

DEVELOPMENT AND TESTING OF A WIRELESS CONTROLLED CAR USING THE INTERNET AS COMMUNICATION MEDIUM

A.J. Lubbe and P. Kluge¹

Department of Industrial and Systems Engineering
University of Pretoria, South Africa
andre.lubbe@up.ac.za

ABSTRACT

Recent innovations in wireless networking technology have brought low cost, high bandwidth and low latency communication to the consumer. Apart from the standard use of connecting computers to the Internet, the new wireless networks also offer the possibility of sending a combination of two-way video, audio and control signals almost instantaneous between any two places in the world.

This paper provides an overview of the techniques and software integration necessary to develop a mechatronic device that can be controlled remotely through the available wireless networks. A prototype mechatronic device, in the form of a remote controlled vehicle, was developed and successfully tested to demonstrate the integration of motion control with live video feedback through the Internet.

OPSOMMING

Die onlangse vooruitgang in draadlose netwerktegnologie het laekoste, hoë-bandwydte en lae tydsvertragingkommunikasie binne bereik van die gebruiker geplaas. Behalwe die standaardgebruik om rekenars aan die Internet te verbind, bied die nuwe draadlose netwerke ook die vermoë om tweerigting-video, audio- en beheerseine feitlik oombliklik tussen enige twee plekke op aarde te stuur.

Hierdie artikel beskaf 'n oorsig oor die tegnieke en rekenaarprogramintegrasie wat nodig is om 'n megatronikatoestel te ontwikkel wat deur afstandbeheer oor die beskikbare draadlose netwerke beheer kan word. 'n Prototipe megatronikatoestel, in die vorm van 'n afstandbeheerde voertuig, is ontwikkel en suksesvol getoets om die integrasie van bewegingsbeheer en intydse videoterugvoer deur die Internet te demonstreer.

¹ This author was enrolled for the final year project at the Department of Industrial and Systems Engineering, University of Pretoria

1. INTRODUCTION

A mechatronic system is any mechanical-electrical system using a microprocessor that interprets and controls the motion of the system. With the tremendous pace of development in wireless communication, it is now possible to control a mechatronic system from almost any place on earth where there are cell phone or Internet connections.

A radio-controlled system is distance limited, and direct sight is normally possible, whereas control over the Internet is not limited by distance, and visual feedback is indirect.

Most of the building blocks of an Internet-controllable system are readily available, but they are normally developed with other applications in mind. The challenge is to match all the different components and integrate them into a system with central processing that can communicate across all platforms.

The technique to view visual images through remote cameras on the Internet is already in the public domain, but wireless communication through the Internet and cell phones only came on to the market in the year 2006. To control the remote camera, the camera must be fitted with its own microcontroller and programmed in such a way that the controller system (computer or cell phone) has full control over the actuators of the remote camera.

A mechatronic system, in this case a prototype distance controlled car, was developed to test the possibilities of controlling remote devices using emerging broadband wireless networks. The user can control the vehicle from a computer that is connected to the wireless network. Using a live video feed sent from the vehicle, the user has visual feedback.

The final prototype vehicle can be seen in Figure 1. The vehicle is fitted with a direct current electric motor for propulsion. Left, right, forward, and accelerated motions are controlled from a host computer with dedicated buttons on the computer keyboard. The fast speed would normally only be selected for straight-line movement.

An emergency slip clutch protects the drive motor when the motor is running but the vehicle is unable to move. The braking system of the vehicle is also activated with a button on the keyboard of the host computer.

For visual feedback a video camera is mounted on the vehicle. The video image, as seen through the video camera, is projected on to the screen of the computer. The video camera can turn a full 360 degrees by means of a stepper motor. An arrow on the computer screen indicates the direction of the camera in relation to the front of the vehicle.

The computer and the vehicle are each equipped with a device to connect to one of the wireless services. The network service providers for vehicle and computer need not be the same.

