UTCM

Improving the Quality of Life by Enhancing Mobility

University Transportation Center for Mobility™

DOT Grant No. DTRT06-G-0044

Bluetooth®-Based Travel Time/Speed Measuring Systems Development

Final Report

Darryl D. Puckett and Michael J. Vickich

Performing Organization
University Transportation Center for Mobility™
Texas Transportation Institute
The Texas A&M University System
College Station, TX

Sponsoring Agency
Department of Transportation
Research and Innovative Technology Administration
Washington, DC

Texas Transportation Institute

UTCM Project #09-00-17
June 2010

Technical Report Documentation Page

1. Project No.	2. Government Accession No.	3. Recipient's Catalog No.
UTCM 09-00-17		

4. Title and Subtitle
Bluetooth®-Based Travel Time/Speed Measuring Systems Development

5. Report Date
June 1, 2010

6. Performing Organization Code
Texas Transportation Institute

7. Author(s)
Darryl D. Puckett and Michael J. Vickich

8. Performing Organization Report No.
UTCM 09-00-17

9. Performing Organization Name and Address
University Transportation Center for Mobility™
Texas Transportation Institute
The Texas A&M University System
3135 TAMU
College Station, TX 77843-3135

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DTRT06-G-0044

12. Sponsoring Agency Name and Address
Department of Transportation
Research and Innovative Technology Administration
400 7th Street, SW
Washington, DC 20590

13. Type of Report and Period Covered
Final Report
June 1, 2009 – May 31, 2010

14. Sponsoring Agency Code

15. Supplementary Notes
Supported by a grant from the US Department of Transportation, University Transportation Centers Program

16. Abstract

Agencies in the Houston region have traditionally used toll tag readers to provide travel times on freeways and High Occupancy Vehicle (HOV) lanes, but these systems require large amounts of costly and physically invasive infrastructure. *Bluetooth* is a widely used technology embedded in cellular telephones and in-vehicle applications for exchanging data over short distances. The initial demonstrations of wireless address matching were primarily designed to prove the ability for *Bluetooth* technology to produce matches between two points on a roadway outfitted with the proper *Bluetooth* reader equipment, with a resulting travel time and speed calculated. Costs for *Bluetooth* travel time measurement systems are one to two orders of magnitude below costs for traditional toll tag reader equipment, depending on the application. This cost advantage could significantly lower the threshold for hundreds of agencies and private entities to enter the travel time measurement market, but there is little guidance on the application. Development and testing of various prototype software and hardware platforms were conducted to use the anonymous Media Access Control (MAC) address from each *Bluetooth* device to measure and report real-time traffic conditions. Several issues were examined and resolved in order to develop a true first generation "product." The development of this method and process is pending patent approval.

17. Key Word
Bluetooth Technology, Travel time, High occupancy vehicle lanes, Infrastructure, Media access control addresses, Houston (Texas), Technological innovations, Cellular telephones, Research projects, Traffic speed

18. Distribution Statement
Public distribution

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	53	n/a

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Bluetooth®-Based Travel Time/Speed Measuring Systems Development

Darryl D. Puckett
Research Scientist
Texas Transportation Institute

Michael J. Vickich
Senior Systems Analyst
Texas Transportation Institute

Sponsored by the
University Transportation Center for Mobility™

Final Report
Project 09-00-17
June 1, 2010

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Acknowledgments

Support for this research was provided by a grant from the U.S. Department of Transportation, University Transportation Centers Program to the University Transportation Center for Mobility™ Project 09-00-17.

The authors thank the City of Houston, Texas, Department of Public Works and Engineering, for assistance and in-kind contribution of infrastructure and electrical power that was supplied in existing traffic signal cabinets at arterial intersections. Specifically, the authors wish to thank Mr. Jeff Weatherford, P.E., Deputy Director of Traffic and Transportation, and his staff for their cooperation, guidance, and review of the effort documented in this report.

Table of Contents

List of Figures and Tables

Executive Summary

Since 1996, agencies in the Houston, Texas, region have used an Automatic Vehicle Identification (AVI) system, using electronic toll tags, to provide real-time travel time and speed information for freeways, tollways, and HOV lanes. While AVI provides a very robust data set for travel time monitoring, the AVI system requires:

- a significant investment in physical and communications infrastructure
- can be difficult to install and maintain, and
- utilizes vendor–specific, proprietary software and protocols.

These capital and maintenance costs become prohibitive as traffic monitoring on arterials and rural highways is considered. However, because the travel time data produced by the AVI system are of tremendous value to traffic management, traveler information, and planning applications; research into more cost effective and efficient methods for determining roadway travel times was desired.

The *Bluetooth®* protocol is a widely used, open standard, wireless technology for exchanging data over short distances. The technology is frequently embedded in mobile telephones, Global Positioning Systems (GPS), computers, and in-vehicle applications such as navigation systems. Each *Bluetooth* device uses a unique electronic identifier known as a Media Access Control (MAC) address. Conceptually, as a *Bluetooth*-equipped device travels along a roadway, it can be anonymously detected at multiple points where the MAC address, time of detection, and location are logged. By determining the difference in detection time of a particular MAC address, the travel time between locations can be derived. A significant advantage of the use of *Bluetooth* MAC addresses for travel time monitoring is that typically only one inconspicuous roadside installation is necessary (consisting of field processor with appropriate software and antenna) to capture the unique address of *Bluetooth* devices traveling in all directions of flow.

Staff from the Texas Transportation Institute (TTI) has been involved in several *Bluetooth* demonstrations conducted for Texas and Ohio Departments of Transportation, Harris County, Texas, and for the cities of Houston, Dallas, El Paso, and College Station, Texas. These initial short-duration demonstrations were primarily designed to prove the ability for *Bluetooth* technology to produce matches between two known points, with a resulting travel time and speed calculated for freeway and arterial facilities.

The *Bluetooth* concept is especially attractive because of its low capital costs as compared to AVI or other methods. Equipment costs for the TTI-developed *Bluetooth* travel time measurement systems are one to two orders of magnitude less than that for traditional toll tag reader equipment. This cost advantage could significantly lower the threshold for local agencies and interested private entities to enter the traveler information market. However, there is little guidance on the appropriate use of this new technology. The *Bluetooth*-based systems are ideal for real-time operational and planning use as well as for evaluating roadway performance measures.

While the feasibility of the TTI prototype system has been successfully demonstrated as briefly described above, there is a need to further test and develop equipment and systems to read and process data in a more permanent roadside environment, particularly on arterial streets.

Through this project, researchers evaluated several potential process controllers and identified an existing, commercially available, U.S. manufactured component for the field hardware. When configured with a refined version of the field software developed by TTI research staff, this field hardened device performed successfully. Because of the early positive results, several agencies have now deployed the system and are currently using the device, installed in traffic signal cabinets, for travel time monitoring.

The intellectual properties developed under this grant enabled The Texas A&M University System to submit patent and copyright applications and a subsequent commercial license for the method and process of utilizing *Bluetooth* technology for collecting traffic information. It is anticipated that a commercial product will be marketed and sold before the end of calendar year 2010.

Introduction

Since October 2008, researchers at TTI have been involved in several research efforts to demonstrate the feasibility of utilizing *Bluetooth* MAC address matching to provide real-time travel time data for freeways and arterials. Findings from these demonstrations indicated that the calculated travel time information for the *Bluetooth*-based system was comparable to those results found using technologies such as License Plate Recognition (LPR) and Automatic Vehicle Identification (AVI) (toll tag technology).

These early feasibility demonstrations identified a number of issues that could be addressed with additional detailed research. Based on discussions with agency partners at Houston TranStar (the Greater Houston Transportation and Emergency Management Center), researchers identified the following issues during the first phase of this one-year project for further research:

- Determining field processors that employ reasonable temperature specifications for long-term pilot or permanent deployments in the signal cabinets of municipalities that are expected to implement the technology;
- Exploring the various antenna types available for reading *Bluetooth* devices at the roadside—each with different characteristics that influence the detection rate of devices and thus the quality of traffic data produced;
- Determining the most appropriate field processor/antenna combination and configuration to better define the various anticipated field installations; and
- identifying enhancements needed for software to detect and process *Bluetooth* MAC addresses at the roadside that can be run with little or no maintenance in a traffic cabinet.

It was also anticipated that temporary or semi-portable deployments of the *Bluetooth* reader devices would be beneficial for locations where power or other infrastructure (traffic signal or Intelligent Transportation System (ITS) cabinets, for example) may not be available. The second phase of the project conducted examinations and measurements to explore various means of satisfying those temporary reader needs. The availability of these stand-alone devices could be beneficial for short-term data collection, special event traffic monitoring deployments, and evacuation route monitoring.

The third phase of this research project was completed with continued cooperation of the City of Houston Public Works Department. In this phase, researchers deployed a small network of reader devices in traffic signal cabinets in a section of west Houston. This longer-term pilot deployment allowed for the development and testing of enhancements to both the field and host software components in order to make the system more reliable, while providing a maximum amount of quality data samples from the roadway network for analysis. In addition, it enabled examination of grid/arterial network deployments as opposed to the previous single roadway/highway applications. A prototype web application was also developed to display the representation of congestion levels on arterial streets.

Bluetooth Traffic Monitoring Concept

Bluetooth traffic monitoring is a probe-based technique for determining travel times and speeds between points on a roadway network. *Bluetooth* technology is used for short-range wireless communications and is frequently embedded in electronic devices including mobile phones, GPS units, computers, and in vehicle navigation systems. Each *Bluetooth* device contains a unique identifier known as a Media Access Control address, assigned by the manufacturer of the device. The standard format for a MAC address is six groups of two hexadecimal digits separated by hyphens or colons. A representative example of a MAC address might be "01:23:45:67:89:01."

As a *Bluetooth* enabled device travels along a roadway, a roadside device "reader" logs the unique *Bluetooth* MAC address, along with its location and time of day that the device was detected. When the same MAC address is detected at distinct points on a roadway segment of a given distance, a travel time can be determined by calculating the difference in detection times at those points. Using the known distance between the points along a segment, an average speed can then be determined.

Several key components are required for a system to be able to estimate vehicle travel times and speeds using *Bluetooth* devices as probes, including the following:

- A roadside system must be able to detect and process *Bluetooth* MAC addresses as the vehicle travels along the monitored roadway. Typically, a roadside system will be housed inside of a traffic equipment cabinet in close proximity to the roadway being monitored.
- The system must also include a radio capable of reading the MAC address of *Bluetooth* devices. The radio can be embedded either into the Central Processing Unit (CPU) system board or in the form of an external adapter. External adapters are typically connected to a Universal Serial Bus (USB) port of the CPU processing device. The radio may also be connected to an external antenna for extending the detection range of the *Bluetooth* equipment.
- The detection and processing of *Bluetooth* devices take place on a field located CPU capable of running software for detecting, processing, and forwarding *Bluetooth* device addresses and other information to a central location.

Similar to other probe-based techniques for determining travel times, multiple roadside systems are necessary to provide traffic data from a roadway. Data received from multiple reader locations allow for the re-identification of MAC addresses at adjacent locations and make the subsequent estimation of elapsed travel times possible. In one model, a central software component receives and processes MAC address data from each roadside reader location. The central host software then is used to determine individual travel times and estimate average travel times over time intervals for configured roadway segments based on the given locations of the *Bluetooth* reader systems at the roadside.

As an illustration, two roadside *Bluetooth* detection modules are installed in traffic signal cabinets one mile apart. As a vehicle containing a *Bluetooth*-enabled mobile phone passes the first roadside detection point, the MAC address is read by the first reader, along with the time of detection and location of the reader. This information is instantaneously transmitted to the central software component. The central software component stores the information and then scans system inputs for subsequent occurrences of the MAC address detected at the first location. As the same vehicle passes the second roadside detection point, the MAC address, time of detection and location of the reader is logged and transmitted to the central software component. Knowing the distance between detection points, the central software component is then able to calculate the segment travel time and speed by calculating the difference in detection time at each point for that individual MAC address. Aggregated observations of multiple vehicles between two points can then produce highly accurate estimates of the overall flow of traffic. Figure 1 shows a diagram of the components required and concept used in the *Bluetooth* Traffic Monitoring Concept.

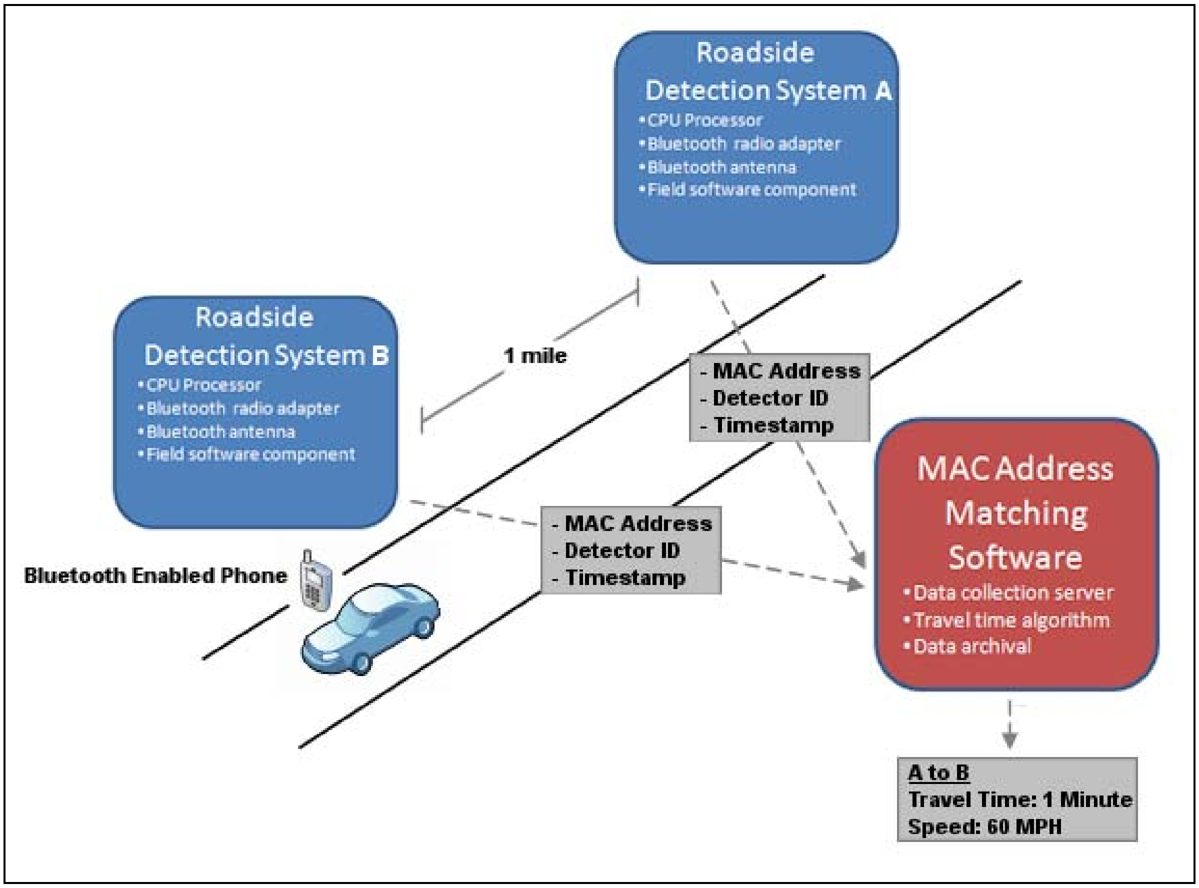


Figure 1. *Bluetooth* Traffic Monitoring Concept

Phase I. Device Development and Testing

Phase I of this research effort focused on five key areas, which included

- issues surrounding field processor capabilities
- antenna capabilities and their impacts on detection rates
- performance of various field processor/antenna combinations and configurations
- enhancements needed to field processor software to optimize detection rates and
- efficiently process *Bluetooth* MAC addresses in a traffic cabinet

This phase also investigated technology requirements to provide temporary or semi-portable deployments of the *Bluetooth* reader devices where power or other infrastructure (traffic signal or ITS cabinets, for example) may not be available.

