Proceedings of the ISRAM Conference, Montpellier, France, May 27-30, 1996, pp. 571-576

# A Completely Wireless Development System for Mobile Robots

**L. Feng[1], J. Borenstein[1], D. Wehe[2]**
*The University of Michigan*
*1) Department of Mechanical Engineering and Applied Mechanics*
*2) Department of Nuclear Engineering and Radiological Science*
*Ann Arbor, MI 48109-2110*

**ABSTRACT**

This paper presents the design and implementation of a *Wireless Development System* (WDS) for mobile robots. The WDS has unique features that can greatly facilitate the development and debugging process for mobile robots. In experimentation with mobile robots, a development cycle usually consists of two parts: *run-time*, and what we call "*debug-time*." Debug-time is the portion of the development cycle in which the experimenter modifies and compiles the software that controls the functions of the mobile robot. Then, during run-time, the robot is actually moving while the experimenter evaluates the effects of the software modifications. The described WDS provides a wireless user-interface for both components of the development cycle, allowing the experimenter full interaction with the onboard computer from a stationary location.

The paper provides information on the three main components of the WDS: One component is a radio control-to-computer interface for wireless joystick operation. This interface is inexpensive, independent of the type of onboard computer, and provides important safety features. The second component is the wireless computer-computer interface, which is specifically designed for IBM-PC type computers. The optional third component of the WDS provides graphical video feedback from the onboard computer during run-time.

**KEYWORDS:** Wireless, Mobile Robots, Development, R/C, Modem

## 1. INTRODUCTION

Mobile robots are beginning to move out of the lab and into real environments [Roman, 1991]. However, as the convenience of a laboratory is left behind, development, debugging, site-installation, and end-user operation are becoming more difficult. While in the laboratory, many mobile robot developers use tethers to link the onboard computer to a convenient desktop monitor and keyboard. When away from the lab, notebook computers on top of the robot are often used to give commands and to develop or modify software. Depending on the circumstances, either approach may not be always feasible. For example, during experimentation sessions it is often extremely helpful to watch certain graphic or numeric data on the screen, while the vehicle is moving. Or, in a hazardous environment application, the operator cannot be physically near the robot to issue commands [Fogle, 1992].

Many researchers in mobile robotics believe that the low-level functions of a mobile robot should be controlled by an on-board computer [Condon, 1992]. These functions include motor control, dead-reckoning, reflexive sensor-based obstacle avoidance, and basic motion primitives. However, if these functions are controlled by an onboard computer, it is necessary to allow a human operator to interact with this computer. There are two distinctly different modes of interaction:

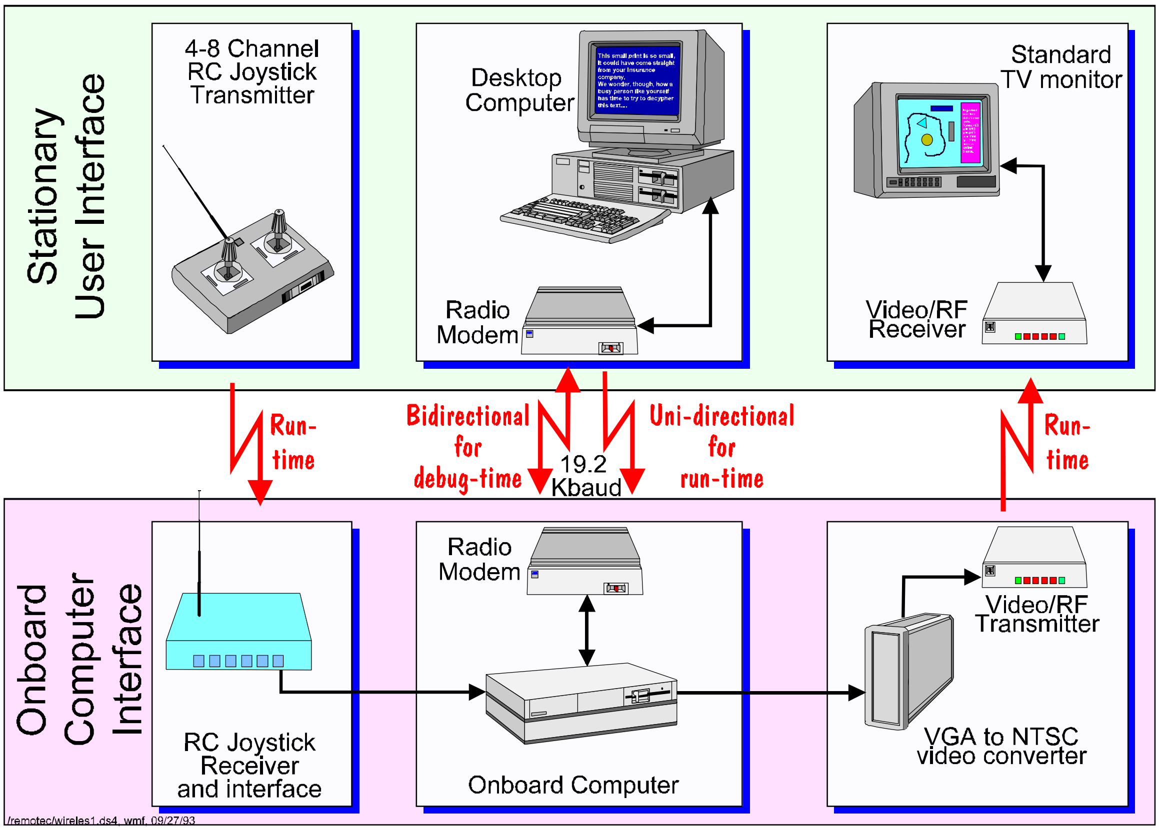


**Figure 1.** The Wireless Development System (WDS) for mobile robots.

- run-time interaction, where an experimenter or end-user can issue operational commands to the robot;
- "debug-time" interaction, where an experimenter edits and re-compiles the software that controls the functions of the robot.

In this paper we provide a technical solution for these requirements, called the *Wireless Development System* (WDS) for mobile robots (see Fig. 1). The WDS has been implemented and tested at the Mobile Robotics Laboratory at the University of Michigan. Clearly, our system is not the only solution to the problem of wireless robot control, but we believe that our WDS offers unique and useful features that will be of interest to fellow experimenters.

Our WDS has three main components. Depending on the circumstances, developers or end-users may need only one or two of these components. In Section 2 we discuss the most versatile component of the WDS, the multi-channel *radio control joystick interface* and its built-in safety features. Section 3 presents the radio-linked computer-computer interface, which is suitable for both run-time and debug-time. We also explain about the video feedback component and its significance for run-time debugging. Finally, in Section 4, we briefly describe non-wireless alternatives: a tether and an onboard monitor. If applicable, these approaches are sometimes more convenient than the wireless system, or they can be used together with some of the wireless components.

## 2. The Radio Control Interface

In this section we discuss a radio control-to-computer interface for reliable and accurate wireless joystick control of mobile robots. This interface also provides the function of a remote kill-switch for vehicle safety.

The wireless joystick control interface is based on a low-cost, commercially available radio control (R/C) system that is normally used for model airplanes. Such R/C systems are ideally suited for our task: They are extremely narrow-band (to minimize interferences with other R/C equipment) and they are designed for reliability and immunity to disturbances. In our system we use an FM transmitter and receiver with seven digital/proportional channels [Hitec]. Complete four-channel or six-channel FM systems from a variety of other manufacturers can be bought for $100 - $200 at any hobby store. The range of these systems is typically 3,000 ft *outdoors*, according to the manufacturers. The indoor range is substantially less, depending on the walls and other environmental conditions.

The receiver in standard R/C systems demodulates the FM signal and provides a pulse-width modulated signal for each channel, once every 20 ms. Each signal is a square wave of 1,100 to 1,900 μs duration. Note that these numbers may be slightly different for R/C equipment from different manufacturers. The width of the pulse is proportional to the position of the corresponding joystick on the transmitter. To measure the pulse width of the demodulated receiver signal accurately and without imposing special timing requirements on the onboard computer, we have developed the interface shown in Fig. 2.

