Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp104-110)

A Wireless LAN-Based PDA Control Platform for Security Robots

HONGWEI LI, CHANGJIU ZHOU
School of EEE
Singapore Polytechnic
500 Dover Road
SINGAPORE 139651

Abstract: - Advances in the development of Personal Digital Assistants (PDA's) have allowed many users to realize the full potential of using this technology. The uses for such a technology are vast and exciting indeed to cover a broad spectrum of applications. For instance, in the near future it is very conceivable that one can use a PDA with a network connection to control home or office devices from any remote location in the world, through wire or wireless medium. On the other hand, the recent development in wireless LAN technology has boosted up such application by equipping convenient wireless network connectivity on the PDAs with fast transmission speed on a universal license- free frequency.

In this research project, we have developed a low cost universal control platform, which is capable of wirelessly commanding and controlling security robots developed in our school by using a PDA through a wireless LAN. The wireless LAN-based platform equips the robots with security features such as surveillance and reconnaissance capabilities by enabling the robots to capture video information and send back to the server wirelessly.

Key-Words: - Wireless LAN, PDA, security robot, action ID, VB.NET, listening socket, service socket, server/client, control platform

1 Introduction

A traditional security system is fixed and requires cameras mounted on the spots which come with wiring connection and can only turn within a limited angle for surveillance. Our solution combines wireless and robotic technologies to successfully develop a mobile security system mounted on an omni-wheeled robot based on a PDA control platform.

Wireless network technology has started to invade our living and working environments. The Wireless LAN-Based PDA Control Platform developed can abe used to develop security robot, service robot and intelligent home robot, which are expected to be the major drives to improve job efficiency and quality of life. Robots can offer variety of services to humans at home as guide, servant, security guard, assistant for elderly, etc. They could also entertain humans as friends, pets or as game players/partners.

There’s great potential to use potable device and Wireless LAN technology in robotic control. Real-time stringent control and high bandwidth data transfer are conflicting requirements to a single processor. In our wireless LAN-based PDA control platform, a division of labor has been distributed.

1. The PIC-based controller is programmed to concentrate on real-time control and low-level interface to actuators and sensors on the robots.
2. The PDA is programmed to concentrate on supervisory control and wireless communication with high bandwidth required by real-time video transmission.

This implementation has many advantages:

1. The PIC-based controller can be optimized to meet real-time requirements.
2. The supervisory control in the PDA is highly configurable using touch screen and stylus GUI interface.
3. The Wi-Fi and Bluetooth capabilities that come with the PDA offer a quick turn-key solution to wireless communication.
4. The dimension and weight of a PDA plus a PIC-based controller is sufficiently small and light to fit into most toy-sized robots.

In short, the Wireless LAN-Based PDA Control Platform has suggested a viable solution for the remote control of robots requiring real-time control and high bandwidth of data transfer.

2 System Hardware Configuration

The overview of system hardware configuration is depicted in Fig.1.

A control and communication platform based on 802.11b Wi-Fi Wireless LAN technologies.

1. A Pocket PC carried on the robot provides serial or USB interfacing with a PIC controller board.
2. Client/Server software developed using Visual.NET Framework: to establish TCP/IP connection to remote base-station, convey the control and communication, manage the bandwidth for audio and video transmission.
3. A web camera on the robot: transmits video through the Wireless LAN and enable the vision for the robot with surveillance capability.
4. RADIUS server and Access Points: provide authentication to users and extend the control distance.
5. Proxy server acts as a gateway to the Internet and hosts a control web-server via the Internet.

A new research line, namely security robot, has been initiated to exploit the application of robots in the areas of security, rescue and military reconnaissance.

