TDRL Projects: Solar/Wind Hybrid Renewable Light Pole, Gravel-Road Traffic Counter, DLL-Based Traffic Software Development Kit

Final Report

Prepared By:

Taek Mu Kwon, Ph.D. (Principal Investigator)
Ryan Weidemann (Research Assistant)
Dan Cinnamon (Research Assistant)

Transportation Data Research Laboratory
Northland Advanced Transportation Systems Research Laboratories
Department of Electrical and Computer Engineering
University of Minnesota Duluth

July 2008

Published By
Center for Transportation Studies
University of Minnesota, Twin Cities

Acknowledgements

This research was supported by the Northland Advanced Transportation Research Laboratories (NATSRL). During the development of solar/wind hybrid renewable light pole, Ryan Weidemann (research assistant) wrote the data logging program that collects power consumption and generation data. He also regularly downloaded the data from the site and provided analysis and graphs. Another student, Dan Cinnamon, wrote the initial version of the Dynamically Linked Library (DLL) for the TDRL Software Development Kit. This version was later refined by Prof. Taek Kwon. The software for the gravel road traffic counter was entirely written by Prof. Taek Kwon.

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

This report describes the results of three projects initiated in Fiscal Year 2006 as a part of the Transportation Data Research Laboratory (TDRL) research projects. The three projects are: (1) Development and field tests of solar/wind hybrid renewable light pole, (2) Development of a gravel road traffic counter, and (3) Development of a DLL-based traffic software development kit (SDK). Each of them is summarized.

Development and field tests of solar/wind hybrid renewable light pole. A prototype of solar/wind hybrid power generator, which consists of a 130W solar panel, a 400W small wind turbine, and six 6V 240Ah batteries, was constructed and field tested for two years to study the benefits of integrating solar and wind power generation as a single unit for supporting power to rural Intelligent Transportation Systems (ITS) applications. The system was entirely built using off-the-shelf components. For the application, a street light was powered. According to the power generation/consumption data collected over two years, there clearly exist many days in which solar energy alone was not sufficient to support the lighting application and the wind provided supplementary energy. Analysis of hourly data revealed that many rainy and snowy days had stronger winds, providing complementary energy resources when solar radiation is deficient. From this field test, it was concluded that solar/wind hybrid generation of electricity provides more reliable power supply for rural ITS applications.

Development of a gravel road traffic counter. In rural gravel roads, tube counters experience frequent air leaks caused by punctures and also reduced sensitivity problems by dirt accumulation around the tube. These problems often lead to inaccurate counting of vehicles. The purpose of this project was to find an alternative technology that works without the mentioned disadvantages of tube counters. The sensor proposed was a coil-wound fluxgate. A prototype counter was constructed using a fluxgate, a microcontroller, and control circuits. The developed system was tested on a County gravel road, as well as from a parking lot under a controlled environment. The test results showed that the proposed system is easy to install and works reliably. Since this technology is based on detection of magnetic fields, its counting ability was similar to that of traditional inductive loop detectors. One difference from the loop detectors is that it is not affected by stopped vehicles near the sensor. If a stopped vehicle is present near the sensor and another vehicle passes by, the stopped vehicle was ignored and only the vehicle that passed by was counted. This technology is presently not available on the market, but this research shows that it is an elegant solution for gravel road counting.

Development of a DLL-based software development kit. The TDRL Data Center runs a data warehouse in which it acquires traffic and R/WIS data from Mn/DOT, archives them, and distributes them to public through Internet. This data has been used by transportation research institutes, university researchers, and Mn/DOT. Researchers from many universities and consultants have requested software development tools for extracting traffic data from the archived traffic data. This project was created in response to those requests and to provide a software tool for developing applications. An SDK package was successfully developed as a Dynamically Linked Library (DLL) that allows

direct embedding into applications, and distributed through Internet. The main target of SDK is software developers and advanced users who develop application software.

Each project described above produced tangible outcomes, in which two of them are in the form of prototype and one of them in a software development package. Some of them took more than two years to complete. At the end, the specific goals set for each project were met, providing plausible solutions for the given problem.

CHAPTER 1: INTRODUCTION

This report describes the results of three projects initiated in Fiscal Year 2006 by the Transportation Data Research Laboratory (TDRL), using the funds provided by the Northland Advanced Transportation Systems Research Laboratories (NATSRL). They are (1) Development and field tests of solar/wind hybrid renewable light pole, (2) Development of a gravel road traffic counter, and (3) Development of a DLL-based software development package. Each project is relatively small and provided as a seed project for the TDRL. Each of them is described as a chapter that includes its own introduction, the main body of work, and its own conclusion. This introduction briefly describes each project.

Development and field tests of solar/wind hybrid renewable light pole. When ITS applications are implemented in remote rural areas, bringing in grid-tied power to the location can be very expensive and becomes an issue. In Minnesota, it costs about $3 per foot in 2007, which translates to $15,840 per mile. Such a high cost is too expensive and unjustifiable for most rural ITS applications. Solar panels (i.e., photovoltaic panels) have been tried for power generation for rural Intelligent Transportation Systems (ITS) applications. However, it often failed during the winter months, mainly due to a short daylight time and deficiency of solar radiation. The solution this project sought was a hybrid generation of power that combines wind turbines and solar panels in to a single system. To test this approach, a prototype generator that integrates a solar panel with a small wind turbine was developed and installed in the Minnesota Department of Transportation (Mn/DOT) District-1 parking lot. As an application, a simple lighting system was implemented, and then the power generation and consumption data were collected for two years. The main hypothesis tested on this project was that wind energy complements solar energy and combining them into a single unified system consequently produces a more reliable power source. Chapter 2 describes the design, implementation, data analysis, and a cost analysis of the solar/wind hybrid system tested.

Development of a gravel road traffic counter. Gravel road traffic counting problem was suggested by the Office of Transportation Data and Analysis (TDA) at Mn/DOT. The TDA office has been responsible for collecting statewide traffic counting data and reporting them to the Federal Highway Administration (FHWA) and other state agencies [6]. The TDA traffic counting data is typically collected over 48 hours, which is then seasonally adjusted to produce Annual Average Daily Traffic (AADT). The main technology used for this data collection has been pneumatic tube counters. However, in rural gravel roads it was found that tubes can be easily punctured or damaged by sharp edges of gravel. Also, noticed was that the dirt accumulated around the tube decreases the sensitivity of tube, causing miscounts. The TDA office has long sought alternative portable counting technologies and products to replace the old tube counters, but none of the existing technologies were suitable for portable gravel road counting applications. A solution proposed by this project is a fluxgate counter which works by detecting disturbance of earth magnetic fields by vehicles. Presently, no traffic counting devices that utilize fluxgate sensors are available in the market. Therefore, TDRL designed and built a prototype fluxgate counter that can be tested on gravel roads. This prototype presents several interesting properties. The sensor size is small (1.25 inch diameter PVC pipe with 15 inch length). Installation is simple and easy. It is simply installed by burying the sensor on the shoulder of the road. Also, it does make any direct contacts with vehicles, and thus the sensor is not likely damaged by vehicle passages. It counts only the objects that contain ferrous material, which would eliminate counting mistakes caused by animal or people crossing. Last but not least, this sensor counts the true number of vehicles, unlike axle counts in tube counters. The details of fluxgate operational principle and the prototype development are described in Chapter 3.

Development of DLL software development kit. The TDRL Data Center regularly acquires traffic and Road/Weather Information System (R/WIS) data from Mn/DOT, archives them, and distributes the archived data to public through Internet. This data has been used by transportation research institutes, university researchers, Mn/DOT, etc. Many researchers and consultants have been requesting availability of software development tools for extracting traffic data from the downloadable archived data. This project was created in response to these requests. Therefore, the objective of this project was to create a software development tool for developing applications using the archived traffic data. The research team successfully created a Dynamically Linked Library (DLL) that can be directly embedded into user's applications. Several examples on how to use the DLL along with a manual were packaged into a single downloadable and installable code. This package is called a TDRL Software Development Kit (SDK) and described in Chapter 4.

In addition to the three projects described above, TDRL performed a preliminary investigation on piezoelectric wires and anisotropic magnetoresistive (AMR) sensors. In the subsequent year, an intersection turning movement counting system based on a mesh network of AMR sensors was developed as a full independent project, and a separate report will be available for this project. This project is still in progress at the time of this writing. The piezoelectric wire technology was evolved into a Weigh-in-Motion (WIM) project.

CHAPTER 2: DEVELOPMENT AND FIELD TESTS OF SOLAR/WIND HYBRID RENEWABLE LIGHT POLE

2.1 Chapter Introduction

With rising oil prices and increased dependencies on foreign oil, U.S. energy policies have emphasized increased use of renewable energy, especially in the area of solar, wind, and hydrogen resources. This report describes a street lighting application developed utilizing a hybrid power-generation technology that combines solar and wind energy into a single, unified power generation system.

In most Midwest states in the U.S., solar panels alone are not sufficient to supply the power needs of many ITS applications during the winter months, due mainly to significantly shorter daylight time and weaker solar radiation. This poses a difficulty in using renewable energy for critical ITS applications. The proposed hybrid technology solves this problem by integrating photovoltaic (PV) power generation with wind power generation. According to seasonal wind resource maps [1] of the U.S., winds are stronger in winter in Midwest states, which suggest that the shortage of electricity caused by a lack of solar energy could be readily supplemented by wind energy. Another complementing weather factor is that winter and/or summer storms generally produce stronger winds, while they reduce solar radiation. Again, this is another complementing relationship between solar and wind energy. Therefore, solar and wind resources complement each other and utilizing them into a single electricity generation system could result in producing more constant rate of electricity, providing a higher power reliability under various weather conditions

Present research identifies that rural ITS applications are one of the sectors that can directly benefit from the proposed hybrid power stations. Recent advances in sensing, communication, and computing technologies have produced various ITS devices that can improve traffic safety and management in rural areas. The types of ITS technologies include traffic-activated advisory signs, advance warning signals/flashers for hazardous locations, variable message signs, animal crossing detection, road-weather information systems (RWIS), etc. However, lack of easily accessible power sources and high cost of brining electric utility to remote rural areas often make the deployment economically not justifiable to many transportation agencies. In Minnesota, the cost of bringing a utility power line to a rural intersection is approximately $3 per foot in 2007, which translates to $15,840 per mile. For a location 20 miles away from a power distribution center, bringing power would cost $316,800. Considering additional monthly recurring fees, theses are too expensive for most of rural ITS applications. On the other hand, the cost of solar/wind generated power is independent of the distance to the location and could be achieved with less than $4,000 of initial investment for most of rural ITS applications. Moreover, no recurring monthly electric bills are required.

The lighting application presented in this report has an important safety benefit in a transportation system in addition to the cost savings mentioned. Rural intersections typically do not have signal lights and have caused more fatalities than intersections in metropolitan areas, according to the GAO Highway Safety report [2]. Despite high fatality rates in rural intersections, adding a signal system is often not justifiable, because they do not meet the minimum requirement of traffic volume specified in state intersection warrants [4]. A common solution to rural intersections thus has been installing a street light to improve night visibility. Adding light is considered a good alternative solution since the most frequent contributing factor to traffic fatalities in rural roads is a visibility problem in late-night driving [3]. On the other hand, the cost of bringing electricity to the location and the associated monthly fee is expensive as described above. With the justifications in place for not putting lights or traffic signals in rural intersections, the death rate on rural roads has been increasing. Death rates on metropolitan roads and intersections have been in decline [2], with the help of modern ITS technologies. Recognizing this problem, development of a low-cost renewable-energy street-light pole was proposed. The research team believes that this is a plausible low-cost lighting solution in rural roads and intersections, providing better visibility to motorists.

2.2 Design of Solar/Wind Hybrid Renewable Street Light

2.2.1 System Design Goals

The design objective of the solar/wind hybrid renewable street-light system is to develop a self-sufficient street lighting system that generates and stores electric power whenever solar radiation or winds are available, and then provides lighting during the nighttime. The designed lighting system should provide light intensity equivalent to 150W incandescent light bulb. The cabinet for the battery storage and controller circuits should be small enough so that it can be readily installable in most locations.

2.2.2 Design Preparation

Availability of solar and wind energy in terms of potential convertible energy is highly dependent on location, and studying the expected annual availability is extremely important and should be the first step before the actual system design. For the solar and wind energy, the National Renewable Energy Laboratory (NREL), which has a web site at http://www.nrel.gov/, provides an excellent resource.

For the solar radiation energy, NREL provides monthly breakdowns and annual averages which should be used as an expected availability of solar energy source. From the NREL home page, searching for a key word "solar radiation" leads to the links for the maps. The map in Figure 1 shows 2004 annual solar radiation for PV panels facing south (most recent data available). According to this map, Duluth, Minnesota provides 4.0-4.5 KWh/m^2/day. In the monthly map (not shown), July had 5.5-6.0 KWh/m^2/day, and January had 2.5-3.0 KWh/m^2/day. Present PV technologies provide about 15-20% conversion efficiency. If we take 4.25 KWh/m^2/day as the average available solar energy in Duluth, Minnesota, the actual useable electricity is estimated at 0.64-0.85 KWh/m^2/day.

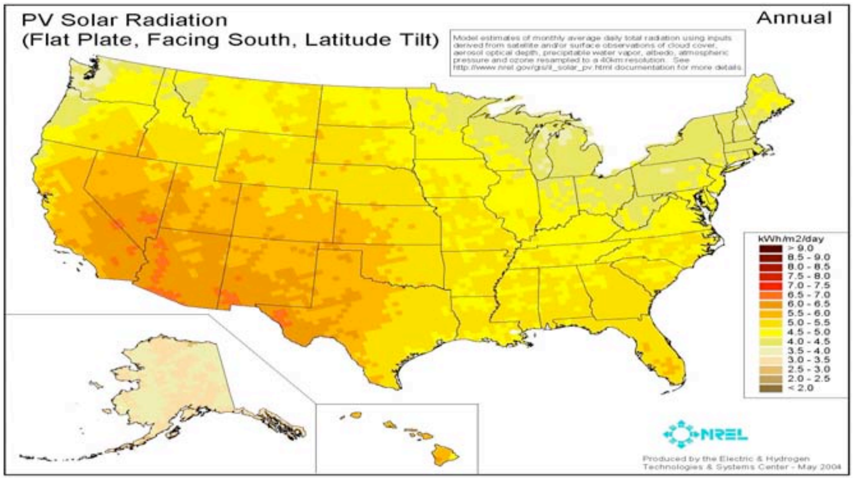


Figure 1: National average solar radiation energy (from NREL website).

Similarly, available wind energy for the given location can be found by searching a keyword "wind map" from the NREL home page. For the assessment of wind energy, wind power density (W/m2) is classified into seven classes. These classes are summarized in Table 1.

Table 1: Classes of Wind Power Density at 10 m and 50 m

Wind Power Class	10 m (33 ft)		50 m (164 ft)	
	Wind Power Density (W/m 2)	Speed (b) m/s (mph)	Wind Power Density (W/m 2)	Speed (b) m/s (mph)
1	0	0	0	
2	100	4.4 (9.8)	200	5.6 (12.5)
3	150	5.1 (11.5)	300	6.4 (14.3)
4	200	5.6 (12.5)	400	7.0 (15.7)
5	250	6.0 (13.4)	500	7.5 (16.8)
6	300	6.4 (14.3)	600	8.0 (17.9)
	400	7.0 (15.7)	800	8.8 (19.7)
7	1000	9.4 (21.1)	2000	11.9 (26.6)

Wind maps are available in annual and seasonal based average wind resources per respective regions. Figure 2 shows the maps for winter and summer, respectively. The wind maps for spring and autumn are similar to winter maps. From the map, it can be clearly seen that winter has higher wind density. For Duluth, Minnesota region, wind classes are recorded as Class-1 during the summer and Class-2 during the rest of the seasons. The identified wind class is later used in the system design to estimate the amount of energy that could be generated by wind.

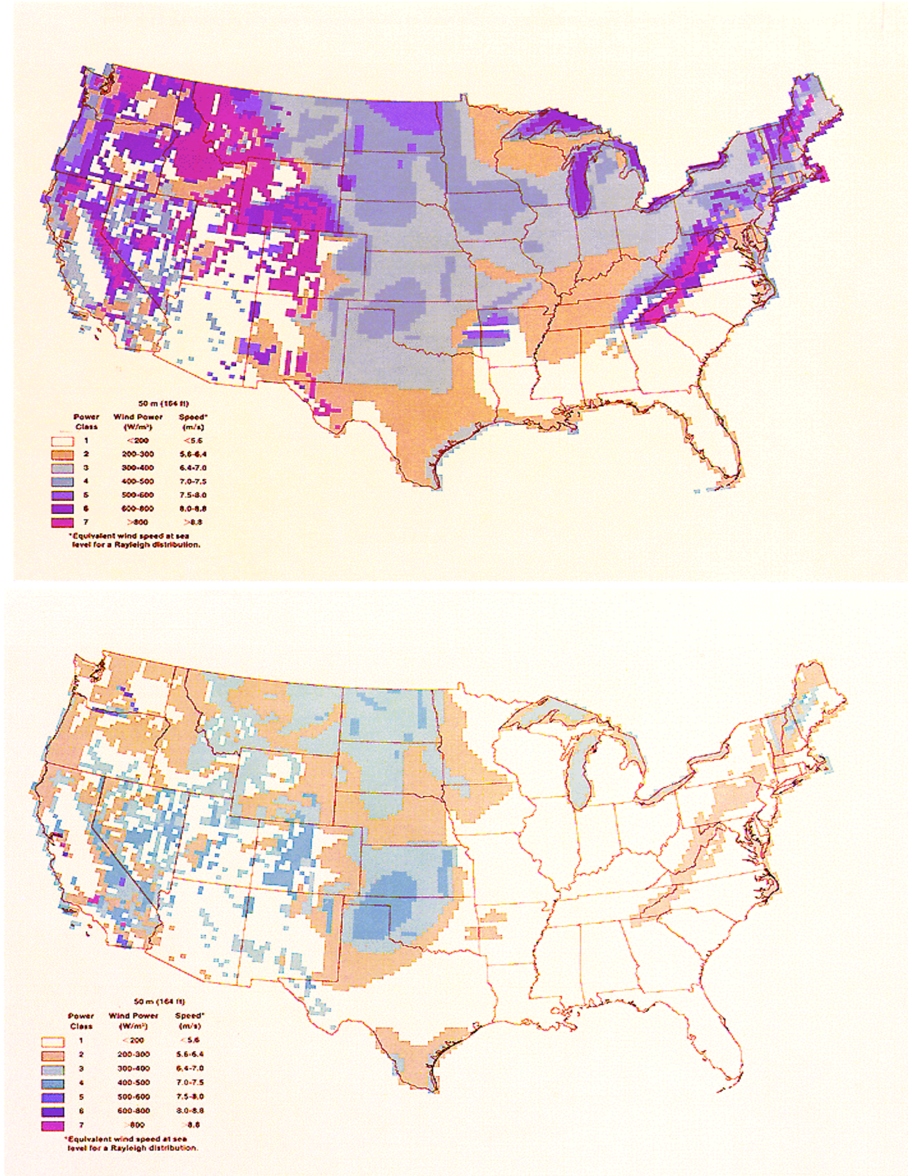


Figure 2: : Winter wind map (above), summer wind map (below).

2.2.3 System Design

The design of a renewable light pole must carefully consider the available wind and solar resources, the size of the cabinet, the choice of lighting system, the choice of PV panel and wind generator, and the overall cost. The design should attempt to strike a balance between the cost and performance. This section presents the overall design components and process.

