



Integrating a Server for Remote Generation of Wake on LAN Packets to Network Management Systems

Gencho Stoitsov, Zhelyan Guglev
Plovdiv University "Paisii Hilendarski",
236 Bulgaria Blvd., 4003 Plovdiv, Bulgaria

Abstract: This abstract suggests the realization of a hardware device and its software provisioning, allowing access to the server application for generating WOL packets in LAN. The WOL protocol implementation does not require large computing power, which allows its integration in a number of alternative devices that have much lower price and power consumption compared to a typical computer system.

Keywords: ENC28J60, Local Area Network, Microprocessor, SNMP, OSI model, PIC18F67K22, Wake on LAN, WOL packets

I. Introduction

The technology Wake On Local Area Network (WOL) represents a communication standard that enables a remote device to be switched into or out of standby mode (stand by) by sending WOL packets in Ethernet (IEEE 802.3) network. The network interfaces supporting this protocol monitor communication traffic for WOL packets addressed to them, even after switching off the controlling system or bringing it in a standby mode [1]-[3]. Using MAC addresses as a key part of the mechanism of WOL protocol makes it difficult to apply outside the local network. There are implementations of WOL, allowing packets generated by external networks to reach devices that are part of another interconnected network. This variation of the protocol is also known as Wake On LAN over Internet. The successful detection and correct interpretation of WOL packets and security settings depend on the behaviour of intermediate devices, building the track to the appropriate local network. The data carried by a WOL package contain the Media Access Control (MAC) address of the device for which it is intended, and sometimes may contain post data with an authorization purpose. The high probability of a WOL package, with a source located outside the local network of the recipient, to fail to reach its target is one of the biggest problems of Wake On LAN technology [1]-[4]. The article presents an improved (hardware and software) version of the proposed approach in a previous article [19] for avoiding this problem. The innovations are related to the addition of remote diagnostics via network protocols ICMP and SNMP [16] and replacing the microprocessor with Microchip PIC18F67K22 to provide the necessary resources.

II. Used hardware platform

The improved hardware platform includes the following modules:

- Microprocessor - Microchip PIC18F67K22 (Architecture: HARVARD, RISC; Maximum speed: 16 MIPS @ 64 MHz; Target Speed: 16 MIPS @ 64 MHz; Program Memory: 128 KB; Data memory: SRAM: 3862 bytes, EEPROM: 1024 bytes) [13]
- Ethernet interface - Microchip ENC28J60 (communication interfaces: Ethernet 10Base-T, Serial Peripheral Interface (SPI); MAC (OSI 2); PHY (OSI 1); RAM buffer: 8 KB) [17]

Using a hardware platform with an architecture different from the traditional von Neumann architecture requires the creation of a specialized software for it. The newly obtained combination can seem to tend to be in the field of network equipment. Most devices in this category support methods enabling their own remote diagnostics as well as diagnostics of their communication environment. Their presence enables the collection of statistical information regarding the performance and stability of the product, which at a later stage can be used for improvement of the platform. The mere collection of information can be automated through the use of a system for network management (NMS).

The software, built on the platform, is a composition of several types of modular applications providing the following functionality:

- Web-based user interface that allows remote access via HTTP protocol
- Manual and automatic modes for generation of WOL packets on the LAN
- User list of target MAC addresses of devices
- Manual configuration of its own network parameters or an automatic configuration via the DHCP protocol
- Different types of statistical and diagnostic information about the platform and its communication environment accessible via HTTP, SNMPv1, SNMPv2c and ICMP protocol. An opportunity for collecting some of the data through a system of network management

III. Hardware architecture

The main goal in the development of the design is that it is maximally simplified, so that it easily meets the requirements for reliability, performance, simplicity and last but not least, the availability of the components. This justifies the choice of integrated circuits with low and medium level of integration, which contain everything necessary to achieve an Ethernet connection, and therefore the functionality described above.

The microprocessor and the network interface of the platform are selected, so that they can operate at the same supply voltage of 3,3 V. This eliminates the need for different supply voltages and components for converting voltages in the communication process between the two integrated circuits.