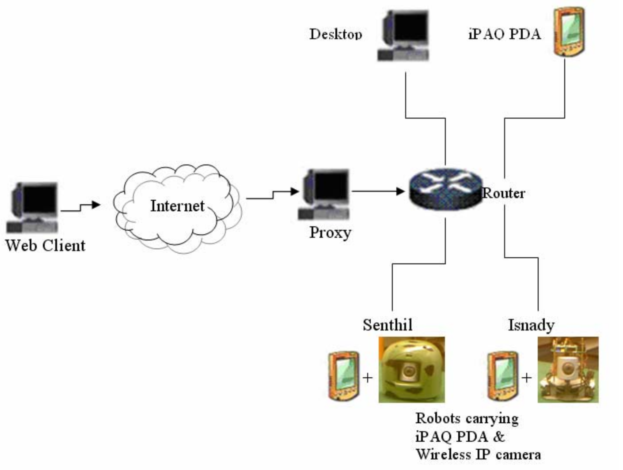

Fig.1 System hardware configuration

IP_address allocation table:

Device	IP Address	Subnet Mask	SSID
Router	192.168.1.1	255.255.255.0	pda
Desktop	192.168.1.100	255.255.255.0	pda
iPAQ PDA	192.168.1.7	255.255.255.0	pda
iPAQ PDA on Senthil	192.168.1.15	255.255.255.0	pda
Wireless IP camera on Senthil	192.168.1.5	255.255.255.0	pda
iPAQ PDA on Isnady	192.168.1.16	255.255.255.0	pda
Wireless IP camera on Isnady	192.168.1.6	255.255.255.0	pda

Table 1 IP_address allocation table

The desktop, HP *iPAQ* PDAs and wireless-LAN cameras are connected to the router and the proxy server wirelessly.

Fig.2 The mobile security system (security robot)

3 Network protocol

The communication model used between the robots, the control server and all the other accompanying clients is TCP/IP over Wireless LAN. TCP/IP is a mature protocol with the following features:

1. It is reliable and connected-oriented
2. Acknowledgement with re-transmission of error packets
3. Widely supported in most networking devices

A client/server model is adopted in the programming for the following reasons:

1. The server/client model for wireless network has been well in place for years and has been rapidly deployed in many environments, thereby proving the ease of deployment.
2. Efforts in developing the peer-to-peer model for wireless network are still in the infancy stage and at the best still under testing. This makes the peer-to-peer model unsuitable for the immediate deployment in the near future, although highly desirable.
3. A centralized model is highly desired in this case for monitoring and troubleshooting purpose

Therefore, devices in the network are divided into two categories: the server (The control server) and the clients (PDAs, PCs and Devices supporting an internet browser). In the server device, effective communication is handled by two network sockets, namely the *listening socket* and the *service socket*. The *listening socket* is first created by the server application. When it is started, it will help the server application to listen for clients. If any attempt is made by the client devices to connect the server, it will choose whether to accept the connection. If it does accept the connection, a *service socket* would be created. The *service socket* would be used to maintain and service that dedicated connection with the particular connected client device. So, it is usual for the server to have multiple *service sockets* when there are multiple client devices connected to the network.

In the event of disconnection by the clients, the *service socket* serving that particular link would close the link and destroy itself from the memory. In the case of the server application being closed, all outstanding links with the clients would be destroyed and the accompanying *service sockets* would be destroyed as well. After that, the *listening socket* would cease to function and will also be destroyed before the application exits.

In the client device, only the *service socket* is required for network communication. It is created at the beginning of the client application. If required, it will try to establish a network connection to the server device through TCP/IP socket. If the server device accepts the connection, a link would be established between the *service socket* of the clients and the *service socket* of the server. This link would be essential as this is the only link for the devices to send/receive data to/from each other. In the event of disconnection, the *service socket* on each side would close the link and destroy itself from the memory.

4 System software configuration

The following diagram in Fig.3 shows the GUI of the control platform:

Fig.3 GUI of control platform

All software that used in the various devices is written in VB.NET, a Microsoft.NET language.

VB.NET was primarily chosen because of the ease in programming, leading to significant reduction in development time. Also, thanks to the .NET framework (and varieties of the framework such as the .NET compact framework and the mono framework), it can be rapidly deployed on clients with various platforms (even non-Microsoft) and non-PC devices, such as hand phones and PDAs.

4.1 Robot Control

This is common among all devices in the network. As mentioned before, all communication between the devices is done using the TCP/IP protocols, through the *service sockets* of the client application and the server application.