2.2.3.1 Selection of Lighting System

In the design objective, the lighting requirement was to provide the lighting equivalent to an intensity of a 150W incandescent light bulb. Among many types of light bulbs, compact florescent light (CFL) lamps are known to have about four times more conversion efficiency than the equivalent wattage of an incandescent lamp. Therefore, 50W CFL should provide about 200W equivalent incandescent light, which exceeds the design requirement. It should be also noted that CFL lamps last about 10 times longer. Considering the efficiency, the model PL-50 produced by Philips Co. was selected for the light. PL-50 (or PL-L 50W/841 2G11) is a double-tube 50W fluorescent light that was designed for outdoor applications and works well under cold temperatures. The color temperature is specified at 4100K, which is equivalent to a white light (sun light). The initial lumens are rated at 4,000lm, and the average life is rated at 10,000 hours. The ballast for this light consumes about 5W. Figure 3 shows the Philips PL-50 light.

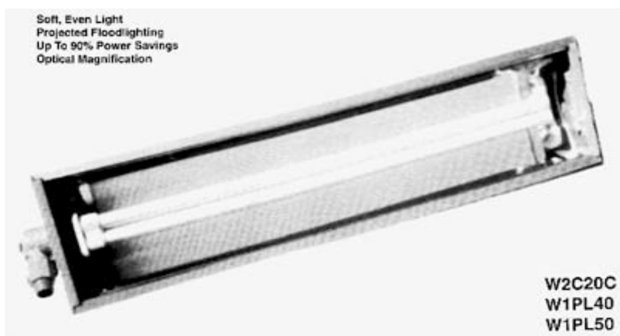


Figure 3: Philips PL-50 light.

2.2.3.2 Batter Capacity

The type of battery selected for energy storage for this project was Absorbed Glass Mat (AGM) deep cycle batteries. In these batteries, acid is absorbed between the plates and immobilized by Boron-Silicate glass mats (very fine fiberglass mats). Nearly all AGM batteries are "recombinant" - meaning that the Oxygen and Hydrogen recombine inside the battery. These use the gas phase transfer of oxygen to the negative plates to recombine them back into water while charging and prevent the loss of water through electrolysis. The recombining is typically 99+% efficient, i.e., almost no water is lost. Since there is no liquid to freeze and expand, they are practically immune from freezing damages. Typically AGM batteries allow 300 plus cycles of 100% discharges before it dies. If AGM batteries are recharged before it drops below 50% of its full capacity, they deliver a very long life (10+ years).

The selected battery type for this project was 6V 240Ah AGM batteries. Lower voltage (6V) batteries were selected because the battery bank can be built using less number of parallel connections and shorter lengths of battery-to-battery cables. Parallel connections require thicker cables, increasing the overall cost and space.

The battery bank capacity is determined based on the power requirement and the available cabinet space of the lighting system. If we assume to provide 8 hours of lighting per day using the lighting system discussed, it would require 440Wh or 37Ah at 12V. The cabinet has enough space for six-6V, 240Ah AGM batteries providing a total of 720Ah. In the control circuit, we do not allow more than 50% discharge to keep the battery life full, which would result in 360Ah of useable battery capacity at 12V. For the lighting, an inverter is used, which is about 90-95% efficient. Considering the inversion efficiency of 90%, the final useable energy is about 324Ah at 12V. Another source of loss comes from the power monitoring and recording microcontroller system, which requires 86mA per hour. Since it runs for 24 hours, it consumes about 2A per day. Therefore, the total number of amp hours required for the lighting system is 39Ah. The number of days that the battery bank can support is finally calculated as:

$$Number\_of\_days = \frac{Useable\_battery\_capacity}{Daily\_Ah\_requirement} \quad (1)$$

In the designed lighting system it is (Useable battery capacity) / (Daily Ah requirement) = 324Ah / 39Ah = 8 days. In summary, the battery bank consists of six-6V 240Ah batteries and it can support up to eight days without charging.

2.2.3.3 PV Panel and Wind Turbine

The energy production by solar and wind resources can significantly vary season to season. It is therefore important to analyze how much realistic electric energy can be generated from both sources throughout the year before the selection of PV panels and a wind turbine. This sub-section describes the selection and estimation process that was done in this project.

The PV selected for this project was KC130TM produced by Kyocera, and its key specifications are summarized in Table 2. The design process should start by selecting a PV panel based on the specifications and then computing the realistic energy generation for the given application.

Table 2: Specification of KC130TM

Maximum Power	*130W*
Maximum Power Voltage	17.6V
Maximum Power Current	7.39A
Open Circuit Voltage	21.9V
Short Circuit Current	8.02A
Length x Width x Depth	56” x 25.7” x 2.3” (1425 x 652 x 58mm)
Weight	11.9Kg (28.8lbs)

Although the PV panel is rated at 130W as its maximum power, the maximum is only attainable under solar radiation of a full sun near the equator. In Minnesota, the maximum convertible power of the 130W PV is about 96W under the full sun, which is 75%. This estimation is based on actual monitoring of the PV panel. Another estimate we need is the number of equivalent full solar radiation hours per day to compute the average rate of power generation per day. In Minnesota, 4.5 hours for spring, summer and autumn, and 2 hours for winter could be used, based on the national solar map (Figure 1). The actual daylight hours are longer than 4.5 hours and the radiation strength varies, but the useable hours provide an estimate of the total producible power per day. Finally, the expected daily producible power by the KC130TM is estimated as, 96W * 4.5hrs = 432Wh for spring, summer and fall, and 96W * 2hrs = 192Wh for winter. Since the lighting application requires 440Wh per day, it can be concluded that the PV panel itself is not sufficient, and the rest must be supplied from the wind turbine.

The wind turbine selected for this project was the 400W Air-X model (manufactured by Southwest Windpower). This unit contains internal electronics in the generator that work as a charge controller. The internal regulator senses the voltage from the battery and determines whether or not to continue charging. Once the battery voltage matches the regulation set point, the regulator will “stop” the turbine from charging. The key specifications for this wind turbine are summarized in Table 3.

Table 3: Air-X Wind Turbine Specification

Rotor Diameter	*46 inches*
Weight	13 lb
Mount	1.5” schedule 40 pipe
Voltage	12V (or 24V)
Startup Wind Speed	7mph
Rated Power	400W at 28mph
Blades (three)	Carbon Fiber Composite
Body	Cast Aluminum
Survival Wind Speed	110mph

According to the wind map (Figure 2), Duluth Minnesota (where the system is installed) is rated Class-1 (0-9.8mph) for summer and Class-2 (9.8-11mph) for the rest of the seasons. As an average, 5mph wind speed for summer and 10mph for winter can be selected from the wind map. Figure 4 shows a graph of monthly energy output in terms of KWh per average annual wind speed, which is a data sheet provided by the manufacturer. According to this graph, the monthly energy output is 1-3kWh for summer and 17-28kWh for the rest of the seasons. Translating it into daily output results in 33-100Wh for the summer and 567-933Wh for the rest of the season. We combine these estimates together with the solar generation to see if electricity generation is sufficient for daily needs. For a conservative estimation, the lower side of the graph is used. We also add local knowledge to fine-tune the estimate. In autumn and spring, solar radiation is not as strong as summer, about 75% of summer would be a reasonable estimate for Duluth, Minnesota. For wind energy, fall and spring only have about 50% of winter. Considering these adjustments, the final estimates are tabulated in Table 4.

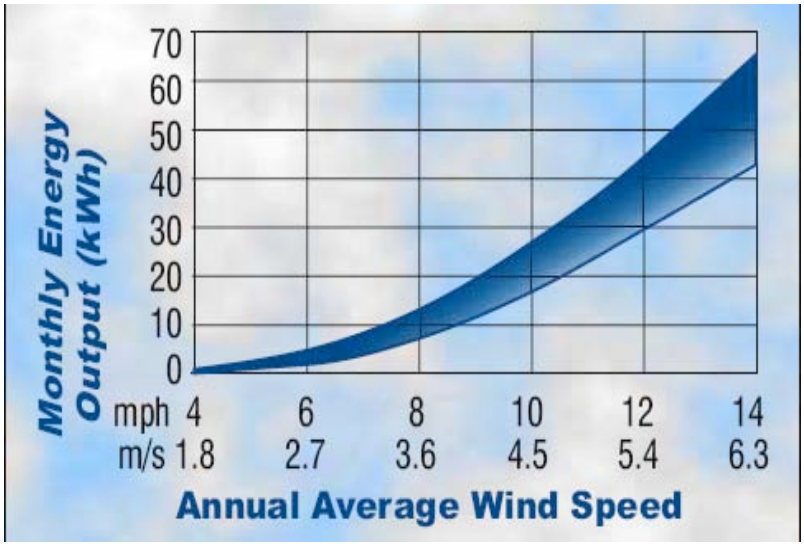


Figure 4: Monthly energy output of Air-X per annual average wind speed.

Table 4: Daily Power Generation and Power Needs for Each Season

	Winter	***Autumn***	***Spring***	***Summer***
PV Generated	192Wh	324Wh	324Wh	432Wh
Wind Generated	567Wh	284Wh	284Wh	33Wh
Total Generated	759Wh	808Wh	808Wh	465Wh
Power needs	440Wh	440Wh	440Wh	440Wh
Excess energy	+319Wh	+368Wh	+368Wh	+25Wh

It should be noted from Table 4 that the combined energy generation has sufficient excess energy except for the summer. However, in the summer the nights are shorter, and thus power needs are less. Although the storage can hold up to eight days of energy needs, the excess energy after fully charged is wasted since the charge controllers do not allow overcharging.

2.2.3.4 Charge Controls

Both PV panels and wind turbine require a charger controller to store the generated electricity into a battery bank. Most of controllers provide multi-stage charging capability along with an overcharge protection function to protect the battery life. Since the application requires turning the light on at night, the research team decided to use the PV panel as a photo detector for detecting the switching time. If the PV controller does not detect any current generation from the PV panel, it triggers a relay that turns on the inverter which supplies power to the light. Once the light is switched on, it should stay on for the duration of the dark hours at night. For this control, the PBRT 12-15 LA-15 produced by the ETA Engineering was selected as the PV controller. This PV charge controller generates a relay output that disconnects the DC load on low light and low battery voltage. Also, it shunts the incoming PV currents to a heat sink when the maximum charging voltage is reached to protect from the overcharging.

The small wind turbine selected for this project (Air-X) includes a regulator circuit inside the generator body. Therefore, no separate controller is needed. This design saves on cabinet space, but it has a disadvantage that if the control circuit is broken, the whole wind generator must be taken down to repair. The internal wind generation controller senses the battery voltage and determines whether or not to continue charging. The regulator stops from charging when the regulation voltage matches the set point. Since a 12V battery bank is used, the set point is set at 14V, which is a fully charged voltage.

2.2.3.5 Charge Monitoring and Data Logging Circuits

One of the objectives of this research is to investigate how reliably the proposed solar/wind hybrid renewable system can support the electricity needs of a street lighting application. It is therefore essential to monitor and record how much electricity is actually produced and consumed. The approach used in this project is to sense the current from each source and load using shunts. The shunts selected for this project are rated at 50mV/500A (MKB-500-50) and shown in Figure 5. This shunt is essentially a very low resistor and the voltage across the shunt terminals determines the current. The resistance of this shunt and the current are determined by:

$$R = V/I = 0.05/500 = 0.1\ m\Omega \quad (1)$$

$$I = V/R = V/0.1\ mA \quad (2)$$

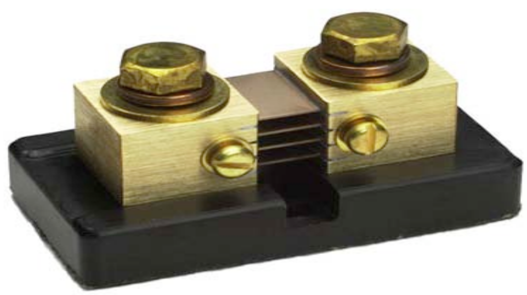


Figure 5: 50mV/500A shunt.

For this project, two shunts are needed to monitor the PV panel and the wind turbine and the voltages are monitored using Analog to Digital (A/D) converters. Since the load is controlled by the PV controller, no separate shunt is needed. For the A/D conversion, PentaMetric by Bogart Engineering was used. This unit samples the shunt voltages, converts them to current values, and then transmits the data through an RS232 interface. However, this unit does not have memory to save the shunt data. Therefore, data logging device is separately needed.

Data logging involves 24/7 continuous monitoring of the current flow into the battery bank from the two sources and the current flow out to the load. For this purpose, we selected TS-7260 Single Board Computer (SBC), which runs on an ARM-9 32bit processor with an embedded Linux OS. Important criteria of selecting TS-7260 were large available flash memory that is required for storing a years worth of data and a real-time clock for time-stamping of the data. Also, additional required parts used are a small keypad and an LCD display. This board is shown in Figure 6, which has the following specification:

- Imbedded Linux OS
- ARM9 CPU: Cirrus EP9302 (32bit, 16KB instruction cache, 16KB data cache)
- Clock: 200MHz
- 64MB SDRAM
- 32MB on-board NAND flash
- 2 USB
- 3 serial ports
- 10/100 Ethernet
- 30 DIO pins
- 2-channel on-board 12bit A/D converter
- PC/104 expansion bus
- Real time clock
- 4.5 – 20V DC Unregulated input
- Size: 3.8 x 4.8 inches

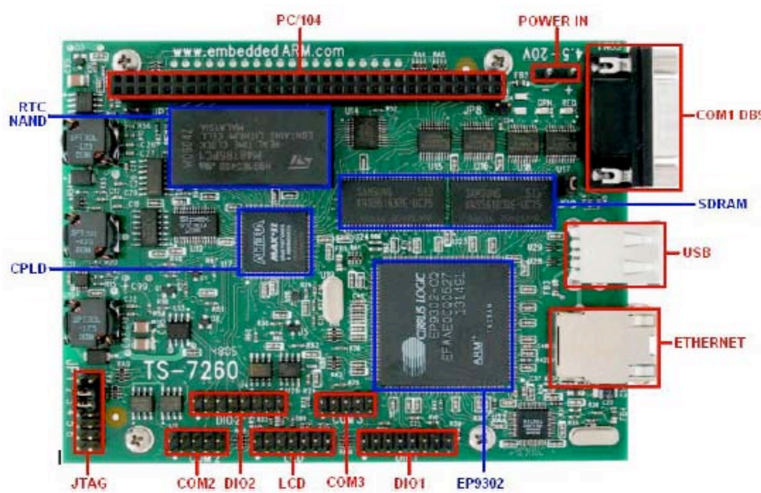


Figure 6: TS-7260 SBC.

This board is programmed on a PC using a gnu c-compiler gcc available at Cygwin. Cygwin is a simulated Linux OS that runs on a Windows environment. In order to produce a binary compatible with the ARM9 processor, a cross compiler called CrossTool is used. The compiled binary code is downloaded to the SBC either through an Ethernet or a serial port. To increase the data logging capacity, a 256MB flask memory stick was added through a USB port. This flash memory expansion provides data storage logging for more than 10 years for this power application. The enclosed data logger along with the PentaMetric is shown in Figure 7. In the picture, a small aluminum tube with mounting hole is a temperature sensor that records the temperature of the battery.

Figure 7: Data Logging System.

2.2.3.6 Wind Load for PV Panel and Wind Turbine Mounting Pole

In designing the pole, one of the important factors is the wind load that the pole must reliably support. Wind load is computed based on so-called "Basic Wind Speed" which is simply a statistical average wind speed. A generic form is given by:

$$F = A \times P \times C_d \qquad (3)$$

where F is force in pounds, A is the projected area of the item, P is the wind pressure in pounds per square feet (psf), and C_d is a drag coefficient (C_d = 2.0 for plates = and C_d = 1.2 for cylinder). The basic wind pressure, P, is computed by:

$$P = 0.00256 \times v^2 \qquad (4)$$

where v is the wind speed in mph. A simple lookup table constructed based on this relation is shown in Table 5.

Table 5: Wind Pressure

Basic Wind Speed(mph)	70	80	90	100	110
Wind Pressure(psf)	12.5	16.7	20.7	25.6	31.0

The area occupied by the PV panel in our design is 4.7' x 2.1' = 9.87 square feet (f^2). The area formed by the wind turbine rotor is 11.5 f^2. The PV panel is mounted at a

60 degree angle from the ground, and the area covered by wind turbine rotor is not a filled space, so taking those in account, we estimate the projected and effective area as:

$$\text{PV panel} = 9.87 \times \sin 60° = 8.5 \text{ f}^2 \quad (5)$$

$$\text{Wind turbine effective area} = 11.5 \times 2/3 = 7.7 \text{ f}^2 \quad (6)$$

If we consider 120mph wind speed as the maximum wind speed, the wind load created by the PV panel and the wind turbine is computed as:

$$\text{Wind load (F)} = 16.2 \times 36.9 \times 2 = 1{,}196 \text{ pounds} \quad (7)$$

The additional wind load is created by the pole itself. The pole is designed by the manufacturer but this wind load specification should be provided.

The next important issue is where to position the PV panel, the wind turbine, and the light arm in the pole. A logical choice is that the wind turbine must be positioned at the highest point, and the light arm should be positioned below the PV panel in order to avoid the shade of the light arm on the PV panel. The pole height was chosen 37 feet, the PV panel was positioned at 28 feet above ground, and the light arm was positioned at 23 feet 9 inches above ground. The final design produced by the manufacturer according to the specification is shown in Figure 8.

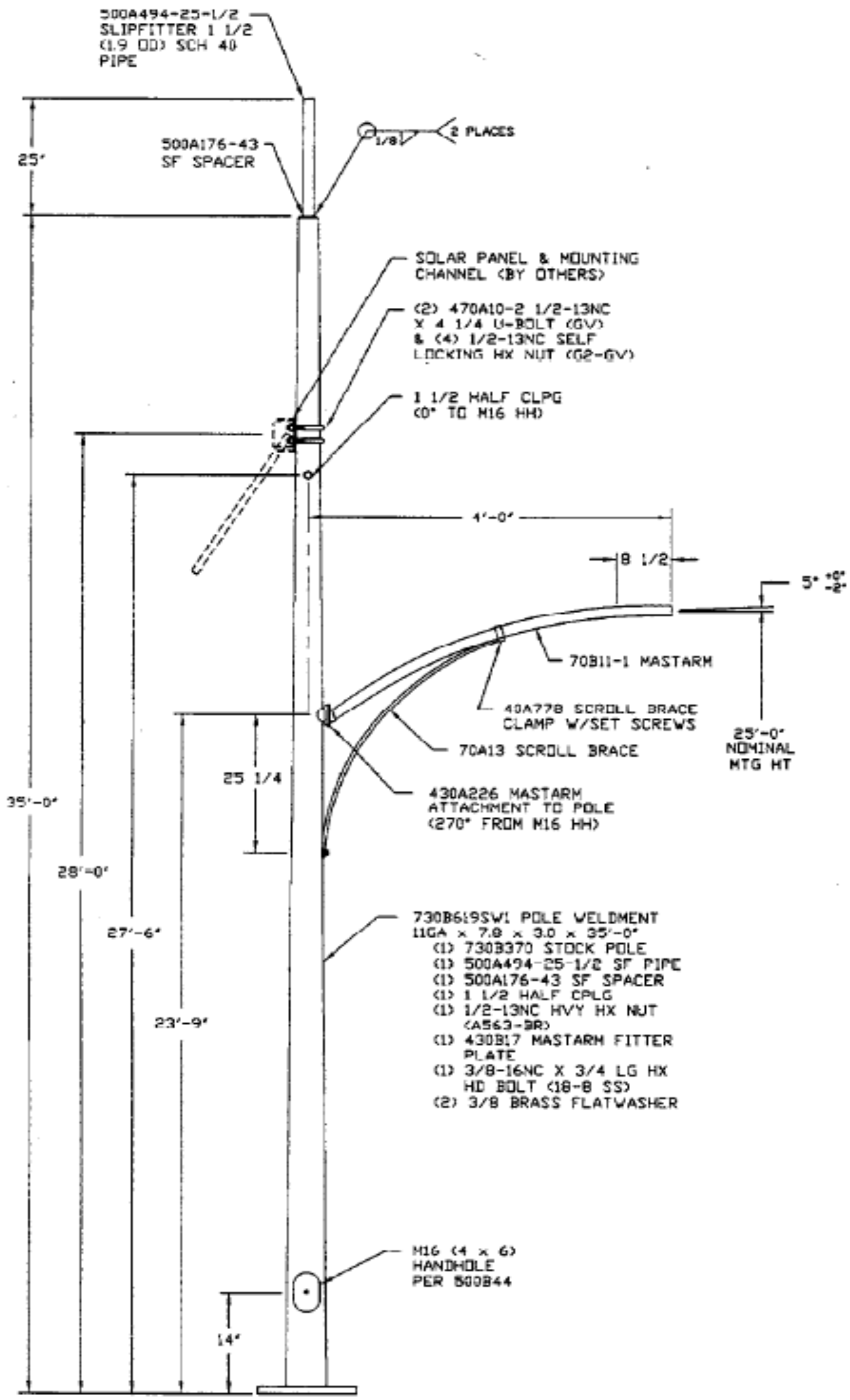


Figure 8: Mounting pole design.

