

Smart Vehicle Control System based on ARM and μ C/OS-II

Shingate S* and Chavan YV

Padmabhooshan Vasantdada Patil Institute of Technology, Pune, Maharashtra, India

Abstract

The ARM7 controller is used in many applications. In this paper it is used as the core controller, to control the entire vehicle. A voice recognition module will be used for human interaction with the vehicle. This module will be at the transmitter side i.e. with the person, which gives the desired commands. The controller used at the transmitter side is PIC controller. This signal will be received by the controller at the receiver end placed on the vehicle for controlling. In controlling mainly four operations will be performed i.e. forward, stop, left, right in this prototype. To provide safety IR sensors will be used which gives feedback at the receiver end whenever there is any obstacle. For real time operation μ cos-ii will be used to enhance the performance of system.

Keywords: Control system; Embedded; LPC2148; PIC 16F876A; Wireless Robot; μ cos-ii

Introduction

Improvements in hardware technology have resulted in low-cost controllers which are composed of a single chip with embedded memory, processor, and peripherals. The advancement in technology is in a rapid progress. New ideas are proposed every time in different sectors. If we consider the automobile field there is a tremendous rise in light and heavy vehicle. Many automobile companies are coming with new ideas in order to increase their sales and to gain top level in market.

ARM architecture has been designed to allow very small, yet high performance implementation. This simplicity leads to very small implementations which allow devices with very low power consumption [1,2].

Now a day's most industries are using this controller to develop their product. One of the examples includes the I-phone 5 mobile which uses ARM 9 processor [3].

ARM is a RISC architecture which has the following features:

1. A large uniform register file.
2. A load-store architecture, where data processing operations only operate on register content, not directly on memory contents.
3. Simple addressing modes.
4. Uniform and fixed length instruction fields.
5. High performance, low code size.
6. Low power consumption and silicon area.

ARM based embedded system has good performance and portability; therefore it has been widely used in various industries. Different operating systems can be ported easily on this controller.

Block diagram

The block diagram of the system is shown in Figure 1 and each of the modules is explained in brief in the following sections.

Modules Used

Raspberry Pi module

The Raspberry Pi board houses an ARM chip at the core. The board is used by many embedded and robotics enthusiast for rapid

prototyping of ideas [4]. The board has support for external storage (used to boot our Linux kernel), USB ports (used for WIFI dongle and camera interfacing), IO ports operated at 3.3 V (used for motor driver control and sensor control) (Figure 2).

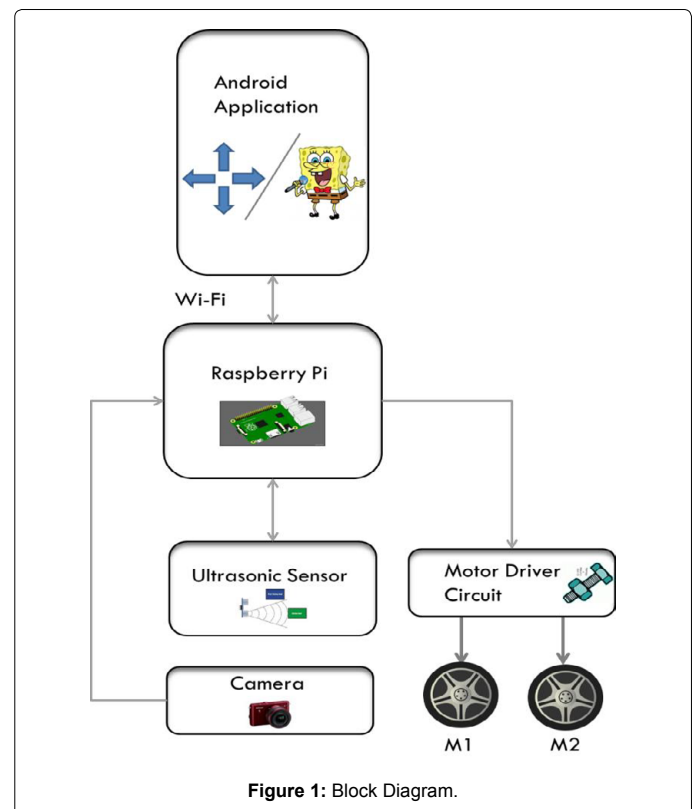


Figure 1: Block Diagram.

