International Journal of Information Technology and Knowledge Management
July-December 2010, Volume 2, No. 2, pp. 461-463

SERIAL COMMUNICATION PROTOCOL FOR EMBEDDED APPLICATION

Gade S S[1], Kanase A B[2], Shendge S B[3], Uplane M D[4]

Serial communication protocol is developed for the small embedded application to have better communication in between personal computer and hardware module. The liability of this protocol is lies in its two different rules for data transmission and reception. The beauty of this protocol is that it reduces the code size of embedded application and most important is no data loss or collision. This method is support both half and full duplex mode. The protocol has been developed and tested in laboratory.

Keywords: Data Format, Embedded Systems, Frame, Microcontrollers, Microprocessors, Serial Communication.

1. Introduction

An embedded system is hidden electronic circuitry present in electronic product. Embedded system is also called as dedicated controllers. Most of the consumer electronics products are consists of embedded controller, memory module and other useful peripherals depending on its application [1]. The firmware which is running on embedded platform has special architecture. The software architecture is selected on the basis of nature of application and system complexity. The RTOS is good solution for this type of development [2, 3].

In most of time there is necessity of interfacing embedded module with personal computer. This is the situation where developer has to concentrate on communication protocol; this means that a set of rule is to be followed by in between application program and firmware in personal computer and embedded system respectively [4].So as to have good data transfer between personal computer and embedded module a technique is adopted which is the part of this proceeding work.

2. System Block Diagram

The hardware module has been designed and tested in the laboratory for the embedded application of on line auto tuning of PID controller using successive approximation method [5]. This application has two major functioning parts one is hardware and second is application program running on windows platform. The software needed for this application is designed as per the system requirement and a new set of rule is created to have better communication in between these two and various results are also compared [6]. Block diagram of the hardware of developed system is shown in the figure 1.

Hardware

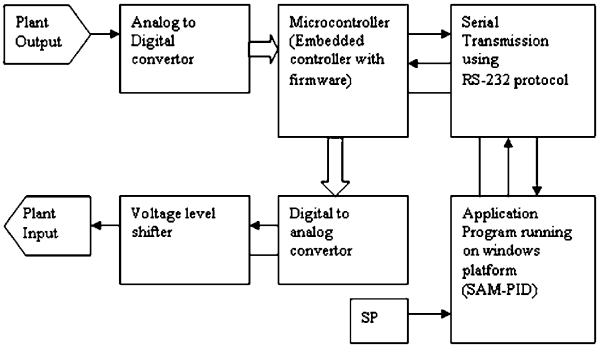

Fig. 1: Block Diagram of Hardware Module

Software

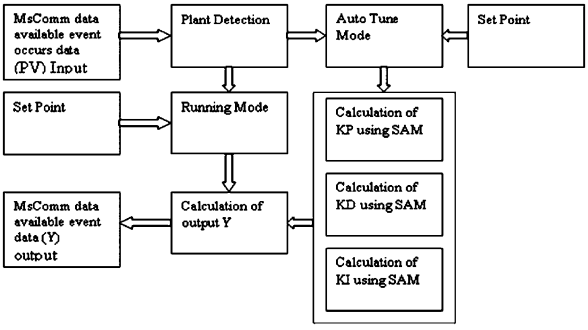

Fig. 2: Data Flow in Application Program

[1]E&TC Department of Annasaheb Dange College of Engineering, Ashta Sangli Maharastra

[2]E&TC Department of Karmyogi Poly-technique College, Pandharpur, Dis : Solapur Maharastra

[3]Computer Department of PVPIT, Budggaon Sangli Maharastra

[4]Electronics Department of Shivaji University, Kolhapur Maharastra

Email: [1]Sachin_gade123@yahoo.co.in, [2]Ajeetkanase@yahoo.co.in, [3]Sangam333@gmail.com, [4]Uplane_suk@hotmail.com

The hardware structure in figure 1 shows the various circuitries are connected to meet the desired application [5, 6].

The embedded application is designed in such a way that it can communicate with application program on personal computer. The data flow is shown in figure 2 [5, 6]. The serial communication is build using MSComm visual module of VB (visual basic) tool set.

3. MSCOMM MODULE

When a serial port is opened, the program creates receive and transmit buffers. To manage these buffers, the communications control has a number of properties that is set at design time using the control's property pages. For example, it's probably not a good idea to have a communications event (an OnComm event) for every byte read; instead, the RThreshold property to the number of bytes is set to read before triggering that event. The communications control's buffer management properties are InBufferSize, OutBufferSize, RThreshold, SThreshold, InputLen, and EOFEnable [7].

Visual Basic 6.0 Module

In VB 6.0 port opening and closing as well as reading and writing to the port using property looks simple. The next version of this tool set is dot net framework [7, 8].