2.3 Implementation

2.3.1 Hardware

Implementation involves pouring the concrete base for the pole and a pad for the cabinet, installing the wind turbine and the PV panel on the pole, burying a ground rod for earth ground, and wiring the system. The battery cabinet was insulated using Styrofoam and was custom made (30” x 30” x 12”) to create a rack for batteries. The picture of the fully implemented cabinet is shown in Figure 9. Six-6V lead acid AGM batteries, charge controller, and a data logging system are housed in the cabinet. The completed light pole system installed is shown in Figure 10. The renewable light pole operating at night is shown in Figure 11.

Figure 9: Cabinet for Holding Batteries and Control Units.

Figure 10: Completed Renewable Light Pole System.

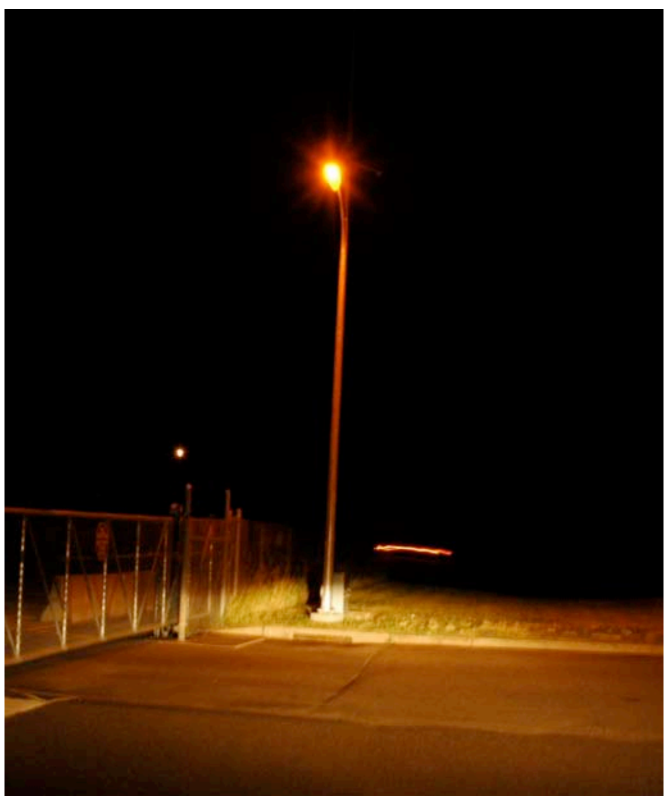


Figure 11: The Renewable Light Operating at Night.

2.3.2 Data Logging

Logging of power generation and consumption data was done using the microcontroller system described in Section 2.2.3.5. The data was stored in a flash memory stick connected through a USB port of the microcontroller. Since the system runs on Linux, a USB driver is available from the OS and it is mounted as a regular file system. The log data downloaded is written to a file named with that day's date, and it contains 24 hours of data. For example, log data recorded on October, 23 2006 is labeled *20061023.txt.* The data is recorded in ASCII text separated by commas. The complete columns are summarized in Table 6. The data is recorded at 15 minutes interval which occupies a single line. For each day, 96 lines of data are recorded.

Table 6: Columns of the Data File

Column #	*Description*
1	Timestamp, hh:mm:ss
2	Battery voltage, V
3	Average battery voltages of the 15minute interval, V
4	PV currents, amps
5	PV amp hours collected in 15 minute interval, Ah
6	Wind turbine currents, amps
7	Wind turbine amp hours collected in 15 minute interval, Ah
8	Inverter currents, amps
9	Inverter amp hours collected in 15 minute interval, Ah
10	Temperature of the battery cabinet in Celsius

Table 7: Key Functions

Key	***Description***
1	Displays Battery 1 Volts
2	Displays Average Battery 1 Volts
3	Displays PV Amps
4	Displays PV Amps/Hour
5	Displays Wind Turbine Amps
6	Displays Wind Turbine Amps/Hour
7	Displays Inverter Amps
8	Displays Inverter Amps/Hour
9	Displays Temperature
0	Resets PV, Wind Turbine, and Inverter Amps/Hour
CLEAR	Clears display and displays Home Screen
ENTER	Return Home(First Row, First Column)
UP arrow	Return Home
DOWN arrow	Return Home
HELP	Displays 'Help Menu' and then displays Home Screen
2nd	Return Home

The system includes a LCD that displays of the current values when the key is pressed. The function of each key is summarized in Table 7.

2.3.3 Cost Analysis

Finally, the cost of the system is discussed. The most expensive part was the light pole due mainly to a customized design, and the next expensive item was the batteries. Actual purchased cost of components is summarized in Table 8 with exclusion of miscellaneous items such conduits, cables, connectors, etc. In the cost summary, if this lighting system is implemented for actual uses, the data logging microcontroller is not needed. Since the pole and lamp are required items in grid-tied lighting systems, the additional items in the renewable lighting system are a wind turbine, a PV panel with a mounting rack, batteries, an inverter, and a PV regulator. The total cost of these additional items is $2,640. This means that a renewable light pole cots $2,640 more than the grid tied light pole.

Another cost aspect is installation, which is not included in this figure. A grid-tied light pole installation can be very expensive if it is in a remote area, because of the cost of bringing the utility lines and maintenance. As described in the chapter introduction, the cost of bringing a utility power line is approximately $3 per foot in 2007, which translates to $15,840 per mile. Since the cost of the renewable light pole is not affected by the distance, the final cost of renewable light pole can be significantly cheaper. The distance for the price break even point occurs at 880 feet or 293 yards, i.e., if any light pole has to be installed at a distance farther than 293 yards in Minnesota then the cost of renewable light pole would be cheaper.

Table 8: List of Purchased Components

Items	***Cost***
Wind turbine, Air-X 12V	$539
PV Panel, KC130TM	$609
PV Mount Rack, UniRac 400209	$170
6V 240Ah AGM Batteries ($166 x 6)	$996
37’ Light Pole	$1,280
Inverter	$202
PV Regulator (PBRT 12-15)	$123
Microcontroller with shunts	$465
PL-50 housing and lamp	$221
Total	**$4,606**

2.4 Data Analysis

This section describes analysis of the power data collected from the data logging microcontroller. The power data was collected from November of 2006 to June of 2008. During this time, the Wind Turbine broke from 1/1/07 – 5/29/07 and from 7/11/07 – 11/11/07, which is about 9 months. On the other hand, the PV panel was working without a single problem for the entire duration. This repair record shows that the reliability of wind turbine is worse than that of PV panels. Having many moving parts in the wind turbine is likely the cause of the problem.

Monthly totals of solar panel Amp Hour (Ah), wind turbine Ah, and inverter Ah from November 2006 to July 2007 are shown in Figure 12. It can be clearly noticed that the wind turbine did not generate any energy from November 2006 to May 2007. On July 2007, the wind turbine worked only for 11 days. During this period, the wind turbine was uninstalled and sent back to the manufacturer for a repair. By examining the graph, it can be noticed that the inverter consumed power during the period that the wind turbine was not working, which indicates that the light was using the energy generated by the PV panel. It can be also noticed that whenever the wind turbine is supplying extra energy, the light is using that excess amount.

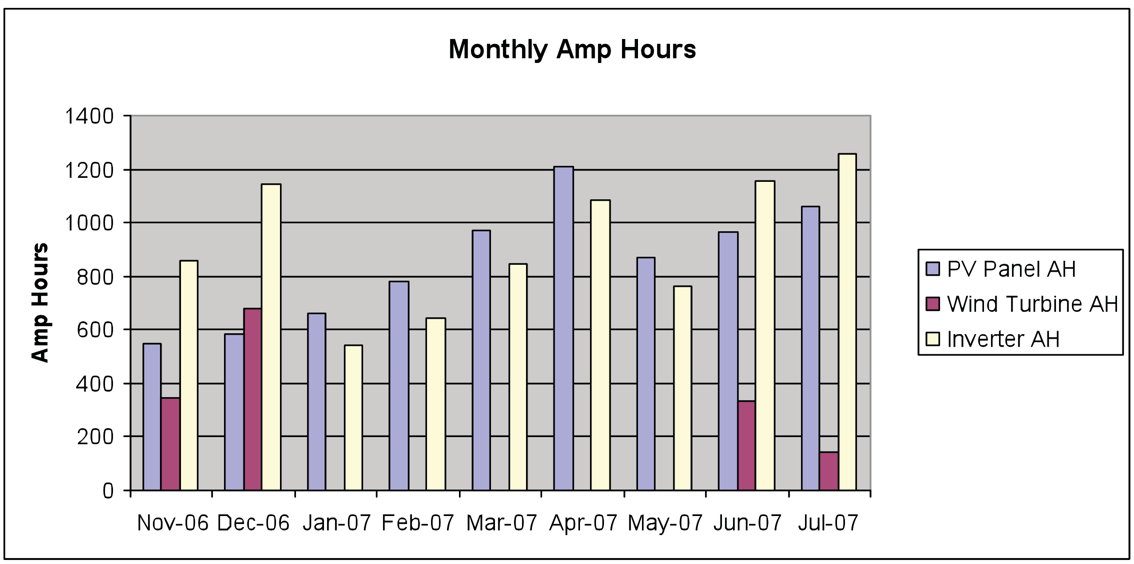


Figure 12: Monthly Ah Totals from November 2006 to July 2007

The months that both the PV and wind turbine worked without any malfunctions are shown in Figure 13. Notice that Ah consumption of the inverter is less during the winter months (Nov, Dec, Jan, and Feb) and more during the spring/summer months (Mar, Apr, May, and June), which corresponds to the amount of energy generated. The PV panel generated an average of 473Ah per day during the winter months and 646Ah per day during the spring/summer months. The wind turbine generated and an average of 537Ah per day during the winter months and 615Ah per day during the spring/summer months. The detailed daily Ah data for the entire data collection period is provided in Appendix-A.

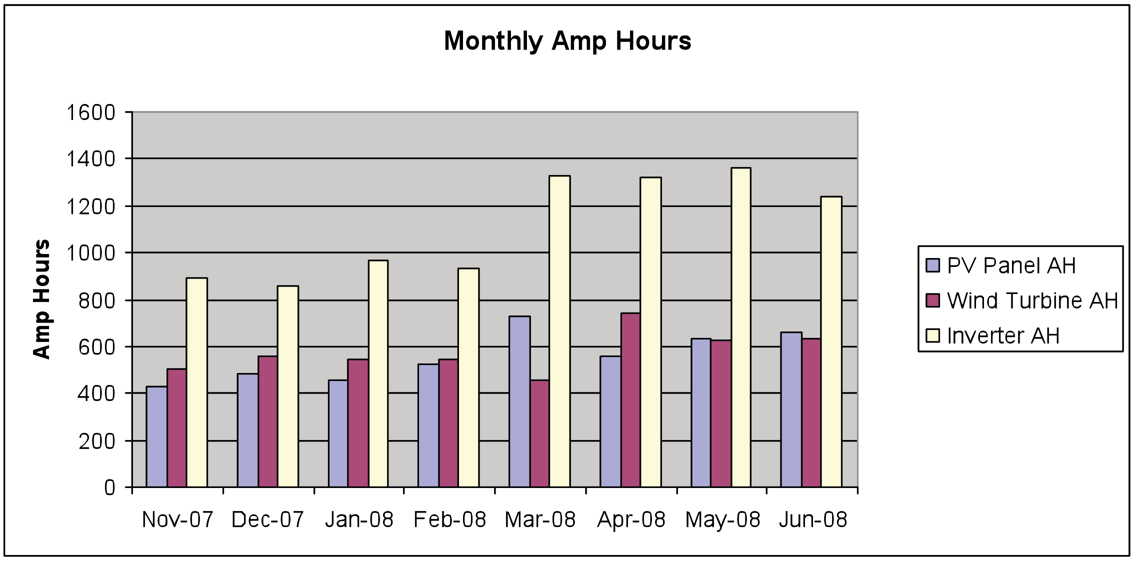


Figure 13: Monthly amp hour totals from November 2007 – June 2008

The graph in Figure 14 shows a plot of solar panel Ah, wind turbine Ah, inverter Ah, and battery volts over a three day span. The 3 days plotted were from 6/7/2007 – 6/10/07. These dates were chosen because in June the wind turbine was working properly. Examining the Ah plots first, the inverter turns on at about 9:00 PM every night, runs for 8 hours and turns off at about 6:00 AM. In this graph, inverter Ah are originally negative but converted to positive numbers for ease of comparison. The PV panel Ah are zero at night, between the time of 9:00 PM and 6:00 PM. They start increasing after sunrise and reach there max at about 2:00 PM. Wind blows at all hours of the day so the wind turbine Ah have no definite pattern to follow. According to the National Weather Service Forecast Office [5], 6/07/07 was a very winding day with the peak wind speed reaching 51mph. This is evident on the graph of wind turbine Ah showing a peak of about 9Ah around 7:00 PM. The battery volts plot rises during the day when the solar panel Ah are high and falls at night when the inverter is on. Also notice that during the last two days, the battery voltage is higher than the first because there are both solar and wind resources present charging the batteries.

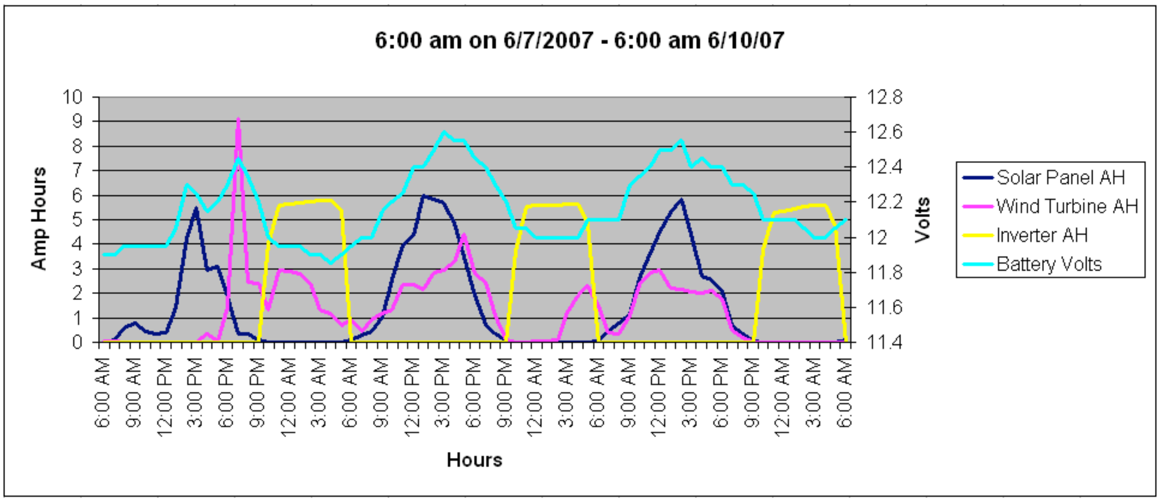


Figure 14: Three day Ah analysis.

The next graph (Figure 15) shows for how many days each month the inverter was on or off. "Inverter on" means that the batteries were charged enough to keep the light on for 8 hours. Anything less than that was considered "inverter off". During this 8 month period the inverter was off for a total of 30 days. Of these 30 days, the inverter did not turn on at all on three days, or 1.3% percent of the 8 month period. The other 27 days, the inverter was on for at least an hour. For detailed daily power data, please refer to Appendix-A.

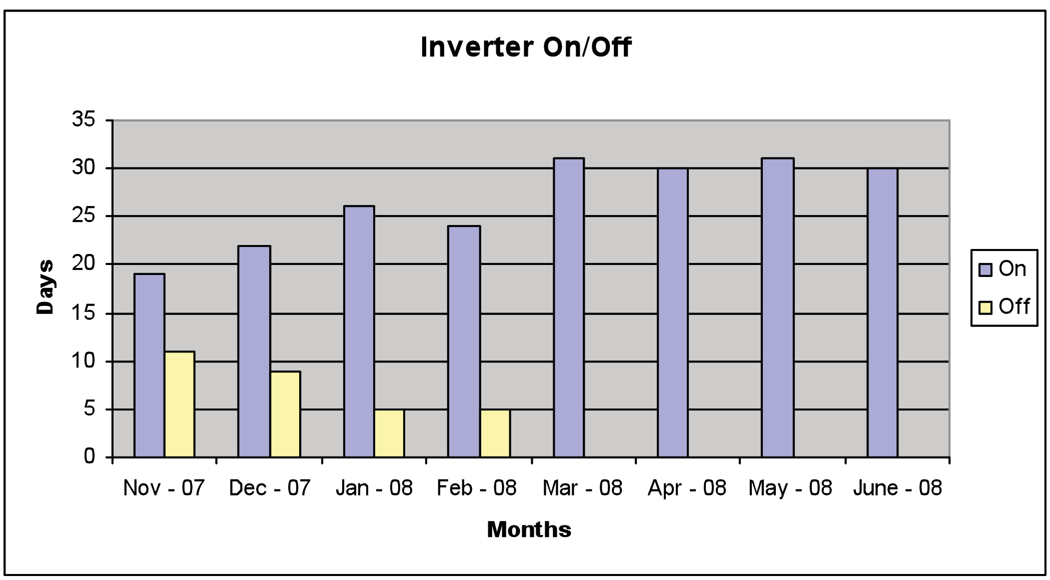


Figure 15: Days inverter On/Off, from Nov 2007 to June 2008.

The next six graphs will show daily values for PV panel Ah, wind turbine Ah, inverter Ah, and battery voltage. These daily values will be compared to climatology data in Table 9. This climatology data was collected from National Weather Service Forecast Office website [5].

