Senior Design Project: Development of the Shuttle Tracker

Nick Guerette

May 5, 2005

Contents

Chapter 1

Introduction

Swarthmore College runs two shuttle vans each at twenty-minute intervals every evening to safely transport students to and from off-campus dormitories. These shuttles run on a regular schedule, making a complete trip every twenty minutes. However, students are able to call the Public Safety department and request a ride from one place on campus to another after dark, and normally it is the off-campus shuttle that is sent to provide these rides. However, when a shuttle driver receives a request for such a special-purpose trip, he or she often (in the author's experience) detours to perform the trip immediately, regardless of whether this will cause the shuttle to miss its normally scheduled stops. In addition, the shuttles sometimes stop running due to weather or mechanical problems, or other anomalies.

Thus, it is common for the shuttles to not run on schedule, and this causes students who are waiting for a shuttle to get frustrated, and often call the Public Safety department to inquire about the shuttles' current locations. In addition, when a shuttle does not arrive on time, students waiting at the shuttle stop are put in the position of deciding whether to wait for the shuttle to possibly arrive, or begin walking to their destinations. Of course, not having any information about the current location of the shuttle makes this a difficult decision. If one leaves on foot, there is the risk that the shuttle will arrive immediately afterward, and the time spent walking will be wasted. However, if one remains at the stop waiting for the shuttle to arrive, it is possible that the time spent waiting will be greater than the time that could have been spent walking the trip. Because of this difficulty, in two years of living in an off-campus dormitory, the author almost never tried to utilize the shuttles, and instead took the opportunity to develop impressive leg muscles making the uphill trip to campus by bicycle.

The motivation for developing the Shuttle Tracker was to address this problem, and both reduce students' frustration and save time for Public Safety officers by implementing a system to track the location of the shuttle vans and make that information easily available to students. One benefit to having such a system would be a possible improvement in punctuality of the shuttles, if drivers know that students can easily check the last known location of a shuttle. Perhaps a greater benefit would be a reduction in frustration, though, when the shuttle does miss a stop. When a shuttle doesn't arrive on time, if students waiting for it can see that it left its previous stop moments ago and is just a bit behind schedule, they can confidently make the decision to wait for it to arrive rather than begin walking. If, however, they can see that the shuttle is sitting in a parking lot, or that the location of the shuttle has been unknown for some time, it will be much easier to make the decision to begin walking home. Overall, hopefully utilization of the shuttle would increase and wastes of students' time and the resources required to run the shuttles would decrease if such a system were implemented, because students would be more likely to try to take the shuttle if they knew that they weren't going to be left in the dark about whether or not the shuttle is in fact going to arrive.

Work on this project began during the summer of 2004 with a grant from the Richard M. Hurd '48 Engineering Student Research Endowment, and work completed during that summer was reported in the proposal to continue work on this project to fulfill the author's Senior Design requirement. Thus, much text in this report describing portions of the project completed during the summer of 2004 was adapted from that proposal.

Chapter 2

System Overview and Major Design Choices

Any system for tracking the location of shuttle vans and making that information available to potential riders in various locations will require components that accomplish three overall tasks:

- automatically determining the locations of vans,
- distributing van location information to users in various locations, and
- effectively displaying van location information to users.

The high-level design of the Shuttle Tracker system required choosing or developing componennts that could accomplish these tasks, integrate easily with each other, be built or purchased with the available budget, and be simple enough to allow the possibility of building and testing the entire system in the time available.

2.1 Choice of Vehicle Tracking Method

At the heart of any system for tracking and reporting the location of the shuttle vans is an automated method for determining the locations of the vans. Two such methods were considered, with very different costs, implementation details, and capabilities. The decision of which method to use involved determining how much information was desired (whether it was necessary to know the exact location of the vans at all times, or simply to know when they arrive at and depart from a stop), what sort of materials the budget available to the project would afford, and the level of complexity involved in building the system compared with the time available to do it.

2.1.1 GPS: Exact Location Information

One method considered for determining the locations of the vans would have utilized Global Positioning System receivers installed in the shuttle vans, which could receive transmissions from satellites in geostationary orbit to report the position of each van with a resolution of approximately 20 meters anywhere on the planet. The strength of this method is of course in being able to determine the location of a van, no matter what that location is. However, this information is reported by the receiver which is located on the van, and this necessitates the extra complexity of transmitting that information from the van to a location from which it can be distributed to users.

