyesight are required... more information on this technique is available at [http://elm-chan.org/docs/wire/wiring\\_e.html](http://elm-chan.org/docs/wire/wiring_e.html), [http://www.izabella.freeuk.com/html/construction\\_techniques.html](http://www.izabella.freeuk.com/html/construction_techniques.html)):

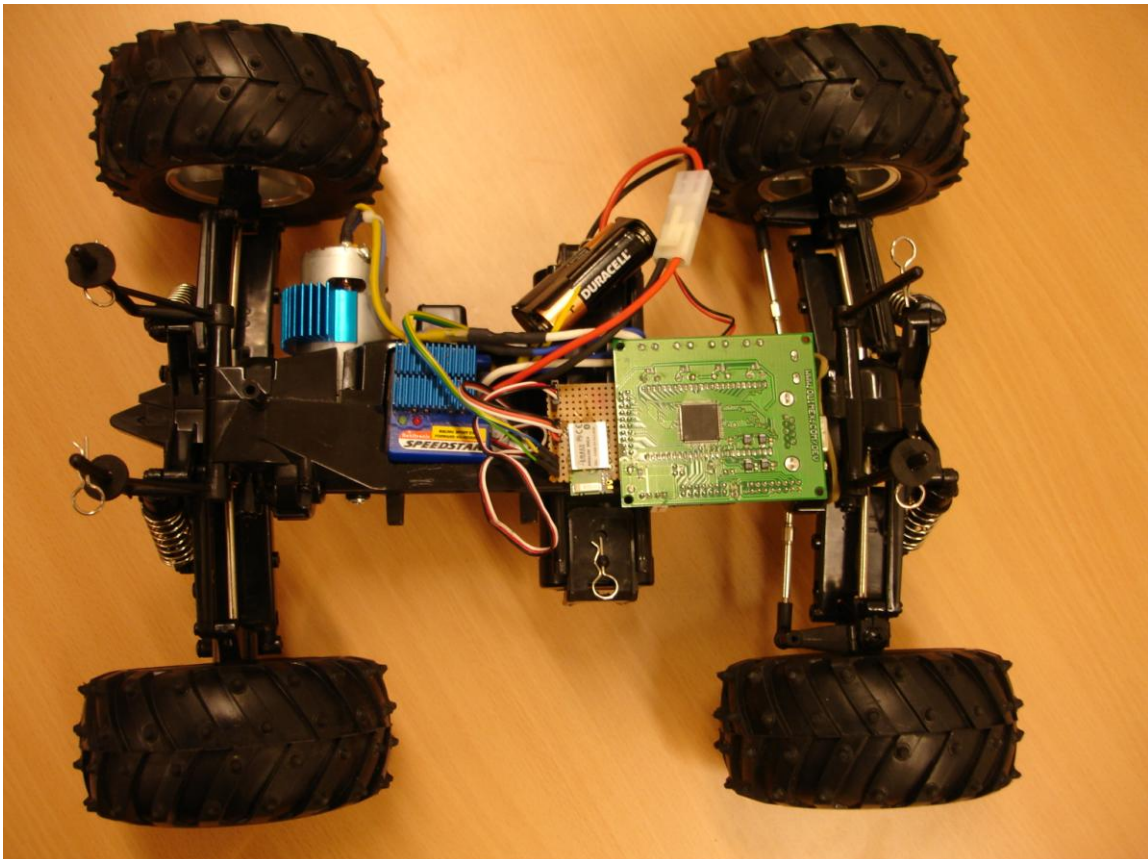




Bottom side:

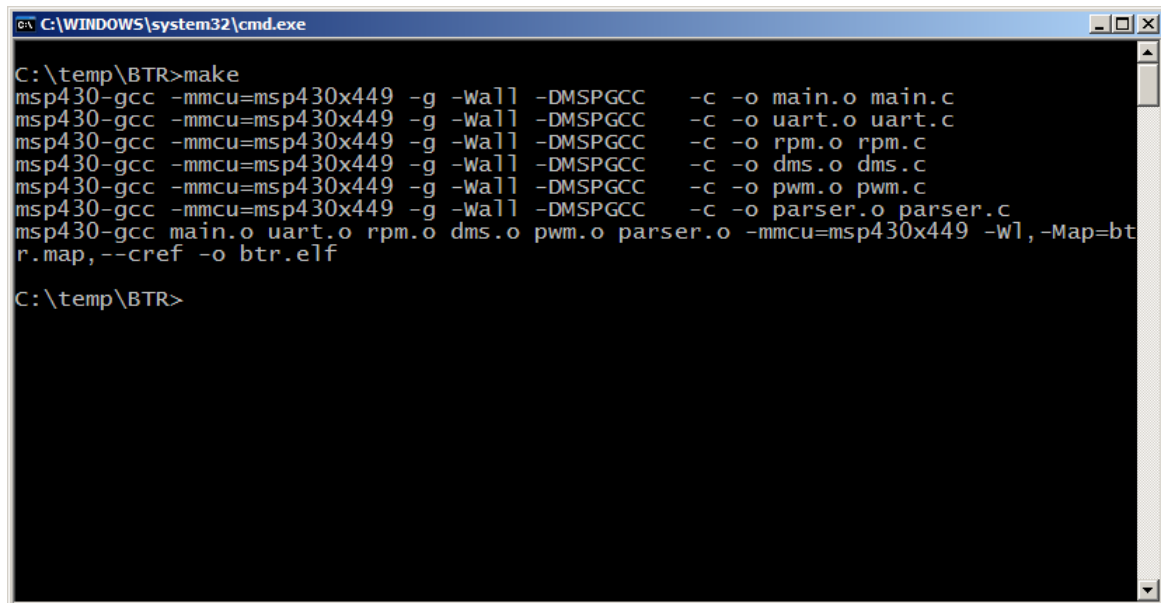


Now the perf board containing the Bluetooth module and the MSP430 board can be connected:



The software is written in C using the MSPGCC (<http://msp gcc.sourceforge.net/>), a port of the GCC to the MSP platform. The JTAG debugger is from Olimex (MSP430-JTAG <http://www.olimex.com>) and is used to download the firmware into the MSP430.

The firmware comes with a makefile – unzip the sources to any directory and run make:

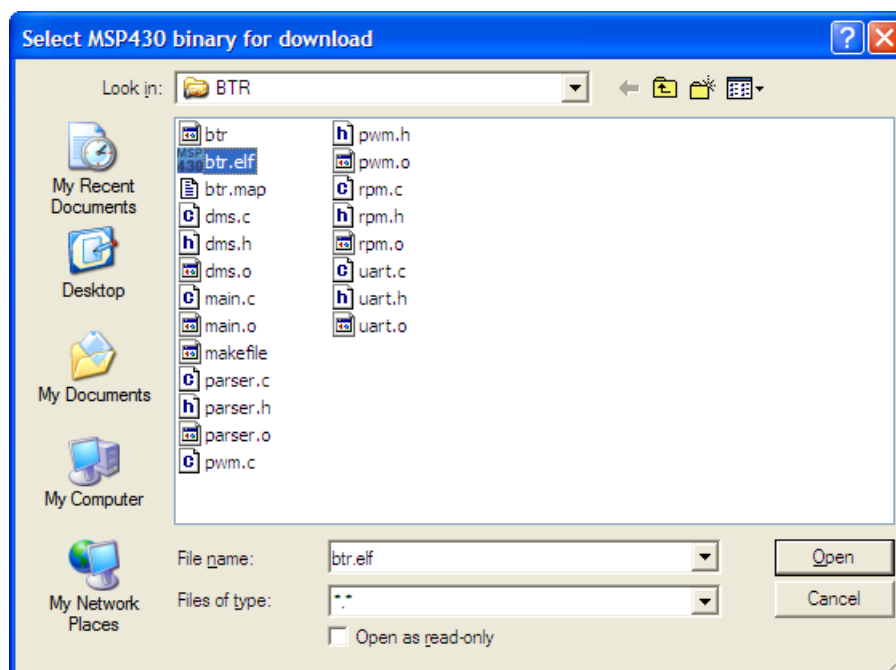


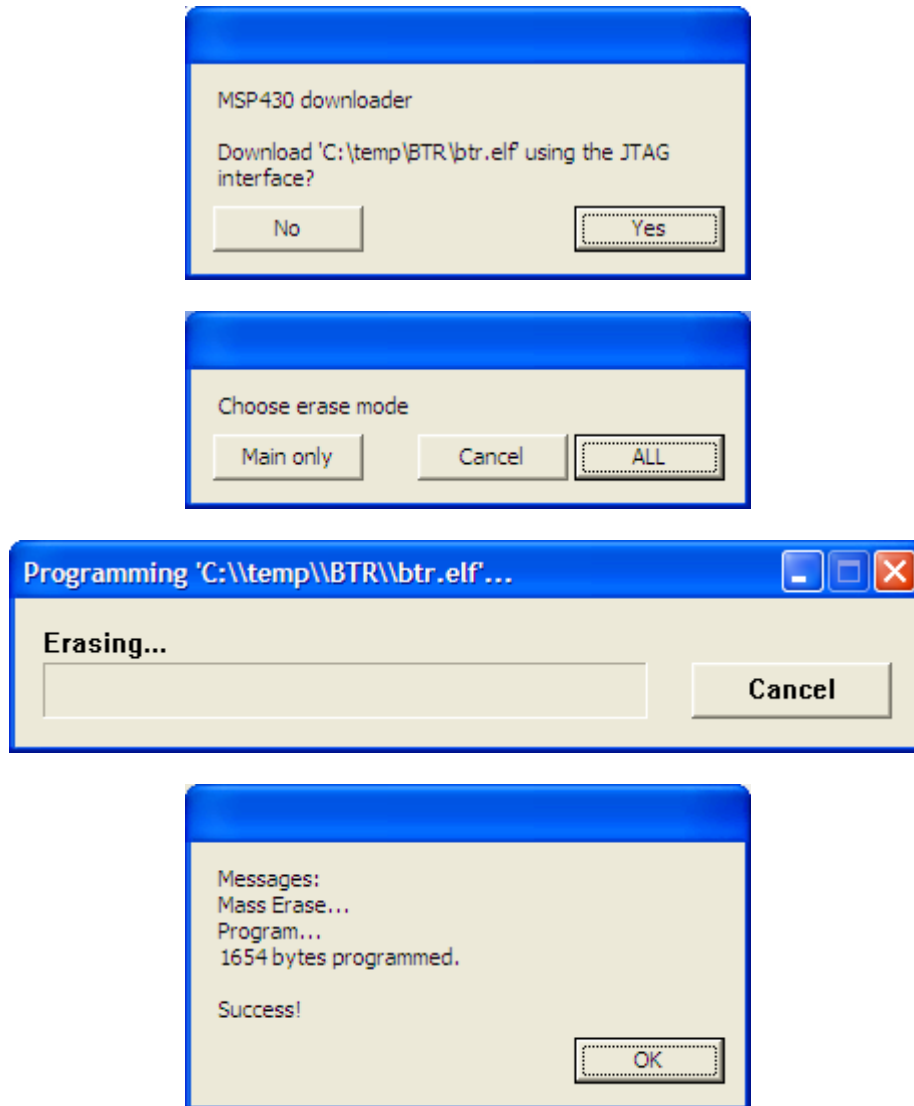
```

C:\temp\BTR>make
msp430-gcc -mmcu=msp430x449 -g -Wall -DMSPGCC -c -o main.o main.c
msp430-gcc -mmcu=msp430x449 -g -Wall -DMSPGCC -c -o uart.o uart.c
msp430-gcc -mmcu=msp430x449 -g -Wall -DMSPGCC -c -o rpm.o rpm.c
msp430-gcc -mmcu=msp430x449 -g -Wall -DMSPGCC -c -o dms.o dms.c
msp430-gcc -mmcu=msp430x449 -g -Wall -DMSPGCC -c -o pwm.o pwm.c
msp430-gcc -mmcu=msp430x449 -g -Wall -DMSPGCC -c -o parser.o parser.c
msp430-gcc main.o uart.o rpm.o dms.o pwm.o parser.o -mmcu=msp430x449 -Wl,-Map=btr.map,--cref -o btr.elf
C:\temp\BTR>

