Master Thesis Presentation
Future Electric Vehicle on Lego
By Karan Savant

Guide: Dr. Kai Huang
Overview

- **Objective**
- Lego Car
- Wifi Interface to Lego Car
- Lego Car FPGA System
- Android Application
- Conclusion
Objective of Thesis

• Interfacing the Lego car with an Wi-Fi Module and controlling it remotely via Smart-Phone/Tab
• Configure the Wi-Fi Module as an Access point
• Re-engineer the Lego Technic 9398 into 4-wheel independent steering/driving
• Develop an Android Application to Implement control based on the Android device inbuilt sensors
• Implement a closed loop control for Car speed measurement using BEMF
Overview

• Objective
• Lego Car
  • Wifi Interface to Lego Car
  • Lego Car FPGA System
• Android Application
• Conclusion
Lego Car

Currently Being Used
It has one DC and one Servo Lego Motors

New Configuration
It has Four DC and Four Servo Lego Motors

Driving Modes

(a) default mode
(b) slow mode
(c) parallel mode
(d) rotational mode
(e) parking mode
(f) emergency mode
Overview

- Objective
- Lego Car
- Wi-Fi Interface to Lego Car
- Lego Car FPGA System
- Android Application
- Conclusion
Why WiFi?

• Better range than most other wireless protocols
• More secure
• Easily available today
• Many hardware options available
• Wifi has higher bitrate so it is more suitable for transferring the camera video signal
• Some Smartphone OS like Apple iOS requires special chip to be put in the application circuit for it to be Bluetooth enabled
Features:

- Hardware interface: UART and SPI slave
- Full onboard TCP/IP stack (no external drivers required)
- Supports Adhoc and infrastructure networking modes
- Real-time clock for time-stamping, auto-sleep, and auto-wakeup modes
- Runs directly from batteries or regulated power supply
- Configuration over serial or wireless interface using simple ASCII commands
- Over the air firmware upgrade
- Secure Wi-Fi authentication schemes (WEP / WPA / WPA2)
Wi-Fi Module - FPGA Interface over UART
Overview

• Objective
• Lego Car
• Wi-Fi Interface to Lego Car
• Lego Car FPGA System
• Android Application
• Conclusion
The key features of the board:

**Featured device**
Altera Cyclone® IV EP4CE22F17C6N FPGA
Altera serial configuration – EPCS16(16Mbits)

**Memory devices**
32MB SDRAM
2Kb I2C EEPROM

**General user input/output**
8 green LEDs
2 debounced pushbuttons
4-position DIP switch

**A/D Converter**
NS ADC128S022, 8-Channel, 12-bit A/D Converter
FPGA Hardware Configuration

Nios II Core Processor CPU

ADC Controller controls the NS ADC128S022, 8-Channel, 12-bit ADC

EPCS Flash Controller

PIO LEDs

Motor Control IP
Controls the Motor Speed according to the PWM Duty Cycle

SDRAM Controller
Controls the 32Mb SDRAM

UART Module sets the UART to run at Baud rate: 115200 bps
8 Data bits, No Parity, 1 Stop bit.
FPGA Software - NIOS II

NIOS II Processor

Motor 1: BEMF Sense, Current Sense
Motor 2: BEMF Sense, Current Sense
Switch 1
Switch 2
Forward, Reverse
Forward, Reverse
LED's
Debug UART TX
WiFi UART TX
WiFi Module
From WiFi Module
WiFi USART RX
To PC
To WiFi Module
Wi-Fi Interface Protocol
In order to make the communication secure the following message structure is used:

**Message Structure**

*Message ID||Vehicle id||MAC ID||COMMAND||Data||*

* Start of message
* end of message
|| Delimiter

**Vehicle ID**

AGxxxxxx: Vehicle ID is made using the ip assigned to the Wifi module

eg: ip 192.168.5.5 has Vehicle ID AG005005
So ip 192.168.xxx.xxx has Vehicle ID AGxxxxxx
Message Exchange

Smartphone

Startup and Authentication

* Mxxx||AGxxxxxx||MAC ID||PASSWORD||Data||*

* Mxxx||AGxxxxxx||MAC ID||ACK||Data||*

Speed Control

* Mxxx||AGxxxxxx||MAC ID||START||Data||*

* Mxxx||AGxxxxxx||MAC ID||ACK||Data||*

* Mxxx||AGxxxxxx||MAC ID||ROL||Data||*

Above messages continue cyclically until Quit message is send

* Mxxx||AGxxxxxx||MAC ID||QUIT||Data||*

* Mxxx||AGxxxxxx||MAC ID||ACK||Data||*

FPGA Board

If Acknowledgement is not received the Message is resend
Overview

• Objective
• Lego Car
• Wi-Fi Interface to Lego Car
• Lego Car FPGA System
• Android Application
• Conclusion
The Android platform supports three broad categories of sensors:

**Motion sensors**
These sensors measure acceleration forces and rotational forces along three axes. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.

**Environmental sensors**
These sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers.

**Position sensors**
These sensors measure the physical position of a device. This category includes orientation sensors and magnetometers.
Android Orientation Sensor

The orientation sensor is software-based and derives its data from the accelerometer and the geomagnetic field sensor.

The orientation sensor lets you monitor the position of a device relative to the earth's frame of reference (specifically, magnetic north).

The orientation sensor provides azimuth (yaw), pitch, and roll values.
At Start button Press
Record the Device Coordinates as the Reference Coordinates

At Set Speed button Press
Record the Pitch Value
Keep Calculating the Difference:
Pitch = Reference Pitch – Current Pitch
PWM Duty= Constant * Pitch
PWM is used to control the Ecar Speed

At Turn button Press
Record the Roll Value
Keep Calculating the Difference:
Turn = Reference Roll – Current Roll
Ecar Remote

Language: English
First Name: Karan
Last Name: Savant
Email ID: aransavant@gmail.com
eCar ID: AG219001
Password: ****

Register
eCar Control

Current Speed

Set Speed

Turn

Start

Exit App

Azimuth: 255.00
Pitch: -26.00
Roll: -10.00
Andriod Programming Links

Here is the link i found which gives a step by step tutorial for setting up the android tools.
If you do not have Eclipse IDE or JRE(Java Runtime Environment) installed please follow this tutorial first.
http://www.vogella.com/articles/Eclipse/article.html

Then you have to install the ADT. The following link describes how to install and configure the ADT(Android Development tools). Please only follow the steps mentioned under "Updating an existing Eclipse installation"
http://www.vogella.com/articles/AndroidInstallation/article.html

There are also tutorials for android development and common problems you can find on this link.
http://www.vogella.com/android.html