In order for this system to be useful, either the transmission of information from the van would have to be reliable at long distances (perhaps 0.5 to 1 mile), or many receivers would be required with smaller distances between them. A radio transmission powerful enough to send data over a distance of 0.5 mile, which would be required to be able to receive data at any point along the normal route of a shuttle at Swarthmore (which is a small campus; in addition, a greater range would be required if the van were to be able to report its location when going on special side-trips, a feature that would be desirable and possible with a GPS-based system), would likely require a cost-prohibitive FCC approval and/or license. However, having multiple receivers spaced more closely together would be more costly. One extra capability a multi-receiver system would have, though, is being able to obtain information about the approximate location of a vehicle simply based on which receiver picks up its transmission. This idea is the basis of the other vehicle location tracking method considered.

2.1.2 Proximity Detection: Imperfect Information at Low Cost

The other method considered for automatically tracking vehicle locations, dubbed the proximity method, involved installing simple, low-power radio transmitters in each van, which would periodically transmit serial data containing a unique ID number for the van. The device installed in each van would have no information about the van's location, but the transmission of an identification number makes the van's presence automatically detectable. Receivers would be installed at locations to which the shuttles may travel (at least at each of their stops, and ideally also at the lot where they are parked), and the location of each shuttle would be detected based on which receiver picks up its transmitted ID number.

This method has the obvious disadvantage that a shuttle can only be detected at particular locations where a receiver is installed. However, if users are able to be provided with information not only about what building a shuttle was last near, but what time it was near that building, they can still make informed decisions about the likelihood that the shuttle will arrive on time. In addition, if users are also able to view a measure of the relative received signal strength of a transmission from a van, they could make further inference about the distance of the van from the receiver location. Since the GPS method was going to involve expensive receivers (at least in the range of $100 per unit) and the complication of transmitting location information continually from each van, it was determined to be an infeasible method of van location detection for this project, and the proximity method with signal strength reporting was chosen.

2.2 Distribution of Van Location Information

Twisted-pair ethernet networks using Internet Protocol (IP) for passing data between ethernet network segments are undeniably the most common method for networking computers at the present time, and thus using an existing ethernet/IP network as a method for distributing information about van location to users is a natural choice. Thus, each RF receiver and each device displaying information about the location of shuttles requires a connection to an ethernet/IP network. An extra wrapper protocol is necessary around IP to manage flow of network data to and from multiple applications on a single computer, and the two main protocols at this layer (called the "transport" layer) are Transmission Control Protocol (TCP), which provides error-checking and validated reception of data spanning multiple IP packets, and User Datagram Protocol (UDP), which really is simply a wrapper around IP packets (although it does allow data spanning multiple packets, due to the fact that IP allows for "fragmentation" of data across multiple packets), and allows sending simple datagrams marked for which application generated them on the transmitting computer and which application they are destined for on the receiving computer. Since the data to be sent by the devices in the shuttle tracker system is very simple, and missed data isn't catastrophic, UDP was chosen as the transport-layer protocol.

Since an implementation of the Shuttle Tracker will have multiple receivers and display devices, a central server is required to keep track of all the devices involved in the system, and distribute the van location information from the RF receivers to the display devices (from here on called "display clients", in line with the server-client model of computer communication). Both the RF receivers and the display clients need to be able to find the server on the network when they first come online, and need to make sure over time that a connection to the server is maintained. The server, in turn, needs to be configured to know how many vans and receivers are present in the system, in order to keep track of the last known location of each van, and provide intelligible location information based on the identity of receivers. As display clients come online, the server needs to also keep track of each display client that is online, and send updates about van locations to each display client, with the possibility that the vans for which each display client is displaying information may change periodically according to a configuration set in the server. The server also needs to stop sending updates to display clients that have given no indication of still being online for a certain period of time.

2.3 Van Information Display Clients

The most useful place for information about shuttle van locations to be displayed is at the shuttle stops where people will be waiting for a van to arrive. Thus, dedicated display client devices should be mounted at each of the shuttle stops. In addition, since the system design also calls for an RF receiver to be at each of the shuttle stops, the RF receiver and display can be integrated into a single device with a single embedded ethernet connection.