Table 9: Daily Climatological Data

Day	Temperature(°F)		Precipitation (inches)		Wind Speed (mph)		Cloud Cover(%)
	Max.	Min.	Water	Snow	Avg.	Peak	
11/28/06	38	32	0.75	0.0	17.8	41	100
12/31/06	35	28	0.67	2.6	12.1	28	100
6/12/07	82	69	0.00	0.0	7.4	17	10
6/15/07	85	72	1.03	0.0	8.6	29	50
6/19/07	68	52	0.00	0.0	17.6	38	40
7/10/07	67	50	0.23	0.0	17.5	60	60

Figure 15 shows the daily data for 11/28/07. This day was rainy, windy, and cloudy according to Table 9. The graph shows that the solar panel Ah were near zero the entire day. The wind turbine Ah were hovering between 3 and 6 for the first half of the day. The battery voltage stayed above 12.2V as a result. In the afternoon, the wind died down, the inverter turned on at about 5:00 PM and the battery voltage dropped below 12V. This is an interesting example since it shows that the electricity on this rainy day was almost entirely generated by the wind turbine while very little was generated by the PV panel.

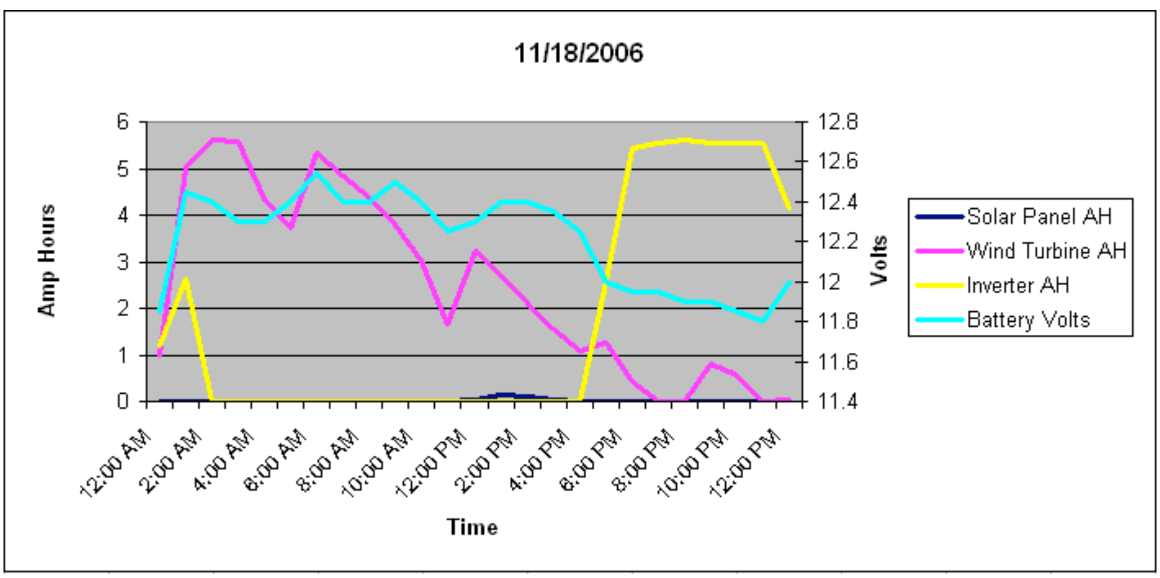


Figure 16: 11/18/2006 hourly Ah data.

On 12/31/2006 there was an ice and snow storm, and the Ah graph is shown in Figure 17. The wind turbine actually broke during this storm at about 11:00 PM. Since there was no sun, the solar panel Ah were near zero all day. The storm didn't start until about 8:00 PM. During the storm, the wind turbine Ah peak at about 14Ah and the battery voltage peaked at bout 15.5V. Again, this example shows that wind and solar energy work each other as a complementary on ice and snow storms.

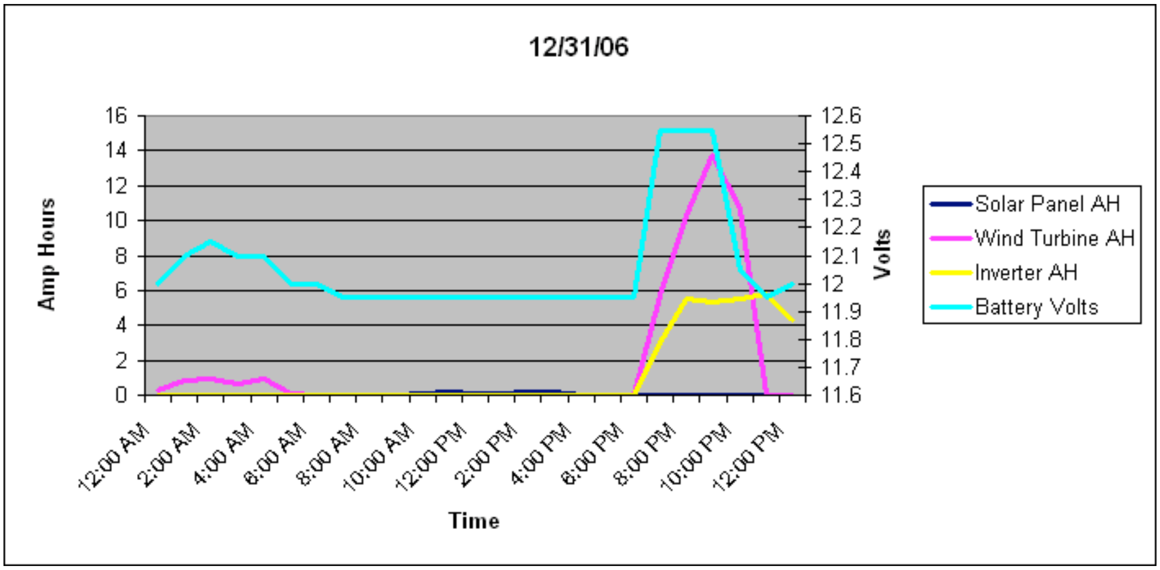


Figure 17: 12/31/2006 hourly Ah data

Next example shows a day with no wind case. This occurred on 6/12/07 and the hourly Ah graph is shown in Figure 18. This day was very sunny, only 10% cloud cover, very calm, and no precipitation according to Table 9. Since there was no wind, the wind turbine Ah were near zero all day. The clear skies provided an excellent day for solar energy and the graph indicated that. The PV panel Ah creates a nice bell curve during the day. The solar panel Ah peaked at almost 6Ah at about 3:00 PM. The battery voltage started out around 12V in the morning but once the solar panel Ah began to rise, so did the battery voltage.

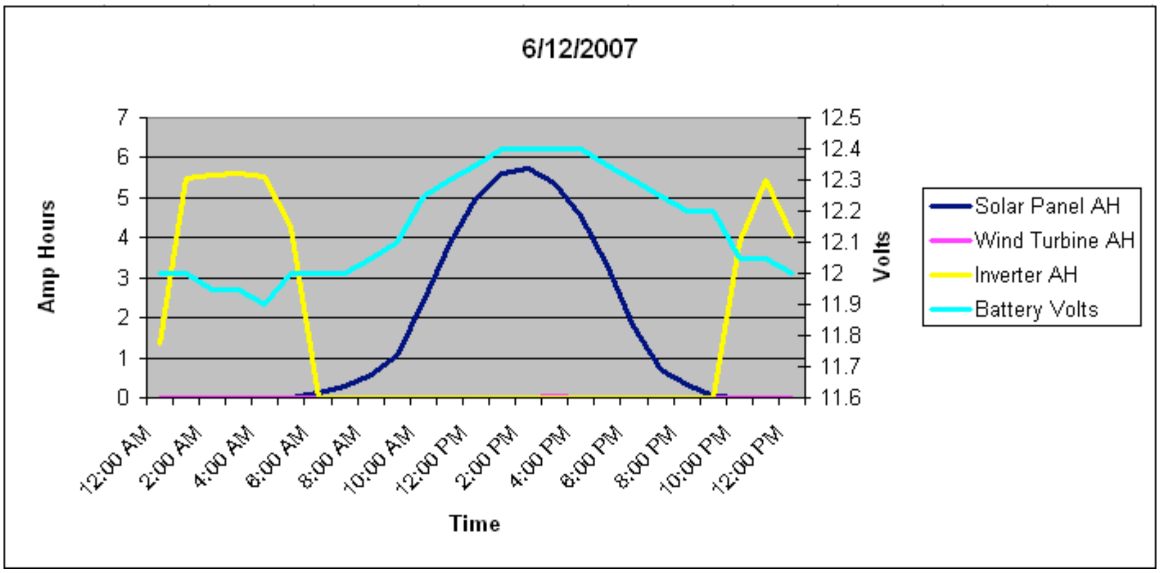


Figure 18: 6/12/2007 hourly Ah data

Figure 19 shows the hourly Ah data for 6/15/07. This day produced over an inch of rain, low wind, and was partly cloudy. The solar panel Ah plot has a nice rising edge in the morning but towards mid-afternoon it looks like the sky became cloudy and introduced the gradual fall of the curve. The wind turbine Ah increased in the afternoon about the same time the clouds rolled in. This was probably the beginning of the storm that produced over an inch of rain. Again, a complementary relation between wind and solar energy can be observed.

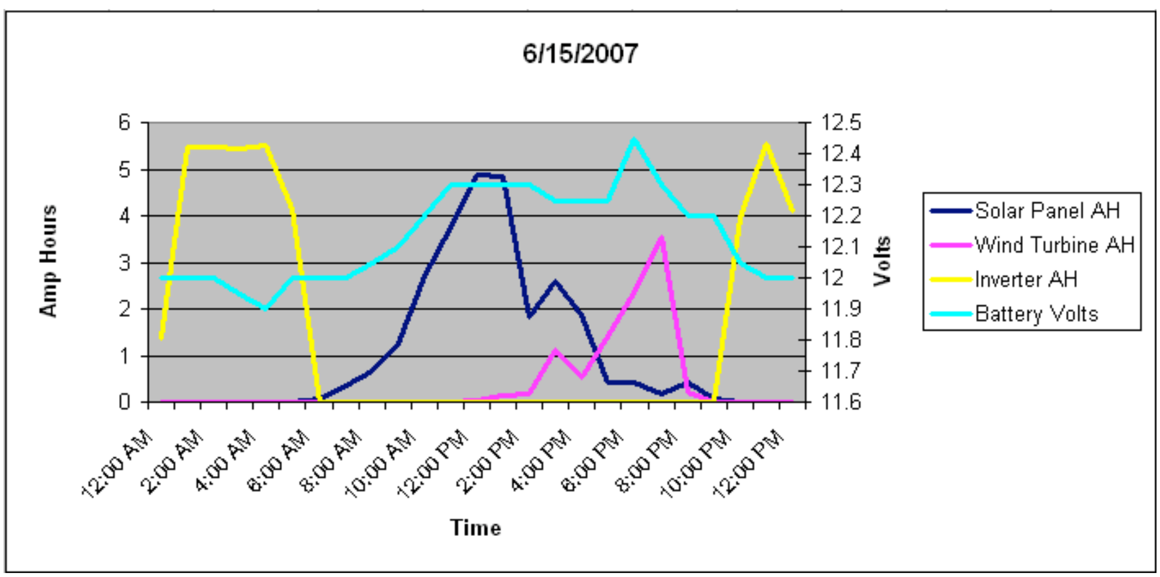


Figure 19: 6/15/2007 hourly data

Figure 20 shows the hourly Ah data for 6/19/07. This day produced both high wind and lots of sun. There was no precipitation. The wind turbine Ah slowly increased from early morning to about noon and peaked out at over 6Ah. Then it slowly decreased in the afternoon and night. The PV Ah resembles its usual bell curve charging the battery bank from 6:00 AM to 9:00 PM. Since the wind turbine Ah were present over a long period of time and the solar panel Ah were present, the battery voltage charged to over 12.6V. This example shows that sunny days can be also windy, providing both solar and wind resources.

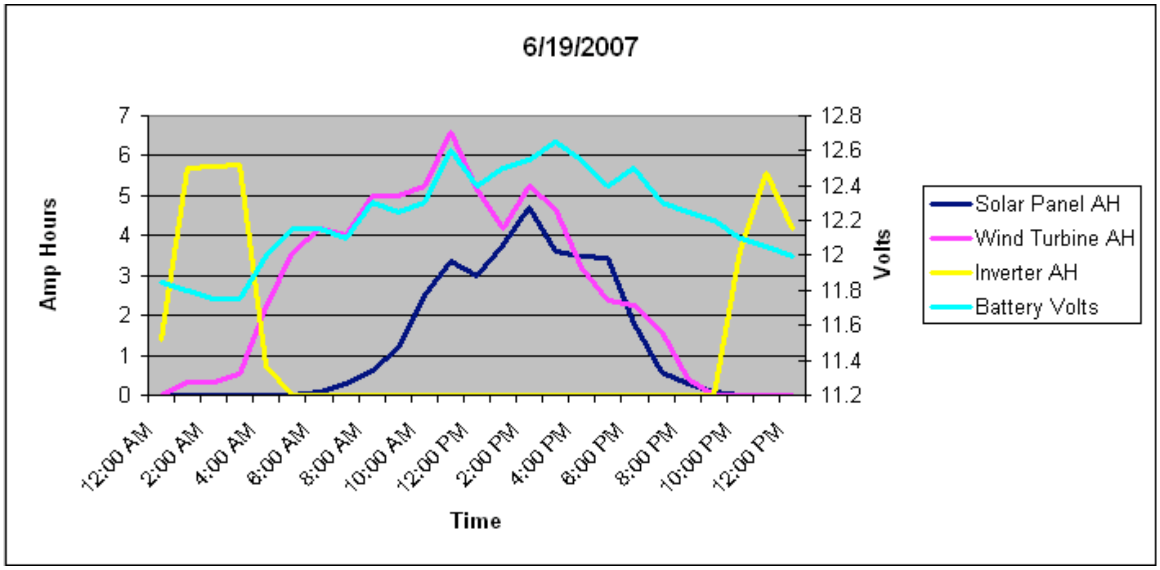


Figure 20: 6/19/2007 daily data.

The next example shows the day the wind turbine broke, which occurred on 7/10/2007. This day was partly cloudy with a little rain and very high winds. From the graph in Figure 21, it can be noticed that the wind turbine broke at 7:00 PM. This day had a 60% cloud cover not allowing the solar panel Ah to peak above 4Ah. The wind picked up at about noon and charged the batteries to over 13V making up for the lack of sunlight. Unfortunately, the regulator/charge circuit inside the turbine was partially burned. This turns out to be the weakness of the Air-X wind turbine model used in this project. This model places the regulator/charge circuit that converts AC to DC and monitors charging inside the generator housing which moves when winds blow. Therefore, the circuit must endure high vibration and water. Although the company specifies the surviving wind speed at 110mph (Table 3), it broke at 60mph in this case. The company is now producing new models with a separate regulator/charge controller box that can be placed inside the cabinet where the circuit is protected.

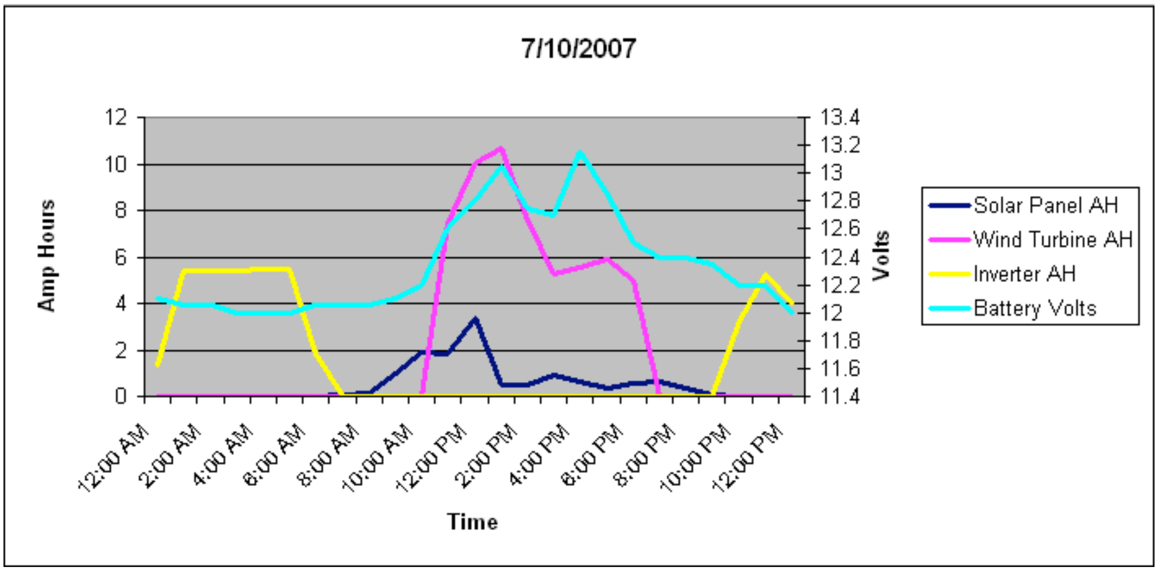


Figure 21: 7/10/2007 hourly Ah data

2.5 Chapter Conclusion

The first objective of this research was to design and implement a wind/solar hybrid renewable street light system that can self-sustain without a connection to an electric grid. The second objective was to investigate whether such a system is economical and practical to provide a street lighting application. The third objective was to learn how the wind and solar energy sources complement each other under field test conditions.

A self-sustainable solar/wind hybrid light system was successfully designed and constructed using off-the-shelf components with a total cost of $4,600. The system was installed at the parking lot of Mn/DOT District-1 and monitored for two years. According to the cost analysis shown in Section 2.3.3, the break even installation cost against a grid-tied light pole occurred at 293 yards, which suggests that for most remote ITS applications solar/wind hybrid system is cheaper than a grid-tied system. Moreover, according to the energy generation/consumption data collected over two years and the examples given in Section 2.4, there exists many days that solar energy itself was not sufficient to support the lighting application and the wind provided supplementary energy. Analysis of hourly data shows that many rainy and snowy days had stronger winds, providing complementary energy resources when solar radiation is deficient.

In conclusion, for many remote ITS applications a solar/wind hybrid generator along with sufficient battery storage should provide a reliable renewable power source. It is recommended that the system is designed to maximally utilize the resources by analyzing the availability of annual wind and solar radiations. The design steps provided in Section 2.2 could be used as a reference.

CHAPTER 3: DEVELOPMENT OF A GRAVEL ROAD TRAFFIC COUNTER

3.1 Chapter Introduction

Traditional pneumatic tube counters (which is often simply called, tube counters) are known to be problematic in rural gravel roads, due to air leaks caused by punctures and/or sensitivity decreases caused by dirt accumulations around the tube. The sharp edges of gravel and the force induced by vehicle weights easily damages the tube. Decreases in sensitivity or air leaks are not easily detectable and often cause miscounts. Some manufacturers provide protective jackets for tubes, but the low cost advantage is reduced by adding more parts and installation takes more efforts to put on a jacket. Another characteristic of tube counters is that it counts the number of axles, and vehicle counts is often obtained by dividing the total axle counts by two. Depending on the mix of trucks, this simple approach leads to a counting error of as high as 30% (according to Mn/DOT traffic engineers). Therefore, tube count data is often adjusted using vehicle classification data.

Mn/DOT TDA office collects traffic count data from approximately 32,000 locations around the state, and the main technology of choice is still tube counters because of its low cost and simplicity in installation. Investigation of alternative technologies for gravel roads was proposed by Mn/DOT TDA. Although many types of vehicle detection and counting technologies exist today, e.g., inductive loops, micro-loops, videos, ultra sound, passive infra red (PIR), microwave radars, lasers, etc., most of them are designed for permanent installations and not suitable for portable short duration counts (typically 48 hours). They typically require installation of a mounting pole, excavation of grounds, or cutting of pavement, all of which are time consuming and expensive for portable counting applications. Pole mounted counters, such as radar and video counters, are expensive and require calibration and aiming, which is considered too time consuming and too costly for short duration counts. Therefore, tube counters are still the main technology for most transportation departments for portable vehicle counting.