2. PROPERTIES OF THE FINAL PRODUCT

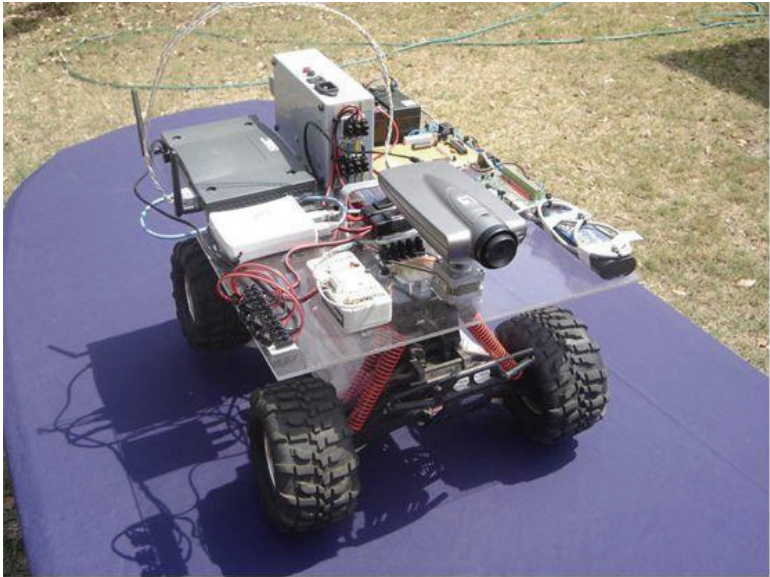


Figure 1: The completed vehicle with control parts

On-board batteries supply power to all electrical and electronic components, to allow the vehicle unrestricted movement.

A microcontroller on the vehicle takes care of all the decision-making and control functions. In contrast with a cell-phone controlled vehicle [Lubbe: 1], where single sound signals are processed in series, the wireless network allows digital data as input and output to the microcontroller. It is possible to control many functions simultaneously - for example, to brake while the vehicle is turning - which is impossible with multi-channel devices where the channels operate in series.

3. SUMMARY OF THE STATE OF THE ART ON WIRELESS CONTROL THROUGH THE INTERNET

Wireless interface of mobile computing devices (Wi-Fi) is widely available today for use in so-called 'hotspots' where the radio signals are available, and information about Wi-Fi can be found in over five million Internet sites. The connection between computers through Wi-Fi to communicate data and video images is almost at a plug-and-play stage. Almost 99% of all Wi-Fi connections is used for communication purposes, and the other 1% is used for mechatronic control purposes.

Mechatronic control applications can be divided broadly into two groups: first, control between two computers, and second, control between a computer and a microcontroller. Most modern computers are already equipped with Wi-Fi hardware devices and software is freely available. On the other hand, microcontrollers do not come standard with Wi-Fi, and computer software cannot run on microcontrollers because of incompatibility between 32-bit and 8-bit data. All microcontroller software must be developed for each application. Only in large mechatronic devices is it feasible to use a computer for control purposes, because the computer must be piggy-backed on the mechatronic device.

Mechatronic control is only possible through the use of actuators. Thousands of actuators are commercially available, and each one comes with a set of control rules. Software to control the actuators does not depend only on the type of actuator but also on the application. When an actuator, like a step motor, is programmed, the control signal must have a specific amplitude and a fixed number of cycles, and a second signal must carry the direction of rotation. These signals must be directed to specific output bits. When the signal is stopped, the controller software must decide whether to return the step motor to its starting position - like when the steering wheel of an automobile is turned and then released - or whether to keep the step motor in the final position. It is therefore impossible to get generic software to control actuators.

Thousands of devices that can be controlled through Wi-Fi are commercially available. You can buy devices to open your automatic gates, control your sprinkler system, or switch on your stove. Most of these devices have only limited feedback. To steer a mechatronic car, visual feedback of the surroundings of the car, as well as a way to sense direction, is of outmost importance. In order to incorporate this feature, not only does the controlled device need to be programmed, but the controller computer as well. Again, standard software is not available because of the great number of microcontrollers on the market.

A similar mechatronic car was developed by Jonathan M. Giffan [Giffan: 7]; the unveiling took place on the Internet on 25 April 2007. Technical information was not given because the product can be bought in D.I.Y. format. Some of the readers commented that this product could revolutionise the hobby industry.

4. MECHANICAL LAYOUT

The chassis, wheels, steering mechanism, suspension system, and brakes of a radio-controlled car were used as basis for the Internet-controlled vehicle. A 12-volt electric motor was fitted to this platform for propulsion. The design speed was achieved through the correct selection and installation of a set of gears. A slip clutch was designed, built, and fitted between the 'gearbox' and the drive axle. A physical platform, on to which all the control parts could be fitted, was secured to the chassis.

On the platform, the video camera was mounted directly on the vertical shaft of a stepper motor to achieve controlled horizontal rotation (see Figure 1).

