

FPGA based Controller Design for Mobile Robots

K. Anitha¹, M. Srinivasa Rao²

^{1,2}Department of ECE, Prasad V Potluri Siddhartha Institute of Technology, Vijayawada, A.P, India

Abstract:- Mobile robotics and embedded systems are two research areas that have been receiving a considerable attention in years. Combining these two research topics is a very interesting and promising task. Some of the problems of controlling robots using embedded systems are designing device drivers, provide network communication and develop complex control algorithms under hardware limitations. This paper presents a conception of mobile robots using rapid prototyping, distributing the several control actions in growing levels of complexity and computing proposal oriented to embed systems implementation. This kind of controller can be tested on different platform representing the mobile robots using reprogrammable logic components (FPGA). Different modules can be interfaced using FPGA controller. Here we are constructing a simple robot model, which can measure the distance from obstacle with the aid of sensor and accordingly it is able to control the speed of motor.

Keywords — Infrared sensor, Obstacle Avoidance, Motor driver L293D, Spartan3E FPGA, VHDL.

I. INTRODUCTION

The emergence of reconfigurable Field Programmable Gate Arrays (FPGA) has given rise to a new platform of complete mobile robot control system. With FPGA devices, it is possible to tailor the design to fit the requirements of applications (for example, exploration and navigation functions for a robot). General-purpose computers can provide acceptable performance when tasks are not too complex. A single processor system cannot guarantee real-time response (particularly in the absence of considerable additional hardware), if the environment is dynamic or semi-dynamic. Here we focus on the design of the mobile robot platform, with two driving wheels mounted on the same axis. An FPGA-based robotic system can be designed to handle tasks in parallel [3]. The mobile robot consists of many units like Mechanics (chassis, housing, wheels), electromechanical parts and Sensors.

Robots carry out various tasks. During these tasks the robot moves and orients. While navigating, it uses signals from the environment and the contents of its own memory to make the correct decisions. This form of navigation may be manifold depending on the given task and problem [1].

Wheeled mobile robot:

Wheel control is less complex than the actuation of multi-joint legs, and wheels cause minimal surface damage in comparison with treads. Wheeled mobile

robots (WMRs) are more energy efficient than legged robots on hard and smooth surfaces [6]. They require fewer and simpler parts and are thus easier to build than legged mobile robots.

A robot is capable of locomotion on surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface. Most designs require minimum two motors for driving (and steering) a mobile robot. The combination of two driven wheels allows the robot to be driven straight, in a curve, or to turn on the spot.

II. HARDWARE DESCRIPTION OF MOBILE ROBOT

Architecture of Mobile Robot:

Within the proposal of mobile robotics platform, the use of FPGA Controller, with control software especially developed for the necessary applications, is considered using structured libraries to design, simulation, and verification with MODELSIM, we convert the model to function prototyping using FPGA hardware.

This system included both hardware and software development. The output of ADC was connected to the FPGA board and it was used as the input of the source code. The assembly language that is used in this system is VHDL. After the simulation and the synthesis process, the program has been implemented on the FPGA board.

The Spartan 3E family of Field-Programmable Gate Arrays is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications [4]. They are ideally suited to a wide range of consumer electronics applications, including broadband access, home networking, display/projection and digital television equipment.

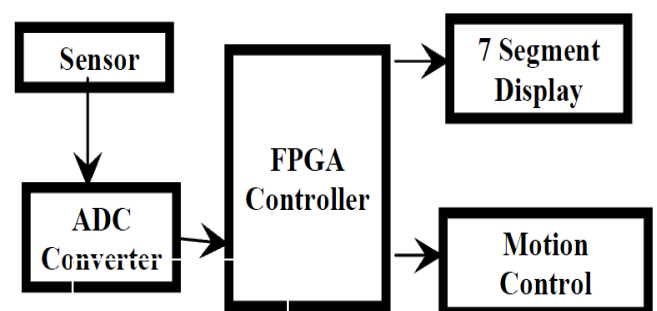


Fig.1: Block Diagram of mobile robot.

FPGA's avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs.

Logic Implementation on FPGA:

Basically two approaches are followed to implement logic on FPGA board. They are Look up table approach and Multiplexer approach.

