

The Wireless Remote Control Car Based On Arm9

V.S.Kumar Reddy Marudi¹, K Joginaidu²

¹Student, ²Assistant Professor

¹ Ece Department, Mallareddy Institute Of Engineering And Technology.

² Ece Department, Mallareddy Institute Of Engineering And Technology.
Hyderabad, Ap, India.¹, Hyderabad, Ap., India².

Abstract: The Internet of Things (IoT) are of great importance in promoting and guiding development of information technology and economic. At Present, the application of the IoT develops rapidly, but due to the special requirements of some applications, the existing technology cannot meet them very good. Much research work is doing to build IoT. Wi-Fi based Wireless remote control has the features of high bandwidth and rate, non-line-transmission ability, large-scale data collection and high cost-effective, and it has the capability of video monitoring, which cannot be realized with RF. The research on Wi-Fi based remote control car has high practical significance to the development of the Internet of Things. Based on the current research work of applications the characteristics of Wi-Fi, this paper discusses controlling the car by using Wi-Fi module along with the conditions can be monitored through remote PC or Lap top which supports Wi-Fi technology. In PC or Lap top two tabs are present. In the first tab we can monitor the conditions and in the second tab four buttons are present to control the car in forward, back ward, left side and right side directions.

Keywords: S3C2440 (ARM9), Wi-Fi Module, Camera, DC motors with driver IC and laptop with Wi-Fi module.

I. Introduction

In recent years, the applications of mobile robot have gradually become more diverse, which makes the robot closer to people's daily life. At present, the middle and small scale motion robot are usually designed based on single chip microcomputer without operating system, such as FIFA/RoboCup middle and small scale football robot, bionic snake robot, multi-foot crawling robot and so on. In order to make the research results of the robot can actually be used in solving more realistic problems, the increasingly complex control process and some special application which is called for high real-time requirement has presented challenges to the existing motion robot design pattern [1].

The performance bottlenecks of the intelligent mobile robot focused on the following three aspects. The First contradiction is between the development cost and the hardware performance. In order to reduce the cost of the robot's hardware in design, we often use cost-effective hardware to build the robot system. However, when the controls became more complex, precise increase gradually, the performance of the original hardware would be inadequate. Rebuild the robot will not only increase the cost of inputs, but also waste the robot's life cycles seriously. The next contradiction is between complexity of controls and the real-time performance. When the hardware performance is insufficient to complete the control of the higher complexity task, the system often sacrifices time to compensate the lack of hardware performance. So that it brings a serious problem in real time. The third contradiction is different fields use platform is difficult to unity. When the mobile robot used in different occasions, as the users of the technology from different backgrounds, different habits, a lot of software and hardware tools due to the limited of hardware performance, or the interfaces is not enough will cause difficulties for practical use.

II. System Design Model

A. Software design module

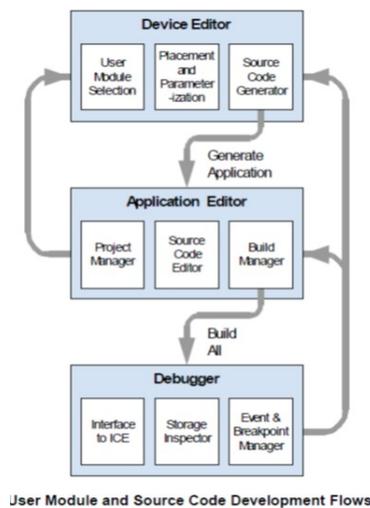
The development process for the PSoC device differs from that of a traditional fixed function microprocessor. The configurable analog and digital hardware blocks give the PSoC architecture a unique flexibility that pays dividends in managing specification change during development and by lowering inventory costs. These configurable resources, called PSoC Blocks, have the ability to implement a wide variety of user-selectable functions.

Each block has several registers that determine its function and connectivity to other blocks, multiplexers, buses, and to the IO pins. Iterative development cycles permit you to adapt the hardware as well as the software. This substantially lowers the risk of having to select a different part to meet the final design requirements. To speed the development process, the PSoC Designer Integrated Development Environment (IDE) provides a library of pre-built, pre-tested hardware peripheral functions, called "User Modules." User

modules make selecting and implementing peripheral devices simple, and come in analog, digital, and mixed signal varieties. The standard User Module library contains over 50 common peripherals such as ADCs, DACs Timers, Counters, UARTs, and other not-so common peripherals Such as DTMF Generators and Bi-Quad analog filter sections. Each user module establishes the basic register settings that implement the selected function. It also provides parameters that allow you to tailor its precise configuration to your particular application.