```

Using the JTAG debugger, the binary can be downloaded with msp430-downloader.exe which can be found in the mspgcc/bin folder:





After the download, the MSP430 immediately starts executing which means that the Bluetooth module can now be discovered by the mobile phone and that the firmware now awaits commands from the mobile phone.

## ShakerRacer Phone Software

The phone software was written in Python for S60. It is able to connect to the car via Bluetooth and then sends control commands to the car.

### Compatible phones

At the time of writing, the following phones are compatible to ShakerRacer:

- Nokia N95 (with Firmware v20+)
- Nokia N95 8GB
- Nokia N82 (+8GB)
- Nokia N93i
- Nokia 5500 Sport

## Setup

**General note:** make sure that your phone is configured so that it allows installation (this setting can possibly be deactivated on some E-series devices or operator-branded firmware variants). To check this, go to the “Application Manager” (usually located in the “Tools” or the “Applications” folder of the menu). In the manager, go to “Options” → “Settings”. Make sure “Software installation” is set to “All” instead of “signed only”.

To run ShakerRacer on your mobile phone, install the latest *PyS60 (Python) runtime*. It is available for free from: <http://sourceforge.net/projects/pys60/>. You need to download PythonForS60\_1\_4\_2\_3rdEd.SIS (or later).

As the car is controlled only by movement of the phone and doesn’t require any key input, the backlight would fade out after a short period of time. For keeping it on, ShakerRacer requires the *Miso extension for pys60*. Download and install it from: <http://pdis.hiit.fi/pdis/download/miso/>

**Note:** at the time of writing, the certificate of the Miso-package was no longer valid. If installation does not work, self-sign the package or use your developer certificate in case you have one.

Next, install *ShakerRacer.sis* on your phone.

**Advanced use:** To edit the script on the phone or if you need to adapt the settings in source code, you can also install the *Python Script Shell*, which allows executing Python scripts on the phone. Then copy ShakerRacer.py to E:\Python\. Now, simply run ShakerRacer from the script shell environment.

**Configuration:** ShakerRacer was developed as a small showcase demo, therefore the application structure isn’t very sophisticated and first-time configuration has to be done in the source code.

The following keys can be used to adapt certain settings at runtime:

- Joystick center: (De)activate the emergency stop
- \*: (De)activate debug display
- #: (De)activate landscape mode
- 0: (De)activate inverted acceleration
- 3: (De)activate the exhibition mode
- Right softkey: exit ShakerRacer

## Emergency stop (Joystick center)

Press the joystick once to activate the emergency stop mode. From now on, the phone will only send 0-values to the car – regardless of the actual tilt status. Please note that emergency stop does not explicitly brake, as this would require sending negative acceleration values, which could trigger driving backwards. Therefore, if the car is in full speed and you activate the emergency stop, it might take some time till the car comes to a stop.

## Debug display (\*)

Activate this mode to see the debug output on the phone. This includes current acceleration values, as well as several settings like landscape/portrait mode, speed limit ...