In look up table approach, the logic is implemented with LUTs. A LUT is basically a memory device with certain capacity. As per the design requirement, the LUT is filled up with specific value to implement the design. The LUT approach is widely used in most of the FPGAs.

As the logic capacity of FPGA increases, synthesis for FPGAs is becoming more important. To efficiently exploit increased logic capacity of FPGAs, synthesis tools and efficient synthesis methods for FPGA targets become necessary. One solution to designing large designs efficiently is to use VHDL synthesis. Several synthesis tools exist for mapping these descriptions to various FPGA families.

FPGA implementation and programming:

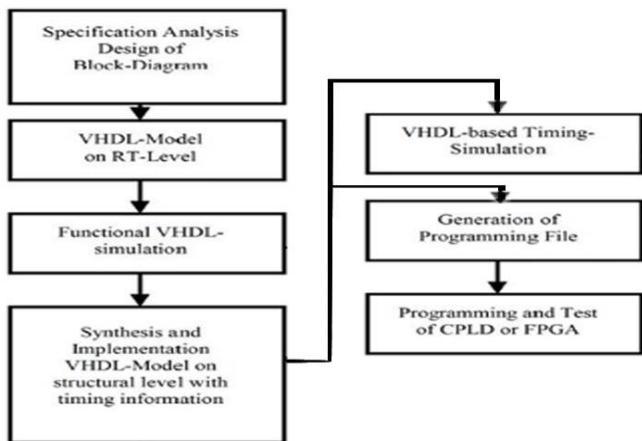


Fig.2: Design flow on FPGA

To define the behavior of the FPGA, the user provides a hardware description language (HDL) or a schematic design. The HDL form is more suited to work with large structures because it's possible to just specify them numerically rather than having to draw every piece by hand.

Once the design and validation process is complete, the binary file generated is used to (re)configure the FPGA. This file is transferred to the FPGA/CPLD via a serial interface (JTAG).

In a typical design flow, an FPGA application developer will simulate the design at multiple stages throughout the design process. Initially the RTL

description in VHDL or Verilog is simulated by creating test benches to simulate the system and observe results. Then, after the synthesis engine has mapped the design to a netlist, the netlist is translated to a gate level description where simulation is repeated to confirm the synthesis proceeded without errors.

Finally the design is laid out in the FPGA at which point propagation delays can be added and the simulation run again with these values back-annotated onto the netlist.

III. IR DISTANCE MEASURABLE SENSOR

The Sharp GP2Y0A21YK sensor shown in Figure is an IR device that can provide distance measurements. This sensor has a shorter range of 10 to 80cm (~ 4 to 32 inches). The measured distance appears as an analog signal at the output.