The CPU clock speed of 64 MHz is achieved through a built-in RC oscillator with an operating frequency of 16 MHz, which is passed to a frequency multiplier of coefficient 4, executed through a built-in module for a phase-locked loop (PLL). The described configuration eliminates the need for additional external components and the stability of the resulting frequency is acceptable for the purposes of the platform.

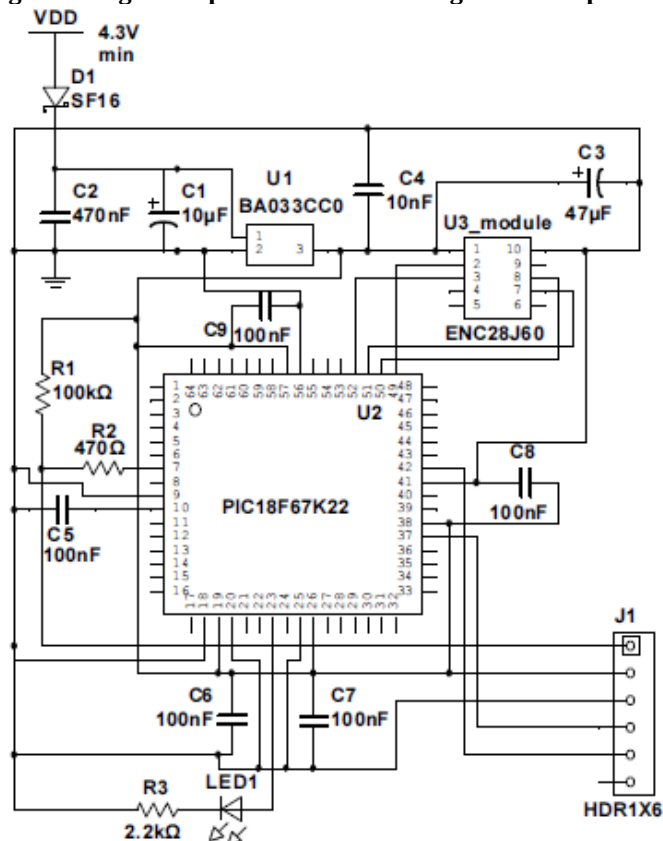
The network interface of the platform is a separate module, whose data and control outputs are used directly by the microcontroller. The internal communication between the module and the microcontroller is carried out via a serial peripheral interface (SPI) [15], as the mere transmission of the data is done by embedded hardware SPI modules. The speed of data sharing on SPI reaches about 16 Mbps, which is acceptable, taking into account the speed with which the network interface transmits data over Ethernet - up to 10 Mbps.

The power supply of the platform is via a DC source and it is recommended to use an AC / DC adapter with the possibility of giving a current of 150 mA at minimum output voltage of 4,3 V. The tolerance to the supply voltage is in the interval 4,3 V - 25 V as it allows the use of a wide range of AC/DC adapters.

The power protection is simple, it is done only with Schottky diode.

A general plan of the circuit diagram of the platform, ignoring details of the device and a method to connect the Ethernet module and the parameters of the other electronic components:

Figure 1 A general plan of the circuit diagram of the platform.



With J1 is marked the programming interface of PIC18F67K22, which is of type ICSP (In Circuit Serial Programming).

IV. Software Architecture

Because the project has a web-based user interface, the parts that make it up can be divided into two main categories - user interface (front-end) and control part (back-end). Since the implementation of the system complies with the architectural characteristics and the parameters of the hardware platform on which it works,

there is a little separate intermediate layer acting as a communication medium between the interface and the control part.

The user interface uses the latest versions of some of the most popular web technologies like HTML5, CSS3 and Javascript.

The realization of the back-end of the platform is achieved by C language and version 2013-06-15 of TCP/IP Stack of Microchip [18], which is recommended for use with microcontrollers of series PIC18. For a compiler is selected Microchip C18 v3.47, as it is compatible with the above version of TCP/IP Stack and is specifically designed for PIC18 microcontrollers.

