2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)
Nov. 23-26, 2011 in Songdo ConventiA, Incheon, Korea

Implementation of Smartphone Environment Remote Control and Monitoring System for Android Operating System-based Robot Platform

Sung Wook Moon [1] ,Young Jin Kim[2], Ho Jun Myeong[3] , Chang Soo Kim[4], Nam Ju Cha[5], and Dong Hwan Kim [6+]
[1,3,6] Department of Mechanical & Automation Engineering, Seoul National University of Science and Technology, Seoul, 139-743, Korea
(Tel : +82-2-970-6362; E-mail: dhkim@seoultech.ac.kr)
[2] Nano Manufacturing Research Center, Seoul National University of Science and Technology, Seoul, 139-743, Korea
(Tel : +82-2-949-0744; E-mail: trocad@seoultech.ac.kr)
[4,5] DUNET, Dong-a Daily News Bldg., Chungjeongno 3-ga, Seodaemun-gu
Seoul, 120-715 Korea
(Tel : +82-2-361-1973; E-mail: moon0529@dunet.co.kr)

Abstract – In this work, Android operating system based robot platform and smart phone operated control and monitoring system are introduced. To implement this work, Cortex-A8 series S5PV210 embedded processor and Android operating system are correlated. The robot has an autonomous and manual travel and is controlled by only smart phone. In the Android OS (Operating System), the camera image is compressed to JPEG format and the image file is delivered to a smart phone through 802.11x wireless LAN communication which utilizes TCP/IP communication socket programming. Later, the transferred image data are converted into BMP format, which enables a real time image display.

Keywords – Android operating system, Caretex-A8, embedded, Smartphone, JPEG, TCP/IP, Image

1. Introduction

In recent time, information technology tends to be more smart and human-friendly. The main theme is the appearance of smart phone. The smart phone prevails in human life and a human-assisted robot also takes the smart phone as a main processor. Traditionally, a robot should possess a high performance computer or microprocessor as a central controller. However, a PC or microprocessor makes the robot bulky or limited in computation. As a smart phone appears in market, the robot controller has tremendous change in design and installment. As we know well, the smart phone has remarkable power in computation as well as a very convenient operation such as camera monitoring or wireless internet access by WIFI. Various application programs are open to market or developed by a robot designer. With the help this powerful smart phone applications, the robot becomes smarter and more intelligent [1]. With a smart phone being developed and popularized, the robot product becomes also commercialized. One of the typical robot applications is iPOLiS mobile made in Samsung Techwin [2]. This iPOLiS robot application program is used by a simple haptic operation to operate a CCTV remotely which has pan, tilt, and zoom operations. Without an extra control kit manipulation, users simply download the application program to a smart phone, operating the CCTV. The CCTC is a kind of robot in the aspect of motor operation and control which are basic components in a robot system.

New concept of network based robot is called URC (Ubiquitous Robotic Companion). It means a robot providing a service at any place and time with users. The URC technology utilizes a network, expanding services the robot can do. Also, it solves cost problem when all functions which should be equipped with a robot in the traditional manner can utilize a network. With this network, the functions are allocated to other agents [3] to increase efficiency. Instead of implementing complicated functions by installing a high performance processor, all functions are carried out by accessing relatively simple remote servers. This is an example of using URC technology [4].

In present work, a robot system was designed by integrating a smart phone technology and URC. The commands for robot operations and camera image monitoring were implemented using the developed smart phone application program. We selected Android OS for the robot operation system and loaded it on smart phone.

As we know well, Android OS is an open source and lots of driver programs for camera, WIFI, microphone, and display have been already open to users or developers, which makes robot development easier and less costly [5].

2. System structure and Embedded Android system

2.1 System structure

In this work, new robot system is introduced, which controls the robot by smart phone, and monitors the environment delivered from the camera on the robot in real time. Fig. 1 shows the developed robot system structure.