There are several other aspects that must be considered in designing a rural traffic counter. First, lanes are not clearly defined or marked on rural gravel roads, and the vehicles may not follow the assumed lane. This means that vehicle counters must detect vehicles in some sort of an arbitrary range, i.e., they must be able to detect as long as a vehicle passes through some part of the road. The second important aspect in rural roads is that the detectors should not count animal or people crossing as vehicles, which is a problem with PIR sensors. In addition, the equipment must be well protected from damages caused by wildlife. Another aspect and a requirement in rural roads is that the detector should draw only a small amount of power so that it can operate on a small portable battery (typically less than 5Ah battery). Most of the today’s commercially available vehicle counting technologies fail to meet all or some of the mentioned requirements.

After studying the present detection technologies, the PI found that coil-wound fluxgates satisfy all of the requirements for gravel-road traffic counting [7]. The wide detection range satisfies the needs of traffic detection in rural gravel roads with no clear lane boundary. Since fluxgates can only detect ferrous objects, the vehicle counting is not affected by animals or people. However, vehicle counters using this technology are not presently available. Based on this preliminary study, TDRL developed a prototype using a fluxgate sensor, and it was named "Bulldog Traffic Counter." It consists of a fluxgate sensor and a console that processes the signal. The sensor is housed in a 14 inch long PVC pipe with a 1.5 inch diameter and is easily buried on the shoulder of the road. It then detects any vehicle that passes within a width of about 40 feet from the sensor. This region is large enough to encompass the entire road. The vehicle detection range is adjustable by controlling the sensitivity threshold. Installation of the sensor is simple and easy and takes less than five minutes. Since the sensor never directly touches any part of the passing vehicle, its durability is not affected by the number of vehicles that pass by. With a small portable 5Ah battery, the system is estimated to run about 100 hours, which is more than twice the minimum 48 hour requirement. One important and critical advantage of the Bulldog Traffic Counter over the conventional tube counter is that it counts the number of vehicles instead of the number of axles. This solves the old problem of inaccuracy in tube counters caused by axle counts. Another advantage is that, since magnetic fields are not affected by rain or snow, traffic surveys could be performed during any time of the year. It even works equally well on snow covered roads, which would not feasible using tube counters. The rest of this chapter describes the working principle, design, implementation, tests, and conclusion.

3.2 Sensor Working Principle

The earth can be thought of as a huge magnet that is aligned between its north and south poles. This huge earth-magnet generates magnetic field as shown in Figure 22, but with relatively weak field strength (0.1-0.001 Gauss). The alignment between the earth magnetic axis and the earth spin axis is slightly off, i.e., about 11.5 degrees off in the direction as shown in Fig. 22. Earth's magnetic field has long been used as an important guide for navigation. However, it can be also used for vehicle detections, since magnetic fields are deformed by ferrous material. This deformation has also been used for detecting mines, concealed weapon detection, and tracking large objects [8].

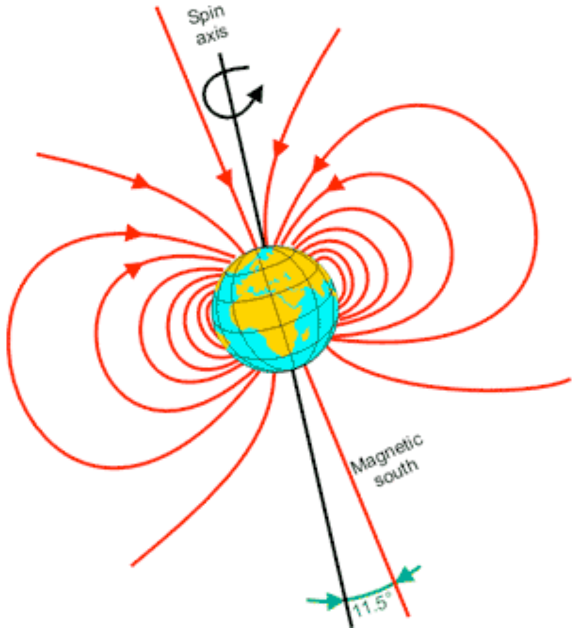


Figure 22: Earth magnetic fields

When the earth's magnetic field is viewed in a small area (such as 10m x 10m), the field in general appears as a uniform directional field with a constant density because of a small area relative to the huge size of earth. Since the ferrous materials are magnetically permeable, the earths' magnetic field is deformed and creates a density difference. Consider a road with no vehicles; the magnetic field will then appear uniform as shown in Fig. 23.a. If a magnetic field sensor, called a magnetometer, is placed, it will detect the strength of the earth magnetic field. Next, consider that a vehicle is present as shown in Fig. 23.b near the magnetometer. The magnetic fluxes will then be concentrated around the vehicle (shown as a rectangle) and thus the magnetometer will detect a lesser density in comparison to absence of the vehicle. Next, suppose that the measured magnetic flux density is thresholded above a level that a vehicle is detected, giving logic 0 for absence of a vehicle and logic 1 for presence of a vehicle. Then, the magnetometer will be able to count the number of vehicles. The sensor used for the Bulldog counter is a fluxgate which is a form of magnetometer and works using this principle. It simply counts up whenever the fluxgate detects the earth magnetic field disturbance.

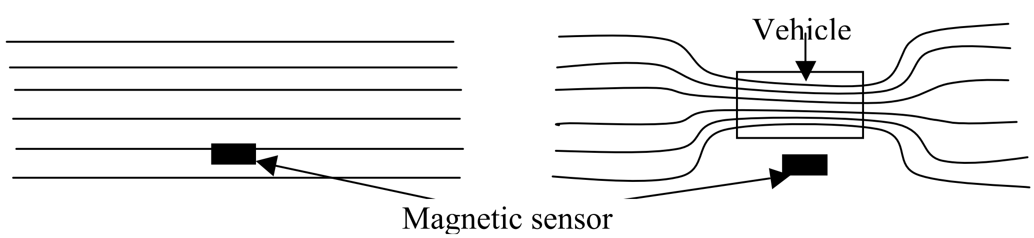


Figure 23: Effect of earth magnetic field: left (a.) --without a vehicle, right (b.)--with a vehicle.

The ambient magnetic fields of earth (i.e., the magnetic field with no ferrous material presence) often do not stay constant and can be influenced by geological structure of the area or by stopped vehicles. Therefore, there is a need to track the ambient magnetic field. The Bulldog counter continuously tracks the ambient field, and the counting is done based on the change of the adjusted ambient field. For example, if a stopped vehicle is present near the Bulldog sensor and another vehicle passes by, it will automatically adjust the value of the ambient field and will detect and count only the vehicle that is moving.

Next, the working principle of fluxgate is described [9-11]. For illustration, Figure 24 is used. The soft magnetic material of the sensor core is periodically saturated in both polarities by the Alternating Current (AC) excitation field, which is produced by the excitation current $I_{exe}(t)$ through the excitation coil with frequency, *f*. Because of that, the core permeability $\mu(t)$ changes with 2*f* frequency, and the Direct Current (DC) flux associated with the measured DC magnetic field B_0 is modulated. If the measured DC field B_0 is present, the associated core flux is also change with 2*f*, and the voltage V_{ind} is induced in the pickup coil having N turns. This voltage is proportional to the measured field. The details on magnetization computation are complex and only suitable for electrical engineers with magnetism background. Reference [7] is recommended for further information.

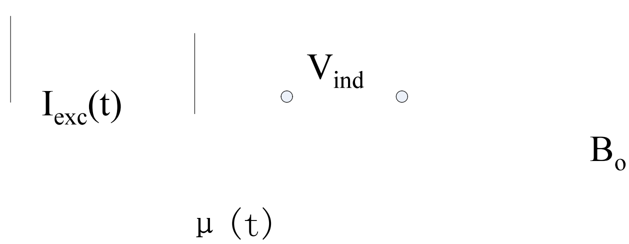


Figure 24: Illustration of fluxgate working principle

3.3 Implementation

Since the UMD lab does not have coil winding tools, the coil wound fluxgate was purchased from a vendor (Winland Electronics Inc). The coil wound on core is inserted in a 1 and 1/4 inch diameter PVC pipe with a length of 15 inches. This sensor is shown is Figure 25.

The overall system design was simplified by using a microcontroller, so that analog signals are directly processed using a digital algorithm. The inside of the box, shown in Figure 25, contains a simple analog circuit that controls the supply voltage using a regulator, which is fed to the sensor and direct connection of sensor output to a microcontroller. It includes an LCD display and a keypad for entering user inputs. For convenience, a buzzer was added for audible tests to indicate detection of a vehicle. The buzzer output can be disabled by a switch on the right. The middle switch is a sensitivity control switch which simply controls signal amplification. The switch on the left is an On/Off switch that connects or disconnects the battery. The microcontroller used is a C8051F020 that runs on a 22MHz clock. The choice of this microcontroller was made because of the on-chip A/D converters and 64K bytes of flash memory that can be used for recording the count data. In summary, the final Bulldog counter includes the following hardware and software components.

- C8051F020 RISC microcontroller with 22 MHz clock
- Real-time clock
- 64K bytes of flash memory
- Fluxgate probe
- Fluxgate excitation signal generator circuit
- Recording of vehicle count data in a nonvolatile memory
- 15 minutes interval count and total count
- Real-time timestamp
- PC downloadable data through a serial port
- Key pad and LCD display for control
- Buzzer function for audible tests

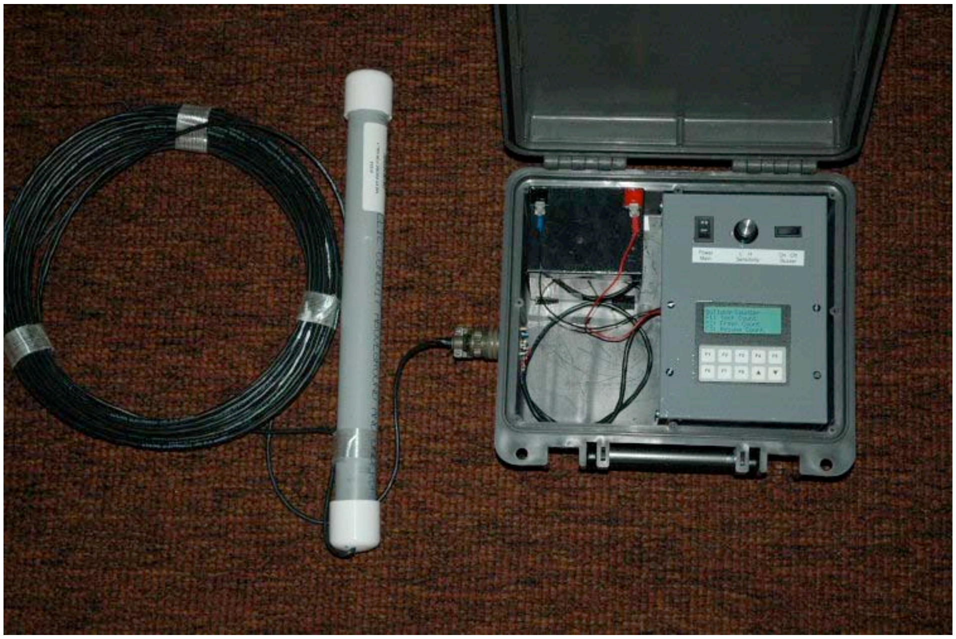


Figure 25: Prototype Bulldog gravel road counter

3.3 Software Design

3.3.1 Operating System

The LCD can display only 4 lines of 20 characters, which limits the displayable information. The membrane keypad has eight functional keys F1, F2, …, F8 and two arrow keys marked ▲ and ▼. When the power is turned on, the console waits 60 seconds to stabilize the circuit. This delay is needed to cancel out the magnetic field in the core and to let the signals settle down. If it is not a cold start, this warm up time is not necessary but it was implemented as a safeguard. The software was designed by following the user interface needs. Therefore, the user interface is described, which can be easily converted into codes.

The LCD is interfaced using an I2C bus, and the characters are simply transmitted serially. After the initial delay, the LCD display shows the following message.

```
Bulldog Counter
F1: Test Count
F2: Fresh Count
F3: Resume Count
```

Since one screen is not sufficient to display the entire menu. The ▼ key is used to display the rest of the menu items.

```
F4: Read Clock
F5: Set Clock
F6: PC Download
F7: Read Data
```

The user can press the ▼ key to toggle between the two screens. The seven functions corresponding to each function key (F#) are divided into three counting functions in the first screen and four utility routines in the second screen. From this main menu, the user can select the desired function. Next, the main functions are described.

Clock setting, F5

The real time clock function is powered by a super capacitor when the main power is off. This helps not to drain the battery when the system is not in use. However, the clock can last only few days after the power is turned off because the clock power is supplied only from the super capacitor. Consequently, clock setting routine had to be implemented.

The clock is set using the two arrow keys, i.e.,

▲ changes the value
▼ moves to the next item

The system first displays the real time value using a format MM/DD/YY where MM is a two-digit month display, DD is a two-digit day display, and YY is a two-digit year display. The arrow keys are used to set the date.

Next, the time is set using a format hh:mm:ss where hh is hour in 24 hour system, mm is a two-digit minute, and ss is a two-digit second. Arrow keys are again used to set the clock. At the end of the clock setting, it asks a confirmation using the following screen.

```
Set or Ignore
F1: Set Time
F2: Ignore
```

Pressing F1 sets the time and date the user entered, and pressing F2 ignores the settings the user just entered. After pressing one of these two keys, the screen returns to the main menu.

Reading Clock, F4

To read the internal clock, the user simply presses F4. It will then display the date and time in a standard format, e.g. Tuesday, June 7, 2007 with time 2:10 PM, 22 sec is displayed as:

```
Tus, 6/7/2007
14:10:22
```

The military 24-hour system is used for hour display. Pressing any key from this screen will bring the screen back to the main menu.

It is important to check the internal clock before any counting, since the clock looses time when the charge in the super capacitor runs out. If the user turns the power of the Bulldog counter on and off once every few days, the clock will maintain the proper time. Since this type of clock maintenance is inconvenient, if a real system is developed for field use, the clock should be directly powered by a battery rather than using a super capacitor.

Vehicle counting functions, F1, F2, F3

Three options of counting functions are available:

F1: Test Count ……. Counts vehicles but it does not store the data in the flash memory.

F2: Fresh Count …… Erases the old count data in the flash memory and then stores new vehicle counts. It stores 15 minutes counting interval to the flash memory. This data is not lost after turning off the power.

F3: Resume Count … Same as the Fresh Count except that it does not erase the old count data stored in the flash memory. It simply resumes from the previous count records.

During the counting operation, the following screen is shown:

```
Counting …
Tot:
15m:
```

In the screen, “Tot:” indicates the total (accumulated) counts since the one of the counting key is pressed and “15m:” indicates the last 15 minute count.

Pressing the F8 key ends the counting, and the screen returns to the Main menu.

Read the recorded Data (F7)

Press F7 to read the data stored in the flash memory. The display will show the following information.
Line 1: count start time
Line 2: total amount of hours and minutes counted
Line 3: total number of vehicles from the start to end of the duration
Line 4: count record up to 48 hours

“Count up to 48 hours” is the sum of 192 15-minute counts starting from the complete 15 minutes available.

PC Download (F6)

The count data stored in the flash memory can be downloaded to a PC using the serial port of the Bulldog. A serial cable is needed for this function.

3.3.2 Internal data format and downloading

Flash data is formatted in the flash memory using binary as follows:

EEEE Yr Mo Dy Hr MM SS	; 8 bytes for the start date and time stamp
#### #### #### …..	; after that all data are recorded as two bytes, ; representing counts for every 15 min ending time
EEEF Yr Mo Dy Hr MM SS	; at the end of the counting record ending time stamp ; is placed.

The block of begin/end pair of timestamps are repeated when multiple periods are recorded using the “F3: Resume Count” function.

Flash Memory Allocation

Count data with safe limit is allocated at the flash memory location A000 – DFFF. This space can store 16Kbytes of data which would allow 20 days of 15min data. Flash memory location FDFD and up are a flash lock area which is a reserved area.

Data download protocol

Data can be downloaded using Windows Hyper Terminal with the protocol described below.

PC sends “>”
Bulldog replies with “<” and display “PC Connected”

PC sends “[“ which is a symbol for request data
Bulldog sends ASCII coded Hex data from the flash

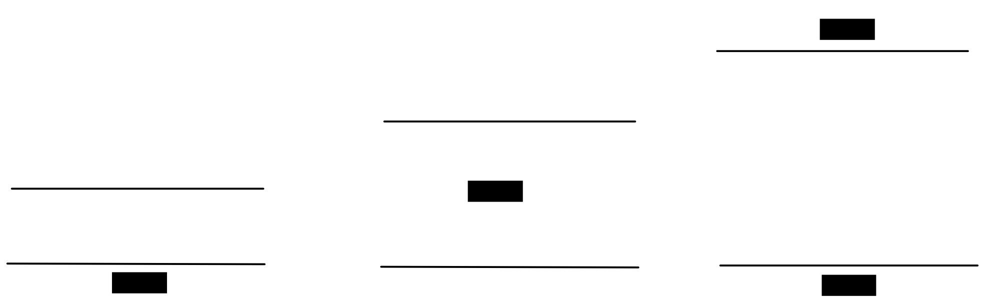
Bulldog sends “]” indicating the end of data transmission.

PC sends “!” indicating the end of the terminal session.
Bulldog sends “bye” and exits to the main menu.

This ends communication between the Bulldog and PC.

3.4 Installation on Gravel Roads

In a typical two-lane rural gravel road, the sensor probe can be buried on one side of the road as shown in Fig. 26.a. The sensor is waterproof and it can be easily buried. If the width of the road is large, the sensor can be buried in the middle of the road as shown in Fig. 26.b. If the width of the road is very large, two sensors could be placed as shown in Fig. 26.c. The sensor console could be tied to a tree or a pole.



a b c

Figure 26: Sensor probe placements

3.5 Road Test

A simple road test was conducted on July 29th, 2008. The gravel road chosen was the County Road 8 in Saint Louis County, Minnesota, which was recommended by the County traffic engineer. The test location is shown in Figure 27, which is located near the T-intersection between the Hanson Road and County Road 8.