Requirements and Considerations for MAC Address Processing Equipment

Initial *Bluetooth* reader feasibility efforts undertaken in advance of this research project utilized simple, inexpensive, consumer-grade notebook computers as the roadside CPU. These devices were used to detect *Bluetooth* devices due to their low cost, portability and small size. However, as environmental temperatures began to rise above 90 degrees Fahrenheit inside the suitcase-type container used in testing, the notebook computers began shutting down to prevent overheating. A significant requirement for the *Bluetooth*-based traffic monitoring system is that the *Bluetooth* detection device be able to withstand temperatures that may exist in the traffic cabinet. Therefore, in this first phase of the project, the researchers investigated various devices that would meet the environmental demands of longer-term or permanent field installation in a traffic cabinet.

For the prototype demonstrations, the field software used to detect and process the *Bluetooth* devices at the roadside was developed under the Microsoft Windows© operating system and utilized the Microsoft *Bluetooth* software stack to interface with the radio module, which interrogates for the MAC address. An initial limitation on the research was to use processors that were capable of running the Windows operating system. However, the use of Microsoft Windows for the operating system was not ultimately a system requirement.

The initial criteria for the selection of permanent field processing units for testing were as follows:

1) Field hardened device capable of withstanding the extreme temperature variations typically experienced in a traffic cabinet. At a minimum, the device should be capable of withstanding temperatures ranging from -4°F to 167°F.
2) The device must contain:
    a. an Ethernet port to utilize the most common types of communication devices, and

    b. a USB port to utilize the most common types of external *Bluetooth* adapters available.
3) The device must be fairly compact in size to fit inside an existing traffic cabinet without interfering with, disrupting, and/or displacing existing equipment.
4) The device must be at a price-point where cost is not a deterrent to wide scale deployment, enabling owning agencies to realize significant cost advantages over existing traffic data collection technologies (such as AVI or LPR).
5) The device must be capable of running a Microsoft Windows-based operating system to run the field software (this requirement was later abandoned as a lightweight Linux environment proved more appropriate for the application).
6) The device must have remote accessibility using common applications such as Secure Shell (SSH) and File Transfer Protocol (FTP) as part of the operating system.

The field software responsible for processing *Bluetooth* MAC addresses and communicating back to the central software host uses very few system resources and does not require a CPU with enhanced performance. For the purposes of the demonstration, all the processors evaluated had specifications that far exceeded the requirements of the roadside software application.

Two processors were initially tested for the field hardware device: 1) Aaeon AEC-6840, and 2) Advantech ARK-4180. These two processors were in use by area agencies in mobile traffic data applications and researchers were familiar with their capabilities. In addition to initial testing with these two processors, researchers undertook a search for additional processing platforms. The primary purpose of the search was to find a low power processor that could run on 12 volts so that it could run from power provided by lower-cost batteries and/or solar power configuration.

Research (at that time) indicated that processing units with potential for application in the *Bluetooth* travel time monitoring system were imminent, but had not yet made it into production. In addition, all but one of the potential processors was manufactured outside the United States. However, an embedded process controller was identified that met the requirements stated above — the model TS-7800 manufactured by Technologic Systems. This device provided a standard set of peripherals (such as an Ethernet port for network communications and USB ports for external peripherals such as the *Bluetooth* adapter) that would likely be used in the system. In addition, the device was shipped with the full Linux kernel installed on the onboard flash and was capable of running a wide range of services if needed, including remote clients like SSH and FTP for maintaining and troubleshooting the devices from a central location.

Among other factors, but primarily because of the tremendous cost advantages the TS-7800 product offered over the other candidates, the researchers decided that the Microsoft Windows version of the *Bluetooth* roadside software should be migrated to a component capable of running the Linux operating system. The TS-7800 processor had provision for multiple platforms for software development, including C, Java, Perl, and Python. The researchers decided to utilize the Python programming language to construct the software component. Python is a widely supported, powerful, yet easy to use language capable of running on multiple platforms—including Linux and Microsoft Windows.

Unlike the other processors considered and/or tested, the TS-7800 is manufactured and assembled in the United States and readily available at an appropriate price-point considering the features provided. Once the Linux-capable version of the field software was ready, this device was installed in the field in conjunction with the other processors for testing.

Field Tests of Process Controllers

The initial field test location was on South Main Street in Houston, Texas, and was the same as the test bed used in a previous comparison of LPR with the *Bluetooth* technology (*1*). In the test segment, South Main Street is a six lane major arterial that extends south, from the Texas Medical Center, to US 90A. For the test, the devices were installed to measure traffic conditions for both the northbound and southbound directions of South Main Street. Figure 2 shows a map of the roadside testing locations located on South Main Street.

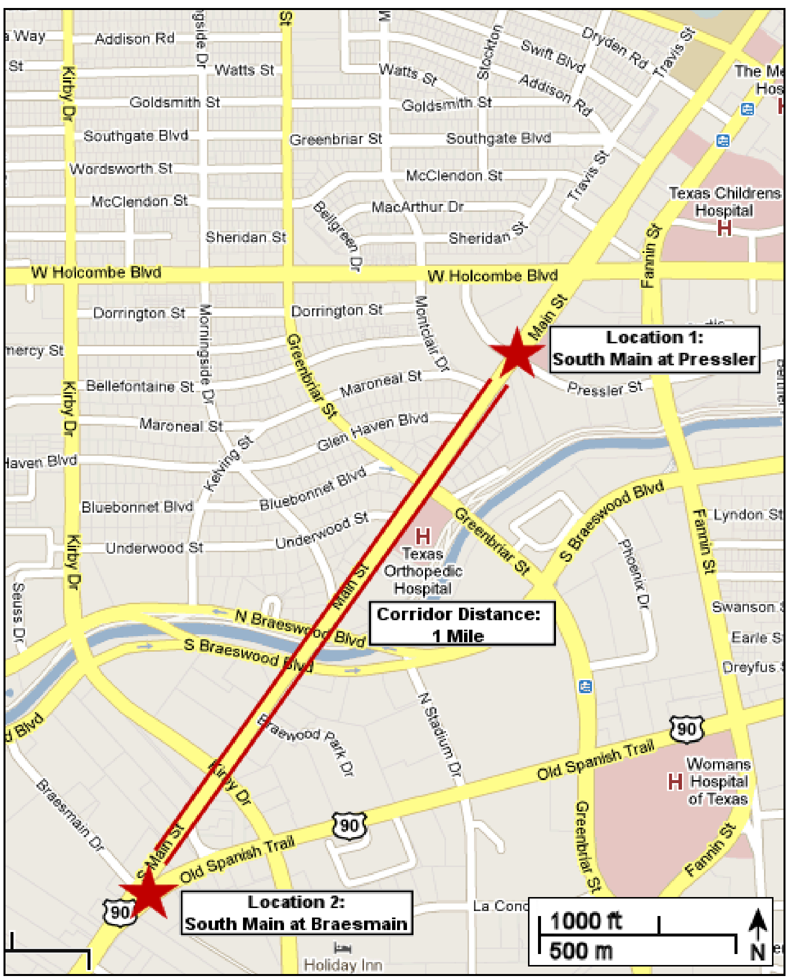


Figure 2. Initial Testing Locations

All initial field tests were conducted using similar field software in each of the candidate field processors. These devices were installed in an available traffic signal cabinet where power was available. The field processor units were programmed to operate unattended and to recover from power outages as necessary. Each unit transmitted the MAC address data stream in real-time to the host software component using cellular communications. Each processor was required to

- perform a *Bluetooth* inquiry process at the roadside
- collect MAC addresses resultant from that inquiry process
- log the time and location of the device and
- 'push' that information back to the host server using the lightweight network protocol known as User Datagram Protocol (UDP)

While perfectly acceptable from an environmental standpoint, the Aaeon AEC-6840 and Advantech ARK-4180 processors contained considerably more functionality than that required for the simple nature of the *Bluetooth*-based traffic data collection application. However, it was the Microsoft Windows operating system running on these devices that presented issues during the field tests. The complex nature of the Microsoft Windows system, its inherent coordinated sub-systems, and graphical user interface dependencies contributed to processor failures on multiple occasions. The multitude of plug-and-play capabilities of Microsoft Windows operating system are not needed by the TTI *Bluetooth* traffic data collection software.

During these initial tests, random occurrences of dropped communications with peripherals — both the cellular network and USB *Bluetooth* adapter — proved troublesome. While the exact causes of these failures were not determined during the tests, it was determined that the Microsoft Windows device drivers used for controlling the peripherals were not suited for running in an "always-on" environment. These drivers would run stable for short-term data collection activities but were more likely to fail the longer they ran unattended. For instance, one failure of the system was caused by Microsoft Windows' inability to recognize the appropriate video card while there was no video being used. Specification sheets for all the equipment recommended from this phase of the project are included in the report appendix.

As mentioned in the introduction, in the latter stages of our earlier feasibility studies it became obvious that temperature (particularly in Texas) would be a critical performance variable. The consumer grade equipment (such as the low-cost notebook computer) that was initially selected for testing would not be appropriate for longer-term deployments.

Process control equipment tested to satisfy high temperatures was considerably more expensive, so one element of the field equipment testing was to determine what temperature extremes could be expected and endured. A battery operated temperature logger was deployed during the later summer months of 2009 at several of the field deployment sites in traffic signal cabinets. Figure 3 shows a sample of the recorded temperatures. The highest recorded temperature inside a non-ventilated cabinet was 123°F. This represents a differential to the outside temperature of 23 degrees. A high temperature requirement (like 60°C/140°F) would be warranted if temperatures were expected to exceed 140°F, and this was not observed in this study.

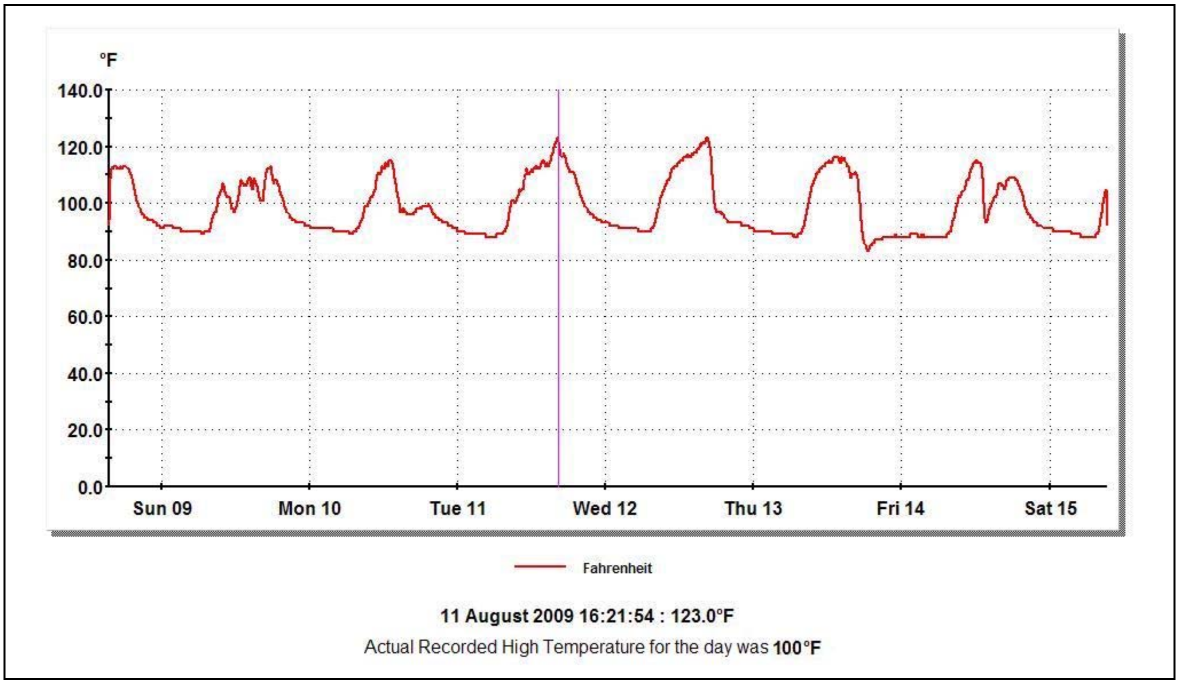


Figure 3. Internal Cabinet Temperature Recordings

Considering the results from the process controller testing, the most appropriate processor that meets basic requirements for the TTI *Bluetooth* Traffic Data Collection roadside application is the TS-7800 offered by Technologic Systems. It has proven to be reliable and consistent over the period of this research project with little to no maintenance required. The processor operates using a full-featured Linux distribution that is cost-effective for developing software applications and not overburdened with unnecessary applications or capabilities. At the time of this writing, the TS-7800 device was priced at about one-third of the cost of the other devices tested.

In addition, since there is no apparent justification for a requirement above 140°F, any additional cost to obtain the higher temperature rating is unwarranted. The TS-7800 offered by Technologic Systems has a temperature rating of -4°F to +167°F and is more than satisfactory for this application. Specifications for this device are included in the appendix.

Bluetooth Adapter Assessment

Bluetooth is a standard communications protocol intended for use over relatively short ranges. The typical classes of devices are specified as Class 1, 2, and 3, and have estimated ranges of 300 ft, 30 ft, and 3 ft, respectively. However, these effective ranges appear to vary in field deployment based on the particular field equipment. The *Bluetooth* devices use a broadcast radio communications platform, so they do not require line of sight.

The feasibility studies utilized a Linksys USB *Bluetooth* adapter based on its commercial availability and relatively low cost. The Linksys model (USBBT100) used in the tests is equipped with an articulated external antenna rated at 1.2 dBi (decibels relative to isotropic, a measurement of antenna gain). This model offered more range and directional flexibility than the other current market offerings equipped with only an embedded patch antenna.

While adequate results were obtained and the cost was not exorbitant for this product, another adapter was identified by the Harris County Public Infrastructure Division. The product is manufactured by Sena and is a Parani-UD100.

The Sena device is equipped with a standard 1 dBi antenna. More importantly, this standard antenna is mounted on the device using a standard RP-SMA connector enabling quick connection to a variety of external cable and antenna combinations. It will be demonstrated later in this report that having flexibility for antenna options could prove beneficial under certain circumstances.

The Parani-UD100 is recommended to be used in the overall application. It has a temperature rating of -4°F to +158°F and offers the capability for optional external cabling and various antenna configurations.

Antenna Evaluation

A key component of this research was to determine the effective range of the field equipment, particularly with respect to antenna selection in combination with various processors. Initially it was presumed that longer-range coverage would be required based on the relatively short specified range of the adapter and antenna and equipment. For this reason, only Class 1 devices were used in this project. By design, most of the *Bluetooth* enabled devices are for short-range operation. However, this is typically defined as a few feet to facilitate connection between two devices (for example, a headset and a cellular telephone).

Exhaustive radio propagation studies were not conducted as part of this research due to the expense of the necessary equipment. However, a testing plan was developed that would enable the empirical testing of a few combinations of processor devices and a variety of antennas. To provide limited background interference, the antenna performance study was conducted in a remote portion of western Harris County where little *Bluetooth* radio activity was present.

The processor was deployed in a prototype portable case with an external antenna and cable. This arrangement positioned the antenna at an appropriate height and facilitated the ready exchange of various antenna combinations for the test. Figure 4 shows the actual deployment.

Figure 4. Antenna Testing Deployment

To conduct the test, a researcher was deployed at various distance intervals along an isolated roadbed with three separate *Bluetooth* enabled devices with discovery mode enabled. At a distance of 30 feet, all devices registered almost the same as measured by the Receive Signal Strength Indicator (RSSI). RSSI is a measurement of the power present in a radio signal received at a particular point.

From the vantage point of the reader station, the researcher with the devices appeared, as shown in Figure 5, at 500 ft.

Figure 5. Antenna Testing View from Reader

Utilizing a specific computer program application developed for this task, RSSI was collected for 10 cycles from each of the processor/antenna devices at each range. The actual RSSI readings were saved to a file, and then the standard inquiry process was initiated and allowed to run for 2 minutes. The first examination was conducted with the standard 1dB omni-directional antenna, beginning at 30 ft and at subsequent 50 ft intervals, and extending to 350 ft from the receiver. At 350 ft from the receiver, returns began generally to be unacceptable. The same process was then conducted with 3dB omni-directional antenna, 5dB omni-directional antenna, and a 13dB Yagi (directional) antenna. Figure 6 shows the results of the testing with various antennas.