5. MICROCONTROLLER

The programming language used in the Faculty of Engineering at the University of Pretoria is MATLAB. Microcontrollers can also be programmed with MATLAB, but only a limited number of controllers are supported by MATLAB. With this in mind, the Motorola HCS12 8-bit controller was selected from a list of four possible controllers as the best for this particular application.

The output ports of the Motorola had to be modified to an I²C system [Gilmore: 2] to enable the connection of a number of output functions, such as stepper motor controllers, servo motor controllers, and others, each with a unique address.

Furthermore, this particular controller has the built-in capability to produce up to four independent 50 Hz pulse-width-modulation (PWM) signals [Motorola: 3] that can be used to control the servo motors; these in turn control the steering, brakes, and acceleration of the vehicle.

6. ELECTRONIC COMPONENT INTEGRATION

6.1 Wireless communication

Recent innovations in wireless networking technology have brought low cost, high bandwidth and low latency wireless networks to the consumer. Services like Vodacom's HSDPA, iBurst, Sentech's MyWireless and up and coming services like Telkom's WIMAX now enable permanent high-speed Internet connections. Most of South Africa's urban regions are already covered by at least one of the broadband wireless services, with rural coverage anticipated with the launch of WIMAX and other 4G wireless technologies.

Apart from the standard use of connecting computers to the Internet, wireless networks now also offer the possibility of sending two-way video, audio and control signals over the Internet, offering a number of technological innovations.

To connect to the wireless network, the computer must be equipped with the necessary communication device, such as an iBurst PC Adapter. The controlled vehicle must have a similar communication device with its own Internet address.

6.2 Connecting the wireless signal to the controller

The software on the host computer that generates the control signals for the microcontroller on the vehicle is called the User Interface Application (UIA).

A TCP/IP signal is sent from the UIA over the network to a USB server on the vehicle. (TCP stands for Transmission Control Protocol, and is an international standard for exchanging data between two computers; IP (Internet Protocol) is the international standard to communicate between different devices over a computer network. The USB server translates the signal to a digital form, which is communicated through a USB bus to a USB interface board. The interface board is connected directly to the serial input bus of the microcontroller. A driver library comes standard with a USB interface board, allowing for easy language-independent software development.

To connect the IP camera on the vehicle to the network, a separate hub and bridge is needed for connection to the USB server. The USB server, interface board and hub/bridge are three commercially available pieces of electronic hardware.

7. SOFTWARE

7.1 User Interface Application (UIA)

The UIA is a stand-alone software application that was created to enable the user to control the remote vehicle from a personal computer.

In summary, the UIA consists of the following components:

- Initial device detection and validation
- Component-wide keystroke monitoring
- Predefined key combination blocking
- USB board interaction
- External media player software

When starting the UIA, a report is generated on the success or failure of the connection of the IP camera and the USB interface board. If both are connected successfully, the user can press the ‘Take Control’ button on the screen to proceed to the control part of the UIA.

When the user enters the vehicle control part of the UIA, the application switches to a full screen window with a background that contains the video feed from the remote camera.

All the control keys that are available to the user are shown in Table 1.

Key	Function
Vehicle control	
Arrow up	Forward propulsion, slow speed
Arrow down	Break
Arrow left	Turn left
Arrow right	Turn right
Shift	Fast speed
Camera control	
Z	Rotate camera counter-clockwise
X	Rotate camera clockwise
Miscellaneous	
ESC	Quit UIA

Table 1: Control keys and functions

To give the user more feedback on the vehicle and camera movements, two key indicators were placed in the bottom left and right parts of the screen: one to indicate camera movement, and the other to indicate vehicle movement. Whenever the user presses a valid key, this action is displayed in either of the key indicators.

This informs the user that the UIA has received the command, and also reminds the user which keys control what aspect of the vehicle or camera. The camera and vehicle can be controlled simultaneously.

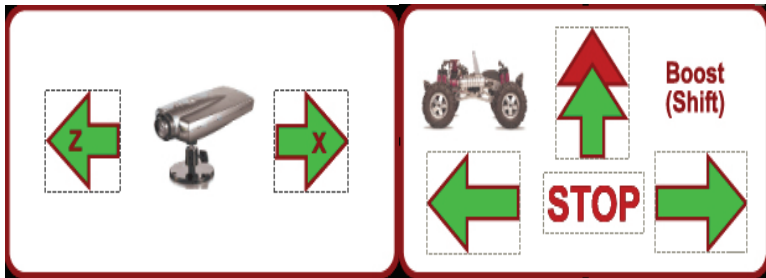


Figure 2: Key indicators for camera and vehicle movement

The UIA was programmed and designed using Microsoft Visual Studio C#.Net 2005.