The key component of this interface is the single-chip AM9513 system timing controller [AMD]. The AM9513 chip has five general purpose 16-bit counters and an internal oscillator that can be selected as an input for individual counters. The chip costs about $20.

In our application, the five counters of the AM9513 are used to measure the width of the square wave signals from the FM receiver. This works as follows: In the initialization phase of the robot control program (e.g., after boot-up or re-compilation), a counter is programmed to operate in the *event counter mode* with auto-read/reset. The mode of a counter is

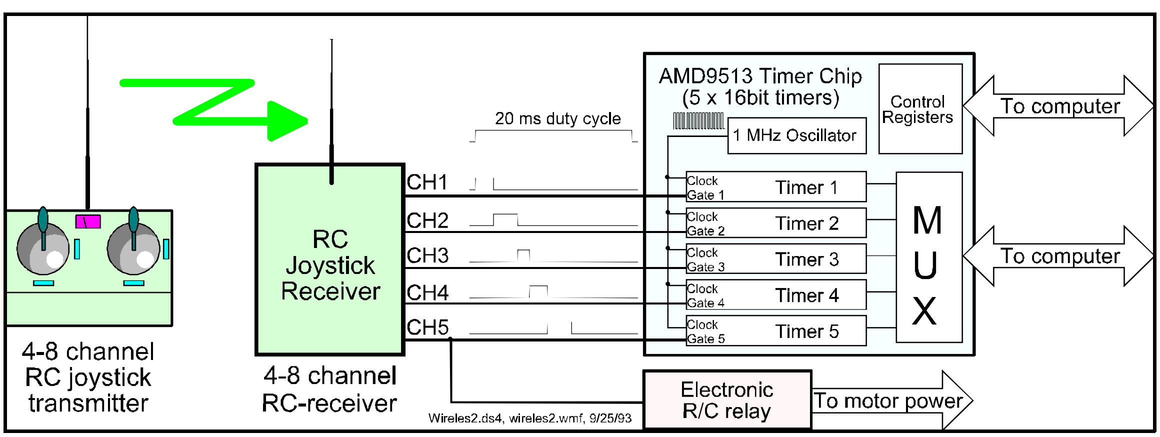


**Figure 2.** Wireless joystick interface, using the AMD9513 timer chip.

programmed by writing control words into the *Master Mode* and *Counter Mode Registers*. An initial value, from which the counter will count down during operation, is loaded into the *Load Register*. The square wave signal from the receiver is connected as the gate input, and a 1 MHZ oscillator is used as the input source (labeled "clock" in Fig. 2) to the counter. Once the onboard computer issues the "*start counting*" command, the counter will count oscillator pulses, for as long as the gate signal is active (receiver output signal is high). The counter will not count while the gate is inactive (i.e., the receiver output signal is low). When an active-going gate edge (i.e., the receiver output signal goes from low to high) appears at the gate input, the counter will reload itself from the *Load Register* (i.e., registers are not shown in Fig. 2). As a result, the counter is retriggered with each rising edge of the receiver output square wave signal. Once retriggered, the content in the counter will be transferred to the *Hold Register* and the content of the *Load Register* (initial value) will be loaded into the counter, and the new counting process starts. The duration of the remote control signal is the difference of the content of the *Load Register* and that of the *Hold Register*. The content in the *Hold Register* can be read by the onboard computer at any time.

Since the routine operation of the AM9513 chip is completely independent of the onboard computer, the computer can read the last valid pulse-width signal for each channel asynchronously, i.e., without the need to interrupt computer operation at the 20 ms intervals prescribed by the duty-cycle of the R/C receiver. One should note that most standard R/C transmitter/receiver units provide a resolution of 1/512 (i.e., 9 bit) for the joystick readings. Since the receiver output signal is pulse-width modulated, a timing resolution of (1,900-1,100)/512 = 1.56 µs would be required to measure the joystick position with the same resolution it is transmitted. With the 1 MHZ oscillator, the timing resolution of the AM9513 counters is 1 µs, which meets this requirement.