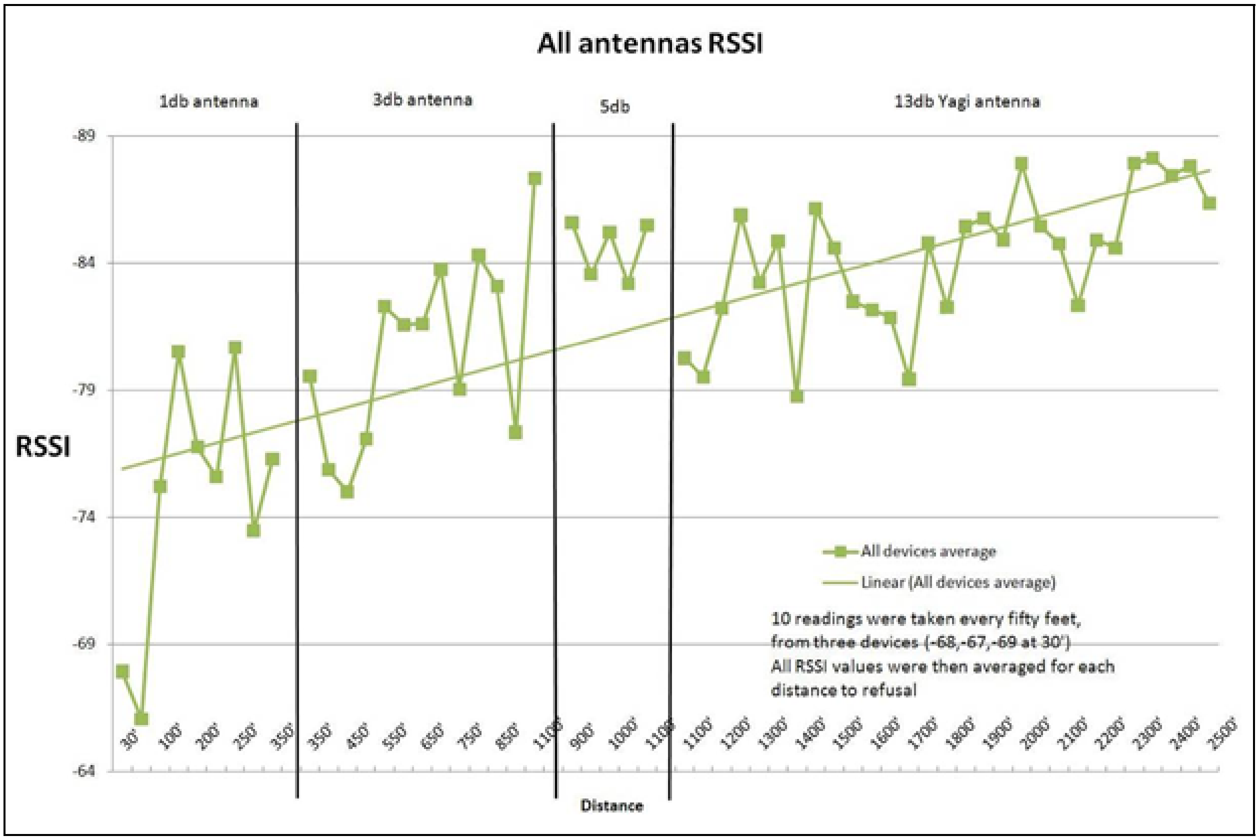


Figure 6. Antenna Testing Results

The most consistent capture rate using the TTI *Bluetooth* inquiry process was observed when the RSSI was -80 or less. As shown in Figure 6, a moderate increase in effective range (and resulting capture rate) can be gained by the use of a 5dB omni-directional antenna.

In a prototype installation in a signal cabinet, located at one quadrant of a major intersection, the 300 ft range would cover an area shown by the illustration in Figure 7.

Figure 7. Approximate Representation of 300 ft Antenna Range at an Intersection

For most geographic situations, the standard 1dB gain antenna, when coupled with a Class 1 adapter, will adequately capture MAC addresses within 300 ft or about 100 m. There is a precipitous drop (50 percent) in the capture rate between 300 and 350 ft. The coupling of a Class 1 and 1db antenna is recommended for most traffic applications except for unique situations that might require longer range. More study is needed to determine how or if other techniques could be employed for very extreme our unusual conditions. For instance, some special consideration is needed in order to separate traffic flows on frontage roads from those on freeway main lanes, or from traffic flows on High Occupancy Vehicle (HOV) or managed lanes from general purpose lanes. This distinction is currently not possible with omni-directional antennas. However, specific Yagi-type antennas or omni-directional antennas with shielding or concentrators might offer solutions, but have not been tried by the authors, so far.

Phase II. Portable Device Development and Testing

An initial *Bluetooth* travel time monitoring feasibility demonstration in Dayton, Ohio, (although not a part of this specific project) utilized inexpensive notebook computers inside a weatherproof case for the data collection deployment (*2*). The Ohio demonstration provided some observations and information that proved germane to the conduct of this research project, particularly regarding the limitations of some of the equipment used. One particular finding was that the battery in the notebook computer allowed for data collection of no more than three consecutive hours. However, the *Bluetooth* equipment case was deployed along the side of I-75 in Dayton and successfully demonstrated that the collection of *Bluetooth* MAC addresses was feasible as an alternate to floating car studies, which were conducted at the same time. As shown in Figure 8, these results for the morning trip, were an additional indication that this methodology held promise for robust and accurate travel time monitoring capability.

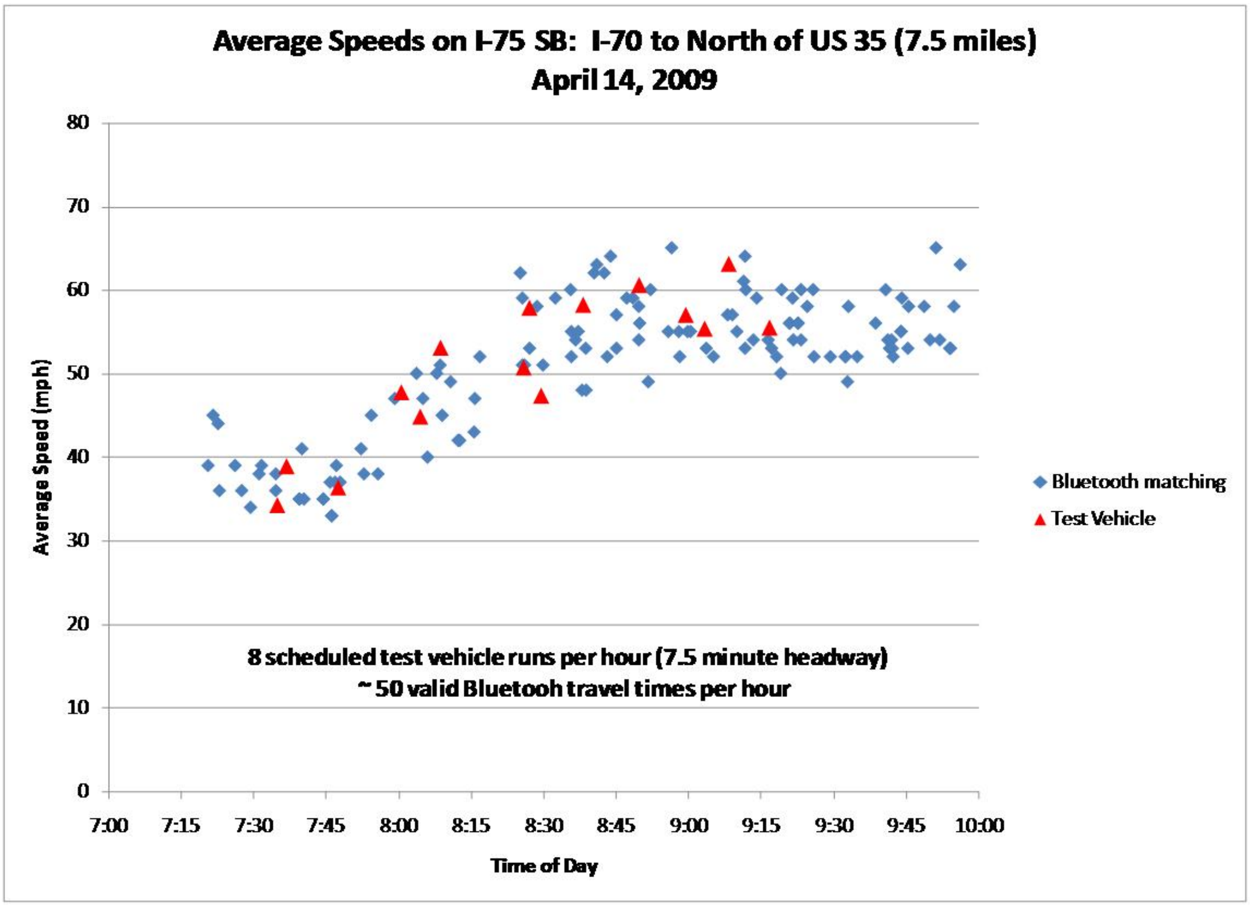


Figure 8. Dayton, Ohio, *Bluetooth* vs. Floating Car Travel Time Comparison Results

In addition to the need for a longer life battery and a field hardened processor, researchers recognized that deployment of the equipment on ground level resulted in fewer matches from the opposing travel lanes when they were obstructed by concrete barrier. In the Ohio test, the

Bluetooth reader antenna was inside the case at ground level located in proximity to a guardrail. Figure 9 shows the deployment equipment, and Figure 10 shows an example of this type of ground level deployment.

Figure 9. Dayton Testing Equipment

Figure 10. Dayton Ground Level Deployment

In several other deployment locations for the Ohio test, the equipment package was deployed on the top of a barrier, as shown in Figure 11.

Figure 11. Dayton Mid-Level Deployment

When mounted on top of the concrete barrier, the additional height of the *Bluetooth* reader afforded the ability for the antenna to gather a significantly higher number of reads from the traffic flow on the travel lanes farther away from the device, even when it was separated by a median barrier. It appeared that positioning the antenna at (or near) the windshield height of typical passenger cars provided enhanced collection opportunities.

The TS-7800 process controller identified as preferred in Phase I of this research project is rated for ambient temperatures up to 167°F and consumes just fewer than three watts with a USB cellular modem attached. Again, as in Phase I where the successful demonstration in signal cabinets proved very effective, this same processor was chosen to be utilized for the portable examinations.

Arithmetically, a field unit that draws 3 watts at 12 volts would nominally consume 0.25 amps per hour. Again, arithmetically, seven days of operation would require at least a 42 amp-hour battery at a continuous 12 volts. To confirm the theoretical electrical demands from the field equipment, various battery configurations were tested in a laboratory environment. In practical use, the amperage draw appears to increase as the battery is drained so that there is a disproportional reduction in capacity the longer the drain is allowed to continue. Figure 12 shows an example of a battery depletion curve.

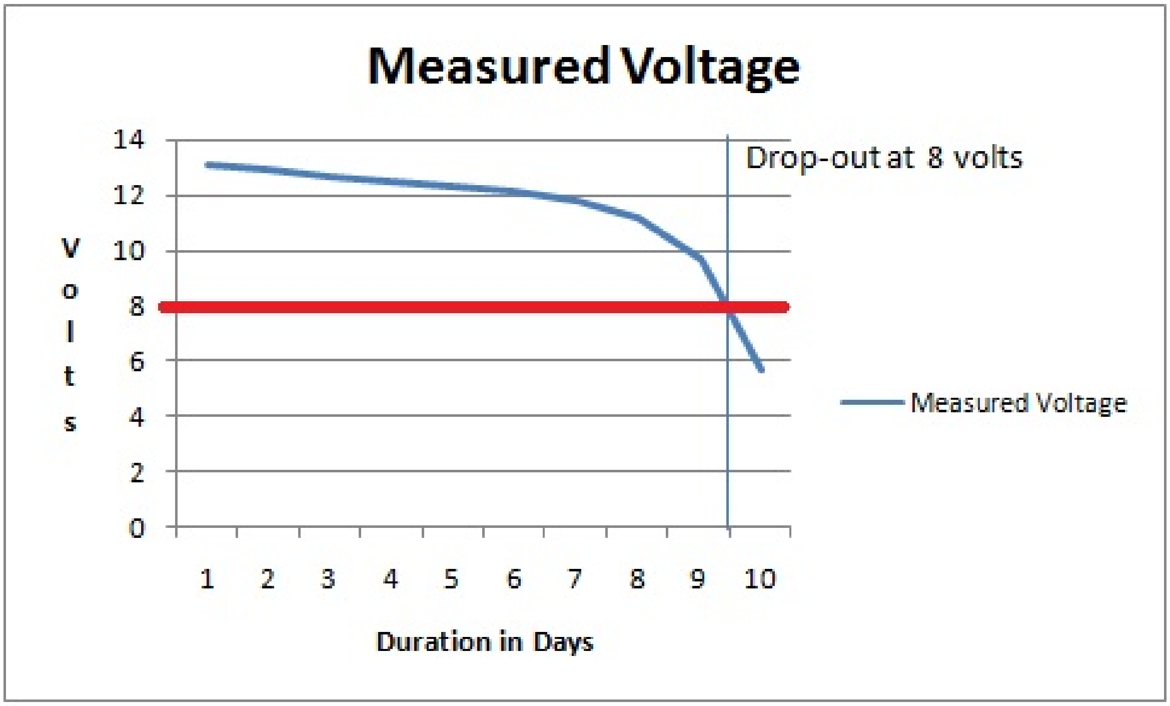


Figure 12. Measured Battery Drain

The chosen process controller, with a USB cellular modem, was measured to consume 2.88 watts at 13.1 volts with a fully charged 12 volt battery. The processor will operate on voltages between 8 and 24 volts direct current (DC). When the battery has reached a voltage of just less than 8 volts, the processor will begin a self-shutdown routine. Given that traditional battery life is considerably shortened with deep cycles of depletion, the battery manufacturer recommended against discharging the battery beyond half strength. Given the processor electrical demand and battery capacity, a 96 amp-hour battery was selected for use in the field testing.

Due to the successful deployment in Dayton, a Pelican-brand case was chosen for its light weight, ruggedness and weatherproof characteristics. Figure 13 shows the actual portable field configuration with processor, battery, cellular modem, and case.

Figure 13. Portable *Bluetooth* Travel Time Monitoring Equipment Configuration

The prototype portable unit was deployed in April 2010 on State Highway 6 in west Houston. This reader was used in conjunction with another traffic signal cabinet mounted *Bluetooth* device reader, so that a travel time monitoring segment was constructed 1.4 miles in length. Figure 14 shows the deployment location of the portable reader adjacent to a guardrail.

Using cellular communications, MAC address data were collected for nine continuous days before the battery voltage was reduced to the point that the processor terminated the application. Temperature readings were also recorded (similar to as before in the signal cabinet) and maximum temperatures did not exceed 110°F. This represented a maximum differential to the outside temperature of 18°F, just a little less than that experienced in the signal cabinets.

Figure 14. Highway 6 Portable Deployment

The portable format *Bluetooth* travel time monitoring prototypes performed as expected and can be deployed in circumstances where power is not available. However, providing a large enough deep-cycle battery to supply power for at least one week results in a portable unit weighing in excess of 100 pounds.

One of the prototype units had a small solar panel attached to the case to see if using a solar solution to charge the unit could result in a smaller battery requirement. In initial testing, the solar panel did not prove adequate to extend the battery life in the configuration that was utilized. While not specifically tested for this project, our experience with longer-term solar-powered installations suggests that a small size cabinet could be configured with a commercial-grade solar panel and the entire assembly attached to an appropriate structure for medium-term duration deployment (weeks to months). For deployment durations that are longer than one or two weeks, a portable, yet mounted, solar-based solution might prove to be a better deployment result than the suitcase approach.