The software modules providing network functionality of the platform, such as support for Ethernet, ARP, IPv4, ICMP, TCP, UDP, WOL, DHCP, SNMP, HTTP, are realized with the direct use of those embedded in Microchip TCP/IP Stack, or through their expansion or own design, but taking advantage of the functionality provided by the TCP/IP Stack.

V. Cooperative multitasking

Cooperative multitasking is an approach in which applications responsible for different types of functionalities are divided into modules that work "simultaneously" [4], [5].

Since the platform, on which the modules are run, does not have an operating system to take care of their "simultaneous" performance, this functionality is achieved as in each module operates a simple finite-state machine that stores the status and the progress on the task being performed. The algorithm, which is responsible for solving the task itself, is divided by the programmer in small blocks of steps, with the idea each block to use a minimal amount of CPU time. When invoking the module only one of its blocks is executed, and the state after the execution is stored to be used as a starting point for the next invocation of the module. If for the module there are no tasks, or the data that it needs to continue performing the task are not yet available, the CPU resources are immediately transferred to another module. The modules are executed sequentially and cyclically in the process of work of the microcontroller, which gives the user an impression of simultaneous operation. That part of the code, which is responsible for invoking the modules, can be defined as a kind of a simple task manager (task scheduler).

VI. Remote diagnostics via network protocols ICMP and SNMP

The possibilities for remote diagnostics are achieved by implementation of 3 independent software modules, which can be activated and deactivated individually by the user.

A. ICMP Implementation

The Internet Control Message Protocol is a widespread standard that provides functionality to check for the availability of a given host and measuring the latency of the route, which connects it [6], [7].

The platform has an ICMP server module and a client module. The server module meets the ICMP echo requests sent to it, returning an ICMP echo reply. The client module has a user interface, accessible via HTTP protocol only after authentication of the user and allows sending ICMP echo requests to devices on the local network, or to those outside it (if the network settings allow it). The possibility of an ICMP communication with internal LAN devices contributes to easier remote administration.

B. SNMP Implementation

The technology of Simple Network Management Protocol is a protocol allowing exchange of administrative and diagnostic data between different network devices. So far there are four versions of SNMP:

- SNMPv1 – the first official version that has become the standard in the management of network resources
- SNMPv2 – extends the capabilities of version 1 and adds new significantly more complex security measures
- SNMPv2c (Community-Based) – expands version 1 and uses security measures of version 1
- SNMPv3 – mainly improves security and the possibilities for remote configuration

Versions 1 and 2c of the SNMP protocol are more vulnerable to sniffing attacks, but the fact that they are much easier to implement and configure in any kind of platforms makes them the most widespread to date [8].

The module that implements support for SNMP uses protocol versions 1 and 2c. The data it provides are received by the client only at his request, and traps and inform requests are not supported.

VII. SNMP Management Information Base

Any system that supports SNMP has hierarchically categorized, formal definitions of the data it can provide. The description of that data is stored in the Management Information Base (MIB) files that are of a text type with the possibility of referencing with each other and with syntax rules derived from the Structure of Management Information (SMI) – a subset of Abstract Syntax Notation One (ASN.1) [8].

The final result in a general, abstract and very simple plan can be viewed as a tree structure of object identifiers (Object Identifier, OID) as to each one corresponds a number. In addressing the OID, the path to it is indicated (through their respective numbers, each separated by a point). There are no restrictions on the number, depth and balance of the tree of OIDs stored in a given MIB [8].

The hierarchy, which is maintained at level OID, is controlled partly by the organization "Internet Assigned Numbers Authority". Any other organization has the right to maintain their own "branch" through registration, and therefore to obtain the right alone to model its interior while respecting the Convention imposed by the SNMP protocol [8]-[10].

The developed platform takes advantage of the internal hierarchy of the branch registered by Microchip, specifically designed for their TCP/IP Stack.

A. Data types in the SNMP MIB

The main types of data that SNMP MIB provides are about 10 types, basically derived from 32 bit, 64 bit integer types or strings of variable length of up to 256 bytes [8].

The general categorization of the most commonly used types differentiates them into two fundamental types, with significant differences between them - gauge and counter:

- Gauge - it provides a convenient way to refer to a specific value in an exactly known time interval.

