

Embedded Internet based system

Ognyan Dimitrov, Ferhan Balpinar, Nina Bencheva, Yoana Ruseva

Embedded Internet based system: *A complete solution of an Internet based embedded system is proposed. The PIC18F67J60 microcontroller is implemented because of its built-in Ethernet controller module. The developed software is based on the Microchip TCP/IP Stack which is modified for the specific application. The system is accessible from users connected to the local network as well as via the Internet. Events that have happened in digital inputs cause sending e-mails or SMSs to predetermined addresses. The system is open for enhancing its features.*

Key words: *Microchip, Microcontroller, Embedded system, PIC18F67J60, FLASH, RAM, TCP/IP Stack, protocol, Ethernet controller.*

INTRODUCTION

The networking of embedded system solutions via the Internet is becoming increasingly important. There are many different fields of application, ranging from remote maintenance and the servicing of distributed monitoring and control systems to automated updates via the Internet.

To connect the microcontroller to the Internet, it is necessary to implement a protocol stack on the device. To function, this protocol usually requires a powerful processor, a complete operating system, and a large amount of memory. In this case, the implementation of a stand-alone Internet protocol stack could be a solution. According to market analysis, more than 75% of the embedded applications use a proprietary or free operating system, about 25% do not use an operating system. Big operating systems usually have Internet protocols already built-in, whereas deeply embedded systems do not yet provide a TCP/IP (Transmission Control Protocol/Internet Protocol) stack. However, a 8-bit or 16-bit microcontroller may not have enough resources for the implementation of an operating system. Software implementations without operating systems can be found in less complex applications often using small microcontrollers with strict hardware and runtime restrictions. The choice of writing standalone code is often made because of the need to optimize memory usage, code size, and run-time behavior [2].

HARDWARE IMPLEMENTATION

One embedded microcontroller system with Internet connectivity is developed and implemented by TERACOM Company. In Fig. 1 the block diagram of the system is shown. It is based on the PIC18F67J60 microcontroller. Example applications that can take advantage of the PIC18F67J60 Ethernet controller include: telecommunications (VoIP phone adapters); inventory management (vending machines, hotel mini bars); remote diagnostics/alerts (appliances, factory machines, POS terminals, power supplies, servers/networks); security (asset monitoring, fire and safety, security panels, access control, fingerprint recognition); remote sensing/actuators (industrial control/automation, lighting control, environmental control). PIC18F67J60 has 128K bytes program memory and 3808 K bytes static RAM [4]. It features an embedded Ethernet controller module. This is a complete connectivity solution, including full implementations of both Media Access Control (MAC) and Physical Layer transceiver (PHY) modules. The Ethernet module incorporates a number of packet filtering schemes to limit incoming packets. It also provides an internal DMA module for fast data throughput and hardware assisted IP checksum calculations. The Ethernet module consists of five major functional blocks: a PHY transceiver module, a MAC module, an 8-Kbyte RAM buffer (divided into separate receiver and transmitter buffer spaces), an arbiter to control access to the RAM buffer and a register interface. The module provides programmable: automatic retransmission on collision, padding and CRC generation and automatic rejection of erroneous packets. The

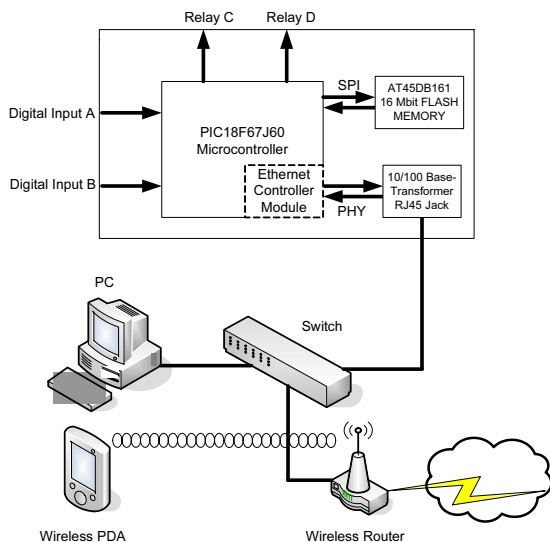


Fig. 1. Block diagram of the system

does not support Automatic Duplex mode negotiation. If it is connected to an automatic duplex negotiation enabled network switch or Ethernet controller, the module will be detected as a half-duplex device. To communicate in full duplex, the module and the remote node (switch, router or Ethernet controller) must be manually configured for full-duplex operation. As there is not a built-in EEPROM in PIC 18F67J60 for the information storage, an external memory is used. The AT45DB161D is utilized in the proposed embedded system. It is a 16 Mbit serial-interface sequential access Flash memory [1]. The AT45DB161D supports RapidS serial interface for applications requiring very high speed operations. RapidS serial interface is SPI compatible for frequencies up to 66 MHz. The memory is organized as 4,096 pages of 512 bytes or 528 bytes each. In addition to the main memory, the AT45DB161D also contains two SRAM buffers of 512/528 bytes each. In this external memory the system configuration data are programmed. It is necessary to generate any Web pages in advance and convert them into a compatible format (MPFS) for storage. If the MPFS image is to be stored in an external memory, a programming method may need to be included in the application, especially if the content is expected to change.