Phase III. Longer-Term Device Testing and Software Refinement

The results of Phase I of this research project were used to produce a field hardened, reliable system for detecting, processing, and forwarding *Bluetooth* MAC addresses. Additional numbers of preferred system components (processors, *Bluetooth* adapters, antennas, and cellular modems) were acquired to prove the long-term viability of the system in an operational environment. It was envisioned that significant lessons could be learned from a longer-term deployment, including gaining experience used to improve the quality and reliability of both system hardware and software. This experience would ultimately be used to improve the traffic data quality derived from the system.

As part of this research, a longer-term deployment commenced from December 2009 through May 2010. The first expansion was to extend the detection coverage distance on South Main Street over two successive one-mile intervals. The first new location was at Buffalo Speedway, and another was added at West Bellfort Street. The additional locations brought the linear coverage to approximately six miles on South Main Street (three miles in each direction). Figure 15 shows a map of the expanded coverage on South Main Street.

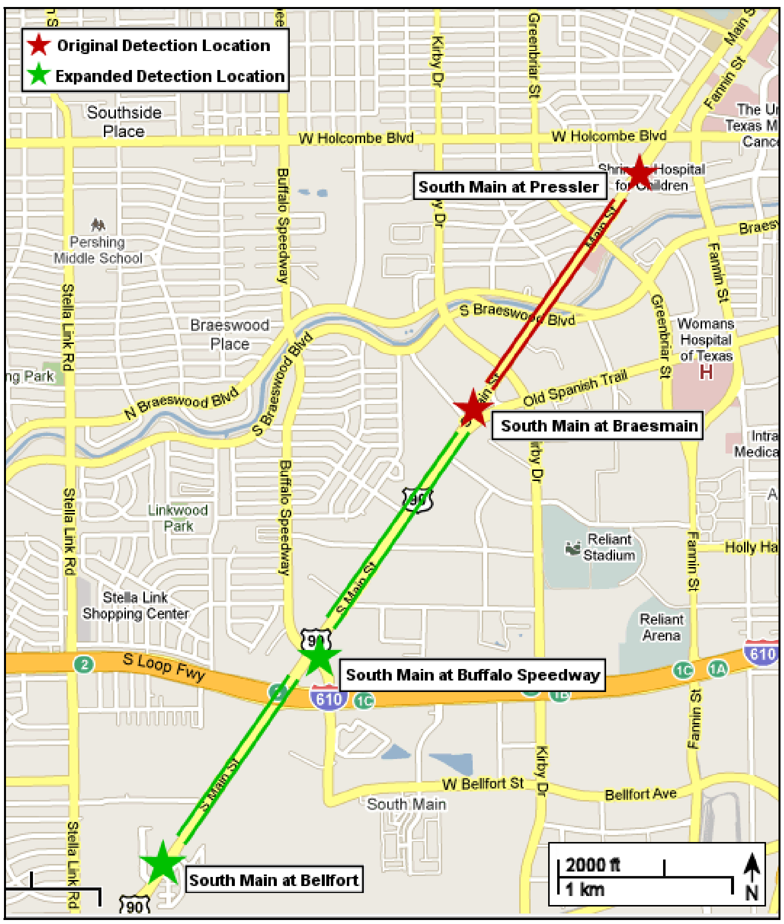


Figure 15. South Main Street Extended Travel Time Monitoring Pilot Coverage

As before, the in-kind contribution and cooperation of the City of Houston Public Works & Engineering Department allowed for the installation of the equipment (and use of the available power) in existing traffic signal cabinets. The equipment was installed in a similar manner to those deployed in the Phase I test on South Main Street. Figure 16 shows an example of the installation.

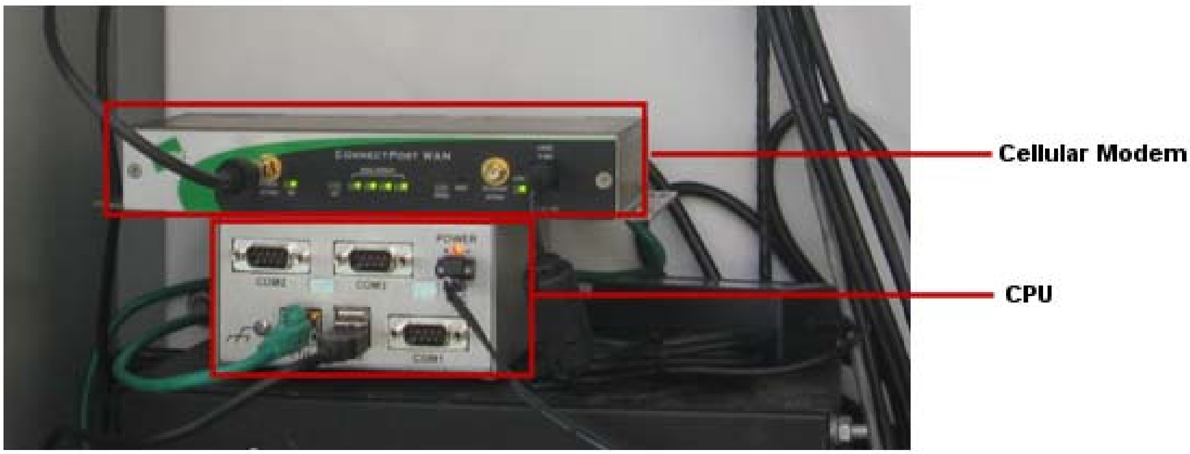


Figure 16. South Main Cabinet Deployment

During the prosecution of the research project, the City of Houston gained funding for a demonstration of ITS applications in the western part of the city. City of Houston Traffic & Transportation staff agreed that the location for the test application on an arterial grid could be accomplished in that area.

Six arterial intersections with favorable characteristics were identified for the arterial and grid network pilot study. These intersections included:

1) Westheimer Road at South Dairy Ashford Road,
2) Westheimer Road at South Kirkwood Road,
3) Westheimer Road at Wilcrest Drive,
4) Briar Forest Drive at South Dairy Ashford Road,
5) Briar Forest Drive at South Kirkwood Road, and
6) Briar Forest Drive at Wilcrest Drive.

Deployment at these intersections would then enable the monitoring of seven segments of five thoroughfares with coverage of about 16 directional miles. Figure 17 shows a map of the detection locations installed during February 2010 in west Houston.

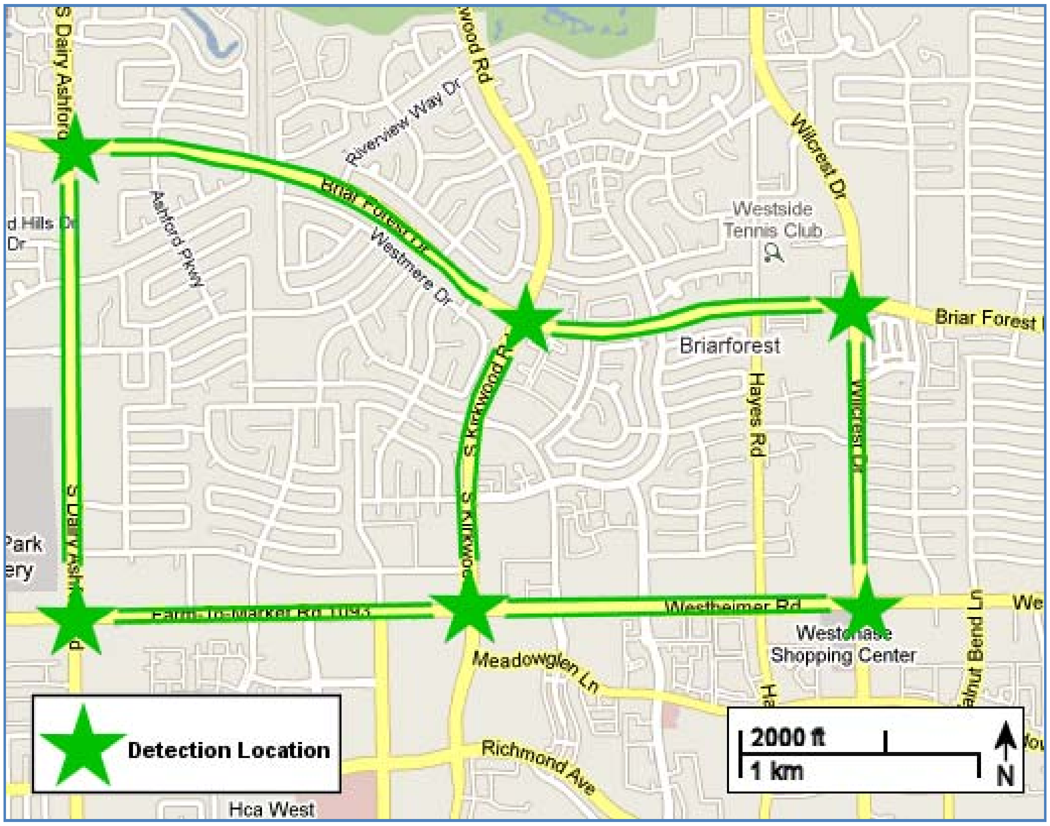


Figure 17. West Houston Arterial Grid Prototype Travel Time Monitoring Coverage

Evaluation and Performance Monitoring Capability

The goals of Phase III were to evaluate not only the long-term reliability of the field system but the quality and quantity of traffic data that the system produced. Researchers developed several software tools to aid in both real-time and historical analysis of the travel time and travel speed data collected by the system.

A web-based charting tool allows researchers to view real-time and historical data from each roadway segment monitored by the host software system. Users of the tool simply visit a Uniform Resource Locator (URL) in their web browser and select various parameters from which to view data.

For example, the chart in Figure 18 shows all of the individual speed samples collected on Westheimer Road traveling westbound from South Kirkwood Road to South Dairy Ashford Road on April 8, 2010. The green dots on the chart represent individual segment travel speed samples generated by the system.

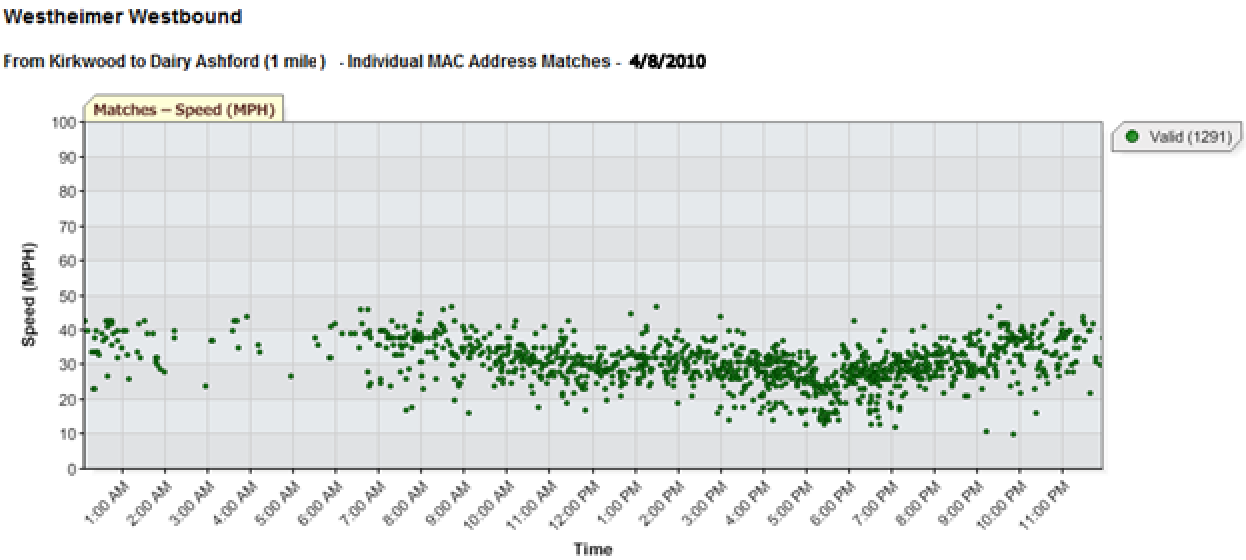


Figure 18. Typical Chart of Matched MAC Addresses on an Arterial Segment

This type of chart allowed researchers to benchmark the number of traffic data samples generated for each roadway link as changes were made to the system. Charts similar to the one shown in Figure 19 were used to show the total number of MAC address reads per hour for each individual *Bluetooth* reader location.

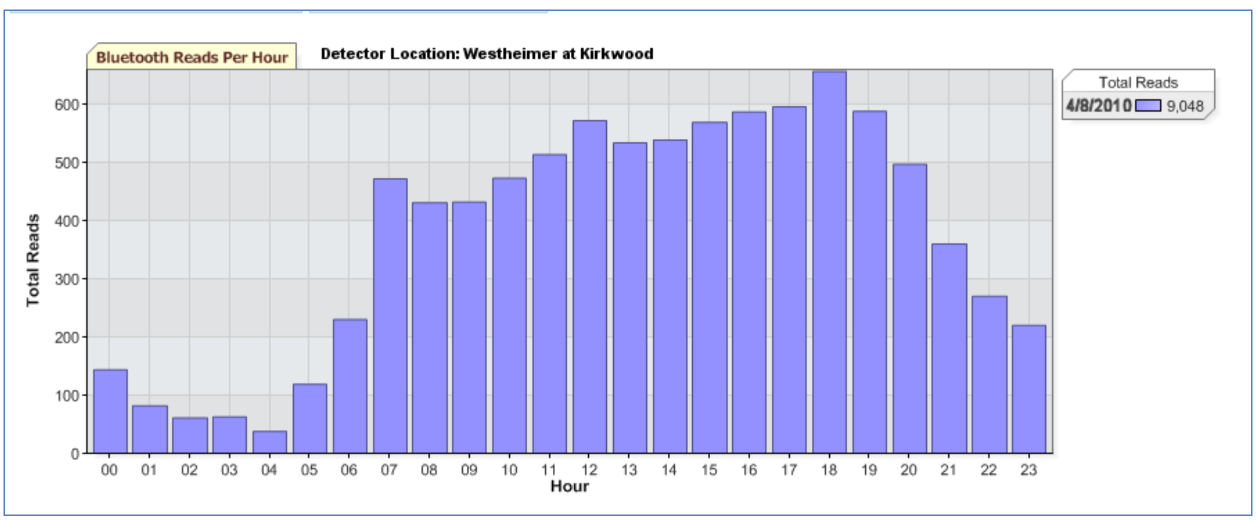


Figure 19. Typical *Bluetooth* MAC Address Read Chart, by Hour of Day

Although the number of individual *Bluetooth* MAC address acquisitions are not necessarily related to the quality of traffic data generated by the system (MAC addresses must be matched between two points to be of value), the totals provided by this type of chart can aid in analyzing the overall performance of individual reader sites.

One of the most valuable performance metrics for determining whether the overall volume of *Bluetooth* MAC address data are sufficient for providing quality information is the sample rate.

The sample rate is the number of traffic data matches per interval (for example, 15 minutes) that the host software component is able to derive. The chart in Figure 20 shows a typical chart of the total number of traffic data samples generated per 15 minute period for a roadway segment.