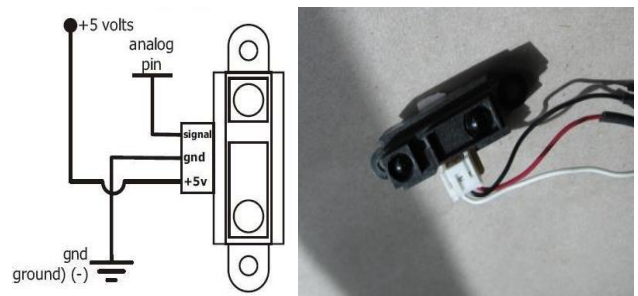


Fig.3: Pin Assignment

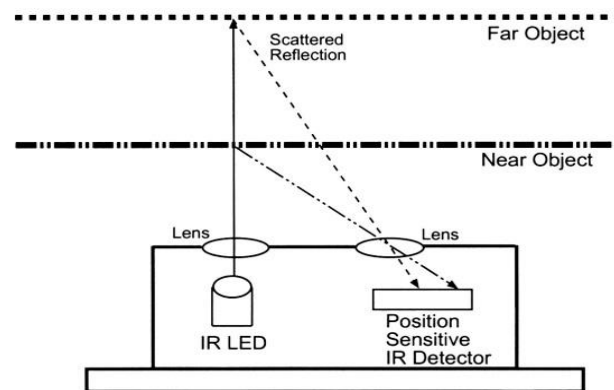


Fig.4: Operation of Sharp IR Ranging Module

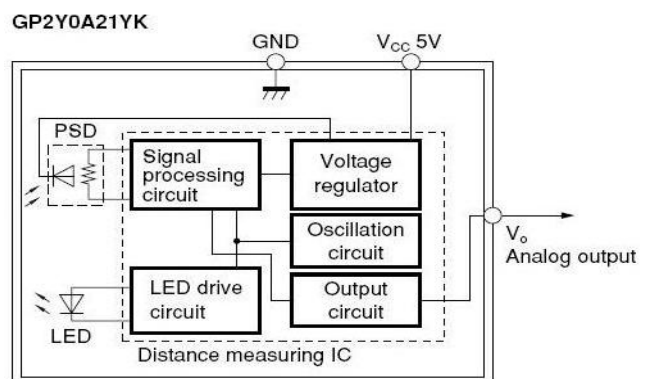


Fig.5: Internal Block Diagram of sensor.

As shown in Figure internally the GP2Y0A21YK contains an IR LED and a position-sensitive IR detector. The IR LED transmits a modulated beam of infrared light.

When the light strikes an object, most of the light will be reflected back to the LED. Since no surface is a perfect optical reflector, scattering of the IR beam occurs at the surface of the object and some of the light is reflected back to the position sensitive detector.

By comparing the near and far object beams shown in figure, it is apparent that position at which the scattered reflected IR beam hits the detector is a function of the reflection angle [7].

Characteristics:

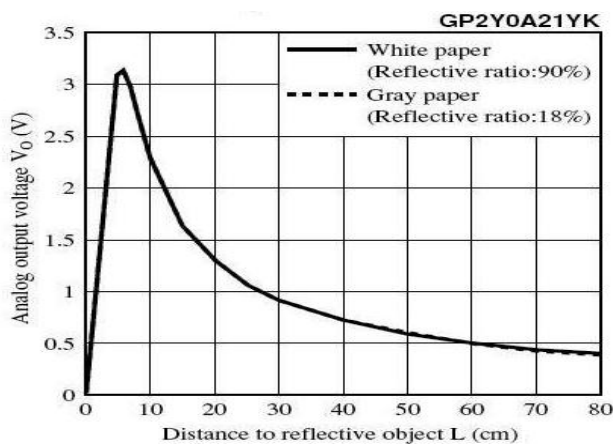


Fig.6: Analog Output Voltage Vs Distance to Reflective Object.

Alignment of sensor:

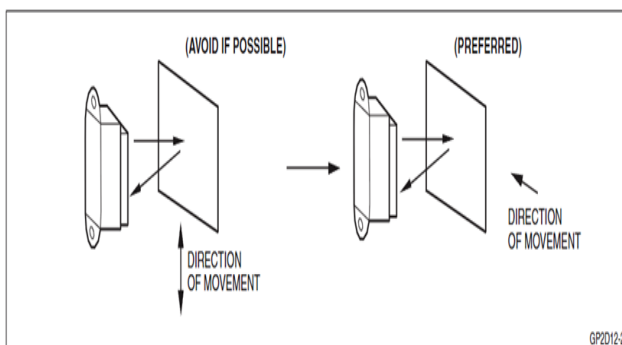


Fig.7: Proper Alignment of Moving Sensor.

IV. DRIVING DC MOTOR

A DC motor requires 9v power supply. The motor will have two lines. Since we cannot connect a DC motor directly to FPGA board, so we use a driver IC.

The driver IC takes the control from board and converts the voltage level to level required to drive the motor. Now the current for the motor is supplied by the IC driver.

A program is written to drive the motor. It is assumed that a switch is connected to digital pin2 and the function of this switch is to change the motor state. Initially the motor is OFF. The motor will run in one direction when the switch is pressed.

L293D Driver IC:



Fig.8: L293D Driver IC.

The diagrammatic representation of L293D is as shown below. L293D is a 16 pin DIP (dual in-line package) IC.