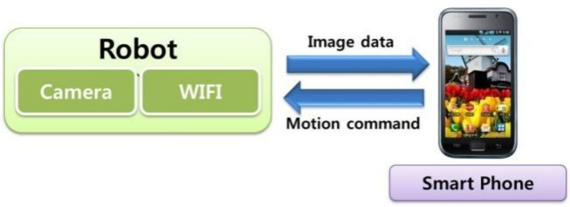

Fig. 1 System structure

+Corresponding author: dhkim@seoultech.ac.kr

In order to monitor the image from the camera mounted on the robot in real time, which is captured on the smart phone the image data should be compressed, later being delivered to the smart phone through WIFI. Also, the basic UI (User Interface) was designed to control the robot. The commands for robot operation which are presented on the phone screen are transferred back to the robot through WIFI. The commands sent to the robot are received at S5PV210 chip which is a kind of embedded PC mounted on the robot body, later being sent to DSP TMS320F2808 by a serial communication for real time control. All robot operations and sensor inputs are executed and analyzed in this DSP.

2.2 Controller design

In order to port Android platform, S5PV210 chip from Samsung Ltd which is an ARM Cortex-A8 based application processor was adopted in the robot. S5PV210 has been used in a Samsung Tablet PC and has 1 GHz speed and second cache for fast processing [6]. Fig. 2 shows the schematic diagram of S5PV210 structure.

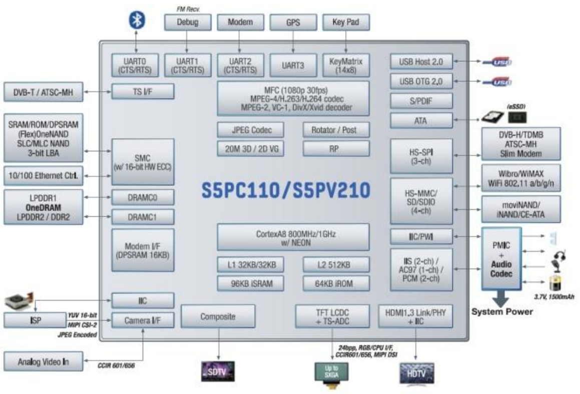

Fig. 2 S5PV210 structure

Since S5PV210 chip constitutes with interfaces for camera, audio codec, LAN (Local area network)-controller, communication, and micro SD-card, it is easy to equip all sorts of devices in one module. Also, with the help of MMU (Memory management unit) in Cortex-A8 series processor it is easy to port OS such as Android or Linux.

The android platform installed on S5PV210 chip plays a role in managing overall system in the robot. However, Android OS does not guarantee RTOS (Real-Time Operating System), which limits a real time control. For real time control, a special chip DSP is selected and was put on the designed board. With this chip a motor driving and sensor measurement could be implemented in real time. A DSP TMS320F2808 from Texas Instruments was selected. Between TMS320F2808 and S5PV210 a serial communication is connected. In other words, DSP plays a role of slave, executing robot commands from S5PV210. Fig. 3 represents the controller structure inside the robot system.

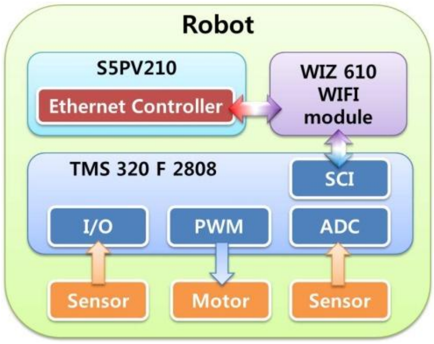

Fig. 3 Controller structure

The TMS320F2808 has ADC (Analog-Digital converter) for sensor measurement and PWM (pulse with modulation) and QEP (Quadrature Encoder Pulse) for motor control, and serial communication interface for communication with other devices.

On the other hand, an Ethernet controller chip for using Ethernet, audio codec for microphone and speaker, WIFI receive module for using WIFI, and 1.3Megapixel camera for image acquisition are all equipped in S5PV210.

2.3 Android Platform

In an embedded OS, Linux, Windows-CE, and Android are well known OS. In our robot system, Gingerbread version as an Android Platform was adopted. Android is a JAVA language based OS, enabling C/C++ language compatibility by JNI (Java Native Interface). As known well, Android OS provides free SDK (Software Development Kit) which operates camera, speaker, audio, and Ethernet and it also has a merit of open source [7]. Furthermore, in an integrated development environment called Eclipse an Emulator [7] is provided. Thus, we utilized this program without a smart phone or target board at first stage.

Fig. 4 shows an Android system architecture which operates on Linux Kernel. In order to input Android to the embedded board, it is of importance to load image file and binary file created after compiling Android, Linux Kernel and Boot Loader to NAND Flash memory or SD-card. Later, booting the system is completed.

