2006-652: A BLUETOOTH-BASED HANDSET WIRELESS DATA ACQUISITION SYSTEM

David Border, Bowling Green State University

A Bluetooth™-based Handset Wireless Data Acquisition System

Abstract

An innovative data acquisition system that is suitable for laboratory work in electrical engineering/computer engineering communication coursework is detailed in this paper. The work makes use of currently available technologies including a Bluetooth™ module in the communication path, and a Windows Mobile 2003 PDA as the system handset. Such items illustrate important data acquisition and data communication elements that are being more commonly found in today's industrial environments. It is shown in the paper how the realization of key system elements spans a mix of hardware, firmware and software subcomponents. The standard elements of the project are discussed in the paper: the source, the channel and the receiver.

The analog source data stream for the project is generalized using multiple function generators to simulate source sensor output. A dedicated microprocessor assembles the digitized data according to the chosen transmission protocol. The protocol stream is transmitted from the microprocessor serial port to the serial port of a commercially available Bluetooth serial input module. A Bluetooth enabled PDA is used for reception and display of the acquired data. To ensure compatibility across a wide number of Windows Operating Systems, and possible extension of the work to a fully distributed and networked control environment, all PDA programming applications were created using the .NET Framework.

I. Introduction

Developers in industrial and consumer electronics along with faculty involved with research and teaching laboratory development are reaping the benefits of lower cost, greater complexity hardware, firmware, and software development tools. This advance is occurring along a broad front. This work focuses on one segment of this advance, handset based data acquisition devices.

The handset based data acquisition work represents a number of interesting technologies from recent electronic innovations. Such technologies are now being used in industry for wireless data logging, mobile data acquisition, and other applications. This work seeks to replicate a version of these concepts for use in a classroom laboratory. The chosen handset for the project is a Windows Mobile 2003 PDA device. The chosen communication channel for the project is wireless, with the transmission protocol being Bluetooth™. The data acquisition processor selected for use at the source was the Motorola MC9S12DT256 16-bit 25 MHz bus microcontroller unit (MCU). Laboratory function generators simulated the raw sensor devices.

While the span of the project is interesting, the variable depth of the project is equally interesting. The PDA Windows Mobile 2003 development environment used is the VisualStudio™, and .Net Framework™. The development environment supports a variety of programming languages, graphical user interface (GUI) forms, and objects. Extensive and complex with wide functionality, the environment permits the generation of end user applications that are lengthy and involved. When compared to the PDA, the depth of work for this project's Bluetooth serial module is not nearly as complex. The wireless communication code resident on the Bluetooth communication module supports configuration through an AT (modem style) command set. The project's MCU is from the widely known Motorola 68HC12 processor family and is currently supported at the application level by Freescale Semiconductor (formerly a division of Motorola). The family of processors comes with a variety of hardware features. The depth of work on this device is determined by the number of controller hardware features implemented and the development language chosen (e.g. assembler, C).

II. System Hardware Elements

For each part of the communication system, the source, channel and receiver, there exists a key hardware element that shares modest hardware commonality with the other elements. At the transmitter source is the MC9S12DT256 MCU. Its target market is the automotive and industrial control industries. The MCU is based on Motorola's 68HC12[1,2] microprocessor marketed first in 2000. The predecessors to the 68HC12 are the 68HC08 and 68HC11 processors. The MC9S12DT256 features the core cpu along with a variety of ancillary components on the chip, such as: ADC(s), asynchronous serial port(s) (SCI), Motorola sponsored synchronous serial peripheral interface(s) (SPI), Pulse Wide Modulation interface (PWM) plus others.

One of the Bluetooth[3] devices used to define the communications channel of the project was the ConnectBlue OEMSPA 13i serial module. It is mounted on a development kit board that allows for convenient prototype wiring. The module supports RS232 signal interfacing (TxD, RxD, plus handshaking) and direct UART signal interfacing (TxD, RxD, plus handshaking). The module has 64KB of SRAM and 512KB of flash. Resident on the device is a Bluetooth embedded host stack. The other Bluetooth module tested was the Ezurio module. It contains serial communication features similar to the ConnectBlue device.