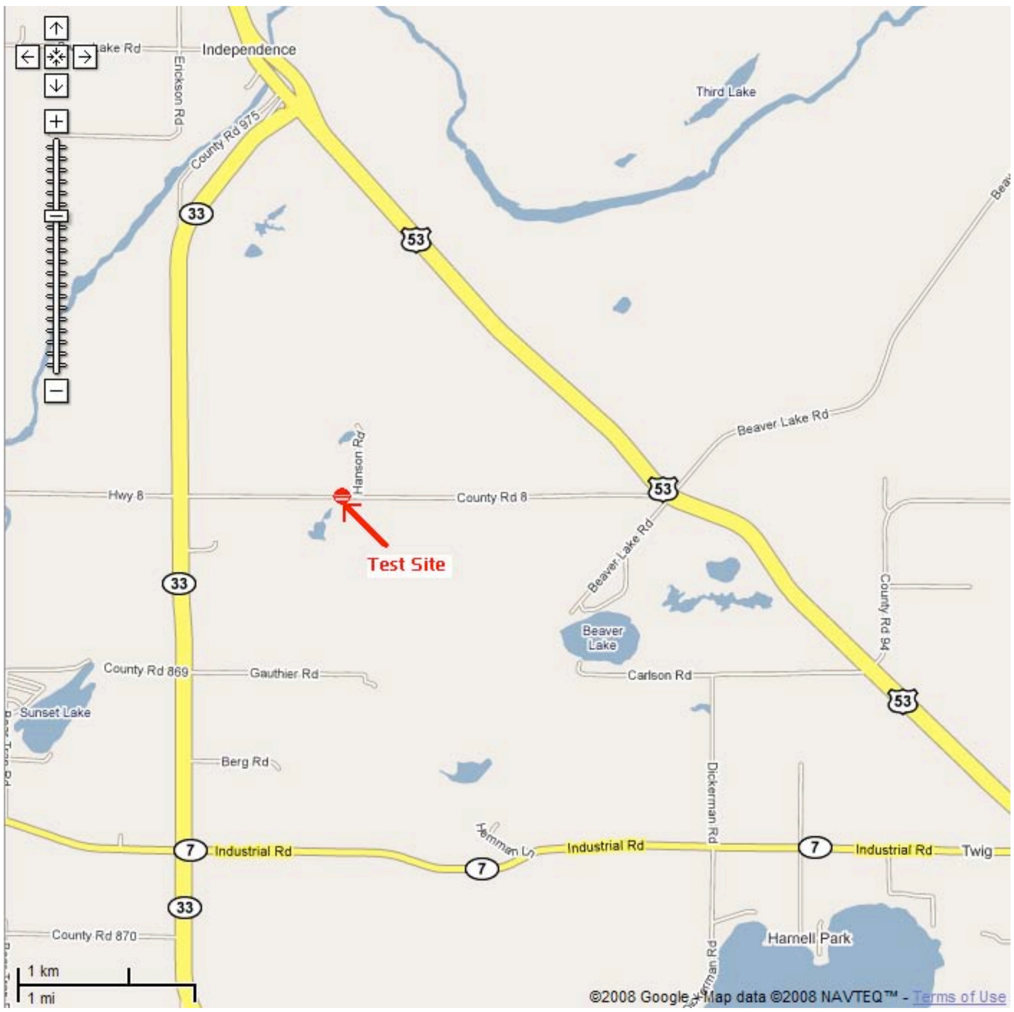


Figure 27: Gravel road test site at the County Rd 8.

County road 8 is a two lane gravel road, and the fluxgate sensor was installed on the side of the road as shown in Figure 28, which follows Figure 26.a configuration. Installation was simply done by digging about an inch deep two-foot channel and burying the sensor. The console was placed about 15 feet away from the sensor, which is shown on Figure 29.

Figure 28: Sensor buried on the side of the road.

Figure 29: Console was place about 15 feet from the sensor.

Figure 30: Manual counting of vehicles with two graduate students for verification.

For verification of counting, two graduate students manually counted the vehicles passed through (Figure 30). The manual counting was done for an hour from 1:15 – 2:15pm on July 29th, 2008 during which the Bulldog counter was turned on. The total number of vehicles passed through the counting area was six by both students, and the number recorded by the Bulldog counter was also six, meaning it accurately counted. Students were bored even within one hour of manual counting, perhaps due to such a low traffic volume on that road and monotonic nature of the task. The manual counting was discontinued after verification of correct count for one hour. Since this road is considered a high traffic gravel road within the St. Louis County (according to a County traffic engineer), the research team determined that tests on other lower volume road is not necessary.

Due to low vehicle counts, this test results cannot be used for the final performance statistics, but it was sufficient to verify its functionalities. All functions, which include counting, time stamping, data reading, and downloading, worked as designed. Another aspect observed was that its installation was simple and quick; it took less than five minutes.

In order to simulate animal and people crossing, students walked around the sensor during the counting time, but it did not trigger the counting. Students had cell phones in their pocket, but it did not affect the counting. Another test conducted was the effect of stopped vehicles. The question raised is, "what happens if a stopped vehicle is present right next to the sensor?" To answer this, a van was parked right next to the sensor and another vehicle was driven through. The result is that the Bulldog counter was able to ignore the stopped vehicle and counted the vehicle passed by. This capability was expected and is due to the following reason. Since the counting of a vehicle is done through detecting the disturbance of earth magnetic fields caused by a vehicle movement and the stopped vehicle only causes the disturbance during it is parking, which is counted as one, the counting of additional vehicles passing by is not affected. More specifically, the magnetic field quickly settles down once the vehicle stops, and after that it does not cause any further disturbance until another vehicle passes by.

3.6 Chapter Conclusion

The objective of this project was to find a vehicle counting technology that does not suffer from the problems associated with tube counters, i.e., air leak by puncture, sensitivity decrease by dirt, and inaccuracy caused by mismatch between the number of vehicles and number of axles. This chapter described development of a new gravel road counter which was designed using a fluxgate. The basic principle, microcontroller integration, system software design, and test results were presented. A prototype was designed and constructed, and basic tests were conducted. The test results showed that the proposed system is easy to install and works reliably. Since this technology is based on detection of magnetic fields, its counting ability was similar to that of the traditional inductive loop detectors. One difference from the loop detectors is that it is not affected by stopped vehicles near the sensor. If a stopped vehicle is present near the sensor and another vehicle passes by, the stopped vehicle was ignored and the vehicle passes by was counted.

If this system is to be commercialized, two improvements must be made. The console box used for the prototype was a plastic box, which is not strong enough for protecting from vandalism. It should be changed to a metal box that is similar to the console for tube counters. The second aspect needs to be improved is the signal processing algorithm. The algorithm sets a hard limit on the maximum signal disturbance period to one second, assuming signal disturbance does not last more than one second by a single vehicle. This assumption fails when a long truck passes by at a slow speed. The solution to this problem is setting the limit not from the beginning of the signal disturbance but from the end of the signal disturbance. Overall, these problems are minor and can be easily fixed. In conclusion, the tested fluxgate technology works well for vehicle counting in gravel roads and could readily replace the traditional tube counters.

CHAPTER 4: DLL TRAFFIC SOFTWARE DEVELOPMENT PACKAGE

4.1 Chapter Introduction

The TDRL's Data Center has been serving as a working data warehouse in which it acquires traffic and R/WIS data from Mn/DOT, archives them, and distributes them to the public through Internet. This data has been used by transportation research institutes, university researchers, and Mn/DOT. Recently, researchers from many universities and consultants requested software development tools for extracting traffic data from the archived traffic data. This project was created in response to those requests and to provide a software tool for developing applications of the archived traffic data.

Starting 2006, TDRL began developing Dynamically Linked Library (DLL) tools that can be directly embedded into applications, providing traffic data retrieval functions from the archived data. DLL is a software library technology which is linked to application programs when they are loaded or run rather than as the final phase of compilation. This means that the same block of library code can be shared between several tasks rather than each task containing copies of the routines it uses. The executable is compiled with a library of "stubs" which allow link errors to be detected at compile-time. Then, at run-time, either the system loader or the task's entry code must be arranged for library calls to be patched with the addresses of the real shared library routines, possibly via a jump table. Advanced users, mainly university researchers and consultants, have inquired TDRL on availability of DLL tools for retrieving data from archived files, and the development of DLL is thus geared towards those users.

4.2 DLL Design Objective

The Transportation Data Research Laboratory Software Development Kit, henceforth called TDRL SDK, is a software library package aimed at easing the process of writing traffic analysis software using *.traffic data file. It has been written to be used in any .NET application. The TDRL SDK has the following features:

- Hierarchical structure including objects to represent roads, stations, and single detectors.
- Fast search capabilities for single detectors.
- Volume and occupancy data retrieval at the road, station, or detector level.
- Automatic multi-day data retrieval, specified by start date and end date.

- Automatic generation of stations and road objects through road and station specification files.

- Custom exceptions thrown with detailed debugging information to assist in locating bugs.

- A full “Ndoc” documentation is provided to help within Visual Studio, as well as for an easy to use web interface.

4.3 TDRL SDK Design

As noted in Section 1, TDRL SDK uses a hierarchical structure to represent all of its elements. Station objects are built from many detector objects, and road objects are built from many station objects. This structure does not include the custom exception objects nor the datafile (i.e., *.traffic file) objects that must be used to extract data. Figure 31 depicts the structure of the library.

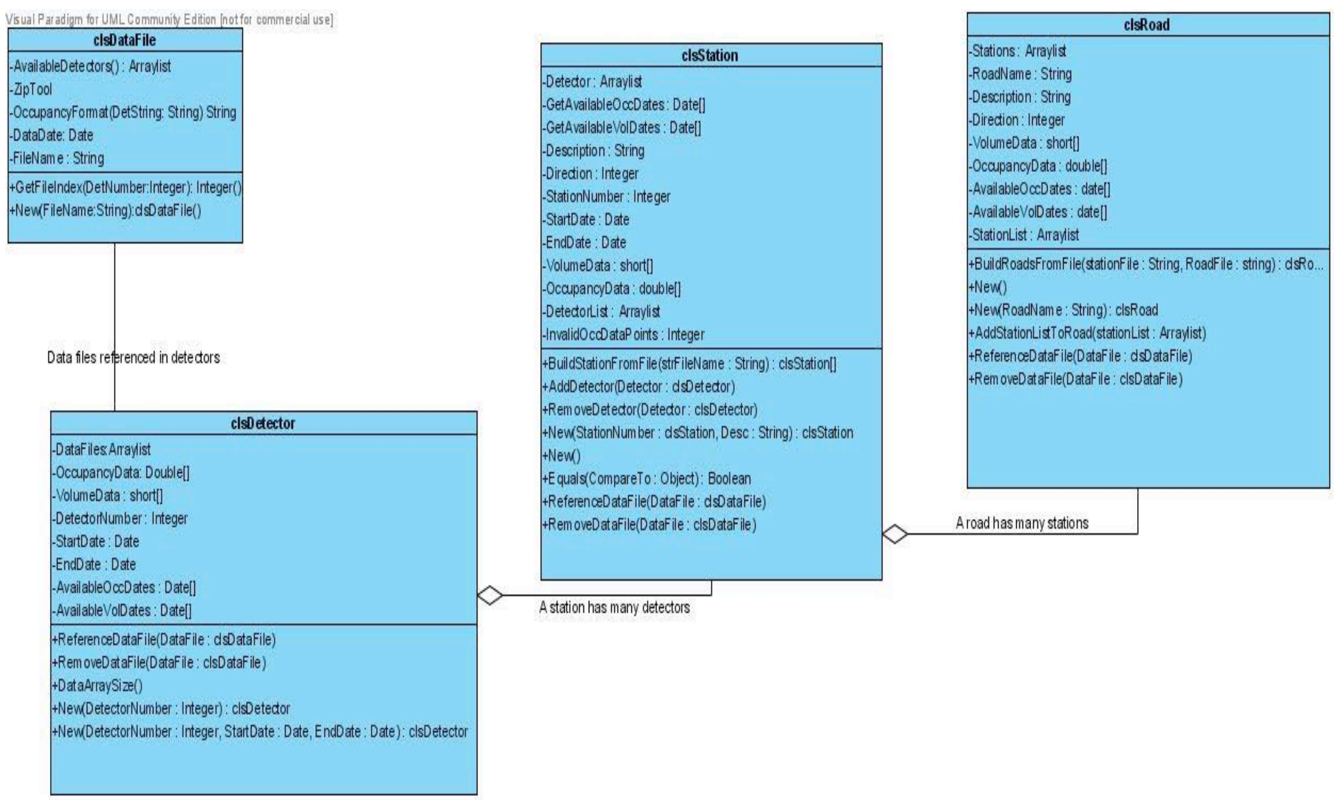


Figure 31: TDRL SDK class diagram.

As shown in the diagram, objects are built up from more primitive objects. Roads are built from stations, and stations are built from detectors. All data extraction is ultimately done at the detector level. That data is interpreted and presented at other levels. A description of each class follows.

clsDataFile - This class is perhaps the most important class of this library. It maintains a link to a datafile that is used for data extraction. It also keeps a list of detectors whose data is included in a particular datafile. Full documentation on this class may be found within the html SDK documentation.

clsDetector - The clsDetector class represents a single detector. The main ability for this object is the ability to extract data. There is also the ability for an object of this class to report a list of dates it knows are valid based upon the datafiles that have been referenced with it. Full documentation on this class and its members may be found within the html SDK documentation.

clsStation - The clsStation class is an encapsulation of multiple clsDetector objects. To add a detector, it can be added manually, or all detectors can be loaded automatically via text file specification. There is also a function to report available dates where data is known to exist for the station. Full documentation on this class and its members can be found within the html SDK documentation.

When data is retrieved from a station, the following rules apply:

- All detectors in a level must have at least a valid datafile referenced to them for the desired date (the data itself need not be completely valid).

- Volume data points are calculated by the sum of each individual data point from each detector in the level. For Example, if the station data point was for 3:10:30 PM, it would be the sum of each detector's 3:10:30 PM data point.

- If an invalid detector data point is found (-1 entry), the station object then falls back on secondary and tertiary detector levels to fill the data point. This utility provides spatial imputation. (This rule is applied to volume data only.). The detector levels are supported up to three levels: primary, secondary, and tertiary. The multi-level is defined and used for automatic replacement in case of failure in the upper level detectors. For example, if one of the primary detectors is faulty, the detectors are replaced with the secondary level. This technique has been used in the Mn/DOT short-duration and continuous count data [Kwon 2004].

- Occupancy data points are calculated by the means of all the individual data points from the detectors in a single level. Since a valid station result can occur even with some invalid detector data, a level number must be supplied from the user. A function is included that will determine the detector level with the least number of errors, and will produce the most accurate results.

clsRoad - The clsRoad class represents a single road. It is made up of multiple station objects. All functions called on the road class are reciprocated down to the station objects, and ultimately the detector objects. Stations can be added to a road in a couple of different ways. First, a function is included to add an array of stations to a road. Second, a function to build roads from a text file specification is also included. Third, stations may be added one at a time. When data is retrieved from a road, an arraylist data type of all the data from all stations is returned. Full documentation for the clsRoad class may be found in the html version of the TDRL SDK documentation.

clsMissingDataException - This exception is thrown whenever data is requested from a detector/station/road, and the corresponding datafile does not exist, or has not been referenced. A detailed debug message is included with this exception to help pinpoint where the error occurred.

clsCustomException - This exception is thrown whenever any other exception occurs. It will give a detailed description of where the error occurred. If possible, a short explanation of why the error occurred (and how to fix it) is given.

4.4 Using TDRL SDK

Setup for using TMC SDK is simple. The user first installs the provided TDRL_SDK.msi file (just by double click on the file). This will install example codes, manual and dll at the directory "Program Files\TDRL_SDK". After that all that is required is to make a reference to the TMC_SDK.dll file. This can be done in 3 easy steps:

- Navigate to the solution explorer in your project.
- Right click on the References icon, select Add Reference.
- Select the TDRL_SDK.dll file from the directory "Program Files\TDRL_SDK" to be added as a reference. Hit the Add button.

It is recommended to put the following line at the top of the user code:

```
Imports TDRL_SDK
```

The Imports statement will prevent codes from requiring a package specifier before declaring a TDRL_SDK object. Instructions on how to use the main features of TDRL SDK is described next.

4.4.1 Obtaining data from a road, station, or detector

There are two things that MUST be done before data can be extracted from any object. First, a clsDataFile object MUST be referenced with the detector/station/road object. This is how the object obtains data. Without a valid data file, a clsMissingDataException is thrown. Second, a start date and an end date including time must be provided. They are the beginning and end points for the data extraction. If there are multiple days between the start and end date, the data files for each date must be referenced. When data is retrieved, an array of type "short" is returned for volume and array type "double" is returned for occupancy. Short stands for short integer, and is a 16-bit signed integer.

4.4.2 Obtaining available dates from a road, station, or detector

When obtaining available dates for a given detector/station/road, simply call the corresponding "getAvailableDates" function (exact function names can be found the html SDK documentation). Keep in mind that in order for a date to be valid, a datafile must exist for that date. For accuracy, datafiles must be referenced before requesting available dates.

4.4.3 Building a list of stations via text specification

TDRL SDK allows automatic generation of station objects through a text file. An example of this text file is provided with the development package. To run the auto generate feature, simply call the clsStation.BuildStationsFromFile function. A String filename is all that is required. An arraylist of stations is returned.

4.4.4 Building a list of roads via text specification

TDRL SDK also allows the automatic generation of roads via text file. To do this, simply call the clsRoad.BuildRoadsFromFile. The return is an arraylist of clsRoad objects. Since roads contain more complex objects (stations) than stations do (detectors), it is required that both a road specification file and a station specification file are given, so that the program can automatically load all the stations that are to be added to the given road.

4.4.5 clsMissingDataException

This exception is only thrown when a data is requested, and the corresponding clsDataFile object doesn't exist. The user should check to be sure that a link to a .traffic file was created, and the clsDataFile object was referenced with detector/road/station object.

The Referencedatafile function only creates the reference if there data exists for that particular detector in the file. For example, if the detector number is 5000 and the data file has no data for detector 5000, the data file is simply not referenced. When this happens, the return of Referencedatafile is false.

4.4.6 clsCustomException

This exception can be thrown for any reason. For more predicable/common problems, a problem cause statement is included with the exception. The exception's DebugMessage property contains this explanation.

4.5 Chapter Conclusion

The TDRL SDK was written to provide a development tool for advanced users of traffic archive files. The DLL included was extensively tested before it was made available to public. With this tool, application developers and researchers can easily incorporate traffic data extraction functions in their codes. Example code slices were written and attached in the Appendix. Retrieving a large amount of station and road data can be made convenient by providing station and detector numbers through an input data file. Example codes are included in the sample code in the Appendix. In addition, a sample application called "AnalyzeThis!" is provided in that examples on how to use the TDRL SDK package can be further learned.

The DLL code was originally written by Dan Cinnamon, an Electrical and Computer Engineering student. Example code slices provided in the Appendix-B were written and tested by Lalit Nookala, a graduate student of Computer Science Department. All code examples are included in the TDRL SDK.

CHAPTER 6: DISCUSSIONS AND CONCLUSION

This report described the three projects initiated in Fiscal Year 2006 by TDRL. Each project produced tangible outcomes that are directly applicable to existing transportation applications. Conclusion of each project is summarized.

The first project involves development and field tests of a solar/wind hybrid power generator for rural ITS applications. A prototype power generator for a street lighting application was successfully built and tested for two years. The main hypothesis tested was, "Does wind power generation complement solar power generation or vice versa, providing more reliable power source?" It was found that most rainy or snowy days, which have limited solar radiation, indeed had stronger winds and generated more power from the wind turbine, complementing the lack of power generation by PV panels. In average, the PV panel generated 560Ah per day, and the wind turbine generated 576Ah per day, which indicates fairly well balanced power generation between two resources. The answer for the hypothesis was mostly "yes." However, there were also days that had no sun and no winds, generating very little electricity. The number of consecutive such days was very small. To counter act such days, a battery bank should always store access energy for use in such days. As a rule of thumb, if the battery bank can store up to seven days of power requirements, it was able to overcome the problem of no sun and no wind days for the lighting application. Experimental results from this research indicate that solar/wind hybrid power generation significantly improves the power reliability in comparison to the use of only one type of resource.

The second project involves the designing of a portable gravel road traffic counter. In gravel roads, tube counters often experience air leaks by puncture and sensitivity decrease by dirt accumulation, both of which cause inaccuracy in counting. As a solution to this problem, this project proposed a coil-wound fluxgate technology for detection of vehicles on gravel roads. A prototype vehicle counter was designed, constructed, and tested on the road. The test results showed that the proposed system is easy to install and works reliably. Since this technology is based on detection of magnetic fields, its counting ability was similar to that of the traditional inductive loop detectors but without the laborious installation. One difference from the loop detectors is that it is not affected by stopped vehicles near the sensor. If a stopped vehicle is present near the sensor and another vehicle passes by, the stopped vehicle was ignored and only the vehicle that passed by was counted. This characteristic works well for rural roads since tractors and vehicles can stop on the side of the road.