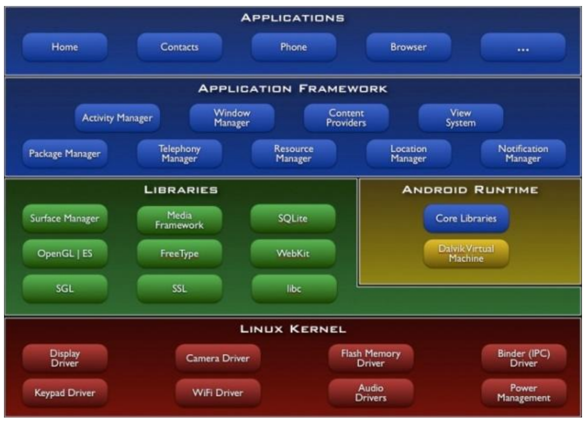

Fig. 4 Android system architecture

In the robot, a mango210 board from crz-technology Ltd was selected and a S5PV210 chip was put on this board. Since this board supports BSP (Board Support Package) we developed the robot control board by integrating Android, Linux kernel, and Boot Loader. The Android loaded in the board is version 2.3 Gingerbread. Kernel version 2.6.35 and u-boot (Universal Boot Loader) as a Boot loader was employed. Even though the provided BSP already has compiled binary file and image file, we directly compiled the source in Linux because we had to modify correct Android system file and Linux kernel.

2.4 Robot mechanism

The developed robot resembles a spider, possessing six legs, being close to the commercial robot [8]. To be an intelligent robot, several sensors, microprocessor, and vision camera are employed in the body.

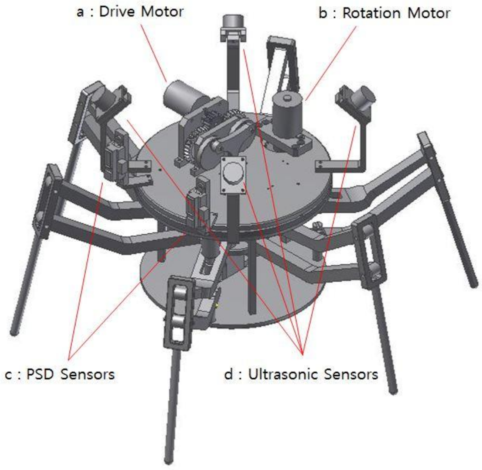

Fig. 5 Robot layout

The robot body is made of ABS (Acrylonitrile Butadiene Styrene) to keep small weight. Other parts which require enough stiffness used steel and aluminum (STS304 or Al6061). Table 1 shows the specifications of the robot.

Table 1 Specifications of robot

Size (W x D x H)	300 x 300 x 220 (mm)
Weight	1.5 (kg)

Two PSD sensors for detection of front objects and four ultrasonic sensors are employed, respectively. Fig. 6 represents the manufactured hexapod robot where sensors, controller, WIFI module, and embedded controller are mounted.

Fig. 6 Developed hexapod robot

3. Image process and transfer

3.1 Image compression

In order to receive camera image frame data from Android OS, the function onPreviewFrame() in Previewcallback class was used [7]. The data frame format delivered from Previewcallback is compressed by YUV420 format. Here, Y denotes luminance, and U and V represent chrominance, and 420 after YUV represents pixel ratio [9]. In other words, the 420 indicates that four Y-components accompany one U-component and one V-component each.

The image size acquired from the camera is around 60-80k bytes, and this large data size makes image transfer by a smart phone difficult without manipulating YUV image signals. Because WIFI communication is utilized between the smart phone and robot, it is more preferable to keep less small size of image data. Therefore, a data format with less image data size is highly recommended.

We converted the YUV image data to JPEG format, yielding the image down to 2k bytes level. The process of converting from YUV420 data into JPEG format is illustrated in Fig. 7.

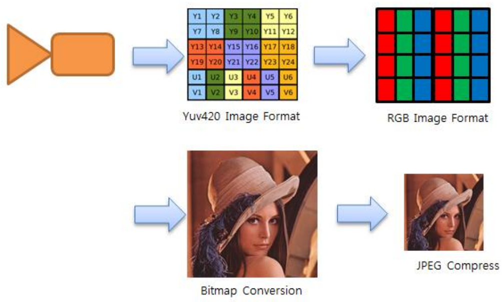

Fig.7 Process of image image conversion from YUV420 to JEPG format

We utilized the well known formula [10] to convert from YUV to RGB data as follow.