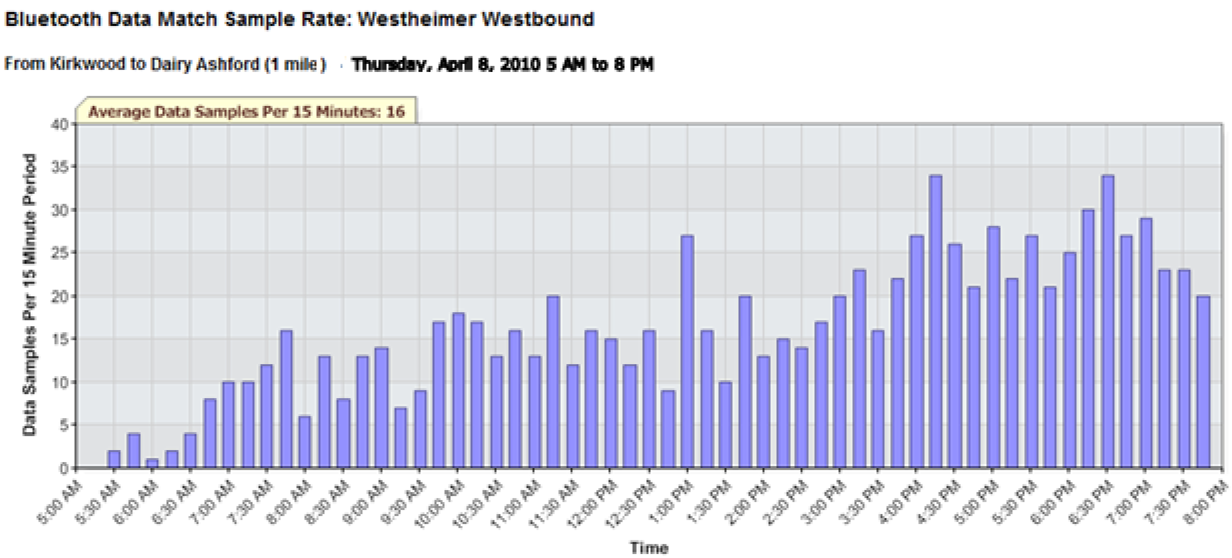


Figure 20. Typical *Bluetooth* Travel Time Match Sample Rate Chart

Ultimately, for traffic management and traveler information purposes, the travel times and speeds may be represented on a mapping platform. To satisfy this requirement and to be able to visualize the data collected by the system in real-time as a consumer of traveler information would, a web-based map was developed that shows the live conditions of the roadway segments being monitored by the system. The map was integrated into the existing Houston TranStar Traffic Mapping System. All features offered by the TranStar traveler information system were available in addition to the traffic conditions being produced by the *Bluetooth*–based travel time and speed monitoring application. Figure 21 shows a screen capture of the map.

The map uses the Microsoft Bing Maps web platform for display. In the screen capture, the map is zoomed in to the detection locations in the west Houston corridor. Users of the map are able to view color-coded lines to evaluate congestion levels. Note that on the legend, “Street Congestion” color codes are referenced differently from “Freeway Speeds.” This is due to the different operating characteristics of the two facilities (such as free flow speeds). It was beyond the scope of this project to discuss the methods utilized to determine the street congestion levels shown on the map, but a general guideline was developed using traffic engineering measure of Arterial Level of Service (LOS) as a basis for the color codes on each arterial roadway segment. Arterial LOS A, B, and C are shown as green, arterial LOS D and E are shown as yellow, and arterial LOS F is shown as red. If data are unavailable, the link is shown in a gray color.

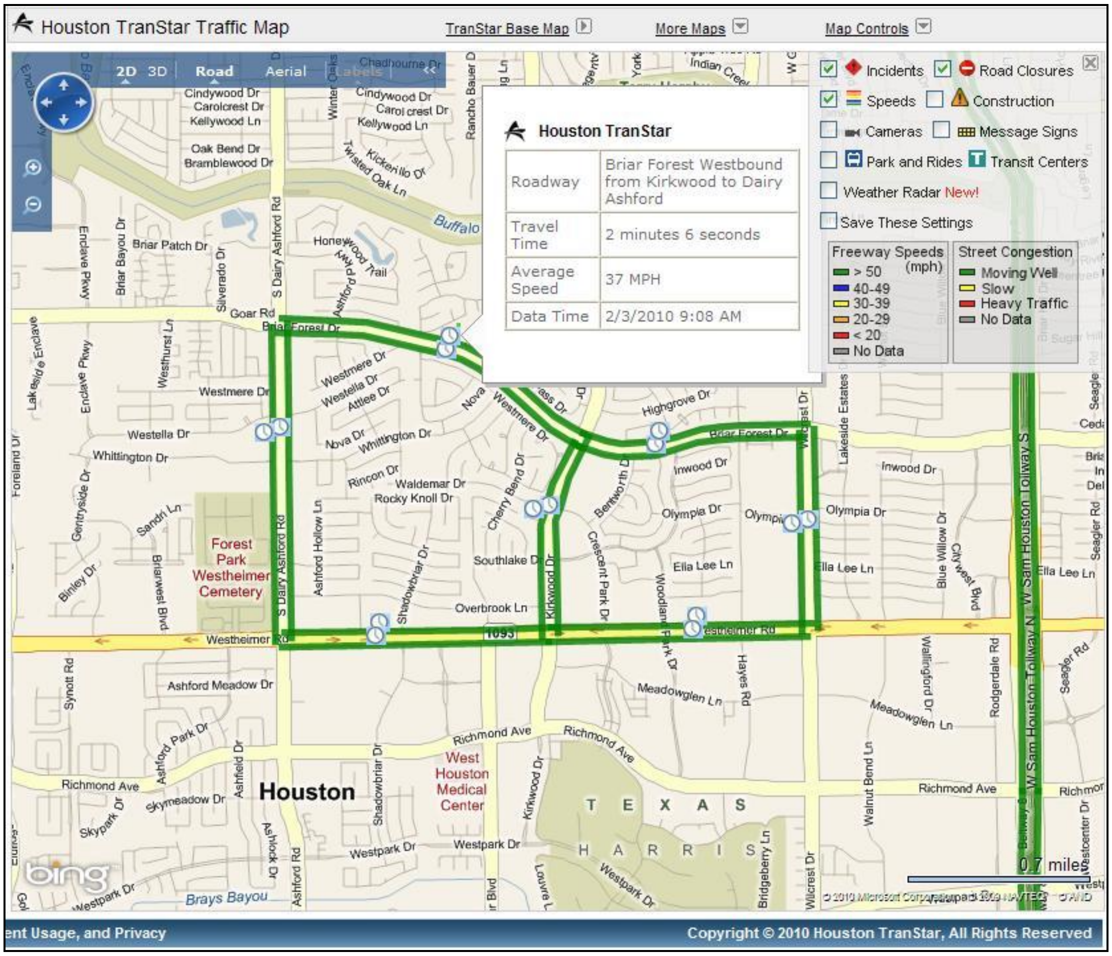


Figure 21. West Houston Demonstration Map

By hovering over the stopwatch icons positioned over the roadway segments, as shown in the screen capture, users of the map are also able to get more detailed information, including

- street (from and to)
- average travel time and speed
- data report time

Field Software Performance and Modifications

In order to evaluate the Linux-based CPU used in Phase I of the project and ultimately deployed in the long-term evaluation of the system, the field software originally written for the Windows operating system was ported to Linux using the Python programming language. In the process of rewriting the software and during the long-term evaluation of the system in the roadside cabinets, several refinements to the field software were made that improved software efficiency and the overall quality of the traffic data produced by the system.

The proof-of-concept demonstrations running on the Microsoft Windows platform and the initial Linux processor examinations used a standard *Bluetooth* device inquiry process that is available using the *Bluetooth* software application programming interface (API) on each system. The API allows external software to make requests to the *Bluetooth* protocol stack embedded in the operating system. In Microsoft Windows, the stack most often present is known as the Microsoft Windows *Bluetooth* Stack, although other stacks (such as Widcomm and Toshiba) might also be available. In Linux, the *Bluetooth* stack known as *BlueZ* is included with the official Linux kernel distributions and is therefore the most widely used. *BlueZ* is the stack that was used with the traffic data collection field software.

Although the API has many rich features that can be used to communicate, accept and receive connections, and send and receive data from *Bluetooth* enabled devices; the only feature used in the field traffic data collection software is the inquiry process that simply queries the *Bluetooth* adapter and anticipates a return of MAC addresses of nearby devices. The standard process for device inquiry with the *Bluetooth* stack on Windows and Linux scans all 79 channels of the 2.4 GHz band allocated for *Bluetooth* devices. By default, this inquiry process takes approximately 10 seconds to return the list of “found” devices. It can therefore be stated that the default process is synchronous, as the inquiry does not return any MAC addresses and cannot initiate any other requests until the entire inquiry is complete. There are several issues with this process as it relates to producing quality traffic data:

1) The actual detection time of the *Bluetooth* devices is unknown because of the 10 second inquiry time. For instance, a device could be detected within any point of the 10 second window, leaving a 10 second margin of error that could greatly influence the travel time estimation of a vehicle. All devices reported to be read in this 10 second interval are all time stamped with the same identical time stamp. The margin of error in the resulting travel time calculations increases as the spacing of the field detector stations decreases.

2) The default inquiry process will only ‘find’ and return a device record for a maximum of eight devices per scan. With this limitation, it is possible that many of the *Bluetooth* devices in range of a detection station will not be recognized by the default software process. Potentially missing devices due to this limitation decreases the probability that a travel time match will be made at a subsequent reader/detection station.

3) Considering that the inquiry process occurs continuously, it is likely that the same MAC address will be detected by the field software multiple times. As the host software attempts to match MAC addresses at the successive detection location for the purpose of travel time estimation, duplicate travel time estimates could result from the field software transmitting identical MAC addresses.

While developing the field software to operate in a Linux environment, an alternate inquiry process was employed to circumvent the issue of synchronous device requests. The new method performs device requests asynchronously, which allows detected devices to be reported to the software immediately after they are recorded by the *Bluetooth* adapter. Once implemented, this process provided an approximate 50 percent increase in the number of non-duplicative reads and increased the resulting match rate for a link by 51 percent.

The chart in Figure 22 shows the increase in travel time samples collected for a roadway segment after the software refinements mentioned above were made.

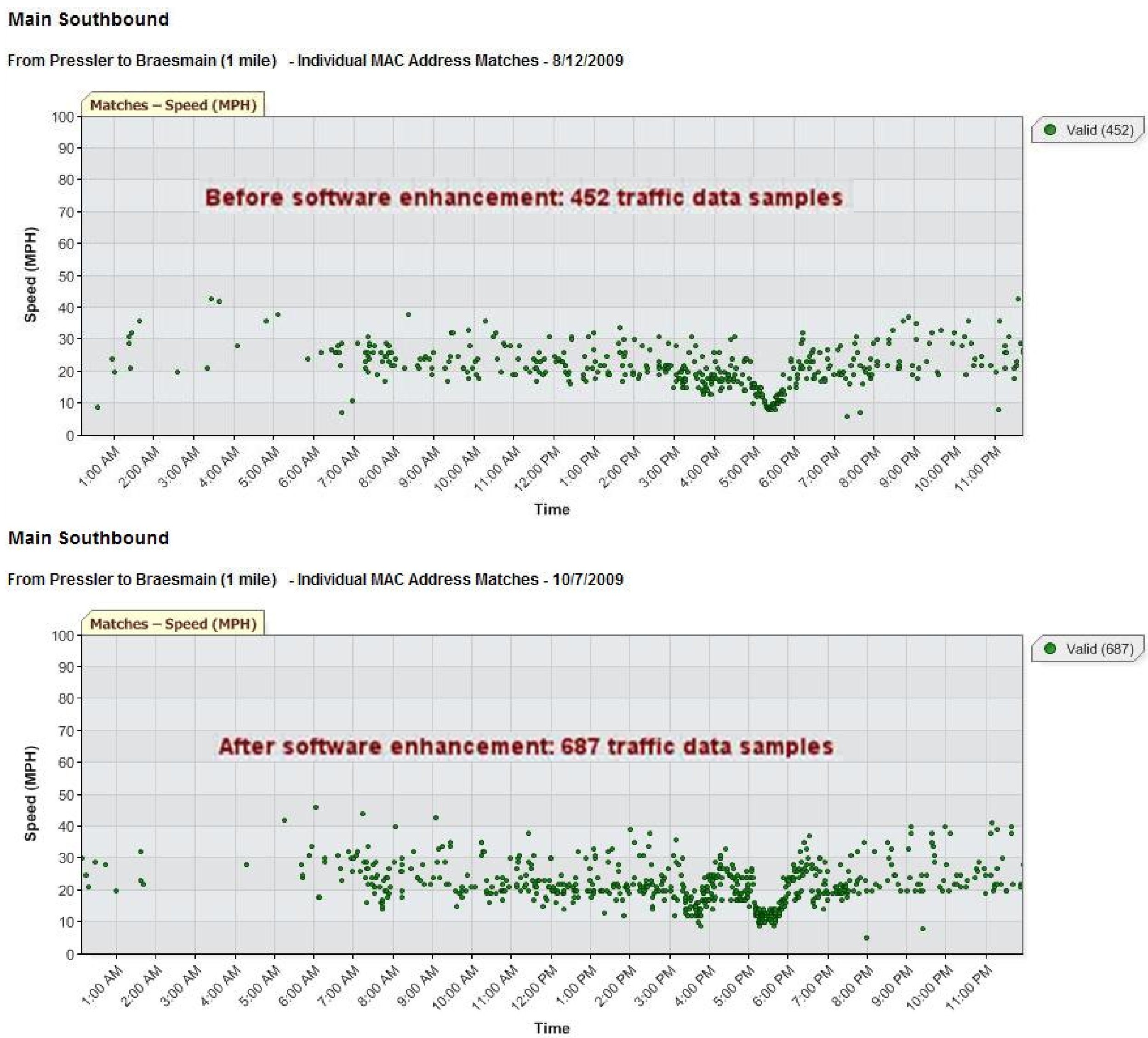


Figure 22. Field Software Improvements

Another field software refinement was added such that duplicate MAC addresses read by the same detector within a specific time window are not transmitted to the host for travel time estimation. For example, if a *Bluetooth* device is read by a detector during an inquiry and then subsequently read on the next inquiry, only the first record will be sent to the host software over the communications link. This means that the host software estimates travel times based on the first MAC address read at each detector. As shown in the first chart in Figure 23, implementation of this feature resulted in 800 percent fewer MAC addresses being transmitted to the host software the hour after it was implemented, offering significant savings in data necessary to be transferred to the host. In addition, the second chart in Figure 23 shows the corresponding travel time data samples from the origin detector station of the same location. As shown, there is no reduction in the number of travel time data samples after the duplicate reads were eliminated. In fact, the chart shows that an increase in travel time data samples may have occurred. This is most likely due to the cleaner nature of the data and the fact that elimination of the duplicates allows more efficient processing of MAC addresses by the host software. Another benefit of the elimination of duplicates is the obvious bandwidth requirement reduction that results from transmitting much less data to the host software component.

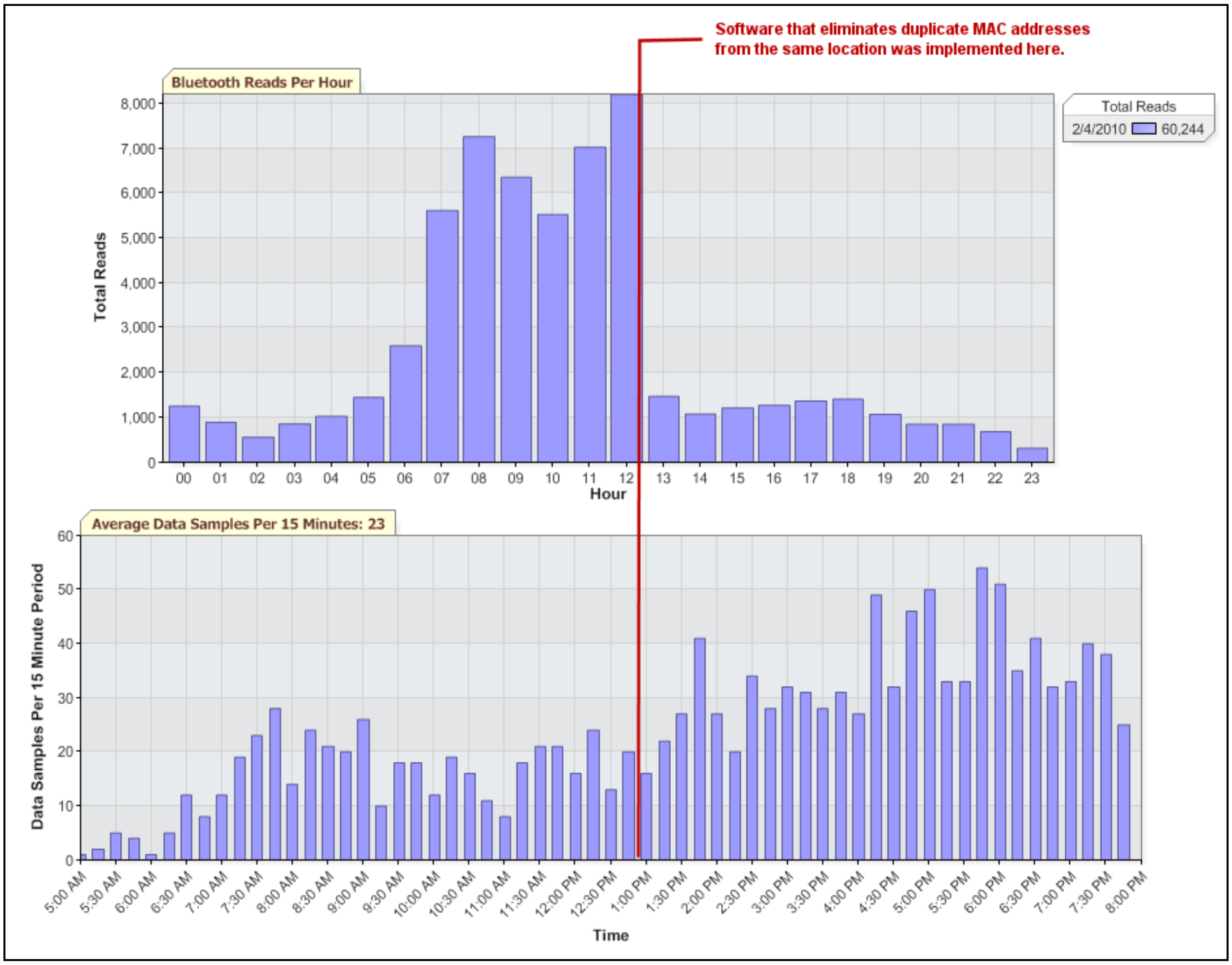


Figure 23. Elimination of Duplicates

Field Data Protocol

There were several initial requirements in developing the data protocol for packaging *Bluetooth* MAC address data and sending it from the field software to the host software.