At the network frame level, data is exchanged between the clients and the server using the format of text string with the structure depicted in Fig.4:

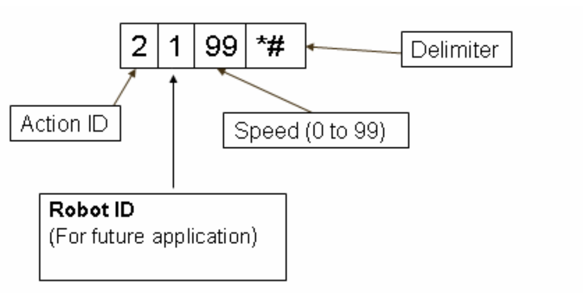

Fig.4 Format of robot-control frame

The action ID refers to the action that is to be performed by the robot. The table below lists the action IDs that correspond to all pre-programmed actions:

Action ID	Actions to be preformed by the robot
1	Move forward
2	Move Backward
3	Move Leftward
4	Move Rightward
5	Turn Leftward
6	Turn Rightward
7	Stop/Halt

Table 2 Robot action ID

The robot ID is an optional feature for future and is not currently implemented.

The speed refers to the velocity of the movement by the robot in any given direction with a scale of 0 ~ 99.

The delimiter stands the end of the data packet.

For example, a data string “2199#*” would mean that the robot is to move backward at the speed of ‘99’.

Upon receiving the control packet, the server application would run a function to convert it into the serial command word in byte and then send directly to the PIC controller of the robot through the serial interface.

The serial command word format is as follows:

65	**51**	**Condition**	**Speed**

Where the value of Condition meets:

1: Forward
2: Backward
3: Left
4: Right
5: Turn left
6: Turn right
7: Stop

The Speed ranges from 0 to 99. If the Speed is set to 0, the robot will stop.

4.2 Multimedia Communication

A value-added feature to the robot would be its ability to capture real-time image and video of its surrounding, using a camera.

A D-link DCS 900w wireless LAN camera was programmed to carry out the video communication through the same wireless LAN link. Any client device with a network connection and a web browser is able to receive the video stream.

A function is included on the desktop client to save the image and video information.

A link test shows the transmission throughput in the following Fig.5:

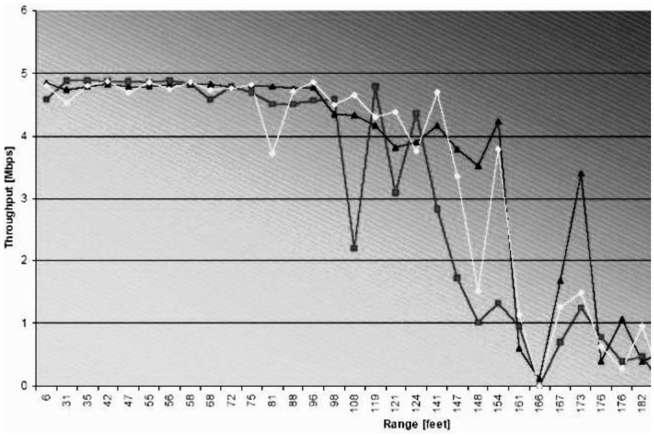

Fig.5 Performance test of Wireless LAN link

4.2.1 Image Saving

Unlike a typical USB web camera which utilities the graphic API of an OS for image and video function, the D-link DCS 900w wireless IP camera is a self-contained embedded wireless Ethernet device with a built-in web server. While USB web camera is dependent on the host PC for everything, the D-link DCS 900w wireless IP camera is able to capture images and stored them in a memory buffer built in the web server.

Therefore, the only way to download those images is through the web server. All we have to do is to simply request for the image from the web server of the camera, which is stored in the memory buffer. Since the web server is a standard http server, we have to first create an http request for the image. Then, we need to define the method on how the server is to respond to the request, which in this case is a ‘GET’ request. Then, the program would have to first convert the received image to a generic image format before asking the users to choose what graphic file format to save.

4.2.2 Video Saving

Officially, the ability to save the streaming video is only available through the manufacturer’s proprietary management software. However, we managed to come up with an alternative way of saving the streaming video.