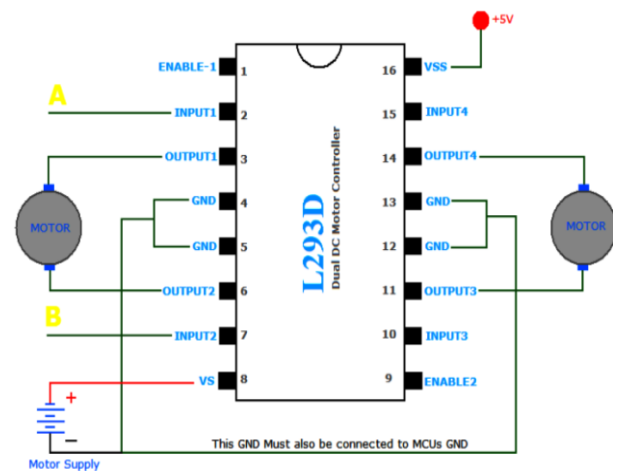


Fig.9: Pin Assignment of L293D IC.

Pin 1 and Pin 9 are enable pins, EN1 and EN2 respectively. The basic purpose of enable pins is to enable a particular channel. EN1 enables the first channel while EN2 enables the second channel. If both the channels are to be used simultaneously, then both the enables pins, EN1 and EN2 should be enabled by applying 5V power supply.

Suppose, to control the left motor which is connected to Pin3 (Output1) and Pin6 (Output2). As mentioned above, it requires three pins to control this motor - Pin1 (Enable1), Pin2 (Input1) and Pin7 (Input2). Here is the truth table representing the functionality of this motor driver.

Table 1: Truth Table for L293D Motor Driver IC

Pin 1	Pin 2	Pin 7	Function
High	High	Low	Turn clockwise (Forward)
High	Low	High	Turn Anti-clockwise (Reverse)
High	High	High	Stop
High	Low	Low	Stop
Low	X	X	Stop

Table2: Logic inputs to activate the L293 chip

Functions	Inputs (EN1 EN2 IN1 IN2 IN3 IN4)
Forward	111010
Reverse	110101
Left	110100
Right	110001

The motor driver can be mounted on the chassis using the connectors as shown below:

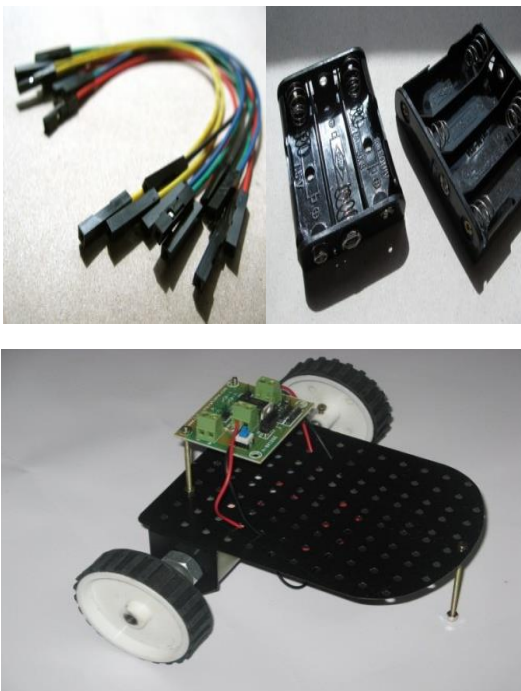
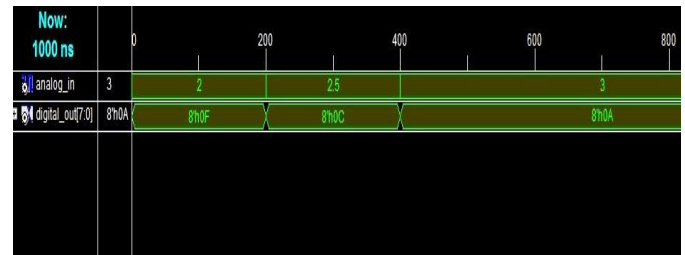


Fig.10: Mounting of L293D

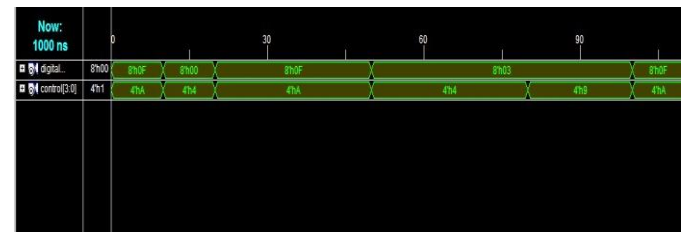
The ADC converter on the Spartan 3E board converts the analog signal from sensor to digital signal which in turn used to display the distance of the robot from obstacle through seven segment and control the motor to drive in a proper way.