## 3. THE WIRELESS COMPUTER CONNECTION

Figure 1 shows the components of the wireless computer-computer interface. One key component is the *ProxLink* bi-directional *radio modem* manufactured by PROXIM [PROXIM]. This is a spread-spectrum radio module with a serial data rate of 19.2 Kbaud. Actual throughput may be somewhat slower, because of the overhead involved in packaging and addressing point-to-point transmissions. In principle, the modem can be thought of as a standard serial (RS-232) cable. A set of two *Proxlink* modems costs $1,300 for the 500-ft range version, and a version with a range of 900 ft is also available.

The *ProxLink* modem connects two computers: the onboard PC and the desktop PC. Both computers run commercially available *Remote Access Software* (RAS), a utility that allows the operator of the desktop PC (called "remote") to control all functions of the onboard PC (called "local"). RAS also allows the screen of the remote PC to show everything that the monitor of the local PC shows. Ideally, a local/remote system with RAS would appear to the remote operator as though he or she was working directly with the local PC. In practice, however, screen updates sent from the local to the remote site slow down the response time. This effect is particularly noticeable when operating in graphics mode.

Most compilers and editors for software development can run in text-mode only, and any one of the commercially available RASs should work well for editing and compiling the onboard computer's robot control programs. The problem begins after the onboard PC's program has been compiled, and starts to run. As a low-level controller, it is essential that the onboard PC runs a fast control loop, typically on the order of 10 - 50 ms. The RASs we evaluated interfere with that control loop, because even when no communication is taking place, these RASs take temporarily control of the local PC. For example, one commercially available RAS we tested (*Close-Up* by Norton-Lampert) was found to take complete control of the local PC for over 100 ms and at least once every few seconds. For timing-sensitive low-level controllers, such temporary loss of control is unacceptable. Another problem is that if the local PC outputs graphic information during run-time, the RAS will transmit the graphic output to the remote PC, resulting in even larger delays of unpredictable length.

In spite of this apparent incompatibility between RASs and graphics output during run-time, we consider the availability of graphics run-time information during experimentation to be of great importance. We have found that plotting the robot's path in real-time, along with other relevant information, is an indispensable tool for debugging and optimizing mobile robot behaviors, or for tele-autonomous operation [Borenstein and Koren, 1989]. In order to provide this feature after all, we have incorporated into our wireless system a video feedback component that works as follows: The digital VGA monitor output from the computer is fed into a *VGA-to-NTSC* decoder. We have tested the *VGA-to-TV Elite*, a device that can be purchased for $300 from Micro-Warehouse [MICRO] and other mail order companies. In our wireless system, the NTSC composite signal from the decoder is fed into a video transmitter, which transmits the video signal to an associated receiver that is connected to a standard TV. The video transmitter/receiver unit is a home-consumer product, originally designed for the purpose of transmitting the output of a VCR to a TV set in another room of a house. Such devices are available from Radio Shack, department stores, and mail-order companies that focus on home electronics. We have tested and used the widely available "*Rabbit*," which costs less than $70. The picture quality of this device is not great and it appears that the radio modem causes periodic interferences with the *Rabbit*. We believe that a more expensive device would provide a better picture, but we haven't tried that yet because the Rabbit is sufficient for our purpose.

With the problem of graphics feedback resolved, we can now take a second look at the remote access software (RAS) and its undesirable interference with the local PC's control program during run-time. After much experimentation, We found an RAS that happens to meet the somewhat peculiar requirements of our WDS. It is an old version (Vers. 2.1 from 1988) of *PC Anywhere* [EKD]. this RAS supports standard PC text mode, but not graphics mode. If the local PC is in graphics mode, this particular version of PC Anywhere will **not send any data** back to the remote PC, nor will it interfere with the timing of the local PC. Yet, keyboard commands sent by the remote PC are received and processed on the local PC, so that the operator still has control over the onboard PC. Thus, a property that is undesirable for most normal applications of an RAS provides exactly the functions needed for our WDS.