Given that an existing ethernet/IP network to which personal computers are connected is being used to distribute van location information, it is also desirable to have software display clients available which users can install on personal computers to receive real-time updates from the Shuttle Tracker server.

Chapter 3

Long-Range RF Identification Considerations

3.1 Choice of RF Modules

There are a number of RF modules designed for transmitting serial data on the market. I settled on the Linx LR series due to the fact that this model has a stated range of up to 3000 feet, and that the receiver has an analog output that indicates the amplitude of the carrier and can be used to provide a relative indication of van proximity. At the time of beginning to work on this project, no LR series transmitters were available from distributors. Since the documentation claimed that the long range of the LR series was mostly due to receiver sensitivity, I chose to use the compatible Linx LC series transmitter modules.

The Linx LR and LC series RF modules are designed to transmit serial binary data, and use an extremely simple method of transmitting data called "on-off keying" (OOK), or a binary version of amplitude modulation (AM). In this method, the transmitter simply generates a carrier wave when transmitting a bit with a value of '1', and doesn't generate a carrier wave when transmitting a bit with a value of '0'.

3.2 Data Encoding Method

I found several vendors that manufacture specialized hardware for encoding and decoding data or button presses to be sent over a serial wireless link, but to minimize cost and hardware complexity I decided to use the universal asynchronous receiver transmitter (UART) modules built-in to the microcontrollers in the receiver and transmitter hardware. A difficulty with this method is that while a standard UART is inactive, its output is high, and because the RF modules being used use on-off keying, simply connecting the output of a UART to a Linx LR or LC transmitter would cause the carrier to be constantly present when data is not being transmitted. This would interfere with transmissions from any other transmitter, and whenever more than one van is present at a stop, none of the vans would be able to be detected.

In order to deal with this problem, the transmitter microcontroller's UART is disabled while data is not being transmitted, and the output to the RF transmitter is set low to disable carrier generation. When the transmitter microcontroller is ready to send identification data, it enables its UART to set the output to the RF transmitter high so that the RF transmitter begins to generate a carrier. It waits in this state for the period of time necessary to transmit one byte, in order to assure the receiver senses a stop bit from any data it had been reading out of random noise and is ready to receive a new byte. The transmitter microcontroller then sends two bytes over its UART. The first byte is the same for all transmitters and identifies the transmission as emanating from a Shuttle Tracker transmitter. The second byte is an identification for the van.

3.3 Random Receiver Noise and Signal Strength Thresholding

An additional problem with using the Linx LR receiver is that when there is no carrier wave from a transmitter present, the receiver module outputs random data. However, since the receiver is not only receiving data but also measuring the signal strength of the received data, the actual presence of a carrier wave can be determined by examining this signal strength.

In order to avoid detecting spurious valid data, the presence of a shuttle is only be acknowledged when the carrier amplitude is above a threshold. The threshold is a value between the median of daily minimum amplitudes during the previous seven days and the median of daily maximum amplitudes during the previous seven days. In the prototype receiver and transmitter setup, the threshold was set at the minimum value seen so far plus 25% of the difference between the minimum value seen so far and the maximum value seen so far. In addition, it is planned that in a final implementation of the system, the frequency of measurements of the received signal strength at a receiver will be high enough that at least three measurements will be taken during the period at the beginning of a transmission when the transmitter is turned on and a carrier wave is generated for at least the length of the transmission of one byte, and then the median of the last five measurements taken will be used as the value to compare to the threshold, for further reliability of received data validation. Another step to be taken is to include a parity byte at the end of each transmission, so that the bitwise exclusive-or of all the bytes of a transmission will be zero, for additional validation.

3.4 Overlapping Transmissions

The Shuttle Tracker needs to be able to deal with multiple vans at a single stop, but if two van transmitters are transmitting simultaneously, they will interfere with each other and neither will be detected. In order to avoid this problem, each transmitter will transmit at different intervals ranging from approximately two to four seconds. In order to accomplish this, the transmitter's ID, which is stored as an 8-bit unsigned integer, will be bit-reversed, and the resulting number used to calculate the length of time beyond the minimum of approximately two seconds between transmissions. Using this method, transmitters that are numbered sequentially starting at 1 will have quite different intervals between transmissions until the number of transmitters in the system becomes large. The result is that if two transmissions overlap once, they are guaranteed not to overlap again for some time.