***Corresponding author:** Shingate S, Padmabhooshan Vasantdada Patil Institute of Technology, Sangwi, Pune, Maharashtra, India, Tel: +91-9561857462; E-mail: snehashingate@gmail.com

Received June 19, 2015; **Accepted** September 21, 2015; **Published** October 15, 2015

Citation: Shingate S, Chavan YV (2015) Smart Vehicle Control System based on ARM and μ C/OS-II. J Electr Electron Syst 4: 159. doi:10.4172/2332-0796.1000159

Copyright: © 2015 Shingate S, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

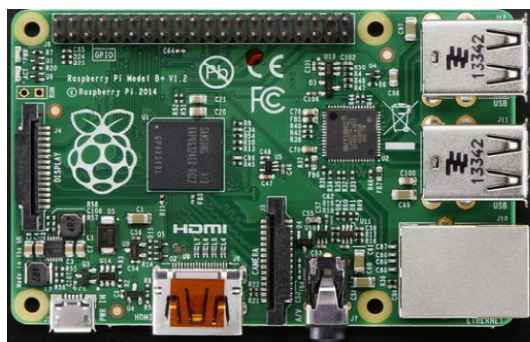


Figure 2: Raspberry Pi Board.



Figure 3: Ultrasonic Sensor.

We have chosen the board as the core because the raspberry pi community is huge and active allowing us access to articles on usage and debugging in abundance. All project requirements are satisfied using the single board.

Obstacle detection module

Ultrasonic ranging module HC-SR04 is used for obstacle detection. The module provides 2 cm-400 cm non-contact measurement function, the ranging accuracy can reach to 3 mm (Figure 3). The module includes ultrasonic transmitter, receiver and control circuit. The basic principle of work:

1. Using IO trigger for at least 10 us high level signal.
2. The module automatically sends eight 40 kHz signals and detects if there is a pulse signal back.
3. The echo pulse pin is kept at high until the signal is back.
4. Distance measurement.

$$\text{Distance} = (\text{high level time} * \text{velocity of sound (340 m/s)})/2$$

Wire connecting direct as following 5 v supply Trigger pulse Input Echo pulse Output Ground.

Motor control module

The module contains L293D at the core. The L293D is a 16 pin IC, with eight pins, on each side, dedicated to the controlling of a motor. There are 2 INPUT pins, 2 OUTPUT pins and 1 ENABLE pin for each motor. L293D consist of two H-bridge. H-bridge is the simplest circuit for controlling a low current rated motor (Figure 4).

The Motor Driver ICs are primarily used in autonomous robotics

only. The raspberry pi board operates at low voltage and requires a small amount of current to operate while the motors require a relatively higher voltage and current. Thus current cannot be supplied to the motors from the board IO pins. This is the primary need for the motor driver IC [5-8].

The L293D IC receives signals from the board and transmits the relative signal to the motors. It has two voltage pins, one of which is used to draw current for the working of the L293D and the other is used to apply voltage to the motors. The L293D switches it output signal according to the input received from the board.

For Example: If the board sends a 1(digital high) to the Input Pin of L293D, then the L293D transmits a 1(digital high) to the motor from its Output Pin. An important thing to note is that the L293D simply transmits the signal it receives. It does not change the signal in any case Figure 5.

Camera module

The JPEG frames are captured using a Logitech camera and motion utility. The server at the receiver end transmits these frames to the client at port 8081.

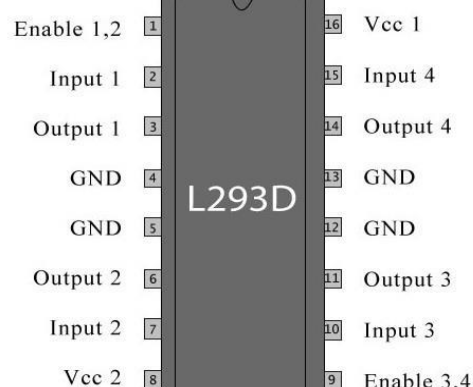


Figure 4: Pin Diagram of L293D.