Example: The air temperature on 22nd August, 2015 at 13 hours 18 minutes 46 seconds 55 milliseconds, 23 microseconds is 36°C.

- Counter - a variable constantly incrementing in time.

The value, with which it increases per unit of time, depends on the specific internal and/or external factors. To derive information, the client must have at least two "measurements" for a particular time span, as their difference (also called delta) respectively indicates the amount by which the value has changed during the given time period. The division of the delta into the difference between the two periods gives the average change per unit of time.

To present floating point numbers is a problem, as they are not supported directly [8].

Two main approaches can solve the problem - transformation of the data into a string by the SNMP agent, or their transfer to the 32 bit or 64 'container', as the responsibility for their interpretation and decoding is left to the respective NMS.

The Counter type variables are often used in the monitoring administered by SNMP. Examples include the total number of packets sent from a given network interface, or the time of system operation elapsed since its last start - data that in practice change too often, making the storage of all discrete values of their modifications worthless.

B. Compilers of MIB files

The way to describe structures and data, used in the MIB form, and the relationships that may arise between individual MIB files, make it complicated and difficult to analyse and use by SNMP agents or the systems for management of networks in real time. The solution to this problem is the preprocessing of the MIB file - called a compilation, which converts the data format of a MIB file in a form that is suitable for machine processing. A useful effect is to perform the validation immediately before and during the compilation of the relevant MIB file [4], [8], [11].

There are different software implementations of the MIB compilers, and it is not uncommon between them to be differences in the way of interpretation. The reasons for this are generally due to internal characteristics associated with the specific purpose of the outcome [12].

The compiler provided by the TCP / IP Stack of Microchip differs significantly from conventional compilers, as the newly data structure is designed for direct installation in the code of the SNMP module of the project. Because the SNMP protocol is built into devices having a significantly limited program memory resources, the compiler itself has very limited abilities in presenting different types of data at a low level, as well as working with "MIB" files. There is a generalization of the supported data types as two general types - dynamic and static. Dynamic data are those, whose condition is not known in advance and usually change over time such as temperature, current time, the workload of communication channels. Static data are always known and do not change over the time in which the SNMP the module works. Examples include the identifier of the platform, software version, hardware revision of the network interface [4], [11].

The compilation of a MIB with the Microchip's compiler leads to the creation of a new file type called BIB, which contains the same data but in a machine format [11].

The compiler of "MIB" files of Microchip does not actually compile authentic MIB files, but requires the use of a pseudo format for describing the data that would normally be contained in a MIB file. This fact forces the developer to describe the same data structures in two different ways - once for the purpose of its application (in particular the architecture of the TCP / IP Stack) and once for integration with standard management systems networks [11].

C. Data returned from the platform via SNMP

Part of the data, returned by the SNMP protocol, are derived from the definitions coming from a standard container of the OID tree called "mib-2", defining in itself other subcontainers-groups that organize the similar MIB objects. Their content can represent object identifiers providing descriptive information about the system, available network interfaces, basic protocol statistics and so on. Groups can be obligatory - maintenance of all object identifiers in them is required, or optional - the entire content of the group may be present or not be present in the concrete implementation of MIB. The conditions that make a group obligatory or optional are connected to

the technologies supported by a device and protocols that can be monitored via SNMP and the available groups on definitions required for them in "mib-2" [8].

Besides the metadata themselves, Wake On LAN over Internet server provides data as the time of operation of the system since its last start, information about the available free space on the drive, about internal temperature of the microprocessor - the latter type is particularly useful for establishing hardware failures or faulty firmware.

VIII. Conclusion

A fully functional product has been created, based on a non-standard platform, successfully integrating a number of established standards and technologies from the fields of network programming, web programming and marking, multitasking.

For the testing of the platform was chosen the standard system for management of networks SevOne NMS 5.3.2.0 trial, which supports SNMP. After 48 hours of testing carried out on the platform at different loading, it was found that it has a stable performance and sufficient speed in work and data transfer. Despite the modest hardware parameters, the time for which the system is ready for use after switching on, is less than 1 second. The transfer of data from the server to the client and in the opposite direction is almost instantaneous in the local network, which fully justifies the choice of a target speed limit to 10 Mbps.