For example, a Pulse Width Modulator User Module configures one or more digital PSoC blocks, one for each 8 bits of resolution. The user module parameters permit you to establish the pulse width and duty cycle. User modules also provide tested software to cut your development time. The user module application programming interface (API) provides high-level functions to control and respond to hardware events at run-time.

The development process starts when you open a new project and bring up the Device Editor, a graphical user interface (GUI) for configuring the hardware. You pick the user modules you need for your project and map them onto the PSoC blocks with point-and-click simplicity. Next, you build signal chains by interconnecting user modules to each other and the IO pins. At this stage, you also configure the clock source connections and enter parameter values directly or by selecting values from drop-down menus.



The last step in the development process takes place inside the PSoC Designer’s Debugger subsystem. The Debugger downloads the HEX image to the In-Circuit Emulator (ICE) where it runs at full speed. Debugger capabilities rival those of systems costing many times more. In addition to traditional single-step, run-to-breakpoint and watch-variable features, the Debugger provides a large trace buffer and allows you define complex breakpoint events that include monitoring address and data bus values, memory locations and external signals.

B. Hardware design module
BLOCK DIAGRAM

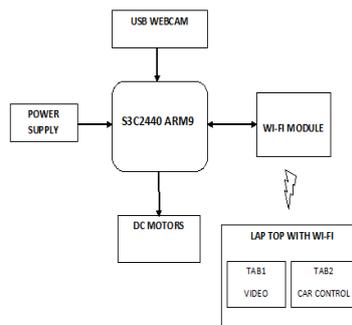


Figure: experimental set-up block diagram

The applicationoftheIoT develops rapidly, but due to the special requirements of some applications, the existing technology cannot meet them very good. Much research work is doing to build IoT. Wi-Fi basedWirelessremote control has the features of high bandwidth and rate, non-line-transmission ability, large-scale data collection and

high cost-effective, and it has the capability of video monitoring, which cannot be realized with RF. The research on Wi-Fi based remote control car has high practical significance to the development of the Internet of Things. Based on the current research work of applications the characteristics of Wi-Fi, this paper discusses controlling the car by using Wi-Fi module along with the conditions can be monitored through remote PC or Lap top which supports Wi-Fi technology. In PC or Lap top two tabs are present. In the first tab we can monitor the conditions and in the second tab four buttons are present to control the car in forward, back ward, left side and right side directions.

III. Experimental Results

The designed embedded remote control vehicle. After platform construction is completed, we integrated test the whole system through a simple gesture recognition experiments. Figure 8 shows the interface of the system console. Due to the obstacle refraction will affect the wireless signal strength the wireless communication distance is approximate 13 meters indoor and about 180 meters outdoor in open area. Video frame rate is 8.62fps-13.16fps (as the wireless signal status). The average recognition time of gesture recognition is 1.52 seconds, the system real-time within an acceptable range.



Figure: experimental set-up implementation.

IV. Conclusion

The project “the wireless remote control car based on arm9” has been successfully designed and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced ARM9 board and with the help of growing technology the project has been successfully implemented.

References

- [1] Volker Graefe and Rainer Bischoff. From Ancient Machines to Intelligent Robots -A Technical Evolution[C]. The Ninth International Conference on Electronic Measurement & Instruments. 2009. Vol.3, 418-431.
- [2] Liggesmeyer P., Trapp M.. Trends in Embedded Software Engineering[J]. IEEE Software, 2009, 26(3):19-25.
- [3] E. D. Lagnese and D. E. Thomas, Architectural partitioning for system level design[C], in Proc. 26th DAC. June 1989, 62-67.
- [4] ATMEL Corporation. ATmega128 Preliminary Summary [EB/OL] (2010-06-21) [2011-01-13]. http://wenku.baidu.com/view/9a4bb6fb84_ae45c3b358c5d.html.
- [5] Kar-Keung D. Young, Yong QuanOu, LunHuaiCai, Jason Kam On Ho and Ken Kin Man Cheng. Real Time Embedded Control System Development for Wireless Mobile Platforms[C]. Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on. 2008.11.18,2022-2027.
- [6] JasminBlanchette, Mark Summerfield. C++ GUI Programming with Qt4(Second Edition) [M]. BeiJing: Electronic Industry Press. 2010.4:11-12.
- [7] AlanCox.Video4LinuxProgramming[EB/OL],(2000)[2011-01-20]. alan@redhet.com.