RazorCar: A FPGA-based Prototyping Platform for Autonomous Driving Systems

Franck Yonga[1], Martin Malchow[2], Michael Mefenza[1], Christophe Bobda[1]
[1]CSCE Department - University of Arkansas - USA
[2]Chemnitz University of Technology - Germany

Abstract

In this paper we present the RazorCar, a FPGA-based radio controlled car that is able to autonomously realize driving scenarios such as obstacles detection/avoidance, self-parking, or street lane following. It is part of a project that aims at providing a prototyping environment for the design of generic hardware/software architectures for self-driving systems. Experiments on a lane detection scenario show that the car is able to drive autonomously by detecting and following a white lane on a track.

Introduction

Traffic accidents take the lives of nearly 1.3 million people every year [1]. The leading cause of all these crashes is unfortunately not imputable to machine but to human mistakes (driving under influence, careless driving). To mitigate or eliminate all tragic consequences related to human error when driving, the use of autonomous driving cars has been considered.

In this work, we present the RazorCar, a FPGA-based prototyping platform for autonomous radio controlled (RC) driving systems. Unlike in other existing projects such as [2], our primary goal is to provide a prototyping environment for the design of generic hardware/software architectures for self-driving systems. This hardware/software decomposition allows designers with experience in C-programming, artificial intelligence and image processing, but with few or no hardware knowledge, to focus on the development of intelligent applications that will seamlessly be mapped to low-level efficient architectures.

The Prototyping Platform

The RazorCar is part of a project that aims at developing intelligent and autonomous driving systems to investigate and validate the concept of self-driving in traffic navigation.

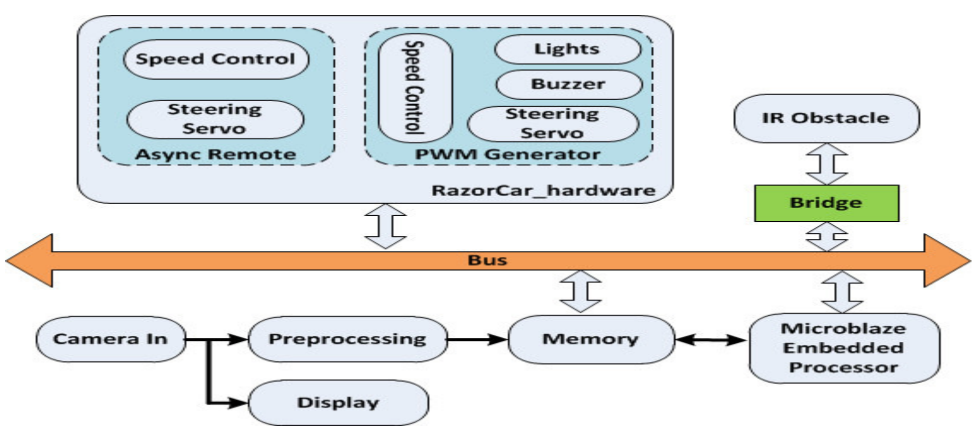


Figure 1. Architecture of the RazorCar System.

Figure 1 gives an overview of a generic RazorCar system. In such a system, the environment is recognized by using both a high resolution digital image sensor and two short range infrared sensors mounted on the front and on the back of the car. Image data coming from the camera are preprocessed and then saved in an internal memory for later use by an embedded general purposed processor. The processor combines the visual information with data provided by infrared sensors in order to infer the location of obstacles in the vicinity of the car. Once obstacles have been detected, driving directions can then be automatically issued to the actuators of the car through the RazorCar_hardware module. This module implements low-level functions to generate the required modulated signals to control the speed system, the steering servo, the lights and the buzzer system of the car. These functions are directly accessible at a high abstraction level through software drivers mapped at the address space of the processor. The car itself can also be remote controlled using a 2.4 GHz Transmitter; thus giving the possibility to switch between manual, semi-automatic and fully automatic control of the car.

The main module of the RazorCar features a Spartan6-XC6SLX45 FPGA. FPGA was chosen because of its capability to process data in parallel (which is suitable for image processing applications) and because of its flexibility, required by the unpredictable nature of the driving environment. Figure 2 shows an implementation of the lane detection and following scenario.