7.2 Microcontroller software

7.2.1 General

To give a microcontroller functionality, software is loaded into its memory. This software runs as long as the microcontroller is switched on, and will respond to inputs by switching different outputs into certain predefined states. New techniques have been developed to make the task of programming a microcontroller easier and quicker. Motorola's new HCS12 microcontroller can now be programmed using simple logical diagrams created in Mathworks Simulink [Simulink: 4].

The process of developing software for the microcontroller now starts by creating a conceptual or logical design of the controller's functionality in Simulink. Different blocks in Simulink will represent functions of the microcontroller like "Get input from port X" or "Delay signal for X milliseconds", etc. Once a complete model of the functionality of the controller has been created, Simulink uses Real-Time Workshop [Real-Time Workshop: 5] to create a C program, which imitates all of the model's functionality. The C program can then be translated into assembly code with Metrowerks Codewarrior [Simulink: 4] for execution on the controller.

7.2.2 Simulink Device Driver Blocks (DDB)

Simulink comes packaged with all the pre-programmed linking blocks that are necessary to create almost any program for the HCS12 microcontroller. Figure 3 shows the layout of the main Simulink model.

In Figure 3, the "IIC IN" block reads the input from the USB interface board. This input is displayed on the LEDs available on the controller through "PORT B". The input signal is also sent on to the following subsystems, which handle their duties separately:

- Camera motor subsystem
- Steering subsystem
- Brake subsystem
- Motor subsystem

The motor subsystem is shown in Figure 4. The other subsystems were designed similarly. The motor subsystem controls the speed of the propulsion motor. Signals generated by the UIA are read by the HC 12 input of the microcontroller, and a Pulse Width Modulated (PWM) signal (PWM45) controls the speed of the motor.

The following speed cycles are possible [Motorola: 3]:

- Bit 2 ON, Bit 3 OFF: Motor ON with duty cycle of 2500
- Bit 2 ON, Bit 3 OFF: Motor faster at duty cycle of 2700
- Any other combination of Bit 2 and Bit 3: Motor OFF

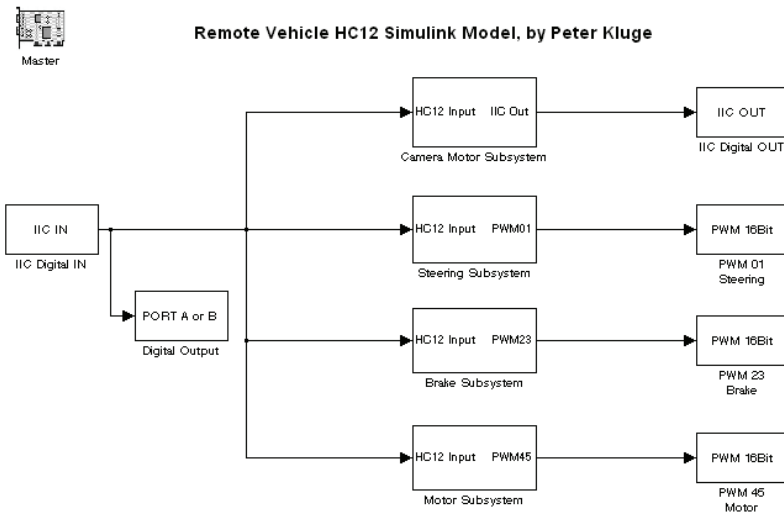


Figure 3: Main Simulink program

Full speed could be achieved by a duty cycle of 3660, but at that speed the vehicle was going too fast to be controlled.

7.2.3 Development of new Device Driver Blocks

The Device Driver Blocks (DDBs) in Simulink are programmed in such a way that Simulink can interpret the functions of the blocks as well as translate the functions into a C program. All the blocks in Simulink have been developed for a standard Motorola HC12 controller.