Chapter 4

Ethernet/IP/UDP Communications

4.1 Choice of Embedded Ethernet Hardware

There are several vendors that sell easily programmed embedded ethernet communication devices designed for use with microcontroller-controlled systems. However, to save costs and also to give me the opportunity to explore low-level network programming, I decided to simply use an ethernet controller like those found on computer ethernet cards which has an 8-bit mode allowing it to be driven by a microcontroller. Several examples of microcontroller programs for operating two common such ethernet cards, the Realtek RTL8019AS and the Crystal CS8900, are available.

A problem with this approach, though, is that the two ethernet controller integrated circuits mentioned above are both in surface-mount packages, and proper facilities are not available to me for mounting surface-mount IC's on printed circuit boards (PCB's). Also, it would be an extra cost to have PCB's made for mounting surface-mount IC's. Because of this, several small companies sell network interface cards designed for microcontrollers that consist simply of an ethernet controller and ethernet jack with supporting components mounted on a small PCB that is intended to connect to another PCB using headers. I chose to use such a network interface card from EDTP Electronics, the "Packet Whacker". Were the receiver and display devices to be mass manufactured, however, it would be less expensive and more efficient to purchase ethernet controllers and jacks and their supporting components separately and mount them directly on the device PCB's.

4.2 Shuttle Tracker Protocol

A format for packets from RF receivers reporting new van information and a format for packets from the server reporting van information to display clients were developed, but not fully implemented.

4.3 Server

The server was not implemented

Chapter 5

Transmitter Hardware

A prototype transmitter designed to be powered through an automotive "cigarette lighter" receptacle was constructed. It contains a PIC16LF873A microcontroller, a Linx LC transmitter, and an ultra-low-quiescent-current voltage regulator designed to allow for the transmitter to remain powered and transmitting periodically while the vehicle is not running without depleting the vehicle's starting battery (current draw while not transmitting should be on the order of several microamperes, and current draw while transmitting should be on the order of several milliamperes, for a period in the range of ten milliseconds every two to four seconds). This prototype was not tested (a separate development prototype attached to a breadboard was used during development and testing of the RF identification scheme during the summer of 2004).

Chapter 6

Receiver/Display Hardware and Software

A dedicated display device able to show the current time, the time and location two vans were last seen, and the signal strength of received transmissions at four locations was constructed.

6.1 Display Drivers

The LED's used for the display are driven by seven daisy-chained Motorola MC14489B LED drivers.

6.2 "Operating System" and Scheduling

The software written for the receiver/display device allows for multiple tasks to happen at once in small increments by maintaining state variables for each task, and repeatedly calling update functions for each task sequentially. The tasks include:

- UART reception of data,
- A/D conversions for measuring signal strength of received data,
- receiving and processing ethernet frames,
- updating the LED display data and sending out the daisy-chained data in order using the synchronous serial port module,

- and sending new van data when available to the server.

Timekeeping is accomplished by using the capture/compare/pulse-width-modulation (CCP) module of the PIC16F877A microcontroller to reset the timer1 module every millisecond on average (timer1 is 16 bits wide and increments 4,915,200 times every second with the oscillator running at 19.6608 MHz, so the CCP compare value is set at 4,915 normally, and set to 4,916 every fifth timer1 reset, to make timer1 overflow exactly five times every five milliseconds). The incrementing of a millisecond-counting register is accomplished with an interrupt that is generated every time timer1 is reset, and further quarter-second, minute, hour, monthday, weekday, month, year, century, and millenium values are incremented when necessary. Any of the applications running can set a time marker against any register up to the year register, and have a flag set when the marker matches. Because the time registers are updated by an interrupt service routine, it is certain that a time marker will never be missed, and applications can use these markers for scheduling time-sensitive tasks.

Also, macros and functions using a stack to allow passing multiple arguments to functions and storing variables before calling a function were written. The timekeeping and stack functions were written and work, the display updating routine was written and works, and the ethernet communication routines are still in the process of being written, but are able to send packets. PIC ethernet communication routines able to both send and receive packets had been previously successfully written and tested using a PIC ethernet communication demonstration board from Microchip, the maker of the PIC microcontrollers. Also, although the RF receiver portion of the receiver/display device has not yet been constructed, a prototype receiver and transmitter setup using PIC prototyping boards was used to successfully develop and test the RF identification routines during the summer of 2004.