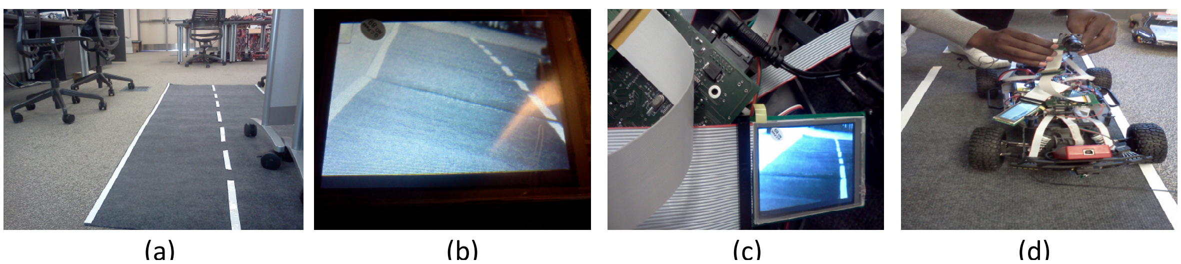


Figure2. The RazorCar system implementing a lane detection scenario. a – The track. b – The camera output. c – The Display connected to the FPGA. d – The car in the lane.

Conclusion and Broader Impacts

In this paper, we presented the RazorCar, a prototyping platform for FPGA-based autonomous driving systems. Besides helping to validate the self-driving concept, this system could also serve as an excellent educational tool in university programs. The system is designed to allow the implementation of complex driving scenarios at a high abstraction level by designers with no hardware experience. The goal is to raise the interest of researchers in other fields, such as machine learning or image processing, in order to enhance the investigation and deployment of high-qualitative solution for intelligent driving systems.

Bibliography

1 World Health Organization, “Decade of Action for Road Safety 2011-2020: saving millions of lives,” 2011. Available WWW: http://www.who.int/roadsafety/decade-of-action

2 University of Oxford UK, “The RobotCar,” (October 2012). Available WWW: http://mgr.robots.ox.ac.uk/robotcar/

Bibliographical Information

Franck Yonga is a PhD student at the Computer Engineering Department of the University of Arkansas. His research interests include embedded systems and hardware/software co-design. E-mail: yfrancku@uark.edu.

RazorCar: A Prototyping Environment for Autonomous Driving Systems

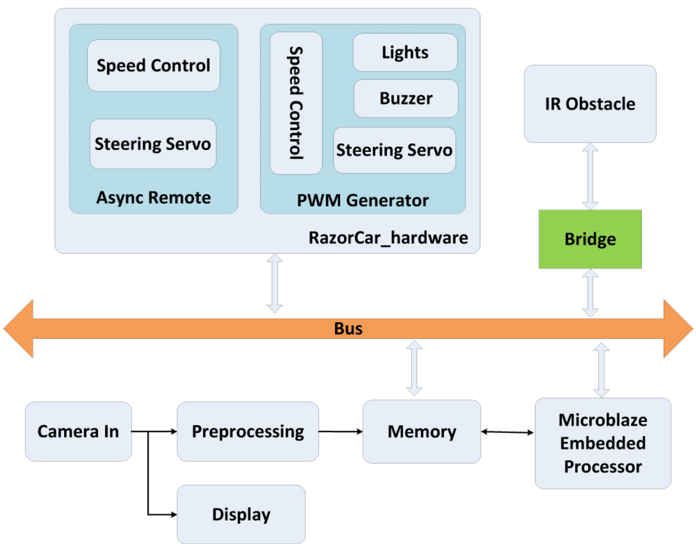
FPGA-based Self-driving Car

Franck Yonga[1], Martin Malchow[2], Michael Mefenza[1], Christophe Bobda[1]
[1]{yfrancku, mmefenza, cbobda}@uark.edu
[2]Martin.malchow@s2009.tu-chemnitz.de

Introduction: **Traffic accidents** take the lives of nearly 1.3 million people every year. The leading cause of all these crashes is imputable to human mistakes (driving under influence, distraction while driving, careless driving). To mitigate or eliminate all tragic consequences related to human error when driving, autonomous driving systems represent a viable solution. We propose the RazorCar system; a prototyping environment for the design of generic hardware/software architectures for self-driving systems. We aim at designing generic embedded architectures that will allow the implementation of complex driving scenarios at a high abstraction level by designers with no hardware experience. The main system of the car is built on a FPGA module for fast parallel processing and real-time response to unpredicted scenarios as required by the nature of driving environments.

RazorCar Project

The **RazorCar** project is the development of intelligent and autonomous driving systems to investigate and validate the concept of self-driving in traffic navigation. The primary goal of this project is to provide a prototyping environment for the design of generic System-on-chip (SoC) architectures for sel-driving systems. This SoC follows a hardware/ software decomposition. The decomposition will allow designers with experience in C-programming, artificial intelligence and image processing, but with no (or few) experience in hardware programming, to focus on the development of intelligent applications that will seamlessly be mapped to low-level efficient architectures. The seamless use of hardware accelerators at a high abstraction level by a general purpose processor is insured through the implementation of software drivers.