Dot Net Module

In visual studio 2005 express edition the serial port communication module is treated as separate thread. This programming defined with some threading technology and as a result of this threading there is necessity to synchronise two threads with each other [8]. Thread synchronisation takes some time as compared with its earlier VB 6.0 module; hence a technique is adopted to solve this problem.

Updating

Fig. 3: Updating Window

First create new project with only serial communication module in VB 6.0 and after that open this project in visual studio 2005 then the window will appear as shown in figure 3 and click next and follow on screen command to update the code from VB 6.0 to visual studio 2005 [8]. This updated project has property that synchronisation of thread in case of serial port is not required and hence transfers become faster than threading technique in VB 2005.

4. SERIAL COMMUNICATION EMBEDDED PROTOCOL

A communication protocol is developed in order to have faithful data transfer between hardware module and personal computer. Communication protocol has two important software module that is application and firmware control running in personal computer and hardware module respectively.

Application Control

Application program is running on windows platform has very simple read operation using MSComm read property and event evoked by serial port. The property settings are as follows

Dock	None
DTREnable	False
EOFEnable	False
GenerateMeml	True
Handshaking	comNone
InBufferSize	1024
InputLen	0
InputMode	comInputMode
⊞ Location	768, 470
Locked	False
⊞ Margin	3, 3, 3, 3
⊞ MaximumSize	0, 0
⊞ MinimumSize	0, 0
Modifiers	Public
NullDiscard	False
OutBufferSize	512
⊞ Padding	0, 0, 0, 0
ParityReplace	?
RThreshold	1
RTSEnable	False
Settings	2400,n,8,1
⊞ Size	38, 38
SThreshold	0

A frame of seven byte is created which consists of CRLF (2-byte) + command (1 byte) + data (4 byte). The frame is purposely kept short for faster data transfer from hardware module to personal computer. Data receive protocol is simple and easy to use. According to command byte at max 255 different type of command can be transmitted by hardware module with each having data of four byte.

Byte by byte protocol is adopted for the data transfer from personal computer to hardware module. This technique is adopted due to slow speed of microcontroller as compared with desktop processor [9, 10]. The routine will send number of bytes, there is no restriction of byte, but first two byte is the count depending on the number of data bytes. After successful receiving of byte hardware module sends acknowledgement confers that byte is received hence there is no chance of data missing or collision. This technique improves the overall throughput of embedded application.

Firmware Control

The firmware is designed with the seven byte frame for the data transfer from hardware to personal computer. The transmission is very simple and hence no extra code is required to handle complicated frame which saves code memory and increases throughput. According to command byte at max 255 different type of command can be transmitted by hardware module with each having data of four byte.

The firmware handles the receiving byte whenever it requires. The serial communication is not handled using interrupt because receiving and transmission event both of having same vector address in MCS-51 family microcontroller [9]. Hence firmware routine and application routine is synchronized during software development. Some care has to taken at development time but this technique improves speed.

5. CONCLUSION

The serial communication protocol is very simple to implement. It is collision free, no data loss occurred and hence easy to use in slow speed microcontroller. The routine developed for firmware to handle serial communication has optimum code length which saves code memory. This protocol has different rule for receiving and transmission and hence in full duplex mode it has better speed. This serial communication protocol is suitable for small embedded applications.

REFERENCES

[1] IEEE Paper on “Software Architectures and Embedded Systems” Nenad Medvidovic Sam Malek Marija Mikic-Rakic.

[2] Gade S S, Mujawar I. I., “Embedded System Software Design”, National Level Paper Presentation at COP, Pandharpur 2008.

[3] Gade S S, Shendge S B, “RTOS Design in Embedded Systems”, National Conference at Nagpur NCOAT NIRMITI 2009.

[4] IEEE Paper on “A Wireless Embedded Sensor Architecture for System-level Optimization” Jason Hill and David Culler.

[5] Gade S S, Shendge S B, Uplane M D, “On line Auto Tuning of PID Controller Using Successive Approximation Method,” Accepted for to be Publish in IEEE Xplore, International Conference on 12th – 13th March, ITC-2010 to be Held at Cochin.

[6] Gade S S, Shendge S B, Uplane M D, “Performance Comparison of Auto Tune PID Controller with Conventional PID Controller,” Accepted for to be Publish in International Journal IJCSC.

[7] Visual Basic 6 Black Book, (Publisher: The Coriolis Group), Author(s): Steven Holzner, ISBN: 1576102831, Publication Date: 08/01/98.

[8] Visual Basic .net Programming, (Publisher: The Coriolis Group), Author(s): Steven Holzner.

[9] Intel Datasheet of MCS-51 Family.

[10] Intel Datasheet of Pentium 4 Processor.