The cost for creating a prototype of the platform totals € 18 - lower when taking into account the technical specifications of those available on the market non-prototype platforms, with the potential to perform the same task such as Arduino Uno, Arduino Ethernet, Teensy, Raspberry Pi.

The measured power consumption of the platform for supply of 13 V is about 1,4 W - to compare with the measurement of Raspberry Pi 2 B + for supply of 5 V found an average consumption of about 1,3 W, and that of the Arduino Uno 2,1 W at 5V power supply.

The potential of the hardware which builds the platform has not yet been fully exploited, leaving possibilities for future development and improvements.

IX. References

- [1] AMD, "Magic Packet Technology," November 1995. [Online]. Available: <http://support.amd.com/TechDocs/20213.pdf>. [Accessed 30 May 2015].
- [2] Cisco, "Internetworking Basics," 1998. [Online]. Available: <http://www.cisco.com/cpress/cc/td/cpress/fund/ith/ith01gb.htm#xtocid166844>. [Accessed 02 June 2015].
- [3] Microchip, "AN1120 - Ethernet Theory of Operation," 02 January 2008. [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/01120a.pdf>. [Accessed 11 June 2015].
- [4] Microchip, "Microchip v2013-06-15 TCPIP Stack Help," 2013.
- [5] Microchip, "AN1264 - Integrating Microchip Libraries with a Real-Time Operating System," 03 March 2009. [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/01264a.pdf>. [Accessed 26 June 2015].
- [6] Network Sorcery, Inc., "ICMP, Internet Control Message Protocol," [Online]. Available: <http://www.networksorcery.com/enp/protocol/icmp.htm>. [Accessed 26 June 2015].
- [7] IETF, "RFC792," September 1981. [Online]. Available: <https://tools.ietf.org/html/rfc792>. [Accessed 27 June 2015].
- [8] D. T. Perkins and E. McGinnis, Understanding SNMP MIBs, Prentice Hall, 1996.
- [9] IANA, "PEN Request," [Online]. Available: <http://pen.iana.org/pen/PenApplication.page>. [Accessed 25 May 2015].
- [10] IANA, "PRIVATE ENTERPRISE NUMBERS," 24 May 2015. [Online]. Available: <http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers>. [Accessed 25 May 2015].
- [11] Shirbhate, Amit; Microchip, "AN870 - SNMP V2c Agent for Microchip TCP/IP Stack," 2009. [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/00870b.pdf>. [Accessed 26 May 2015].
- [12] Cisco Systems, Inc., "26015 MIB Compilers and Loading MIBs," 26 October 2005. [Online]. Available: <http://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/26015-mibcompilers.pdf>. [Accessed 26 June 2015].
- [13] Microchip, "Microchip PIC18F87K22 Family Data Sheet," 10 Юни 2011. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/39960d.pdf>. [Accessed 05 Май 2015].
- [14] Microchip, "MPLAB C18 Getting Started," Microchip, 2005.
- [15] Microchip, "SPI Overview and Use of the PICmicro Serial Peripheral Interface," [Online]. Available: <http://ww1.microchip.com/downloads/en/devicedoc/spi.pdf>. [Accessed 17 Май 2015].
- [16] Microsoft, "TechNet - How SNMP works," [Online]. Available: <https://technet.microsoft.com/en-us/library/cc783142%28v=ws.10%29.aspx>. [Accessed 25 Май 2015].
- [17] Microchip, "Microchip ENC28J60 Stand-Alone Ethernet Controller with SPI Interface," 26 Октомври 2012. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/39662e.pdf>. [Accessed 05 Май 2015].
- [18] Rajbharti, Nilesh; Microchip, "AN833 - The Microchip TCP/IP Stack," 2002. [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/00833b.pdf>. [Accessed 26 Май 2015].
- [19] G. Stoitsov and Z. Guglev, "Server for Remote Generation of Wake On LAN Packets in the LAN," International Journal of Recent Development in Engineering and Technology, vol. 3, no. 2, p. 6-27 August 2014.