The receiver is a PDA Windows Mobile 2003 device. It is used as the handset and it is a standard consumer electronics PDA. It includes Bluetooth capability and sufficient memory capacity and processor speeds to run graphical user interface code. Code for the PDA is developed offline using Microsoft VisualStudio, and .Net Framework. Code compatibility is insured through the use of Microsoft's Compact Framework.

III. Source

The MCU selected for this project is available on a number of hardware development modules. These development modules contain headers or pins, and/or small breadboards to facilitate prototype work. The development module manufactures include: Technological Arts, Axiom Manufacturing and the Wytec Company. These modules provide communication link(s) to PC workstations for the programming of MCU flash and for MCU code debugging. The resident PC applications for the support of MCU code development are referred to as Integrated Development Environments (IDE) and several of these exist for use with MCU development modules. Examples of these include: MCUezIDE, Metrowerks' Codewarrior™, and certain GNU Public Licensing based IDEs.

The data acquisition source made use of Freescale bundled hardware and software products. This bundle includes the Axiom Manufacturing's MCU development modules[4] and Metrowerks'

IDE Codewarrior. In addition to these items, the project made use of the National Instruments Educational Instrumentation Laboratory Suite (NI ELVIS™) compatible MCU Project Board[5,6,7] (Axiom Manufacturing). The MCU Project Board is pin compatible with the Axiom development module and provides an onboard USB interface Background Debug Mode (BDM) module. A BDM interface enables access to Motorola's built in hardware debugging mode present on a number of their processors families (68HCxx and others). On the 68HC12 processor the USB BDM allows realtime hardware probing of executing code, i.e. breakpoints, register and memory alteration and examination, and so forth. The BDM feature is transparently incorporated into the debugging phase of the Metrowerk's IDE. All flash programming for this project was done through the BDM interface.

Analog input to the MCU is provided through two 8 channel 10-bit ADCs. The device has 12 input pins. Eight (8) of the 12 pins are devoted to the 8 analog input signal channels. Two of the 12 pins are used to set the analog input voltage range, one pin denoting the high range value, and one pin denoting the low range value. The final two pins are assigned for ADC power, a +5 V DC supply input and a reference GND input.

The ADC is controlled using the following registers:

- ATDCTL2
- ATDCTL3 (sets debugging directive)
- ATDCTL4
- ATDCTL5

By using the above registers a C programmer can configure such features as:

- Set ATD power on/off
- Set a scan rate
- Read single channel 8 times or read each of the eight channels once
- Set continuous sampling or non continuous (sample and stop) sampling
- Vary bit resolution (10 bits or 8 bits)

During ADC operation improved conversion accuracy can be obtained by scanning each channel eight times and then averaging the values. This technique was implemented in the project.

Frequency generators were used to generate the simulated sensor signals. The project had the luxury of using multiple generators so generation of two, three or four unique waveforms was not difficult. In a laboratory classroom setting it would be expected that fewer generators would be available for each handset used. Under those circumstances the number of waveforms acquired could be reduced, or simple circuits (such as diode circuits) could be used to shape one frequency generator waveform into a number of differently shaped waveforms. The choice of waveforms is not critical to the overall project.

The selection of Metrowerks' Codewarrior for debugging and code development was not a difficult decision. Motorola has had a strong relationship with Metrowerks for a number of years. In a news release about their HC08 processor family (a predecessor to the HC12) they state that "CodeWarrior ANSI C compliant compiler was designed to take full advantage of the HC08 architecture, with wore[sic] than 60 advanced optimization strategies specifically designed to boost performance and reduce code size." As with the HC08, Metrowerks HC12 support features an optimized ANSI compliant compiler (a result of their extensive ANSI-C library porting effort), so C functions such as standard input/output (e.g. sprintf), and math functions (e.g. sqrt) are available for programming use. This allows for quick coding of MCU based operations, such as mathematical operations on the raw signal data and subsequent formatting (encapsulation) of data with header information into an ascii packet. The encoded data packet is then ready for shipment to the Bluetooth channel on a FIFO basis. Finally the strength of Metrowerks-Motorola tie is illustrated by the fact that as of October 2005, Freescale had acquired Metrowerks and is now marketing its products under the new name of “Freescale Developer Technology Organization.”