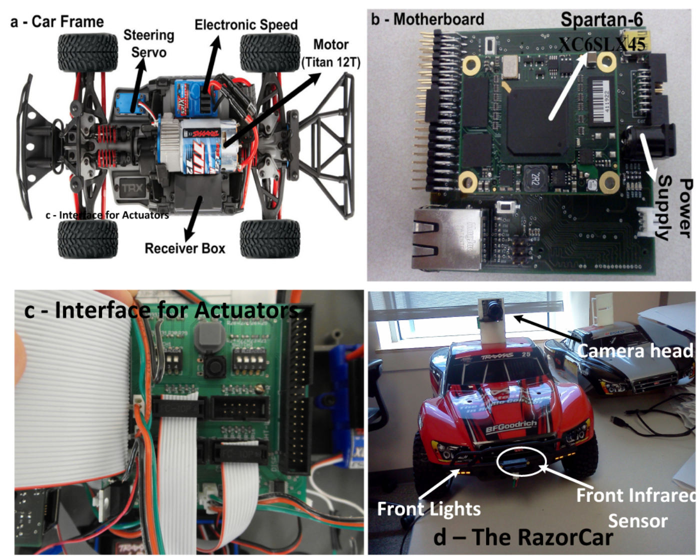


Architecture

- **Microblaze Processor :** Coordinates all operations on the RacazorCar, from retrieving and manipulating abstract data saved in memory (representing detected objects) to providing driving directions to the car through software drivers.
- **RazorCar Hardware :** implements low-level functions that generates modulated pulses to control the actuators of the car such as the speed control system and the steering servo.
- **IR Obstacle :** Interface that provides distance information about all objects surrounding the car.
- **Camera In module & Preprocessing:** Processing chain that transforms input image from the digital camera into abstract data that can be used by the processor. The transformation usually consists of a color conversion and a convolution kernel.
- **Display module:** Displayed input image from the digital camera on a 302 x 240 Thin Film Transistor LCD.

Prototyping Platform

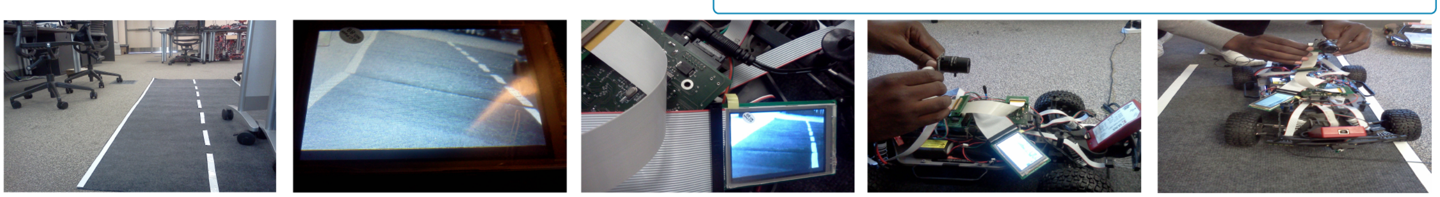
- **Motherboard Module** provides all the connections to the other modules and also the Main Power Supply of the system.
- **Main computing FPGA** features a Trenz Electronic GigaBee XC6SLX, which is a FPGA micro-module integrating a leading-edge Xilinx Spartan-6 LX FPGA.
- **TFT display** : a 320 x 240 thin film transistor LCD with serial peripheral interface for display and debugging purpose.
- **Camera Head Module:** a digital image sensor module based on the MT9V034, a wide VGA CMOS digital image sensor from Aptina.
- **Interface for Actuators:** an extension board providing interface for infrared, steering, speed, lights, and buzzer signals and data.
- **Infrared Sensors:** 2 GP2Y0A21YK wide-angle distance measuring sensors, with a detection area diameter of 80 cm.
- **Radio System:** a TQ 2.4GHz, 5 channels radio system consisting of a handheld transmitter and a receiver box located on the car frame.
- **Others:** a Steering Servo, an Electronic Speed Control system, front and back lights, a buzzer.



Application & Broader Impacts

As Application, we implemented a **Lane detection** scenario in which the car uses input image from the digital camera to detect white lane on a track.
Goals: keep the car inside its driving lane.
Design and implementation : The preprocessing step implements: a Bayer to to RGB conversion + Thresholding + scan line detection.
Major Drawbacks : Too many false positives (Outliers detected as lanes).
Results : the camera processes almost 320frames/s, the speed of the system is 100MHz and the size of the entire design (Slices LUTs) is 33% of the FPGA.
Broader Impacts of the Razorcar Project:

* Excellent educational tool in university programs
* Enhance easy utilization (programming) by designers with no hardware experience.
* Enhance investigation and deployment of high-qualitative solution for intelligent driving systems by researchers of different fields (Machine Learning, Artificial Intelligence,...)



Computer Science and Computer Engineering Department, University of Arkansas, Fayetteville, Arkansas, USA, 2013