SOFTWARE IMPLEMENTATION

Software, based on an architecture referred to as the "TCP/IP Reference model", is divided into multiple layers, where layers are stacked on top of each other and each layer accesses services from one or more layers directly below it. Many of the TCP/IP layers do act not only when a service is requested but also when events like time-out or a new packet arrival occur. Unlike the TCP/IP reference model, many of the layers in the Microchip TCP/IP Stack directly access one or more layers which are not directly below it. A system with plenty of data memory and program memory can easily incorporate these requirements. A multitasking operating system may provide extra facility and therefore, may make implementation modular. But the task becomes difficult when an 8-bit microcontroller with limited RAM and program memory is used. In addition, without access to a multitasking operating system, the user must pay special attention to make the stack

PHY module provides separate outputs to drive the standard Ethernet indicators, LEDA and LEDB. The LED outputs are multiplexed with PORTA pins RA0 and RA1. The Ethernet module meets all of the IEEE 802.3 specifications for 10-BaseT connectivity to a twisted-pair network. PIC 18F67J60 provides a dedicated 4-pin signal interface for the Ethernet module. To complete the Ethernet interface, the Ethernet module requires several standard components to be installed externally. On the TPIN+/TPIN- and TPOUT+/TPOUT- pins, 1:1 center-tapped pulse transformers rated for Ethernet operations are required. In this instance CJ series 10/100 base transformer is used [3]. The Ethernet module

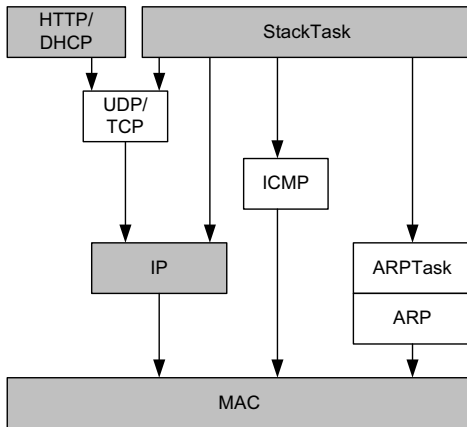


Fig. 2. Microchip TCP/IP Stack

independent of the main application. The developed software is based on the Microchip TCP/IP Stack (Fig. 2) designed to run on Microchip's PIC18 family of microcontrollers [5].

The shaded blocks are modified in accordance with the requirements of the particular application. The stack is written in the 'C' programming language.

Media Access Control Layer (MAC)

The user can specify the sizes for the transmit buffer, transmitting queue and receiving queue using 'C' defines. The additional feature, which has been developed, is MAC filtration. The administrator can add, activate and deactivate up to 8 MAC addresses. The screen of the administration programme

part (Setup) is shown in Fig.3. This part of the program is not accessible for the other users. If there is an error in the MAC frame (byte numbers), the Microchip stack causes a Reset. Such random reset can cause unexpected reactions or damages to the external modules connected to the microcontroller ports. In order to avoid this, the wrong packet is discarded. In the Setup module an additional function for verification of power-on-reset (POR) is included. This function allows multiple updating and saving the configuration data of the system (for example, adding new MAC addresses) without repeating POR.

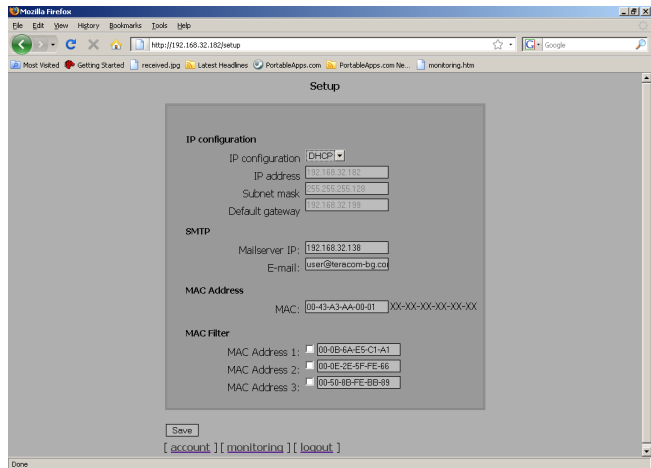


Fig. 3. Dialog window of the Setup programme module

Internet Protocol

(IP) In this architecture, the IP layer is passive; it does not respond to IP data packets. Instead, higher level layers use IP primitives and fetch the IP packet, interpret it and take appropriate action. The IP specification requires that the local host generate a unique packet identifier for each packet transmitted by it. The identifier allows the remote host to identify duplicate packets and discard them. The Microchip TCP/IP Stack's IP layer maintains a private 16-bit variable to track packet identifiers. The open source of the Microchip stack does not check the IP addresses. In the proposed system a new function is implemented giving the possibility to the administrator to allow or to forbid an external Internet accessibility.