The third project was to develop a DLL-based traffic software development kit (SDK). Since the launch of the TDRL data center, researchers from many universities and consultants have requested software development tools for extracting traffic data from the archived traffic data. This project was created in response to those requests and to provide a powerful software tool for developing applications. An SDK package was successfully developed as a Dynamically Linked Library (DLL) that allows direct embedding into applications, and distributed through Internet.

In conclusion, the three projects completed provide practical solutions in three different areas of transportation, i.e., power infrastructure, data collection, and data processing. Although each project was designed only for a specific application, the underlying technology should be applicable to other types of transportation applications. For example, the hybrid power project was later evolved into a full independent project that provides renewable power for a dynamic message sign, and the fluxgate project was evolved into another full project that develops a vehicle tracking system.

For future study, a further research on applying earth's magnetic field to transportation is recommended. Earth's magnetic patterns can provide a three dimensional view of ferrous objects, which can provide information for vehicle classification, identification, and tracking. Earth's magnetic field patterns not only provide the signature of ferrous objects above ground, but it also provides a profile of the objects buried underground [8]. These powerful properties may have interesting future applications in developing transportation and security related sensors.

REFERENCES

[1] http://www.eere.energy.gov/windandhydro/windpoweringamerica/wind_maps.asp, visited 7/21/2008.

[2] GAO, "Federal and state efforts to address rural road safety challenges," GAO-04-663, Report to Congressional Committees, United States General Accounting Office, May, 2004.

[3] T. Maze and N. Hawkins, "Rural expressway intersection safety," CTRE, Iowa State University, *Tech. Transfer Summary*, Oct. 2004.

[4] Minnesota Department of Transportation, *2006 Signal Design Manual*, Office of Traffic, Safety and Operations, Mn/DOT, St. Paul, MN, 2006 (also available from http://www.dot.state.mn.us/trafficeng/signaldesign/index.html).

[5] National Weather Service Forecast Office http://www.weather.gov/climate/index.php?wfo=dlh

[6] T.M. Kwon, *TMC Traffic Data Automation for Mn/DOT's Traffic Monitoring Program*, Minnesota Department of Transportation, Report No. MN-RC-02004-29, St. Paul, MN, July, 2004.

[7] P. Ripka, Ed, *Magnetic Sensors and Magnetometers*, Artech House, Inc., Boston, MA, 2001.

[8] P.V. Czipott, "Stand-off detection and tracking of concealed weapons using magnetic tensor tracking," *Document No: 189583*, U.S. Department of Justice, Aug. 2001.

[9] F. PrimdAhl, "Bibliography of fluxgate magnetometers," *Publications of the Earth Physics Brach*, Vol. 41, No. 1, 1970.

[10] P.H., Serson and F. PrimdAhl, " Bibliography of magnetometers," *Publications of the Earth Physics Brach*, Vol. 43, No. 8, 1972.

[11] F. PrimdAhl and R.L. Coles, "A Bibliography of magnetometers," *Adv. Space Res.*, vol. 8, pp. 79-83, 1988.

APPENDIX-A: DAILY POWER GENERATION AND CONSUMPTION DATA

Date	Solar Ah	Wind Ah	Inverter Ah
11/1/2006	6.59	32.11	-41.77
11/2/2006	13.13	18.48	-41.17
11/3/2006	40.13	5.62	-43.8
11/4/2006	10.31	5.07	-45.14
11/5/2006	28.06	0.1	-45.48
11/6/2006	34.52	6.82	-47.71
11/7/2006	2.14	0.62	-47.68
11/8/2006	6.02	6.74	-45.58
11/9/2006	14.24	43.59	-44.53
11/10/2006	15.7	23.05	-37.14
11/11/2006	22.15	3.46	-18.26
11/12/2006	6.02	0.75	0
11/13/2006	3.06	0	0
11/14/2006	4.84	0.14	0
11/15/2006	2.43	0.96	0
11/16/2006	3.04	3.45	0
11/17/2006	4.22	8.82	-7.15
11/18/2006	6	14.58	-14.87
11/19/2006	39.49	7.23	-39.25
11/20/2006	24.6	2.45	-22.79
11/21/2006	26.26	4.96	-1.44
11/22/2006	42.09	0.7	-40.74
11/23/2006	29.12	27.34	-40.99
11/24/2006	38.02	6.57	-45.28
11/25/2006	37.46	6.48	-44.67
11/26/2006	2.32	11.5	-8.13
11/27/2006	0.92	22.28	-17.89
11/28/2006	0.33	61.03	-43.9
11/29/2006	39.15	37.09	-35.89
11/30/2006	42.63	3.64	-37.18
12/1/2006	8.1	13.57	-40.2
12/2/2006	26.67	28.77	-36.59
12/3/2006	30.58	54.4	-57.87
12/4/2006	8.19	16.81	-34.78
12/5/2006	2.44	4.93	-35.54
12/6/2006	31.19	57.17	-33.14
12/7/2006	39.91	19.7	-46.82

12/8/2006	27.49	11.09	-40.16
12/9/2006	42.62	18.97	-43.71
12/10/2006	35.5	9.92	-44.92
12/11/2006	5.12	19.67	-45.47
12/12/2006	1.64	0.01	-45.95
12/13/2006	22.68	37.04	-43.35
12/14/2006	7.5	46.64	-45.27
12/15/2006	3.62	34.91	-46.74
12/16/2006	4.62	2.06	-9.24
12/17/2006	6.05	17.47	-16.6
12/18/2006	40.77	12.24	-43.53
12/19/2006	42.38	38.41	-46.36
12/20/2006	22.68	12.47	-47.11
12/21/2006	2.8	9.95	-28.76
12/22/2006	1.2	19.45	-16.6
12/23/2006	15.33	40.31	-41.37
12/24/2006	21.16	17.05	-42.58
12/25/2006	41.59	30.59	-43.74
12/26/2006	29.01	12.05	-45.91
12/27/2006	18.5	12.55	-45.29
12/28/2006	0.97	37.27	-29.07
12/29/2006	1.61	0.78	0
12/30/2006	1.66	12.89	-9.6
12/31/2006	0.85	43.92	-30.89
1/1/2007	44.98	0	-45.41
1/2/2007	38.96	0	-38.65
1/3/2007	20.77	0	-19.88
1/4/2007	3.73	0	0
1/5/2007	1.58	0	0
1/6/2007	1.66	0	0
1/7/2007	2.88	0	0
1/8/2007	34.9	-0.01	-25.52
1/9/2007	34.01	-0.02	-26.29
1/10/2007	4.38	-0.02	0
1/11/2007	5.31	-0.02	-1.64
1/12/2007	19.07	-0.08	-10.88
1/13/2007	12.88	-0.02	-7.44
1/14/2007	34.63	-0.05	-29.19
1/15/2007	33.47	-0.05	-26.28
1/16/2007	45.03	-0.06	-34.29
1/17/2007	34.42	-0.03	-34.39
1/18/2007	6.66	-0.05	0
1/19/2007	45.52	-0.04	-36.23
1/20/2007	12.48	0	-11.37
1/21/2007	2.49	-0.05	0
1/22/2007	22.05	0	-16.97

1/23/2007	38.5	-0.01	-33.32
1/24/2007	14.75	-0.06	-9.14
1/25/2007	32.47	-0.07	-27.28
1/26/2007	9.51	0	-6.04
1/27/2007	36.5	-0.06	-31.51
1/28/2007	36.69	-0.05	-31.05
1/29/2007	5.49	-0.03	0
1/30/2007	18.42	0	-32.16
1/31/2007	33.93	-0.01	-36.75
2/1/2007	17.74	-0.01	-13.67
2/2/2007	27.85	-0.02	-18.73
2/3/2007	26.09	-0.18	-19.12
2/4/2007	35.38	0	-27.37
2/5/2007	39.14	-0.04	-25.32
2/6/2007	36.23	-0.08	-27.43
2/7/2007	48.13	-0.1	-26.08
2/8/2007	48.88	-0.04	-34.12
2/9/2007	48.32	-0.02	-27.47
2/10/2007	36.58	-0.04	-35.2
2/11/2007	31.71	-0.04	-41.65
2/12/2007	44.46	-0.03	-41.54
2/13/2007	50.31	-0.07	-38.36
2/14/2007	50.25	-0.05	-35.77
2/15/2007	49.86	-0.11	-37.14
2/16/2007	11.19	-0.02	-44.37
2/17/2007	54.3	-0.03	-32.96
2/18/2007	37.76	-0.06	-44.95
2/19/2007	16.56	-0.03	-34.82
2/20/2007	38.91	-0.01	-35.37
2/21/2007	0.02	-0.01	-6.88
2/22/2007	0	-0.03	0
2/23/2007	3.67	-0.04	0
2/24/2007	3.68	-0.23	0
2/25/2007	4.46	-0.04	0
2/26/2007	10.18	0	0
2/27/2007	13.19	-0.07	-8.91
2/28/2007	10.03	-0.01	0
3/1/2007	4.32	0	0
3/2/2007	10.77	-0.04	0
3/3/2007	59.29	-0.05	-28.18
3/4/2007	49.7	0	-40.96
3/5/2007	56.15	-0.05	-35.63
3/6/2007	10.93	-0.04	-38.99
3/7/2007	55.63	-0.04	-36.88
3/8/2007	8.36	-0.01	-24.76
3/9/2007	41.11	0	-31.84

3/10/2007	56.46	0	-45.8
3/11/2007	38.63	0	-37.27
3/12/2007	40.74	0	-49.08
3/13/2007	24.16	0	-41.88
3/14/2007	27.96	0	-23.18
3/15/2007	9.19	-0.04	0
3/16/2007	56.87	-0.01	-22.56
3/17/2007	56.13	-0.01	-41.58
3/18/2007	25.67	0	-45.03
3/19/2007	54.76	0	-33.28
3/20/2007	34.19	0	-42.37
3/21/2007	3.11	0	-12.89
3/22/2007	41.95	0	-25.78
3/23/2007	44.99	0	-41.3
3/24/2007	45.16	0	-44.4
3/25/2007	6.37	0	-20.19
3/26/2007	32.93	0	-25.26
3/27/2007	54.79	0	-35.17
3/28/2007	5.9	0	-23.96
3/29/2007	9.93	0	0
3/30/2007	3.21	0	0
3/31/2007	2.71	0	0
4/1/2007	4.09	0	0
4/2/2007	16.39	0	-9.01
4/3/2007	7.15	0	0
4/4/2007	29.93	0	-20.83
4/5/2007	55.98	0	-22.79
4/6/2007	46.64	0	-42.18
4/7/2007	42.92	0	-43.41
4/8/2007	54.37	0	-38.1
4/9/2007	53.13	0	-44.9
4/10/2007	38.18	0	-45.81
4/11/2007	25.36	0	-45.08
4/12/2007	49.77	0	-24.29
4/13/2007	44.87	0	-45.96
4/14/2007	42.05	0	-46.83
4/15/2007	50.89	0	-44.81
4/16/2007	48.35	0	-46.9
4/17/2007	50.79	0	-46.68
4/18/2007	50.44	0	-46.77
4/19/2007	49.19	0	-46.66
4/20/2007	46.74	0	-47.37
4/21/2007	8.96	0	-31.81
4/22/2007	16.41	0	0
4/23/2007	46.25	0	-22.19
4/24/2007	45.27	0	-46.77

4/25/2007	38.99	0	-47.33
4/26/2007	43.14	0	-34.92
4/27/2007	46.41	0	-38.64
4/28/2007	46.48	0	-45.42
4/29/2007	47.56	0	-45.39
4/30/2007	20.62	0	-42.04
5/1/2007	45.88	0	-20.84
5/2/2007	31.51	0	-41.74
5/3/2007	46.97	0	-25.29
5/4/2007	27.68	0	-43.54
5/5/2007	6.5	0	-2.09
5/6/2007	25.69	0	-21.2
5/7/2007	8.02	0	-3.44
5/8/2007	46.58	0	-19.89
5/9/2007	45.51	0	-45.65
5/10/2007	44.38	0	-46.34
5/11/2007	27.43	0	-47.18
5/12/2007	43.36	0	-20.38
5/13/2007	37.22	0	-36.07
5/14/2007	32.13	0	-33.04
5/15/2007	8.12	0	-7.54
5/16/2007	24.56	0	-20.14
5/17/2007	44.8	0	-19.86
5/18/2007	42.22	0	-42.65
5/19/2007	12.57	0	-33.05
5/20/2007	19.22	0	-10.71
5/21/2007	25.62	0	-19.31
5/22/2007	32.34	0	-20.82
5/23/2007	13.19	0	-10.7
5/24/2007	9.34	0	0
5/25/2007	42.54	0	-19.13
5/26/2007	9.37	0	-32.67
5/27/2007	30.28	0	-19.19
5/28/2007	25.65	0	-19.8
5/29/2007	4.5	0.07	-25.02
5/30/2007	31.06	0.13	-16.61
5/31/2007	30.62	0	-15.34
6/1/2007	37.9	0.11	-43.83
6/2/2007	9.3	0.61	-37.36
6/3/2007	15.68	3.66	0
6/4/2007	32.08	34.82	-16.72
6/5/2007	32.19	0	-43.52
6/6/2007	11.28	0.73	-45.21
6/7/2007	22.36	23.23	-20.02
6/8/2007	41.62	39.96	-43.19
6/9/2007	37.09	30.01	-41.73

6/10/2007	41.48	0.01	-42.15
6/11/2007	31.71	0.1	-42.11
6/12/2007	40.75	0.03	-41.22
6/13/2007	41.59	0	-41.06
6/14/2007	41.52	0	-41.21
6/15/2007	26.3	9.5	-41.02
6/16/2007	37.83	4	-41.56
6/17/2007	11.47	3.04	-42.37
6/18/2007	15.26	12.35	-44.08
6/19/2007	32.68	66.19	-32.67
6/20/2007	39.38	32.1	-41.58
6/21/2007	41.82	1.55	-40.88
6/22/2007	41.11	1.4	-41.1
6/23/2007	32.74	3.05	-41.95
6/24/2007	40.63	16.1	-40.48
6/25/2007	38.92	5.57	-40.08
6/26/2007	23.5	1.58	-40.06
6/27/2007	29.55	21.5	-41.1
6/28/2007	41.85	4.68	-42.52
6/29/2007	41.86	1.22	-41.51
6/30/2007	39.43	12.44	-41.79
7/1/2007	32.37	4.75	-42.56
7/2/2007	8.14	0.01	-44.67
7/3/2007	21.59	0	-41.32
7/4/2007	38.3	22.47	-41.5
7/5/2007	34.24	20.36	-41.65
7/6/2007	39.45	7.47	-41.25
7/7/2007	42.16	9.14	-41.03
7/8/2007	22.07	12.78	-40.73
7/9/2007	42.14	14.51	-41.78
7/10/2007	12.83	57.39	-42.82
7/11/2007	32.77	0	-42.25
7/12/2007	32.66	0	-42.69
7/13/2007	13.82	0	-42.92
7/14/2007	42.2	0	-42.38
7/15/2007	29.88	0	-42.95
7/16/2007	43.35	0	-42.61
7/17/2007	42.32	0	-42.72
7/18/2007	37.21	0	-42.36
7/19/2007	43.36	0	-43.14
7/20/2007	43.46	0	-43.54
7/21/2007	42.64	0	-43.67
7/22/2007	3.84	0	-44.28
7/23/2007	38.2	0	-3.45
7/24/2007	40.78	0	-15.11
7/25/2007	42.5	0	-42.53

7/26/2007	26.15	0	-42.52
7/27/2007	42.49	0	-44.22
7/28/2007	44.4	0	-42.97
7/29/2007	41.38	0	-43.47
7/30/2007	43.65	0	-43.24
7/31/2007	43.08	0	-43.37
8/1/2007	28.99	0	-43.45
8/2/2007	44.48	0	-41.59
8/3/2007	44.94	0	-41.5
8/4/2007	31.17	0	-43.41
8/5/2007	16.59	0	-23.9
8/6/2007	26.71	0	-18.37
8/7/2007	39.6	0	-22.91
8/8/2007	37.56	0	-37.86
8/9/2007	28.94	0	-16.22
8/10/2007	43.2	0	-18.16
8/11/2007	39.17	0	-48.29
8/12/2007	35.4	0	-45.13
8/13/2007	25.74	0	-33.93
8/14/2007	34.1	0	-21.34
8/15/2007	43.72	0	-35.7
8/16/2007	47.86	0	-45.12
8/17/2007	49.01	0	-45.1
8/18/2007	11.54	0	-34.33
8/19/2007	14.71	0	-10.13
8/20/2007	3.79	0	-1.47
8/21/2007	8.81	0	-0.12
8/22/2007	43.99	0	-25.43
8/23/2007	11.61	0	-34.62
8/24/2007	34.25	0	-22.2
8/25/2007	28.93	0	-7.81
8/26/2007	42.3	0	-22.02
8/27/2007	18.43	0	-47.12
8/28/2007	5.77	0	-15.59
8/29/2007	45.06	0	-23.56
8/30/2007	49.41	0	-38.44
8/31/2007	15.89	0	-36.78
9/1/2007	33.7	0	-23
9/2/2007	46.56	0	-31.14
9/3/2007	37.09	0	-46.45
9/4/2007	6.77	0	-13.84
9/5/2007	42.41	0	-23.33
9/6/2007	38.52	0	-18.06
9/7/2007	39.33	0	-24.71
9/8/2007	11.92	0	-47.88
9/9/2007	33.42	0	-25.98

9/10/2007	48.65	0	-30.26
9/11/2007	43.97	0	-45.52
9/12/2007	43.8	0	-38.3
9/13/2007	18.96	0	-32.73
9/14/2007	34.59	0	-26.72
9/15/2007	42.57	0	-28.45
9/16/2007	47.82	0	-38.91
9/17/2007	8.42	0	-34.02
9/18/2007	6.45	0	0
9/19/2007	15.32	0	0
9/20/2007	1.11	0	-10.55
9/21/2007	10.93	0	-9.78
9/22/2007	52.6	0	-27.02
9/23/2007	49.62	0	-44.28
9/24/2007	31.46	0	-47.56
9/25/2007	9.6	0	-20.56
9/26/2007	39.84	0	-29.25
9/27/2007	34.12	0	-35.39
9/28/2007	52.55	0	-31.1
9/29/2007	15.64	0	-36.09
9/30/2007	4.32	0	0
10/1/2007	9.58	0	0
10/2/2007	9.7	0	-14.19
10/3/2007	50.47	0	-29.15
10/4/2007	43.81	0	-45.07
10/5/2007	1.53	0	-15.47
10/6/2007	4.03	0	0
10/7/2007	12.86	0	-9.8
10/8/2007	4.6	0	-2.12
10/9/2007	7.45	0	0
10/10/2007	12.27	0	-9
10/11/2007	23.27	0	-18.27
10/12/2007	24.8	0	-20.13
10/13/2007	38.02	0	-33.11
10/14/2007	19.59	0	-18.01
10/15/2007	19.34	0	-16.2
10/16/2007	1.34	0	0
10/17/2007	3.86	0	0
10/18/2007	4.58	0	-3.09
10/19/2007	8.68	0	-5.48
10/20/2007	17.36	0	-14.91
10/21/2007	1.51	0	0
10/22/2007	42.45	0	-34.72
10/23/2007	10.62	0	-7.71
10/24/2007	51.4	0	-34.8
10/25/2007	51.83	0	-43.8