The key to that is file manipulation and using the multimedia API of the OS. The basic idea is to disassemble an AVI video file to bmp images and re-assembled the bmp images into an AVI file.

As mentioned earlier, the D-link DCS 900w wireless LAN camera is able to capture images and stored them in a memory buffer in the web server. It can do capturing at a rate of 30fps per second.

Firstly, we use a programming loop to request for images as per normal. Therefore, requests are done much rapidly than before and would be endless unless the user interrupts.

When the user clicks on the stop button, a ‘compiling function’ will be activated, which is a defined VB.NET class based on the idea above. The program will find out the directory where requested image are stored. After gathering a list of images from the directory, the program will create an AVI video object together with some pre-defined properties (like compression, frame rates, size of video). Then, it will combine the images into the video by treating them as individual frames in the video clip. Finally, an AVI file will be built and stored in the client hard disk for viewing.

4.3 Server/Client Communication

The client program used for network communication runs on the desktop and PDA control devices as illustrated in Fig.6:

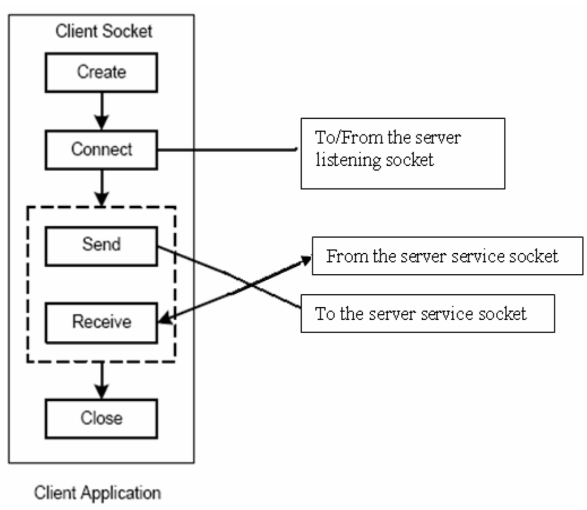

Fig.6 Flow chart of client program

The server program used in network communication runs on the PDA carried by the security robot as illustrated in Fig.7:

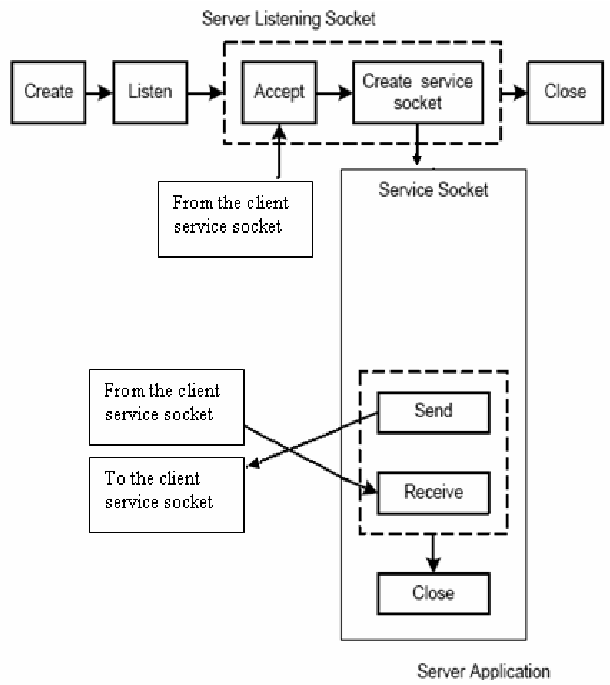

Fig.7 Flow chart of server program

4.4 Robot Control by the Server

A PDA carried by the robot not only performs wireless communication function, but also enable the robot-control through one of the following three methods: remote, semi-autonomous and autonomous. Such algorithm is depicted in Fig.8:

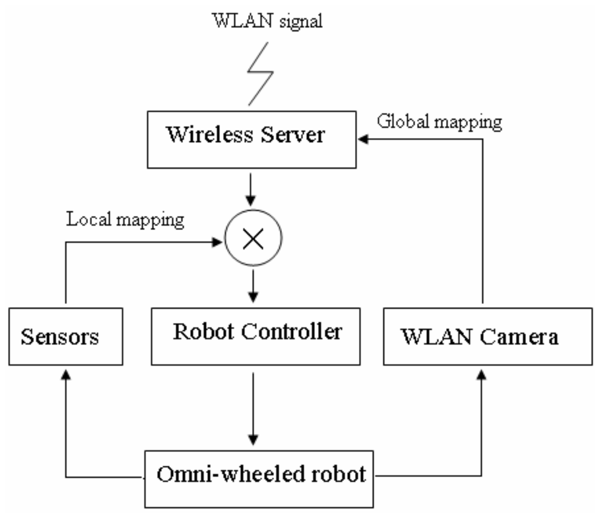

Fig.8 Server control algorithm flow

Two position mapping solutions have been implemented on the robot: local position mapping by the robot sensors and global position mapping/path planning by feedback from video camera.

4.5 Video capturing function

Video capturing program runs on the desktop client.

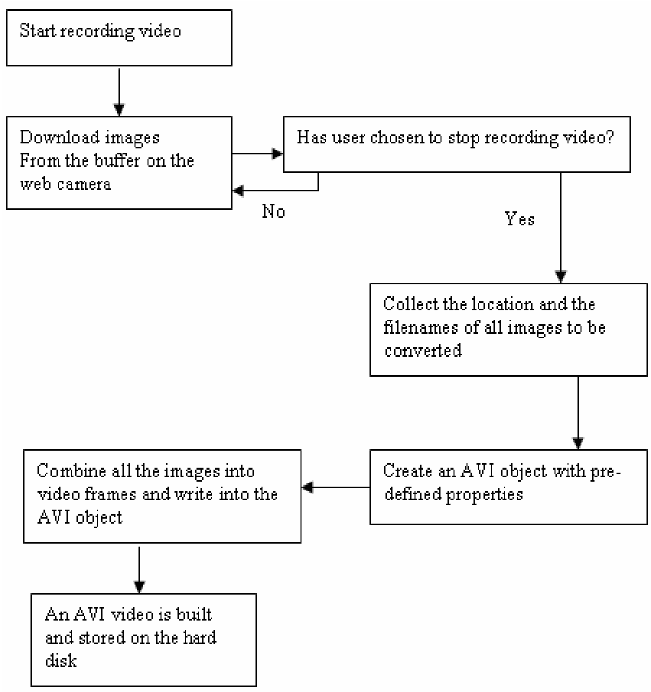

Fig.9 Video capturing program flow chart

5 Conclusions

We have demonstrated the possibility of remotely controlling a robot via a hand-held PDA through Wireless LAN, which dramatically decreases the cost of teleoperation and increases the mobility.

As another contribution, the function of mobile security is achieved and demonstrated by delivering video information via the same Wireless network.

Acknowledgements:
The authors would like to thank the support given by School of EEE, Singapore Polytechnic, to the development of the Wireless LAN-Based PDA Control Platform for Security Robots.

References

[1] P.K. Yue, C. Zhou, etc, “Robo-Erectus: The Story of its development,” *Proc. Of International RoboCup Symposium*, Lisbon, 2004

[2] Brady, K. and Tarn, T.J., “Internet-based remote teleoperation,” In *Proc. IEEE Int. Conf.*, 1998. *Robot. Automat* .pp. 65–70.

[3] Shandong Wu, Yimin Chen, “Remote robot control using intelligent hand-held devices,” *Computer and Information Technology, 2004. CIT '04. The Fourth International Conference* on 14-16 Sept. 2004 Page(s):587 - 592

[4] Xu Yi; Yan Wenjun; Zhu Jing, “Research on robot control system based on PC,” *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress* on Volume 2, 10-14 June 2002 Page(s):1246 - 1250 vol.2

[5] Nilas, P., Sueset, T., Muguruma, K., “A PDA-based high-level human-robot interaction,” *Robotics, Automation and Mechatronics, 2004 IEEE Conference* on Volume 2, 1-3 Dec. 2004 Page(s):1158 - 1163 vol.2