IV. Channel

It was desired to adopt a channel mode that has achieved quasi commodity status in the communications world. For this reason, the Bluetooth standard, and Bluetooth standard devices were chosen. Two modules were tested during project work, the Euzio (class 1) and ConnectBlue (class 2) Bluetooth modules. Both units are marketed as replacement for copper serial cable links. Both modules are mounted onto development kit boards that support voltage level shifting (3.3 V / TTL / RS-232), RS 232 with DB-9 connector and power adaptor. The Ezurio provides plated through holes to access the 40 pins of their Bluetooth module. The ConnectBlue version uses a 20 pin header mounted on the module board itself. The Ezurio and ConnectBlue modules are configured through a modem style AT terminal command set. The exact command sets differ since the command sets are proprietary to the manufacturer.

The Ezurio module has a number of unique interface pins on the module that were not needed for the project, these are: 8 GPIO lines, PCM lines, and 4-wire SPI bus lines. The Ezurio development kit’s featured two 8-bit channel ADCs were left unused.

The AT terminal command set is used to conditionally set a number of firmware resident parameters or to cause certain actions. A partial list follows:

- Master/Slave policy
- Security mode
- Read/Write local device name
- Read/Write serial port communication parameters
- Read/Write power save modes
- Read Bluetooth device address

For our purposes the Bluetooth module was operated in a trivial mode, with the device being visible to all Bluetooth enabled devices within range. Communication testing between the PDA and Bluetooth modules relies on simple troubleshooting techniques using the PDA Bluetooth manager application and hyperterminals (or hyperterminal-like) applications.

V. Receiver

At my home institution students are taught VisualStudio C within the context of their program studies. Therefore a Microsoft centric approach was followed for the receiver portion of the project. An available laboratory computer was used to host the VisualStudio and .Net Framework tools software. An advantage with the inclusion of the .NET framework is that the approach allows programmers to optionally augment their basic work by taking advantage of voluminous features native to .NET.

The Microsoft Visual Studio .NET development environment provides both a program code and graphical drag and drop toolset to ease custom GUI creation[8]. Like many development tools residing on a Microsoft OS, it is initiated by invoking a "new project" dialog. Following this a "windows application" style form is shown in an active workspace. This active workspace window allows the programmer to drag and drop all necessary labels, textboxes, buttons, system drawings and other GUI elements from the toolkit until the physical form shell is built. The basic function of the forms can be tested by use of the native PDA emulator.

The project GUI form for the PDA was chosen to simulate a basic data acquisition screen that would be appropriate for use in a factory setting. Common textbox objects were placed on the form to allow reporting of simulated temperature, pressure, and flow values. All parameters are read only. The program allowed for offline creation of operational profile entries for each of the parameters (points). The operational profile includes the following items:

Point Data Base

- Tag identifier (numeric)
- Tag name (10 characters)
- Point description (30 characters)
- Units
- Alarm High Value
- Alarm Low Value

The GUI form uses the data received via the Bluetooth serial port, along with parameters from the Point Data Base to provide an informative LCD display that shows the sensor values being acquired. A few other items are reported on the GUI also, they include: communication state (based on a watch dog timer), time of day, and configuration options.

The fundamental task of the receiver is to take serial data from the Bluetooth connection and parse it for use with the GUI form. The serial data connection is accomplished for the most part by use of an Open Software solution made available from OpenNETCF.org[9] a subsidiary of OpenNETCF Consulting, LLC. Regardless of the hardware interface used (USB, Bluetooth, Compact Flash, etc.) and regardless of the platform used (PDA, laptop etc.) this software library treats the read operation simply as a com port data read.