10/26/2007	51.25	0	-47.06
10/27/2007	49.38	0	-46.8
10/28/2007	41.12	0	-47.26
10/29/2007	46	0	-38.88
10/30/2007	40.48	0	-45.22
10/31/2007	25.53	0	-32.35
11/1/2007	39.27	0	-36.01
11/2/2007	41.81	0	-37.13
11/3/2007	38.37	0	-38.72
11/4/2007	36.63	0	-34.9
11/5/2007	25.04	0	-21.51
11/6/2007	5.61	0	0
11/7/2007	8.25	0	-4.82
11/8/2007	3.67	0	0
11/9/2007	3	0	0
11/10/2007	0.91	0	0
11/11/2007	26.07	0	-22.31
11/12/2007	37.75	11.44	-43.52
11/13/2007	3.8	57.07	-48.64
11/14/2007	1.68	79.03	-41.28
11/15/2007	6.9	34.96	-69.53
11/16/2007	4.56	1.7	-37.61
11/17/2007	13.97	10.14	-19.54
11/18/2007	3.17	2.05	-2.05
11/19/2007	3.69	9.53	-11.4
11/20/2007	4.67	15.41	-17.78
11/21/2007	2.14	50.64	-47.08
11/22/2007	48.26	18.6	-37.03
11/23/2007	14.18	0.59	-32.37
11/24/2007	8.51	34.39	-42.57
11/25/2007	16.72	28.14	-45.7
11/26/2007	7.59	14.81	-22.34
11/27/2007	20.22	51.99	-66.86
11/28/2007	5.01	25.38	-38.96
11/29/2007	22.25	19.36	-35.39
11/30/2007	11.97	8.67	-28.23
12/1/2007	1.43	31.38	-43.83
12/2/2007	3.6	73.64	-32.82
12/3/2007	4.89	1.61	-40.33
12/4/2007	2.77	23.41	-29.86
12/5/2007	36.45	20.64	-34.31
12/6/2007	1.46	0	-9.78
12/7/2007	34.49	0.01	-25.06
12/8/2007	37.92	-0.04	-25.63
12/9/2007	7.84	0	-7.63
12/10/2007	38.53	0.4	-38.13

12/11/2007	31.48	4.95	-37.42
12/12/2007	35.76	0.58	-36.85
12/13/2007	4.72	58.1	-33.31
12/14/2007	20.89	4.5	-28.88
12/15/2007	29.61	-0.02	-37.91
12/16/2007	28.74	10.95	-39.72
12/17/2007	31.23	3.82	-36.58
12/18/2007	4.3	0.15	-2.43
12/19/2007	2.04	0.1	0
12/20/2007	20.98	0.09	-19.1
12/21/2007	1.52	0	0
12/22/2007	2.33	45.7	-39.02
12/23/2007	0.02	175.35	-46.99
12/24/2007	31.76	8.87	-69.73
12/25/2007	3.05	0.93	-46.03
12/26/2007	3.59	3.04	-43.99
12/27/2007	5.72	51.48	-39.81
12/28/2007	5.17	-0.03	-12.07
12/29/2007	3.78	-0.06	0
12/30/2007	4.16	0	0
12/31/2007	20.69	1.46	-22.58
1/1/2008	20.95	70.4	-21.93
1/2/2008	25.31	39.04	-48.52
1/3/2008	1.73	31.48	-41.79
1/4/2008	19.03	-0.02	-45.39
1/5/2008	3.51	0	-44.58
1/6/2008	7.68	0.13	-14.25
1/7/2008	6.31	0	-5.52
1/8/2008	7.85	24.13	-32.84
1/9/2008	14.56	1.12	-10.77
1/10/2008	7.66	1.03	-5.72
1/11/2008	4.21	34.25	-34.86
1/12/2008	12.42	19.89	-29.2
1/13/2008	20.54	95.96	-33.45
1/14/2008	7.12	82.03	-51.78
1/15/2008	20.19	0.53	-41.87
1/16/2008	4.88	29.91	-39.44
1/17/2008	9.64	8.8	-46.72
1/18/2008	25.51	12.44	-23.29
1/19/2008	26.11	3.82	-27.31
1/20/2008	20.75	8.89	-27.64
1/21/2008	5.82	1.19	-25.34
1/22/2008	17.19	13.55	-24.58
1/23/2008	35.32	9.84	-34.8
1/24/2008	31.85	8.89	-27.06
1/25/2008	37.22	0.17	-41.33

1/26/2008	11.89	0.31	-43.02
1/27/2008	31.66	0.16	-45.52
1/28/2008	7.37	0.64	-15.36
1/29/2008	2.94	92.34	-20.76
1/30/2008	19.64	21.02	-43.56
1/31/2008	8.98	0	-25.31
2/1/2008	3.24	-0.03	-17.93
2/2/2008	4.53	-0.03	0
2/3/2008	21.77	-0.01	-32.4
2/4/2008	4.55	0.55	0
2/5/2008	17.33	38.24	-40.17
2/6/2008	7.5	9.52	-30.62
2/7/2008	20.09	0.34	-18.35
2/8/2008	8.87	-0.01	-5.99
2/9/2008	6.3	68.07	-20.52
2/10/2008	23.27	28.49	-36.2
2/11/2008	26.37	4.66	-30.81
2/12/2008	21.5	0.76	-34.45
2/13/2008	23.33	8.33	-35.61
2/14/2008	16.29	48.7	-27.85
2/15/2008	22.18	6.12	-24.55
2/16/2008	25.51	4.53	-43.91
2/17/2008	19.04	45.97	-41.55
2/18/2008	6.16	90.57	-45.83
2/19/2008	5.75	28.72	-28.99
2/20/2008	0.07	21.08	-30.4
2/21/2008	15.86	8.19	-35.27
2/22/2008	39.92	1.29	-38.61
2/23/2008	42.28	3.85	-42.91
2/24/2008	32.09	5.13	-43.72
2/25/2008	16.93	49.6	-39.74
2/26/2008	5	17.38	-39.88
2/27/2008	9.86	15.53	-39.56
2/28/2008	28.96	0.36	-42.84
2/29/2008	8.53	27.73	-39.47
3/1/2008	44.02	7.77	-41.25
3/2/2008	11.9	29.56	-44.8
3/3/2008	10.42	28.14	-48.48
3/4/2008	14.41	4.55	-42.59
3/5/2008	10.95	27.72	-38.22
3/6/2008	16.73	5.71	-29.58
3/7/2008	18.13	6.49	-27.77
3/8/2008	24.88	4.51	-37.91
3/9/2008	36.48	3.23	-34.33
3/10/2008	13.4	21.35	-43.38
3/11/2008	4.15	28.58	-44.75

3/12/2008	8.25	0.91	-45.71
3/13/2008	28.41	21.78	-44.44
3/14/2008	13.01	23.23	-44.37
3/15/2008	46.11	4.64	-43.09
3/16/2008	47.72	0.15	-43.33
3/17/2008	7.06	0	-46.18
3/18/2008	8.36	1.33	-45.28
3/19/2008	55.55	30.58	-44.29
3/20/2008	53.3	0.54	-43.99
3/21/2008	4.52	32.1	-43.5
3/22/2008	40.43	3.17	-44.85
3/23/2008	35.02	8.04	-42.12
3/24/2008	25.63	17.18	-43.63
3/25/2008	4.64	32.81	-42.84
3/26/2008	10.08	27.42	-43.85
3/27/2008	48.9	0	-42.83
3/28/2008	47.74	0	-43.67
3/29/2008	43.58	1.11	-45.72
3/30/2008	15.31	18.69	-45.47
3/31/2008	5.3	39.02	-43.37
4/1/2008	20.78	34.79	-56.46
4/2/2008	15.37	18.9	-45.8
4/3/2008	36.5	7.26	-44.22
4/4/2008	32.27	8.69	-45.37
4/5/2008	43.26	4.33	-44.97
4/6/2008	1.92	38.45	-44.51
4/7/2008	3.12	35.99	-42.95
4/8/2008	9.42	1.53	-45.08
4/9/2008	20.62	7.2	-45.52
4/10/2008	4.1	101.46	-37.59
4/11/2008	0.09	92.17	-38.65
4/12/2008	5.57	15.17	-38.14
4/13/2008	41.39	0.05	-39.96
4/14/2008	23.93	6.32	-42.75
4/15/2008	8.86	43.26	-44.49
4/16/2008	13.25	28.5	-44.7
4/17/2008	28.58	6.55	-44.35
4/18/2008	12.21	0.86	-45.65
4/19/2008	5.25	4.37	-46.09
4/20/2008	7.99	38.88	-44.53
4/21/2008	41.7	33.54	-46.21
4/22/2008	23.99	62.91	-48.69
4/23/2008	19.33	17.81	-44.39
4/24/2008	4.02	3.1	-46.32
4/25/2008	4.55	36.84	-44.4
4/26/2008	8.41	36.28	-42.45

4/27/2008	20.81	16.31	-43.69
4/28/2008	29.02	2.3	-44.96
4/29/2008	39	3.79	-45.33
4/30/2008	41.91	3.6	-46.27
5/1/2008	10.71	67.64	-42.61
5/2/2008	0.03	79.94	-47.3
5/3/2008	0.26	50.2	-41.84
5/4/2008	35.83	12.42	-44.29
5/5/2008	28.02	6.13	-44.89
5/6/2008	31.22	0.09	-43.17
5/7/2008	1.57	58.59	-44.54
5/8/2008	30.96	0.42	-44.64
5/9/2008	34.26	2.92	-44.29
5/10/2008	13.93	12.18	-46.2
5/11/2008	6.33	61.48	-44.99
5/12/2008	9.72	27.72	-45.91
5/13/2008	7.71	16.69	-44.24
5/14/2008	35.47	4.86	-44.26
5/15/2008	25.65	16.67	-45.21
5/16/2008	26.73	47.79	-43.67
5/17/2008	4.53	54.69	-44.17
5/18/2008	2.57	34.44	-44.17
5/19/2008	19.57	4.82	-44.13
5/20/2008	3.03	52.8	-44.04
5/21/2008	21.34	10.54	-43.47
5/22/2008	31.12	2.91	-42.98
5/23/2008	12.58	17.65	-43.52
5/24/2008	20.84	11.44	-43.25
5/25/2008	18.05	0.53	-44.36
5/26/2008	10.62	2.85	-44.65
5/27/2008	43.29	0	-43.14
5/28/2008	43.76	4.23	-42.26
5/29/2008	19.52	0.24	-44.29
5/30/2008	14.36	1.76	-43.28
5/31/2008	40.36	15.26	-41.92
6/1/2008	37.75	11.09	-41.26
6/2/2008	13.1	10.47	-41.9
6/3/2008	14.86	33.85	-43.45
6/4/2008	7.28	0.15	-44.09
6/5/2008	15.47	38.93	-44.86
6/6/2008	7.79	69.76	-44.98
6/7/2008	35.03	25.56	-41.24
6/8/2008	31.08	2.42	-42.99
6/9/2008	29.51	11.02	-43.04
6/10/2008	22.55	0.05	-32.88
6/11/2008	4.08	54.04	-43.7

6/12/2008	19.42	9.4	-42.5
6/13/2008	11.54	52.49	-42.12
6/14/2008	33.14	17.16	-41.7
6/15/2008	25.74	31.42	-41.43
6/16/2008	15.1	29.97	-41.92
6/17/2008	15.69	23.44	-40.88
6/18/2008	31.42	12.76	-40.92
6/19/2008	30.68	0.42	-40.4
6/20/2008	31.63	14.4	-40.42
6/21/2008	19.48	23.49	-40.46
6/22/2008	22.22	8.09	-40.91
6/23/2008	38.65	0.32	-40.57
6/24/2008	33.03	11.17	-40.31
6/25/2008	33.24	8.46	-40.13
6/26/2008	37.56	3.36	-39.17
6/27/2008	14.79	0.21	-40.98
6/28/2008	34.51	30.98	-42.47
6/29/2008	1.45	70.66	-40.85
6/30/2008	21.38	14.6	-39.89
7/1/2008	12.34	25.9	-40.41
7/2/2008	6.58	44.97	-39.44
7/3/2008	28.5	2.39	-40.42
7/4/2008	32.67	0.98	-40.68
7/5/2008	29.71	6.5	-41.01
7/6/2008	21.19	16.58	-40.75
7/7/2008	24.61	4.24	-41.09
7/8/2008	12.41	28.02	-41.54
7/9/2008	23.62	7.75	-42.47
7/10/2008	19.16	1.34	-41.72
7/11/2008	10.69	12.67	-47.49
7/12/2008	4.19	87.46	-35.97
7/13/2008	2.46	41.98	-42.19
7/14/2008	39.88	0.3	-40.89
7/15/2008	30.08	8.73	-41
7/16/2008	22.61	2.52	-41.58
7/17/2008	10.05	0.04	-42.35
7/18/2008	36.96	9.76	-42.49
7/19/2008	17.04	2.64	-42.98
7/20/2008	35.79	0.06	-43.66
7/21/2008	28.03	0.46	-43.59
7/22/2008	43.35	0.94	-43.31

APPENDIX-B: EXAMPLE CODE SLICES FOR USING TDRL SDK

Example 1: Extract the 30 sec volume and occupancy values for a single detector for one day. It is assumed that the data file is "20030618.traffic".

```
Dim datafile As New clsDataFile("20030618.traffic") 'declare the traffic file from which the data has to be extracted
Dim vol() As Short
Dim occ() As Double
Dim i As Integer, buf As String
Dim det As New clsDetector(100) ' declare the new detector with detector no. 100
det.ReferenceDataFile(datafile) 'reference the .traffic file from which the data has to be accessed
det.StartDate = #6/18/2003# 'set the start date and time
det.EndDate = #6/18/2003 11:59:59 PM# 'set the end date and time

vol = det.VolumeData 'retrieve the volume data for the detector
occ = det.OccupancyData 'retrieve the occupancy data for the detector

buf = "Volume and Occupancy of the detector on " + det.StartDate.ToShortDateString + vbCrLf
buf += "# ," + det.DetectorNumber.ToString + ".vol, " + det.DetectorNumber.ToString + ".occ" + vbCrLf
For i = 0 To vol.Length - 1
    buf += CStr(i + 1) + ".  " + CStr(vol(i)) + vbTab + ", " + occ(i).ToString("0.00") + vbCrLf
Next

If buf Is Nothing Then
       TextBox1.Text = "No data returned"
Else
       TextBox1.Text = buf
End If
```

Example 2: Extract the 30 sec volume and occupancy values for a single station for a single day. The station comprises of detectors 10, 100, and 1000.

```
Dim datafile As New clsDataFile("20030618.traffic") 'declare and instantiate the traffic file from which the data has to be extracted
Dim sta As New clsStation(555, "Test station") 'declare and instantiate the station for which the volume and occupancy data is extracted

Dim det1 As New clsDetector(10)
Dim det2 As New clsDetector(100)
Dim det3 As New clsDetector(1000)
'add one detector at a time to the station
sta.AddDetector(det1)
sta.AddDetector(det2)
sta.AddDetector(det3)
sta.ReferenceDataFile(datafile) 'reference the traffic data file for which the data has to be extracted.

sta.StartDate = #6/18/2003# 'set the start date and time for which the data has to be extracted
sta.EndDate = #6/18/2003 11:59:59 PM# 'set the end date and time for which the data has to be extracted

Dim vol() As Short
Dim occ() As Double

vol = sta.VolumeData 'retrieve the volume for the station
```

```
occ = sta.OccupancyData ' retrieve the occupancy for the station. Both arrays will
contain 2,880 values.
```

Example 3: Extract the 30 sec volume and occupancy values for a road for a single day. The station comprises of stations 111 (10, 100), 222 (20, 200), 333 (30, 300), and 444 (40, 400) where (#, #) are detectors of the station.

```
Dim vol As New ArrayList
Dim occ As New ArrayList
Dim i As Integer, buf As String
Dim stations As New ArrayList ' the station arraylist is required to build the road
Dim testRoad As New clsRoad(6)
Dim datafile As New clsDataFile("20030618.traffic") 'declare and instantiate the traffic
file from which the data has to be extracted

'declare all the detectors that will be used in building the stations that are present on
the road
Dim sta1Det1 As New clsDetector(10, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59 PM#))
Dim sta1Det2 As New clsDetector(100, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59
PM#))
Dim sta2Det1 As New clsDetector(20, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59 PM#))
Dim sta2Det2 As New clsDetector(200, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59
PM#))
Dim sta3Det1 As New clsDetector(30, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59 PM#))
Dim sta3Det2 As New clsDetector(300, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59
PM#))
Dim sta4Det1 As New clsDetector(40, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59 PM#))
Dim sta4Det2 As New clsDetector(400, CDate("#6/18/2003#"), CDate(#6/18/2003 11:59:59
PM#))

'Add stations to the arraylist of stations that will comprise the road
stations.Add(New clsStation(111, "TestStation1"))
stations.Add(New clsStation(222, "TestStation2"))
stations.Add(New clsStation(333, "TestStation3"))
stations.Add(New clsStation(444, "TestStation4"))

'Add detectors to each station
CType(stations(0), clsStation).AddDetector(sta1Det1)
CType(stations(0), clsStation).AddDetector(sta1Det2)
CType(stations(1), clsStation).AddDetector(sta2Det1)
CType(stations(1), clsStation).AddDetector(sta2Det2)
CType(stations(2), clsStation).AddDetector(sta3Det1)
CType(stations(2), clsStation).AddDetector(sta3Det2)
CType(stations(3), clsStation).AddDetector(sta4Det1)
CType(stations(3), clsStation).AddDetector(sta4Det2)

testRoad.AddStationListToRoad(stations) 'Add station list to the road
testRoad.ReferenceDataFile(datafile) ' reference the .traffic datafile
testRoad.StartDate() = CDate("#6/18/2003#") 'set the start date and time
testRoad.EndDate = CDate(#6/18/2003 11:59:59 PM#) 'set the end date and time
vol = testRoad.VolumeData 'retrieve the volume for the road
occ = testRoad.OccupancyData 'retrieve the occupancy for the road

The rest of the codes are simply printing the data
buf = "Volume and occupancy of the road on " + testRoad.StartDate.ToShortDateString +
vbCrLf
buf += "index,"
For i = 0 To stations.Count - 1
   If i = stations.Count - 1 Then
       buf += CType(stations(i), clsStation).StationNumber.ToString + ".vol," +
CType(stations(i), clsStation).StationNumber.ToString + ".occ"
   Else
       buf += CType(stations(i), clsStation).StationNumber.ToString + ".vol," +
CType(stations(i), clsStation).StationNumber.ToString + ".occ,"
   End If
Next

For j As Integer = 0 To vol(0).Length - 1
```

```
   For i = 0 To stations.Count - 1
       If i = 0 Then
          buf += vbCrLf + CStr(j + 1) + "," + CType(vol(i)(j), Short).ToString + "," +
CType(occ(i)(j), Double).ToString("0.00")
       Else
           buf += "," + CType(vol(i)(j), Short).ToString + "," + CType(occ(i)(j),
Double).ToString("0.00")
       End If
    Next
 Next


If buf Is Nothing Then
    TextBox1.Text = "No data returned"
Else
     Dim objFileWriter As New IO.StreamWriter("testRoad.csv")
     objFileWriter.Write(buf)
     objFileWriter.Flush()
     objFileWriter.Close()
     Shell("notepad " & Application.StartupPath & "\testRoad.csv")
            TextBox1.Text = "Data sent out to file 'testRoad.csv' "
 End If
```