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.407 \\ 1 & -0.345 & -0.717 \\ 1 & 1.779 & 0 \end{bmatrix} \begin{bmatrix} y \\ u-128 \\ v-128 \end{bmatrix} \quad (1)$$

Next, the RGB data is compressed into JPEG by employing functions provided in Android OS, resulting in small size JPEG image data. In Fig. 8 the functions and commands are shown for data compression.

```
Bitmap bitmap = Bitmap.createBitmap(width, height, Bitmap.Config.RGB_565);
bitmap.setPixels(bBitmapInt, 0, width, 0, 0, width, height);
bitmap.compress(Bitmap.CompressFormat.JPEG, 50, outstream);
```

Fig. 8 Data compression from RGB to JPEG

3.2 Image transfer and reception

The image transfer from robot to smart phone takes the following procedures. First, set the robot a server, setting the smart phone a client receiving the image, which is a kind of a server-client socket communication system regarding that a client accesses to a server. In Fig. 9, the communication structure done in this work is illustrated, showing actions and data flows.

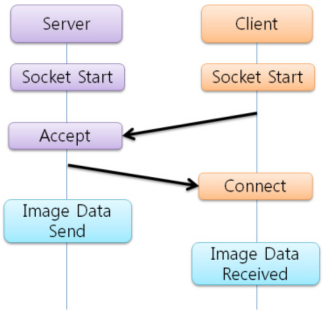

Fig. 9 Client-Server socket structure

When the program executes, the socket is generated in the robot and waits until the client accesses. Once the client tries to access, the server permits the access through the function "Accept". The server acquires the camera image after the server accepts, transferring the compressed image data to the client finally.

4. Experimental results

The proposed smart phone controlled robot system was implemented in real mode. The received video image from the robot camera is diplayed on the slart phone as shown in Fig. 10.

Fig. 10 Video image transfer and reception on smart phone

As a result, it is proved that the video image taken in the robot camera is diplayed in real time and also the control commands from the bottons on the phone are transferred to the robot, moving or stopping the robot without any technical difficulties. There was only 0.2 second time delay between the camera and phone in image delivery. This amount of time delay is almost negligible to human eyes. The other commands for robot movement was successfully implemented.

5. Conclusions

The developed smart phone operating robot system was successfully executed with the help of correlating Android OS and embedded computer. Operation commands as well as images were well delivered and received without severe time delay. All operations were verified enough to be commercially employed. With this success, any kind of Android OS based robot control and monitoring in real time can be impemented and various applications will be extended in the short run.

References

[1] Young-Hoon Jeon, Hyunsik Ahn "Smart-Phone Based Interface for Mobile Robot Control" The Institute of Electronics Engineers of Korea 2010 Annual Summer Conference, Vol. 33, No. 1, pp. 1951~1953, 2010.
[2] http://www.samsungtechwin.co.kr/
[3] Sang-Rok Oh "Network-based Intelligent Service Robot : Ubiquitous Robotic Companion" Communications of Korean Institute of Information Scientists and Engineers, Vol. 23, No. 2, pp. 48-55, 2005.2
[4] Hyun Kim, Young-Jo Cho, Sang-Rok Oh "URC(Ubiquitous Robotic Companion)" Communications of Korean Institute of Information Scientists and Engineers, Vol. 24, No. 3, pp. 5-11, 2006.4
[5] Seokhoon Ko "Android Platform Trend" Communications of The Korea Contents Association, Vol. 8, No. 2, pp45~49, 2010.6
[6]http://www.samsung.com/global/business/semiconductor/support/brochures/downloads/systemlsi/S5PV210.pdf
[7]http://developer.android.com/index.html
[8] Innovation First Labs, Inc. http://www.hexbug.com/
[9] Lujun Yuan, Guobin Shen, Feng Wu, Shipeng Li, Wen Gao, "Color Space Compatible Coding Framework for YUV422 Video Coding" ICASSP, Vol. 3, pp. 185-188, 2004.
[10] Ning Xu and Yeong-Taeg Kim, "Luminance Preserving Color Conversion for 24-bit RGB Displays" Consumer Electronics, 2009. ISCE '09. IEEE 13th International Symposium on, pp. 271 - 275, 2009.