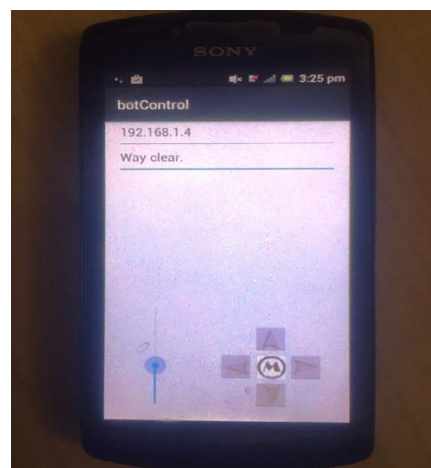


Figure 5: Android Application for Bot control.

Motion, a software motion detector, is a free, open source CCTV software application developed for Linux.

It can monitor video signal from one or more cameras and is able to detect if a significant part of the picture has changed saving away video when it detects that motion is occurring (it can also do time lapse videos).

The program is written in C and is made for Linux (exploiting video4linux interface). Motion is a command line based tool whose output can be JPEG, netpbm files or MPEG video sequences. It is strictly command line driven and can run as a daemon with a rather small footprint and low CPU usage [9,10].

It is operated mainly via config files, though the end video streams can be viewed from a web browser. It can also call to user configurable “triggers” when certain events occur.

4. Video signal transmission, reception and display over controller achieved.

Conclusion and Applications

Transmitter module

The transmitter module contains button controls for left, right, forward, back motion and speed control, a textbox for the obstacle detection status. The module also contains speech recognition for the controls mentioned. The module is implemented as an Android application.

The commands are transmitted to the receiver using UDP packets over WI-FI.

Encoding for the commands is done using JSON.

This system uses voice driven principle which improves human machine interaction and makes the control of the system simple. The use of ultrasonic sensors helps the vehicle to prevent from damage. Use of ARM microcontroller and Linux kernel improves the speed of operations.

The system can be used as a carrier of the mobile robot, residential patrol, bomb detecting and diffusion, site investigation and many other areas.

Software Used

Linux kernel is used for fast, reliable and responsive command processing and sensor interface. Following tasks have been spawned

in the cinit process of the kernel for initialization on boot. Start motion utility for JPEG frame capture and server. Start receiver script for command reception, sensor control. The code is developed using python language. The algorithm contains a total of 3 activities. Each activity is handled in a separate thread. Activities are as follows

1. Command processing.
2. Obstacle detection.
3. Motor control.

A priority queue is used for command processing.

Result

1. Control of vehicle using manual controller and speech recognition achieved.
2. Communication between vehicle and controller over Wi-Fi achieved.
3. Collision detection achieved.

References

1. Gupta RA, Chow MY (2010) Networked control system: Overview and research trends”, IEEE Transactions on Industrial Electronics.
2. Xiong C, Hu J (2012) Design of smart vehicle control system based on ARM and ucos-ii, International Conference on Computer Science and Electronics engineering.
3. Marti P (2010) Design of an embedded control system laboratory experiment, IEEE Transactions on Industrial electronics.
4. Sezaki H (2010) Development of an optical vehicle to grid aggregator for frequency regulation, IEEE Transactions on smart grid.
5. Wolf M (2010) Hardware support for secure processing in embedded systems, IEEE Transactions on Computers.
6. Wang FX, Tan QL, Li QL (2011) Design of the High-Precision Signal Generator Based on ARM, Applied Mechanics and Materials.
7. Zhang GL (2011) Study on Approach of Determining Size of μ C/OS-II Task Stack, Journal of Computers.
8. Wu Z, Wu Q, Cheng H, Pan G, Zhao M (2007) A semantic and adaptive middleware platform for smart vehicles pace, IEEE Transactions on Intelligent Transportation Systems.
9. Nosratini A, Hunter TE, Hedayat A (2004) Cooperative Communication in Wireless Networks, IEEE Communications Magazine.
10. Sandeep SK, Arumugam M (2004) TDMA Service for Sensor Networks, International Workshop on Assurance in Distributed Systems and Networks (ADSN).