Motor Subsystem:

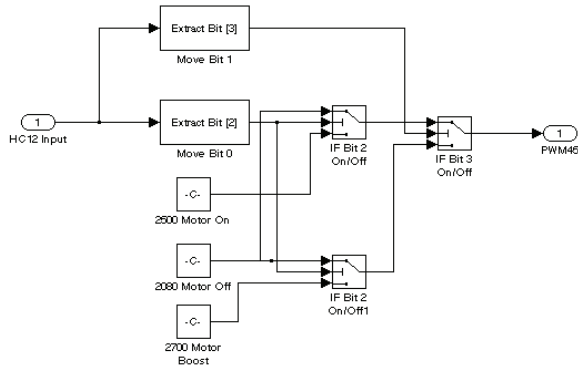


Figure 4: Motor subsystem using Simulink blocks

The standard output of the microcontroller used for this project had to be changed to an I²C output bus to handle many different functions simultaneously, each with a unique output address. Also, a separate input port had to be added for the USB interface board. New DDBs had to be developed to incorporate these new functions in Simulink.

During the development of the driver blocks, guidelines in Simulink's help files were followed closely. Every driver block consists of the following components [Help File: 6], each of which has to be created in order to build a new DDB:

- Subsystem block
- Resource keyword
- S-Function
- Target Language Compiler (TLC)

Creating the reservation and TLC functions for a DDB requires a lot of Matlab and TLC code knowledge.

8. APPLICATIONS

The main application of this design is to control motion over a large distance.

Apart from many general applications like carrying a surveillance camera, controlling the irrigation system on a farm, activating and focusing remote wildlife cameras, patrolling your house when you are not present, and so on, the technique has numerous engineering applications.

The civil engineer or architect can plan to have the security gates of several security complexes controlled from a single remote control room. The gates can be partially

opened to allow pedestrians in, or fully opened for vehicles. The guard can focus the camera on faces or on number plates.

An expert robot programmer could make use of the remote motion control principle to take over control of remote manufacturing robots at different clients without being physically present at the site. Programming problems could be tested and rectified from a remote site.

9. CONCLUSIONS

The assembled vehicle manages to demonstrate the incredible functionality that can be achieved with new broadband wireless networks.

Although most of the hardware components are commercially available, integrating and controlling them as a unit is a major task. Luckily, developing new S-functions for the microcontroller is a once-off task.

During development it was found that, should a commercial version of the same or a similar device be developed, one could, at a cost of roughly US\$1,000, connect directly to the network by replacing the USB server and the USB interface board with a software TCP/IP stack in the microcontroller [Gilmore: 2]. Also, most of the distributed electronic hardware could be integrated on one dedicated electronic board.

For this project a cheap video camera was purchased; an external media player had to be used in the interface application, resulting in unnecessary programming. The video camera should have been able to stream its video to the standard Windows Media Player software.

With this development project, it has been demonstrated that it is possible to match different cutting edge and divergent technologies and integrate them into a system with central processing that can communicate across all platforms.

More specifically, the following gaps in knowledge were filled:

1. Anybody attempting to build a similar type of mechatronic system must realise that about 80% of the needed software is not available and must be developed.
2. With MATLAB it is possible to program a standard microcontroller with the technique of driver blocks; but not all the blocks needed for such a project are available, and they can only be developed with in-depth specialist knowledge. The needed blocks (s-functions) were developed in this project.
3. Not all the hardware is commercially available, and so electronic boards to incorporate the I²C signals were developed.
4. The video feedback over the Internet is not fast enough to allow rapid response for fast moving vehicles. The speed of the motor had to be reduced

to about one meter per second. The same technique cannot be applied, for example, to control remote aircraft in flight.

5. The project demonstrated that the restriction of a one-channel control that is achievable with cell phone control can be overcome with an Internet connection, making it possible to control the camera and the speed of the motor, and to turn the vehicle - all at the same time.
6. The amateur with sufficient knowledge can build a similar device with hardware costing less than R2,000 by making use of commercially available hardware.

10. REFERENCES

- [1] Lubbe, A. J., 2005. The development and evaluation of a prototype cell phone controlled robotic system. *South African Journal of Industrial Engineering*, 16/2, 175-185.
- [2] Gilmore, C. M., 1995. *Microprocessors: Principles and applications*, 2nd edition, McGraw-Hill.
- [3] Motorola Inc., 19 July 2001, PWM_8B8C Block User Guide V01.15, 48.
- [4] Simulink, 2002, *Model-based and system-based design (Writing S-Functions)*, Version 5, The Mathworks.
- [5] Real-Time Workshop, 2004, *Getting started with Simulink*, Version 6, The Mathworks.
- [6] Help File, R2006a, *Creating device drivers for the embedded target for Motorola HC12*, The Mathworks.
- [7] Giffan, J. M., 2007, <http://arstechnica.com/journals/science.ars/2007/04/25/cmu-unveils-hobby-robots-with-wi-Fi-control>