As is common to GUI based Visual Studio coding, OpenNETCF treats read (and send) tasks as system Events. The programmer establishes the read buffer content length that will trigger a read event. For the work in this project, a read threshold of 512 bytes was used. Following a read event, the read event handler forces the received buffer to be parsed (decoded) and populated into a data structure. The read event also triggers a conditional update to the GUI form. By employing this fundamental event driven architecture the LCD display is continuously updated with new numeric data.

VI. Material costs

The material costs for the data acquisition system can be viewed as dividing into three roughly equal items. The first item is the following product bundle: the Motorola microprocessor module

(Axiom), NI-ELVIS MCU development board (Axiom) and Codewarrior IDE (Metrowerks) products. These are bundled by Freescale and marketed through their University Program. The second item is the Bluetooth module and development kit board bundle. The third item is the PDA Windows Mobile 2003. On average, the cost of each of the describe items is about $225.00 (some items lower priced, some higher priced), for a total cost of approximately $675.00. The given materials list assumes that a PC is available to house the Metrowerks' IDE, VisualStudio and .NET Framework applications.

VII. Conclusion

The overall work done consists of a number of components suitable for junior/senior electronics laboratory investigations. These components are covered in curricular elements that already exist in undergraduate programs, e.g. microprocessors, analog-digital converters, serial communication, and programming languages. By casting elements together into a system (the Bluetooth-based handset wireless data acquisition system) the student is challenged to integrate various classroom skills. Providing that there are proper reference materials present (product documentation, programming examples, textbooks) the level of complexity within any given part of the system is subjectively rated as moderate. There exists broad support mechanisms for development work in these technologies, consider these particulars: (1) the existence of a large and active Motorola 68HC12 programming community, (2) the existence of a large community using VisualStudios and .NET technologies, and (3) the current plug and play nature of recently manufactured Bluetooth serial modules.

The data acquisition system has proven to be very reliable during operation. By relying on development kits, all semiconductor components get their power from reliable onboard power circuitry. The amount of field wiring required is limited to bringing the analog signals onto the MCU ADC input pins. Once all devices are turned on, operation is straightforward. The MCU is instantly on following a reset and the Bluetooth module is an always-on device. The PDA, like other Windows devices, generally provides solid operation but does occasionally needs to be given a cold restart to get it to work properly. In repeated trials, the source, channel and receiver worked successfully together, providing constant updates to the handset's display of realtime simulated sensor data.

Bibliography

1. Lipovski, G., Introduction to Microcontrollers, 2nd ed., Elsevier Academic Press Burlington Ma., 2004. ISBN 0-12-451838-9
2. D. Pack and S. Barrett, 68HC12 Microcontroller Theory and Applications, Upper Saddle River, NJ: Prentice Hall, 2002.
3. Haartsen, J. Bluetooth-the universal radio interface for ad hoc, wireless connectivity. Ericsson Review, vol.75, (no.3), L M Ericsson, 1998. p.110-17
4. CSM-12D Educational Module for Freescale MC9S12DX512 and MC9S12DT256 MCUs, Axiom Manufacturing, Garland, TX, July 2005
5. MCU Project Board, Prototyping Board with Microcontroller Interface, Axiom Manufacturing, Garland, TX, June 2004.
6. Montañez, E & Ruggles, S. Getting Started with the Microcontroller Student Learning Kit (MCUSLK),

Freescale Semiconductor, Inc., August 2004
7. Sibigtroth J., and Montañez, E., “Motor Control Demonstration Lab,” International Journal of Engineering Education, Tempus Publications, Dublin Inst Technology, Bolton St, Dublin, Ireland, Volume21, Number 1, 2005. ISSN: 0949-149X
8. Sells, C., Windows Forms Programming in C#, 4th ed., Addison-Wesley Pearson Education Boston Ma., May 2004. ISBN 0-321-11620-8
9. M. Monga, A. Scotto "A Generic Serializer for Mobile Devices", Proceedings of the 20th Annual ACM Symposium on Applied Computing, Santa Fe, New Mexico, USA, 2005