V. SIMULATION RESULTS

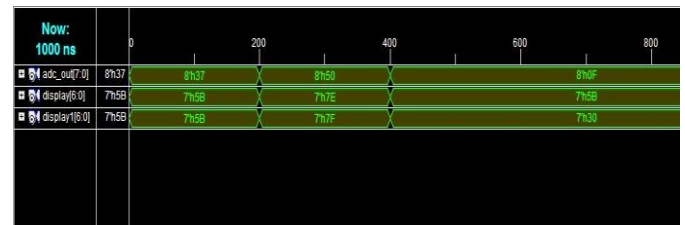
i) Analog to digital converter



ii) Motor Control



iii). Display (seven segment)



iv) Controller

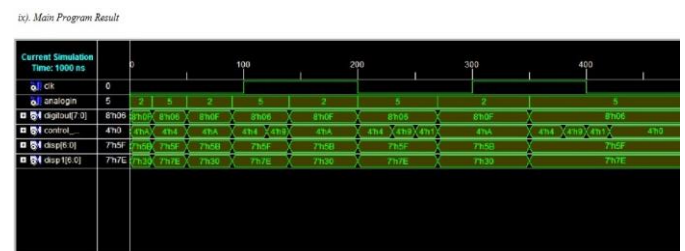


Fig. 9: Simulation Result of Main Program Result.

VI. CONCLUSION & FUTURE WORK

With the advancement in the field of robotics, robots can be used in hazardous condition where the presence of human is unsafe. Autonomous mobile robots can also be used to deliver parts in factories, being complementary platforms in a security system. This paper describes the implementation of a FPGA based controller for a simple mobile robot. This design may be described as a mapping from the input sensors to the actuators which control the robot motions. It is shown that FPGA can be configured to implement the design successfully. The wireless channel may also be added to increase system flexibility.

VII. REFERENCES

- [1] Prabhas Chongstitvatana. "A FPGA-based Behavioral Control System for a Mobile Robot". IEEE Asia-Pacific Conference on Circuits and Systems, Chiangmai, Thailand, 1998.
- [2] Renato A. Krohling, Yuchao Zhou, Andy M. Tyrrell. "Evolving FPGA-based Robot Controllers using an Evolutionary Algorithm". n International Conference on Artificial Immune Systems, pp.41-46, 2002.
- [3] Hannan Bin Aznar, MA, Dimond, K.R. "Design of an FPGA based adaptive neural controller for intelligent robot navigation". In Proceedings of the Euromicro Symposium on Digital Systems Design, pp.283-290, 2005.
- [4] P.H.W. Leong and K.H. Tsoi "Field Programmable Array Technology for Robotics Applications". In IEEE International Conference on Robotics and Bioimetics ROBIO, 2005.
- [5] P. Jin, M.H. Yang, L. Wei, R. Liu, Y.W. Cai, H.G. Liu, H. Seitz, N. Butterfass, J. Hirzinger, G "High performance DSP/FPGA controller for implementation of HIT/DLR dexterous robot hand". In International conference on Robotics and Automation, ICRA, pp.3397-3402, 2004.
- [6] J.M. Rosario, R. Pegoraro, H. Ferasoli. "Conception of Wheeled Mobile Robots with Reconfigurable Control using Integrate Prototyping". Laboratory of Automation and Robotics, Sao Paulo, Brazil
- [7] István Matijevics: Microcontrollers, Actuators and Sensors in Mobile Robots, in Proceedings of 4th Serbian-Hungarian Joint Symposium on Intelligent Systems (SISY206) September 29-30, 2006,Subotica, Serbia.
- [8] Volnie A.Pedroni. "Circuit Design with VHDL".MIT Press, Cambridge, Massachusetts, London, England.
- [9] Chris Lo. "Dynamic Reconfiguration Mechanism for Robot Control Software". Department of Electrical and Computer Engineering, University of Auckland, Auckland, New Zealand.