The HTTP Server is implemented as a cooperative task that co-exists with the Microchip TCP/IP Stack and the user's main application. The HTTP Server also maintains a list of file types that it supports. It uses this information to advise a remote browser on how to interpret a particular file, based on the file's three-letter extension. By default, the Microchip HTTP Server supports ".txt", ".htm" and ".cgi" files. The possibilities for file transfer are extended and the system can exchange various graphical formats. When there is a request from the browser the microcontroller sends the content type and the size of the file. This is an additional feature which does not exist in the original Microchip stack.

Dynamic Host Configuration Protocol (DHCP) The DHCP module is implemented as a cooperative task, performing automatic operations without the involvement of the main application. The actual DHCP integration and control is done by the Stack Manager; it handles all required operations as part of its standard task, using the DHCP APIs to control the module's behavior. DHCP is called only from the Setup module. The protocol TELNET, sitting in the application layer, is removed because it is not used by the system. The same is done for the PPP protocol.

StackTask In order to relieve the main application from the burden of managing the individual modules, the Microchip TCP/IP Stack uses a special application layer module known as "StackTask", or the Stack Manager. StackTask is implemented as a cooperative task; when given processing time, it polls the MAC layer for valid data packets. When one is received, it decodes it and routes it to the appropriate module for further processing. The timing system that Microchip has employed for their TCP/IP stack allows timer interruptions up to 100 ms. The modification made extends the timing value up to several minutes. This is necessary for the user time out – for example, when the user enters the password and there is no other action in 2 minutes, it is obligatory to enter the password again.

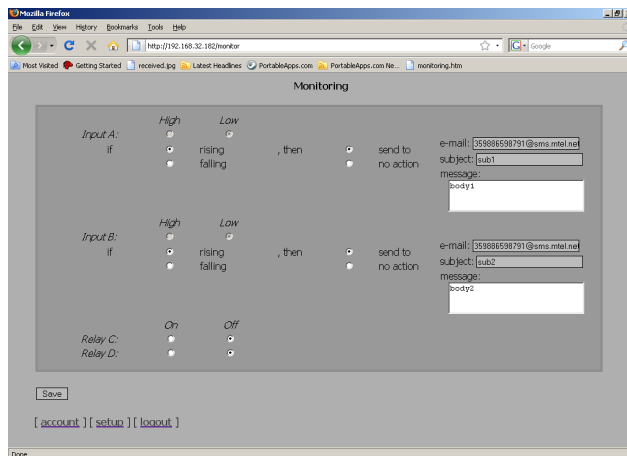


Fig. 4. Dialog window of the Monitoring programme module

The dialog window of the Monitoring programme is shown in the Fig. 4. This part of the programme is accessible for all the users from LAN network whose MAC addresses are activated from the administrator and those using Internet if it is permitted. When a user enters wrong user name or password three times, the system blocks the access for this user for 2 minutes.

At this stage of the development the events that have happened in two digital inputs will cause sending e-mails or SMSs up to three addresses or mobile phones. The number of the symbols of the subject and the body of the messages cannot be more than 39 characters. There are two options for every address – to be active or disabled. The e-mail addresses are specified by the administrator in advance. In the monitoring page every user has administrator rights but they can not enter the other pages (account and setup). Users can control the way the

events occur. It is possible to choose the type of the active action to the pin – level or edge (rising or falling). The events can cause message sending not earlier than 2 minutes after the last sending.

CONCLUSIONS

The proposed Internet based embedded system is a complete hardware and software solution implemented by TERACOM Company but it is open for enhancing its features. It is designed to implement a micro SD card in order to store log files for the events and the exact time of their occurrence, as well as the messages sent and their exact time. For this purpose a real time clock is included in the system. In Fig.1. two digital inputs (A and B) and two digital outputs (relay C and D) are shown but every pin of the parallel ports of the system can be programmed to receive information from or to send control signals to Input/Output devices. It is envisaged that analog information from various sensors can be processed and messages can be sent under predetermined conditions.

REFERENCES:

- [1] AT45DB161D, ATMEL, 3500J–DFLASH–4/08
- [2] Cupris G. et al., Implementation of a UDP/IP (User Datagram Protocol/Internet Protocol) Stack on HCS12 Microcontrollers, AN2304/D, Motorola, 10/2002
- [3] CviLux Corporation, System CJ Telephone Jacks&Modular Plugs
- [4] Microchip Technology Inc., PIC18F97J60 Family Data Sheet, Microchip, 2008
- [5] Nilesh Rajbharti, The Microchip TCP/IP Stack, AN833, Microchip, 2008

For contacts:

MEng. Ognyan Dimitrov, TERACOM Ltd, tel.: 082 862 862,
e-mail: o.dimitrov@teracom-bg.com
Student Ferhan Balpinar, University of Rouse, Department of Telecommunications,
e-mail: ferhan.balpinar@googlemail.com
Pr. Lecturer Nina Bencheva, University of Rouse, Department of
Telecommunications, tel.: 082 888 677, e-mail: nbencheva@ecs.ru.acad.bg
Assoc. Prof. Yoana Ryseva, University of Rouse, Department of
Telecommunications, tel.: 082 888 677, e-mail: iruseva@ecs.ru.acad.bg

Докладът е рецензиран.