1) Since communications method and available bandwidth are unknown in future production environments, keeping the data payload as small as possible will allow the

system to function regardless of the type of communications network on which the readers are installed; and

2) Privacy concerns about the transmission and storage of personal information from the *Bluetooth* devices require that only the MAC address is read and transmitted by the field software despite the other types of information available during a device inquiry such as device name and device type.

In the TTI methodology, each individual field device sends, in real-time, the addresses it detects to a central host software component for the purpose of travel time and speed estimation. The host software then determines the "matching" of addresses between locations to derive traffic data. Each individual record is sent as a single UDP packet to a specified network host and port. To accept a record as valid, the host software requires that each detection record obtained from the field devices conform to a specific format. The format is an ASCII string of text with each data element separated by a comma. Note each record is **not** terminated by a linefeed character. Table 1 below describes each data element.

Table 1. Field Data Elements

Element	Description
Detection Timestamp	The exact time that the reader was detected in Coordinated Universal Time (UTC). The format is "mm/dd/yyyy hh:mm:ss p". Although most instances of the host software will use the timestamp of the data as it is received by the host, this element is still necessary for troubleshooting purposes.
Reader Location	Specifies the defined location of the field device. The current standard is to utilize the primary street and the nearest cross street with the street names separated by the underscore character. For example, a reader located at the corner of Westheimer and Dairy-Ashford would be defined as Westheimer_DairyAshford. The order of the streets is not relevant to the calculation of travel times and is only used to identify the location.
Address Identifier	The unique identifier for each device detected. This is the 48-bit MAC identifier used by each wireless device. For example, 00:1E:7D:E7:6E:6D. Note that the hexadecimal numbers are separated by the ":" character.

An example of several detection records is shown below:

06/01/2010 10:00:00 AM,Westheimer_Kirkwood,00:1E:7D:E7:6E:6D;
06/01/2010 10:00:01 AM,BriarForest_SH-6,04:1E:74:E7:6E:64; and
06/01/2010 10:00:07 AM,BriarForest_SH-6,01:14:7D:E7:6E:6D.

Researchers recommend taking further initiatives to establish a standard protocol for transmitting *Bluetooth* MAC addresses for the purpose of deriving traffic information. A standard protocol would enable greater interoperability between multiple types of vendor equipment and simplify systems integration for the entities using the system.

MAC Address Anonymization and Privacy

As is the case with any probe-based data collection system, storage of personal information with regard to personal privacy is an issue that should be openly addressed. It should be stressed that MAC addresses read by the system are not directly associated with a specific user and do not contain any personal data or information that could be used to identify or "track" an individual person's whereabouts. Users who still have privacy concerns are also able to turn off the *Bluetooth* discovery function of their device, which completely prevents it from being read by this system.

To further address potential privacy concerns, a routine was added to the TTI field software that ensures that all addresses collected by the system are anonymized through encryption immediately upon receipt. This ensures that actual device addresses are not sent or stored anywhere, but rather a random set of characters. As an example, a device with the MAC address *00:24:9F:E1:FE:98* might be anonymized to *MDA6MjM6RDc6REQ6MzI6QkM* at the instant it is read by the field processor software.

Automated Communications Error Recovery

In most cases, the field software and devices have operated throughout the entire evaluation without required maintenance or failures. This equates to over 32,000 hours of runtime when all sensors are taken into account. The less than handful of failures that did occur were related to issues with the communications devices being used to transmit the data back to the host software component; in this case USB cellular modems. In cases when this did occur, a field site visit was necessary to power cycle the CPU to reinitiate communications with the cellular network.

To prevent field site visits, a method was implemented in the field software that automatically reboots the CPU if a communications failure is detected and continues for an interval of one minute. The method requires the failure to be continuous in order to reboot so that the device does not unnecessarily reboot during short, intermittent communications outages. Subsequent to implementing this feature, the site visits required for the evaluation CPUs were reduced to zero.

Recommendations Regarding Field Software and Host Performance

The preferred method to ensure that the maximum amount of MAC addresses are read in a timely manner, the *Bluetooth* software API should optimally be interfaced such that asynchronous device inquiry is implemented. Traditional *Bluetooth* stacks can be used for this process, but the improved TTI field software method increased the total number of travel time samples by over 50 percent in the evaluation. In addition, to operate the communications aspect and host software processes optimally, duplicate MAC address readings from the same location should be eliminated to prevent duplicate travel time samples from skewing the overall averages and reduce the overall bandwidth required by the field system.

Although a simple field to host data protocol was established during this phase of the project, researchers recommend taking further initiatives to establish a formal standard for packaging *Bluetooth* MAC address data for traffic information. This would allow interoperability among vendors and users of different systems.

Any probe-based data collection system should be sensitive to privacy concerns expressed by the public. Even though MAC addresses are a relatively benign piece of information, methods should still be taken to ensure probe information is as anonymous as possible. The wide array of software tools for anonymizing and encrypting data make this relatively easy to accomplish.

Any agency responsible for deploying field equipment on a wide scale will likely be concerned about long-term reliability and maintenance of the equipment. It is both costly (in terms of data loss and work hours) and time consuming to make field site visits to troubleshoot equipment. In order to reduce the number of site visits required, methods should be implemented on the field device that attempt to automatically recover from errors that might otherwise require a technician to physically address at the roadside.

Aspects of the Host Software System

In the TTI-developed *Bluetooth* Traffic Monitoring System, a central software component is utilized for accepting MAC address data in real-time from each roadside system and then estimating average travel times between configured roadway segments based on the location of the *Bluetooth* reader systems at the roadside. The basic functions of the central or “host” software components are as follows:

1) accept MAC address data from each of the roadside detectors via the UDP network protocol;
2) apply the proper algorithms for estimating the travel times between configured roadway links as defined by the detector locations;
3) apply filtering algorithms to travel time data samples and averages so that “outlier” data are not included in travel time estimations; and
4) create various output interfaces so that external applications can utilize the travel time and speed data generated by this software component.

These basic functions were already available as they were developed for the demonstration system prior to this project. However, there were some notable enhancements that were completed as a result of the long-term evaluation of the new field equipment. These functions may not necessarily be unique to a *Bluetooth*-driven travel time monitoring system, but apply to any probe-based system.

Custom Filtering Algorithm Implementation

One of the most important features of any probe-based traffic monitoring system is the ability to filter outlier data from the travel time calculations in order to determine the approximate average travel time over a defined period of time. Outlier filtering is necessary to identify the true average of the travel time data for a roadway segment and eliminate those matches representing longer (or shorter) than expected travel times.

Figure 24 illustrates a schematic for this process. For instance, if a probe were to pass a detector station, stop in a parking lot temporarily, and then continue on to pass the next detector station, the software would recognize the travel time as an outlier and not include it when estimating the average for that roadway segment. Initially, the software used a single filtering technique to identify outlier data for all the roadway segments in the detector network.

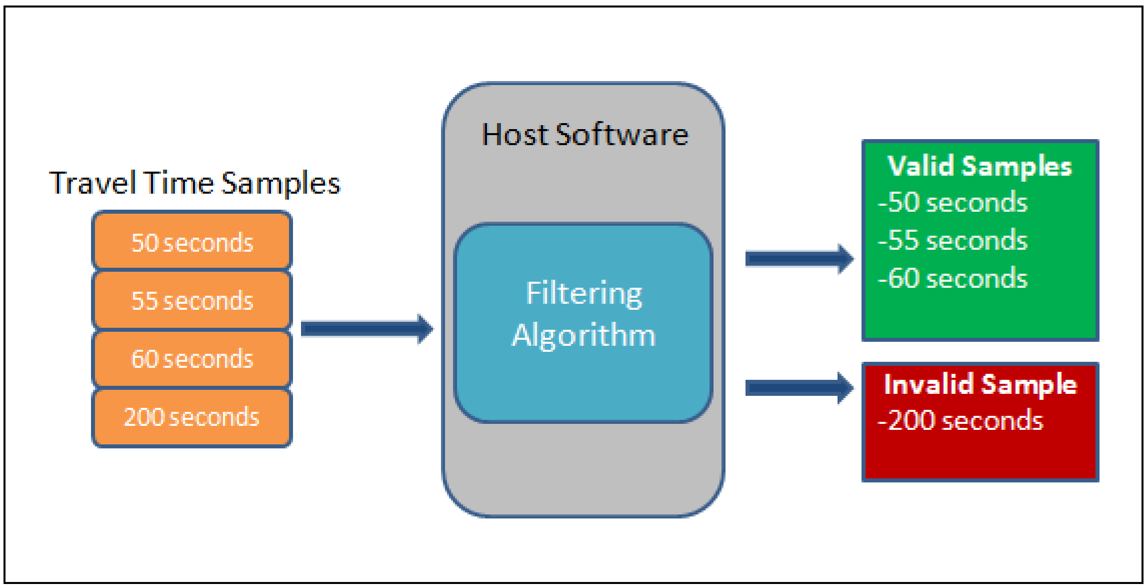


Figure 24. Travel Time Filtering Algorithm Generalized Process

While it was beyond the scope of this project to evaluate various filtering algorithms in detail, the original filtering algorithm compares each new travel time sample to the current average for the roadway segment. If the new travel time differs by more than a certain percentage (for example 25 percent), the travel time would be labeled "invalid" and thus not used in any future travel time estimations. While this technique proved successful on some roadway segments, particularly high volume freeways that do not have a lot of variance in speed, it tends to discard many potentially valid records on arterials where the travel speeds were more varied.

For example, the chart in Figure 25 shows all speed samples collected by the system in a single day, with each dot representing one matched MAC address between two points. The roadway segment shown is an arterial with traffic signals in between the detectors, which induces a wide variance in travel time as some vehicles are stopped and some vehicle progress through these signals unimpeded. The green dots are samples labeled “valid” by the system and the red dots are labeled “invalid” and thus not used in travel time and speed average determinations. The results on the chart were determined by utilizing a filtering algorithm that determines validity by using the 25 percent threshold described above. It is evident from the chart that many of the samples labeled “invalid” by the algorithm should have been marked “valid” as the red and green dots overlap in many cases. Based on these and similar observations, the filtering algorithm needs to be less sensitive, or more flexible, to changes in travel time and speed.

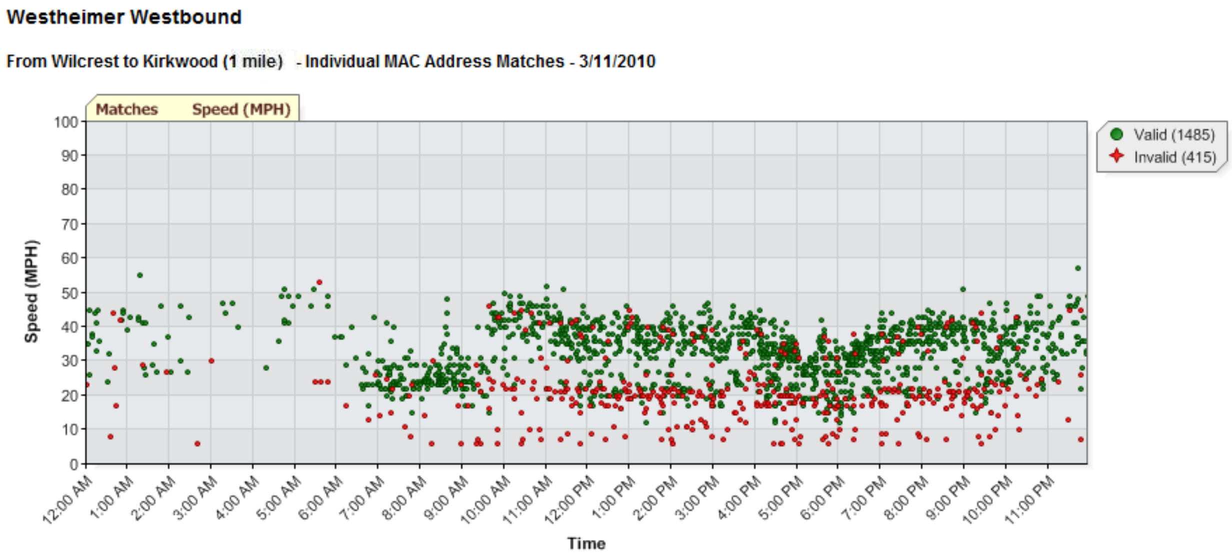


Figure 25. Valid and Invalid Travel Speed Matches Using a Basic Filtering Algorithm

Although this filtering technique worked on other types of roadways with reasonable effectiveness, particularly freeways, it was obviously filtering out valid data on this particular roadway segment. Slight changes to the parameters of the algorithms resulted in less filtering of “valid” samples, as shown in Figure 26.

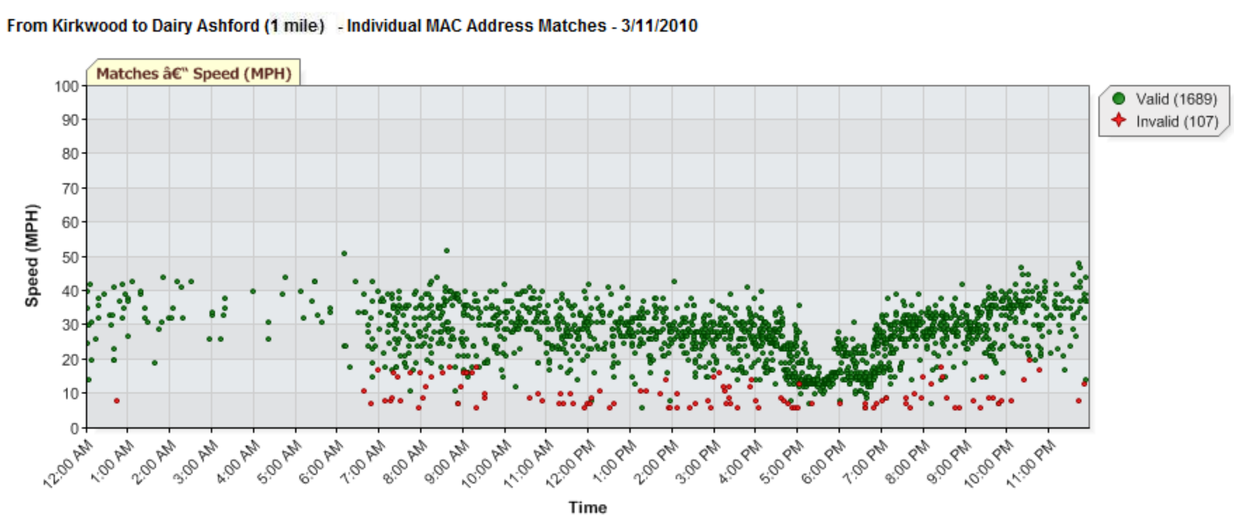


Figure 26. Valid and Invalid Travel Speed Matches with Slightly Modified Filtering Algorithm

When taking into account the varied geometric conditions and other characteristics of the roadway segments in the evaluation corridors, it quickly became evident that a “one size fits all” approach to a travel time filtering algorithm was not the most effective. Therefore, researchers decided that a better approach would be to add functionality to the host software that allowed users to specify the algorithm and parameters that would be most effective. In another example, researchers were concerned about the variance in valid speeds on the roadway segment expressed in the chart in Figure 27.