## 4. NON-WIRELESS ALTERNATIVES

When working in the laboratory, it is usually possible to operate the on-board low-level control computer with the help of a tether. For PC compatible on-board computers we have found that the *Cybex Companion* [Cybex, 1992] makes a particularly useful tether. This product is a combination of cable and signal booster, which allows to connect a keyboard and VGA (or better) monitor up to 250 feet away from the on-board computer. The cost of a Cybex cable with booster is approximately $400. Our lab has three separate tethers with

Cybex cables. Each tether runs from the on-board computer to a ceiling mounted-rotating beam, and then to a stationary keyboard and monitor on the operator's desk. The advantage of this setup is that the robot can travel freely throughout a large work area (about $6 \times 6$ m) without running over the tether. The particular advantage of the Cybex cable is that no second computer is needed and that the system behaves exactly like a single desktop unit. The disadvantages are the limited area of operation and the fact that the tether may have to be unwound periodically, if the boom rotated several turns in the same direction.

Another solution is the use of an onboard keyboard and monitor connected directly to the onboard PC. The disadvantages of this approach are the additional battery drain imposed by the monitor, and the fact that the operator has to crouch over the robot to operate the system. In our lab we use a 9-inch grey-scale VGA monitor [DATALUX] that costs $195 and consumes less than 25 Watts. The low power consumption allows uninterrupted development sessions of 3-5 hours with our Cybermotion and TRC robots. This monitor is very small and fits easily on top of our different robots, but the small display is also tedious to work with during long development sessions. Another disadvantage of this monitor is its lack of color. As we emphasized before, graphic run-time information is of great importance for the analysis of run-time performance, and color is an effective tool for increasing the amount of information that can be conveyed without confusing the experimenter.

## 5. CONCLUSIONS

We have implemented and tested a completely wireless development system (WDS) for mobile robots. This system is of interest to researchers and developers of mobile robot systems, and to end-users of mobile robots operating in hazardous environments. The total cost for all off-the-shelf components of the WDS (except the computers) is approximately $2,000. The system works well and provides important wireless debugging tools such as graphics video feedback at run-time. We are especially satisfied with the R/C joystick interface; we are using this interface exclusively in all our new projects, regardless of whether the robot runs with a tether or under the WDS. We found that editing source code is somewhat sluggish through the radio modem, in spite of the rated 19.2 Kbaud. For long editing sessions the Cybex tether is more convenient.

**Acknowledgments**:
This research is supported by the Department of Energy grant DE-FG02-86NE37969.

## 6. REFERENCES

1. Borenstein, J. and Koren, Y., 1989, "Tele-autonomous Guidance for Mobile Robots." *IEEE Transactions on Systems, Man, and Cybernetics*, Special issue on unmanned systems and vehicles, Vol. 20, No. 6, Nov/Dec 1990, pp. 1437-1443.
2. Congdon, I., et al., 1992, "CARMEL Versus FLAKEY - A Comparison of Two Winners." *AI Magazine*, Winter, pp. 49-56.
3. Cox, I. J. and Wilfong, G. T., Editors, 1990, *Autonomous Robot Vehicles*. New York: Spring-Verlag.
4. Feng, L., Koren, Y. and Borenstein, J., 1993, "A Cross-Coupling Controller for a Differential-Drive Mobile Robot." To appear in IEEE Control Systems Magazine.

5. Fogle, R., 1992, "The Use of Teleoperators in Hostile Environment Applications," Proceedings of the 1992 IEEE International Conference on Robotics and Automation, pp. 61-66, Nice, France, May.
6. Roman, H., 1991, "Robots Cut Risks and Costs in Nuclear Power Plants," IEEE Computer Applications in Power, pp. 11-15, July.

**Commercial Products**

1. AMD, Advanced Micro Devices, Inc., 901 Thompson Place, P. O. Box 3453, Sunnyvale, CA 94088.
2. CYBEX, Cybex Corporation, 2800-H Bob Wallace Avenue, Huntsville, AL 35805.
3. DATALUX Corporation, 2836 Cessna Drive, Winchester, VA 22601
4. EKD Computer Corp., 764 Middle Country Rd., Selden, NY 11784.
5. HITEC, RCD, Inc, 10729 Wheatlands Ave., Suite "C", Santee, CA 92071.
6. MICRO Warehouse, 1720 Oak Street, P. O. Box 3014, Lakewood, NJ 08701.
7. PROXIM, Inc., 295 North Bernardo Avenue, Mountain View, CA 94043.