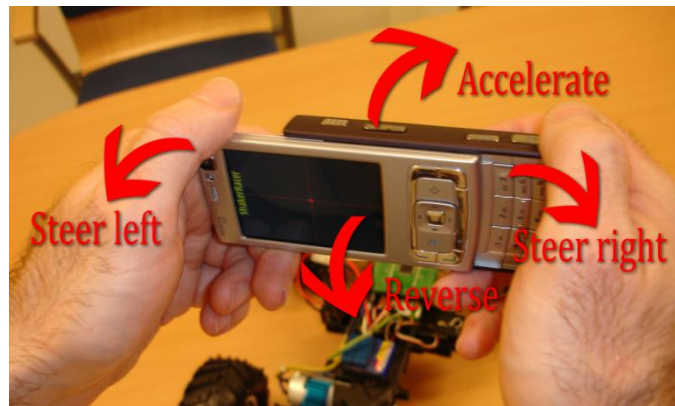
## Landscape Mode (#)

Our tests showed that landscape mode is the more natural way to control the car. However, you can still activate the other control mode by pressing #.



Please make sure that you hold the phone correctly when starting the application. Otherwise, it might happen that the car accelerates at a fast speed as soon as you connect ShakerRacer to the car.

#### *Landscape mode:*



#### *Portrait mode:*



### **Inverted Acceleration (0)**

This setting is especially useful for the portrait control mode (see above). If this setting is activated, the direction in which you have to tilt your phone for driving forward or backward is switched / inverted. Choose the mode that is more natural for your taste.

### **Exhibition Mode (speed limit)**

Especially for using the car in a crowded area like an exhibition or if you don't have much room, you should only use ShakerRacer with the activated exhibition mode. The maximum speed of the exhibition mode can be configured in the source code of ShakerRacer. These values have to be configured according to the strength of your motor and the quality of the components.

### **Setting the acceleration offset**

Depending on the cruise control module of your mobile phone, you might have to modify the acceleration offset. With the standard remote control, this is usually done by a small trimming wheel.

- **Better cruise control:** these modules automatically calibrate themselves during startup. For this reason, the hardware module automatically sends 0-values to the cruise control module when it is activated. Nothing has to be done on the phone; the acceleration offset should be left at 0.
- **Cheaper cruise control:** you have to configure the acceleration offset in the source code of ShakerRacer. It's recommended that you hold the car so that its wheels do not touch the ground while testing. While debug display (\*) is activated, test at which acceleration value the stands still. Set this value as the acceleration offset in the ShakerRacer source code, so that the car doesn't move when you hold the phone upright.

## Debugging the application

If you want to debug the application in a safe way, the source code can be reconfigured in order to make this task easier.

- **use\_bluetooth:** if set to -1 (instead of 1), the application doesn't require to be connected to a Bluetooth target. This makes it easier and faster to test new behavior of the software.
- **draw\_graphics:** usually ShakerRacer runs in a graphical full screen mode. If you extend the code and error messages appear, you wouldn't see them. Deactivate the graphics display in order to see and react to errors.

## Sensor API

By default, ShakerRacer uses the official Nokia Sensor API to read data from the acceleration sensor. No additional software or libraries have to be installed and ShakerRacer works right away.

On certain phones (N95 < Firmware 20, N93i with an older firmware), this API has not been available. In this case, you need to install the R&D Accelerometer Plug-in package and the aXYZ Python package from Cyke64 on your phone. Then set use\_sensor\_api to -1.

Nokia accelerometer plug-in package: [http://research.nokia.com/projects/activity\\_monitor](http://research.nokia.com/projects/activity_monitor)  
aXYZ: <http://discussion.forum.nokia.com/forum/showthread.php?t=120394>

## Team & Contact

ShakerRacer was developed by:

- **Andreas Jakl:** phone software ([andreas.jakl@fh-hagenberg.at](mailto:andreas.jakl@fh-hagenberg.at))
- **Stephan Selinger:** hardware ([stephan.selinger@fh-hagenberg.at](mailto:stephan.selinger@fh-hagenberg.at))

Development was done at the Upper Austrian University of Applied Sciences, Campus Hagenberg – Department of Mobile Computing – <http://www.fh-ooe.at/mc>

The source code of both the hardware and the phone components is released under the GNU General Public License v3 (GPLv3).

Copyright © 2007-08 Stephan Selinger and Andreas Jakl.