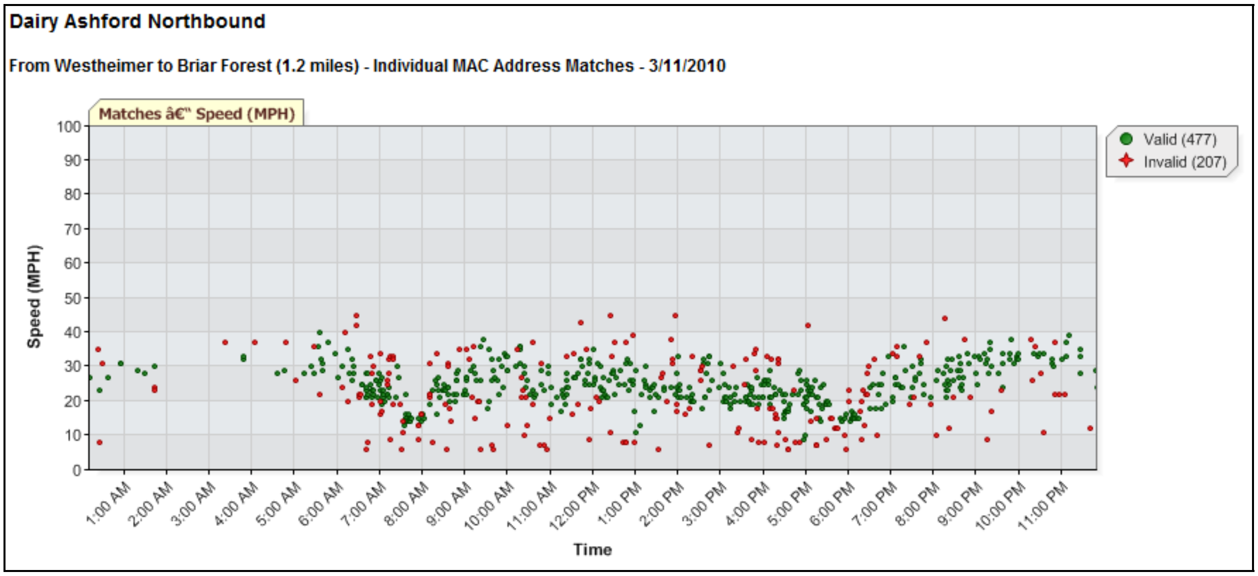


Figure 27. Valid and Invalid Speed Observations before Filtering Algorithm Change

By using another algorithm for this particular roadway segment, the variance in valid speeds was reduced as shown in the chart in Figure 28.

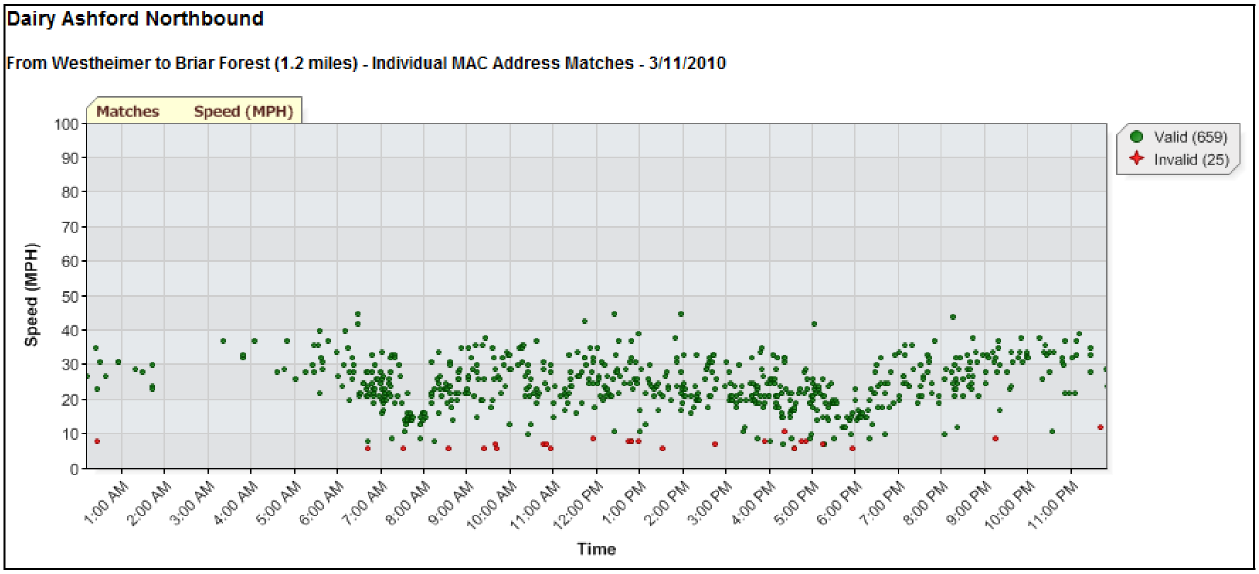


Figure 28. Valid and Invalid Speed Observations after Filtering Algorithm Change

At the time of this writing, there were various filtering algorithms available in the host software component that could be configured separately for each individual roadway segment in the traffic monitoring system. This gives users of the software the flexibility to define the algorithm that best suits the needs of the roadway being monitored and helps the system derive the highest quality data possible.

Automated Field Outage Notification

In order to provide as much uptime as possible for field detectors and to ensure that data interruptions are kept to a minimum, timely notification of detector outages is important. Even though outages were rare during the long-term evaluation of the new field equipment, the researchers thought it was important to provide the ability for the system automatically to notify system operators shortly after a site stopped reporting MAC address data to the host. To accomplish this, a method was implemented in the host software that maintains a list of the last time each detector communicates with the host. If the last communication time exceeds a preset threshold, the host software sends an alert, in the form of an email, to personnel specified in the system configuration for the detector.

This method proved useful on the few occasions where a detector stopped reporting due to a communications outage. The alerts allowed researchers to address promptly the field issue and led to the development of the automatic communications error recovery feature in the field software mentioned previously.

Data Interfaces and Output

The interfaces into a traffic data collection system are important for allowing other software applications to utilize and present the information generated by the system. Prior to the long-term evaluation of the system, the basic interfaces into the *Bluetooth* Traffic Data Collection System consisted of a very basic, limited interface that provided users with the raw data collected by the system. The interface consisted of the raw travel time data samples, or "matches," and the raw MAC address reads. This information was made available in ASCII-based comma delimited text files.

During the long-term evaluation, additional interfaces were added to provide external applications with more robust methods for using the data. First, a method was created in the software that maintains a list of the most current historical rolling travel time and speed averages for each configured roadway segment. The averages are updated every 30 seconds and use the last "x" interval worth of samples where "x" is a time window (for example 15 minutes). "X" is a variable that can be uniquely configured for each roadway segment. After each average is updated, it is output to an XML-based file that can be accessed by external applications by making a simple web request. Figure 29 shows a snapshot of the XML file as seen in a web browser.

```
<?xml version="1.0" encoding="utf-8" ?>
- <match_summary_data>
  - <match_summary>
      <system_id>Houston TranStar</system_id>
      <origin_id>Main_Pressler</origin_id>
      <dest_id>BraesmainBT</dest_id>
      <origin_roadway>Main</origin_roadway>
      <origin_cross_street>Pressler</origin_cross_street>
      <origin_direction>Southbound</origin_direction>
      <dest_roadway>Main</dest_roadway>
      <dest_cross_street>Braesmain</dest_cross_street>
      <dest_direction>Southbound</dest_direction>
      <segment_length_miles>1</segment_length_miles>
      <timestamp>6/11/2010 1:09:45 PM</timestamp>
      <travel_time>194</travel_time>
      <speed_mph>19</speed_mph>
      <summary_mins>15</summary_mins>
      <summary_samples>7</summary_samples>
      <map_display>True</map_display>
    </match_summary>
  - <match_summary>
      <system_id>Houston TranStar</system_id>
      <origin_id>BraesmainBT</origin_id>
      <dest_id>Main_BuffaloSpeedway</dest_id>
      <origin_roadway>Main</origin_roadway>
      <origin_cross_street>Braesmain</origin_cross_street>
      <origin_direction>Southbound</origin_direction>
      <dest_roadway>Main</dest_roadway>
      <dest_cross_street>Buffalo Speedway</dest_cross_street>
      <dest_direction>Southbound</dest_direction>
      <segment_length_miles>1</segment_length_miles>
      <timestamp>6/11/2010 1:09:45 PM</timestamp>
      <travel_time>148</travel_time>
      <speed_mph>24</speed_mph>
      <summary_mins>15</summary_mins>
      <summary_samples>19</summary_samples>
      <map_display>True</map_display>
    </match_summary>
  - <match_summary>
      <system_id>Houston TranStar</system_id>
      ...
    </match_summary>
  </match_summary_data>
```


Figure 29. Typical Travel Time Monitoring System XML File Output

The contents of the file represent the real-time status of the current travel time and speed averages estimated by the system. The file will likely be used by applications looking to present the traffic data in a real-time environment such as a traffic congestion map interface. In fact, the file is currently used as an input into the Houston TranStar mapping interface described previously.

In addition to the real-time snapshot of the traffic data, an output file containing historical summary calculations was also added as an interface. To produce the summaries, a method was added to the host software that calculates travel time and speed summaries for each "x" interval where "x" is an element of time that is user configurable (for example 15 minutes). The result of the summary is an output containing travel time and speed averages for each roadway segment for each "x" minute interval. The summaries are output to a comma delimited ASCII file. The data in Figure 30 show the four 15-minute summaries calculated for the 10:00 AM hour.

```
Westheimer_Wilcrest,Westheimer_Kirkwood,Westheimer,Wilcrest,Westbound,Westheimer,Kirkwood,Westbound,1,6/11/2010 10:00 AM,109,33,15,20
Westheimer_Wilcrest,Westheimer_Kirkwood,Westheimer,Wilcrest,Westbound,Westheimer,Kirkwood,Westbound,1,6/11/2010 10:15 AM,111,32,15,14
Westheimer_Kirkwood,Westheimer_Wilcrest,Westheimer,Kirkwood,Eastbound,Westheimer,Wilcrest,Eastbound,1,6/11/2010 10:30 AM,110,33,15,12
Westheimer_Wilcrest,Westheimer_Kirkwood,Westheimer,Wilcrest,Westbound,Westheimer,Kirkwood,Westbound,1,6/11/2010 10:45 AM,116,31,15,17
```

Figure 30. Typical Comma Delimited Travel Time Monitoring, 15-Minute Summaries

External software can use the summary interface to track the average speed and travel time of a roadway segment over time. The chart in Figure 31 uses the summary file to show the historical average speed on a roadway segment for part of the day.

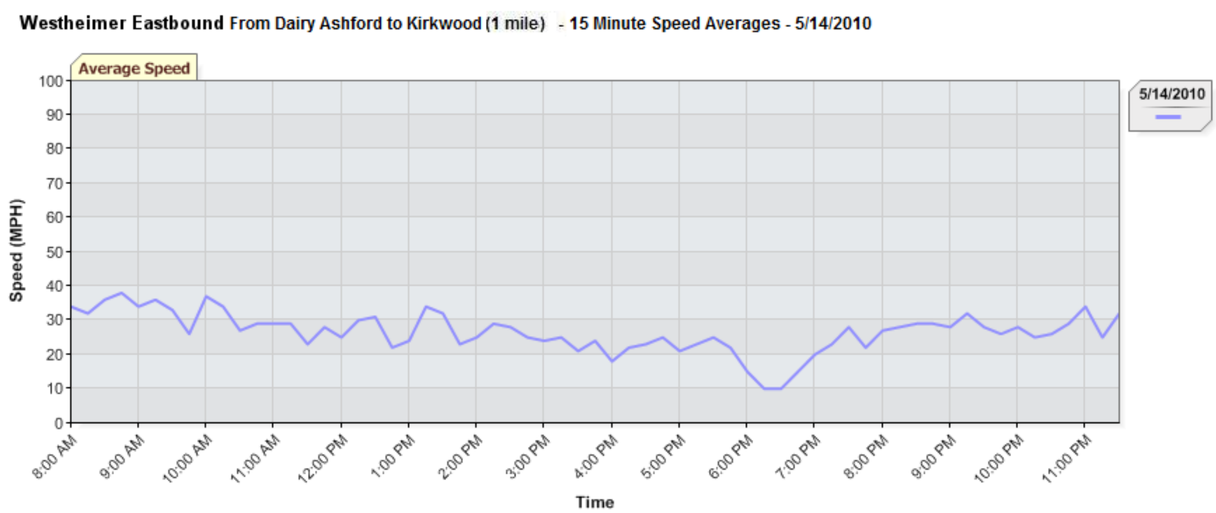


Figure 31. Average Travel Speed Summary, 15-Minute Intervals

Figure 32 shows a diagram showing the current interfaces into the system with existing and potential uses.

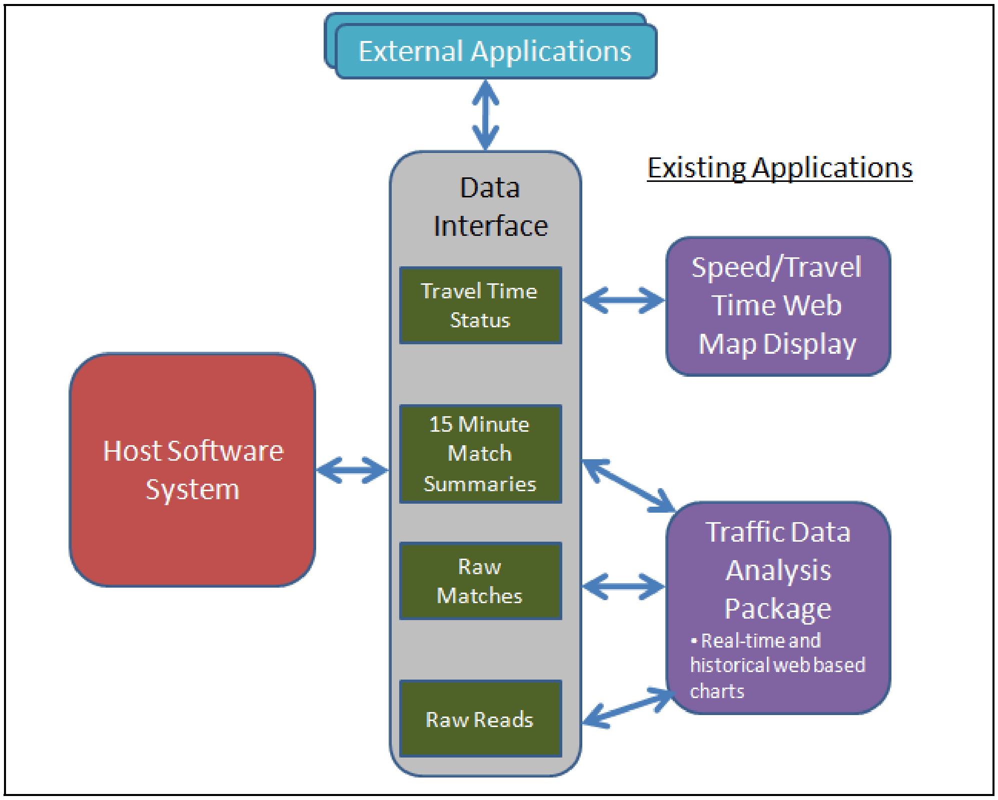


Figure 32. Travel Time Monitoring System Host Software Interfaces

Recommendations Regarding Host Software Capability

In most cases, probe-based travel time monitoring systems will be responsible for reporting performance on various types of roadways, many of which will have different geometrics and operating characteristics. The travel time filtering algorithm that applies to one roadway segment may not necessarily be the most applicable for another roadway (or segment on the same roadway). For instance, an urban freeway segment with high volume may require a different algorithm than an urban arterial segment with traffic signals in between detectors. It is the authors' recommendation that any probe-based travel time system allow for flexibility in the algorithm used for individual roadway segments used across a system.

As the host software is able to monitor the status of all detectors configured to report to it, methods should be implemented for automated notification of detector outages to enable operators to address field issues in a timely manner.

Finally, in order to make use of the data output by the host software, rich interfaces to the data should be developed such that other applications can utilize these data, and archival capability provided to access historical data.

Findings and Recommendations

This section summarizes the findings and recommendations resulting from research conducted as part of this UTCM project. While there are identified topics of further interest that future research will likely address, this project was able to identify some of the technological and process issues that should be considered when developing and deploying basic *Bluetooth*-based travel time monitoring systems.

Findings

The conduct of this series of tests and evaluations has conclusively determined that the use of the *Bluetooth* protocols and process for collecting travel times on arterial roadways can be pursued programmatically by public sector agencies and/or private sector entities that need or desire travel time information. It has been shown that deployment of low-cost, off-the-shelf devices, combined with unique (but not necessarily complex) field and host software elements, result in a very effective means of collecting real-time travel time and speed measures for a variety of operational, informational, and planning purposes. Compared to other alternatives to provide probe-based travel time and speed output, the TTI-developed test methodology is several orders of magnitude less expensive.

Recommendations

The following high-level recommendations may be considered by those considering deployment or further testing of *Bluetooth*-based travel time monitoring systems.

Hardware Requirements for Permanent Deployment

Initial *Bluetooth* reader feasibility efforts undertaken in advance of this research project utilized simple, inexpensive, consumer-grade notebook computers as the roadside CPU. These devices were found to be impractical for deployments longer than a few hours. However, as part of this research and implementation effort, the following criteria for the selection of permanent field processing units for deployment and/or further testing for a real-time provision of travel time and travel speed data are as follows:

1) Field hardened device capable of withstanding the extreme temperature variations typically experienced in a traffic cabinet. At a minimum, the device should be capable of withstanding temperatures ranging from -4°F to 140°F, and these ranges could be modified based on the environmental conditions expected in particular locations.
2) The device must contain:
    a. an Ethernet port to utilize the most common types of communication devices,
    b. a USB port to utilize the most common types of external *Bluetooth* adapters available, and

   c. an antenna connector on the *Bluetooth* adapter to accommodate various types of replaceable antennas.

3) The device must be fairly compact in size to fit inside an existing traffic cabinet without interfering with, disrupting, and/or displacing existing equipment.
4) The device must be at a price-point where cost is not a deterrent to wide scale deployment, enabling owning agencies to realize significant cost advantages over existing traffic data collection technologies such as AVI or LPR.
5) The device must be capable of running a lightweight operating system to run the field software, similar to various embedded versions of Linux.
6) For maintenance and configuration purposes, the device must have remote access capabilities using standard applications and protocols such as Secure Shell and File Transfer Protocol as part of the operating system. Most Linux distributions provide these capabilities.
7) For solar-powered driven environments, the devices should draw minimum power and battery power supplies should be adequate to supply 14 days of power without recharging.
8) For most geographic situations, the standard 1dB gain omni-directional antenna, when coupled with a Class 1 adapter, will adequately capture MAC addresses within 300 ft (or about 100 m). The coupling of a Class 1 adapter and 1db antenna is recommended for most traffic applications except for unique situations that might require longer range, or those with a need to limit range or coverage.

Hardware Requirements for Portable Deployment

The portable format *Bluetooth* travel time monitoring prototypes performed as expected and can be deployed in circumstances where power is not available. However, providing a large enough deep-cycle battery to supply power for at least one week results in a portable unit weighing in excess of 100 pounds.

Based on this research project, portable deployments could consist of readily available field processor units, with cellular communications and *Bluetooth* adapters similar to those found suitable for permanent deployments. Hardware requirements are similar to those listed above for permanent deployments.

Field Software Characteristics

To utilize the preferred method to ensure that the maximum amount of MAC addresses are being read in a timely manner, the *Bluetooth* software interface to the stack API should optimally be interfaced such that asynchronous device inquiry is implemented. By implementing asynchronous inquiry, the TTI-developed field software method increased the total number of travel time samples by over 50 percent.

In addition, to operate the communications aspect and host software processes optimally, duplicate MAC address readings from the same location should be eliminated to prevent duplicate travel time samples from skewing the overall averages and reduce the overall bandwidth required by the field system.

Although a simple field to host data protocol was established as part of the project, it is recommended that further initiatives be taken to establish a formal standard for packaging *Bluetooth* MAC address data for traffic information. This would allow interoperability among vendors and users of different systems.

Any probe-based data collection system should be sensitive to privacy concerns expressed by the public. Even though MAC addresses are a relatively benign piece of information, methods should still be taken to ensure that probe information be as anonymous as possible. The wide array of software tools for anonymizing and encrypting data make this relatively easy to accomplish.

Any agency responsible for deploying field equipment on a wide scale will likely be concerned about long-term reliability and maintenance of the equipment. It is both costly (in terms of data loss and work hours) and time consuming to make field site visits to troubleshoot equipment. In order to reduce the number of site visits required, methods should be implemented on the field device that attempt to automatically recover from errors that might otherwise require a technician to physically address at the roadside.

Host Software Characteristics

In the TTI-developed *Bluetooth* Traffic Monitoring System, a central software component is utilized for accepting MAC address data in real-time from each roadside system and then estimating average travel times between configured roadway segments based on the location of the *Bluetooth* reader systems at the roadside. The basic functions of the central or "host" software components are:

1) accept MAC address data from each of the roadside detectors via the UDP network protocol,
2) apply the proper algorithms for estimating the travel times between configured roadway links as defined by the detector locations,
3) apply filtering algorithms to travel time data samples and averages so that "outlier" data are not included in travel time estimations, and
4) create various output interfaces so that external applications can utilize the travel time and speed data generated by this software component.

In addition to basic functionality of the host software system, there were some notable enhancements that were completed as a result of the long-term evaluation of the new field equipment as part of this research project. The following recommendations result from these additional enhancements:

1) One of the most important features of any probe-based traffic monitoring system is the ability to filter outlier data from the travel time calculations in order to determine the approximate average travel time over a defined period of time. Outlier filtering is necessary to identify the true average of the travel time data for a roadway segment and eliminate those matches representing longer (or shorter) than expected travel times.
2) The varied geometric conditions and other characteristics of various roadway segments require that several filtering algorithms be available for use in the host software. The various filtering algorithms available in the host software component should have the capability to be configured separately for each individual roadway segment in the traffic monitoring system. This gives users of the software the flexibility to define the algorithm that best suits the needs of the roadway being monitored and helps the system derive the highest quality data possible.
3) In order to provide as much uptime as possible for field detectors and to ensure that data interruptions are kept to a minimum, timely notification of detector outages is critical. To accomplish this, a method should be implemented in the host software that maintains a list of the last time each detector communicates with the host. The host software should send an alert to personnel specified in the system configuration for the detector if a fault is determined.
4) Finally, in order to make use of the data output by the host software, rich interfaces should be developed such that other applications can utilize that data and archival capability provided to access historical data.

References

(1) Darryl Puckett and Mike Vickich. *Wireless Address Matching for Travel Time Estimation*. City of Houston, March 2009.

(2) William H. Schneider IV, Shawn Turner, Jennifer Roth, and John Wikander. *Statistical Validation of Speeds and Travel Times Provided by a Data Service Vendor*. Ohio Department of Transportation, Office of Research and Development, January 2010.

Appendix:
Equipment Specification Sheets

Bluetooth USB adapter,

Parani™ UD100

The Parani-UD100 is a class 1 type Bluetooth USB adapter that supports 300 meters of wireless transmission distance by default. The working distance can be further extended up to 1000 meters using optional replacement antenna. Thanks to its longer communication distance than other regular Bluetooth USB adapters, it is suitable for industrial or special applications. Parai-UD100 is compatible with other SENA Bluetooth devices perfectly.

Features

- Bluetooth 2.0+EDR Class 1
- Working distance 300m, up to 1000m using optional antenna
- USB 2.0 interface
- Bluetooth driver CD (Toshiba driver)

Specifications

Standards
- Bluetooth 2.0+EDR Class 1
- USB 2.0

Max Transfer Rate : 3 Mbps (EDR)

Frequency Range : 2.402 ~ 2.480GHz

Transmit Output Power : +19dBm (+6dBm EDR) E.I.R.P

Receive Sensitivity
- Basic 1Mbps : -88 dBm
- EDR 2Mbps : -87dBm
- EDR 3Mbps : -82dBm

Antenna Connector : RP-SMA

Antenna Gain
- Default Stub Antenna : 1 dBi
- Optional Dipole Antennas : 3 dBi & 5 dBi
- Optional Patch Antenna : 9 dBi

Working Distance(In Open Field)
- Stub antenna – Stub antenna: 300 m
- Dipole (3 dBi) – Dipole (3 dBi) : 400 m
- Dipole (5 dBi) – Dipole (5 dBi): 600 m
- Patch antenna – Patch antenna: 1 km

* *working distance can vary depending on install environment*

Bluetooth stack software : Toshiba

Bluetooth Profiles : DUN, FAX, LAP, SPP, HID, HCRP, FTP, OPP, A2DP, AVRCP, GAVDP, HSP, HFP,PAN, BIP

Computer OS Support
- Windows 2000/XP/Vista (32/64bit)
- Linux (3rd party driver required)
- MAC OS X (MAC OS X driver required)

Size : 72(L) x 22(W) x 10(H) mm

Operating Temperature : -20 ~ +70℃

Storage Temperature : -40 ~ +85°C

Humidity : 90% Non-condensing

Regulatory Approvals : FCC, CE, TELEC, KCC, Bluetooth SIG

Warranty : 1 year limited warranty

Exterior

Ordering Information

UD100-G01 Includes
- Parani-UD100
- Stub antenna
- Quick Start Guide
- CD-ROM including Windows Bluetooth driver, User Guide and Datasheet

Optional Accessories
- SAT-G01R : 1 dBi Stub Antenna - RP-SMA Plug Right-hand Thread
- DAT-G01R : 3 dBi Dipole Antenna - RP-SMA Plug Right-hand Thread
- DAT5-G01R : 5 dBi Dipole Antenna - RP-SMA Plug Right-hand Thread
- PAT-G01R : 9dBi Patch Antenna with RP-SMA Plug Right hand Thread Default Cable. (SMA Jack Right hand Thread w/o Default Cable)
- SEC-G01R : 15cm Extension Cable (For SAT-G01R / DAT-G01R / DAT5-G01R)
- RFC-G01R : 1m Extension Cable (For PAT-G01R)

For more information,
please visit us at http://www.sena.com

U.S. / Canada / South America
Tel: 408 448 1097 Fax: 408 907 3738
Sales and Support Toll Free (US/Canada): 8888 US SENA (888 087 7362)
General Information Toll Free (US/Canada): 8888 US SENA (888 087 7362)

EU / Africa
Tel: +33 (0)1 53 53 16 29 Fax: +33 (0)1 70 24 70 90

Asia / Pacific
Tel: +82 2 571 8283 Fax: +82 2 573 7710

Contact us via E-mail
General Information: info@sena.com
Sales & Distribution: sales@sena.com
Technical Support: support@sena.com

SENA
www.sena.com

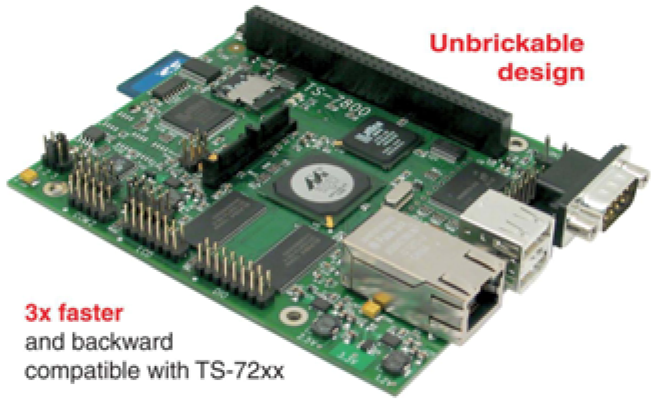
OVERVIEW

The **TS-7800** is a **RoHS** compliant Single Board Computer (**SBC**) based on a Marvell **500MHz ARM9** CPU with internal **PCI** bus and that provides a standard set of on-board peripherals such as **Gigabit** Ethernet, dual SATA and dual High-Speed host/slave USB 2.0.

The TS-7800 also features a 12,000 LUT on-board Lattice **FPGA** and provides extra peripherals such as **110 GPIO** lines and additional serial ports.

On the software side, the TS-7800 uses an in-house improved **Linux 2.6** Kernel that allows **0.69 second** bootup and provides driver support for all on-board hardware. In addition, the **512 MB on-board Flash** enables a full **Debian** distribution to be installed with a complete embedded development environment.

The TS-7800 is backward **compatible** with our TS-72xx computers, providing **3 times** more performance and higher-end features with identical footprint, thus allowing quick platform migration for customer applications.



FEATURES

- **500Mhz** ARM9 CPU
- Internal **PCI** bus, PC/104 connector
- 12,000 LUT programmable **FPGA**
- 128MB DDR-RAM
- **512MB** NAND Flash, high-speed (17MB/s)
- 2 SD socket (1 micro-SD, 1 full-size SD)
- 2 SATA ports
- 2 USB 2.0 **480Mbps** host/slave ports
- **Gigabit** Ethernet, 10/100/1000 speeds
- 5 10-bit ADC channels
- 10 serial ports, 2 optional RS-485
- **110 GPIO** (86 arranged as a PC/104 bus)
- Sealed-battery backed RTC
- Matrix **Keypad** and Alphanumeric **LCD** interfaces
- **Fanless**: -20° to +70°C
- Optional on-board Temperature Sensor
- Low power **4W@5V**
- Sleep mode uses **200 microamps**
- Optional **8-30V input** voltage range (default is 5V)
- Boots to a Linux shell-prompt in **0.69 second**
- runs **Kernel 2.6** and Debian Linux by default

FAST BOOTUP FIRMWARE

The TS-7800 bootstrap uses a unique and clever combination of FPGA hardware logic, specific boot-up firmware and Kernel tweaks which ensure fast boot time, security, high board recoverability and more:

- **Linux-based bootloader** boots Linux 2.6 kernel to shell-prompt in **less than 1 second** after power-on from either SD card or on-board Flash
- Full Debian can be installed into on-board Flash from a USB flash dongle - no need for Busybox
- **Unbrickable** design ensures 100% recoverability from SD card in case of on-board Flash erasure

12,000 LUT FPGA

- Connects to CPU via 50Mhz local **PCI** bus
- Default load uses GPIO pins as a PC/104 bus
- Enables fast board modification or feature improvement via FPGA load customization

LINUX 2.6 AND DEBIAN

The TS-7800 is shipped with Linux Kernel 2.6 and the Debian distribution on on-board **Flash**, enabling a wide range of server services, desktop-like applications and developments tools to run on a **embedded real-time** system.

Technologic SYSTEMS

16610 East Laser Drive #10
Fountain Hills, AZ 85268
TEL 1.480.837.5200
FAX 1.480.837.5300

www.embeddedARM.com
support@embeddedARM.com

University Transportation Center for Mobility™

Texas Transportation Institute

The Texas A&M University System

College Station, TX 77843-3135

Tel: 979.845.2538 Fax: 979.845.9761

utcm.tamu.edu