Embedded System for Measurements and Data Acquisition with Web Interface Based on ARM Processors

Kamen Kostadinov Krastev, Tsvetomir Antonov Gotsov and Aneliya Vladimirova Manukova-Marinova

Abstract - **The aim of the article is to show the advantages of embedded systems and their structures based on ARM(Advanced RISC Machines) processors. The presented system is used for data transmission and data acquisition in a distributed system. Building a modern system requires interactive human-machine interface for observing the information of ongoing processes in a way suitable for analysis. The ability to monitor processes remotely and the lower cost of the system's components make the latter available for use in education, research, civil and even industrial environments. Attention is paid to the possibilities for connection with other data analysis systems and the compatibility between the proposed and other existing embedded systems. One of the great advantages of the ARM architecture is the existing wide variety of IDEs (Integrated Development Environment) and hardware manufacturers.**

Keywords – **ARM, embedded system, data acquisition, microcontroller**

I. INTRODUCTION

The whole system is built on the principle "producer - consumer". At the core of the developed system is a Cortex M4 series microcontroller with embedded peripherals on board. The controller periodically measures values from sensors. Depending on the specifics of the initial processing, the internal parameters and measurement resources are changed, to ensure as much computational efficiency as possible. After validation of the computed results, the latter are transmitted to the server (the server can receive data from more than one slave). If a communication channel to the server is not available, the data is stored locally in non-volatile memory. The system is designed to be flexible and adaptable. With the addition of a few slaves the entire local system can easily be converted into a distributed one.

Modern high-level systems for data acquisition, processing and transmission are capable of receiving data via OPC (OLE for Process Control). With the development of ARM technologies and the potential usage of secure operating systems, the exchange of information between systems with different platforms is accessible and achievable. Modern industry has partially solved the problem of information exchange via OPC server / client architecture.

The server utilizes a single ARM A8 series processor, whose architecture is adapted to numerous types of applications. This type of architecture is used in modern "smart" devices e.g. notebooks, tablets, mobile phones etc. The LINUX operating system gives the foundation of security. Furthermore, it allows the installation of SQL to collect data in a database and HTTP server to visualize the information of the resulting WEB applications. As a result, this paper provides methods and systems for data transmission to industrial and business systems (SCADA, ERP, BMS), through the usage of TCP/IP, OPC, etc. An HMI is used to access the information collected and processed by the system. By MVC (Model-View-Controller) the user declares their intention. The system verifies his legitimacy and access rights, then it provides adequate access and view. Data views are divided into two types: current and archived. The information update period of the current data depends on parameters such as the physical interface of communication channels and the communication protocol used. Archived data is recorded based on certain criteria e.g. rank required, minimum period, etc. The use of modern WEB technologies allows the system to easily send an e-mail, SMS or an emergency alarm. The system data can be exported to CSV, CSV for MS Excel, SQL and many other formats. This gives the mathematical and statistical software (MatLab, Canopy, etc.) access and ability to analyze the data.

II. ARCHITECTURE OF THE EMBEDDED SYSTEM

System structure

The architecture of the system (Figure 1) is a classic case of a distributed system. The functionality is shared between processors. The proposed system allows the incorporation of more than one microcontroller. Each additional controller is accessed by a unique address and data are kept in a separate table from the database.

K. Krastev is with the Department of Electronics, Faculty of Electrical engineering, electronics and automation, University of Ruse, 8 Studentska str. 7017 Ruse, Bulgaria, e-mail: kkkrastev@uni-ruse.bg

Ts. Gocov is with the Department of Electronics, Faculty of Electrical engineering, electronics and automation, University of Ruse, 8 Studentska str. 7017 Ruse, Bulgaria, e-mail: tsgocov@uni-ruse.bg

A. Manukova is with the Department of Electronics, Faculty of Electrical engineering, electronics and automation, University of Ruse, 8 Studentska str. 7017 Ruse, Bulgaria, e-mail: amanukova@uni-ruse.bg

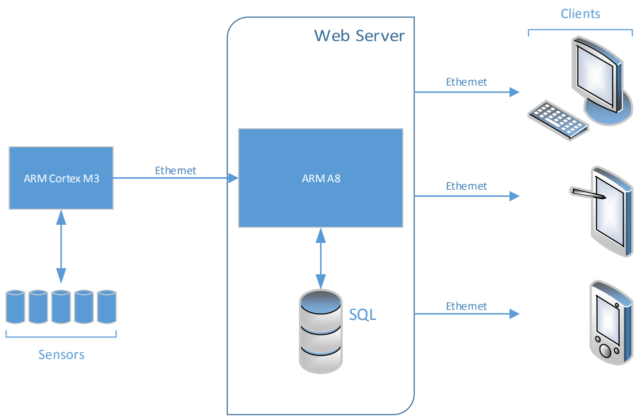

Fig.1. Architecture of the system

• *Functionality and communication at process level*

Figure 2 presents the controller for the data collection at process level.

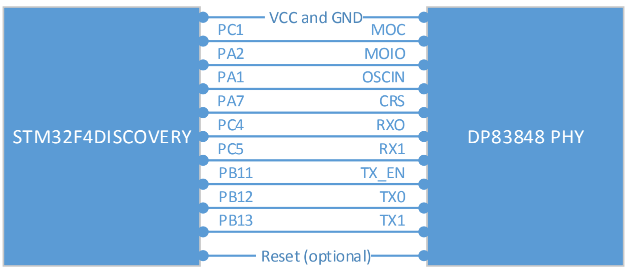

Fig.2. Structural diagram of the process level

Hardware

STM32F4 Discovery is selected as the process unit in the system of data collection. The other structural parts of the board are: STM32F407 microcontroller with Cortex-M4 - 32 bit CPU architecture, which included 10/100Mbs Ethernet MAC (media access control) with a special DMA (direct memory access); backing IEEE 1588v2 hardware; MII (Media independent interface); RMII (Reduce MII). A physical layer (PHY) Layer 1 of the OSI model is placed between the *Discovery board* and the integrated circuit DP83848. The main advantage of Reduce MII (Figure 2) interface is the smaller number of connections between communication busses at the expense of twice the clock frequency.

The periphery of the process level supports up to 70 I / O, 3-12 bits analog-to-digital converter to 2,4 MSPS to 24 multiplexed channels, two Digital-to-Analog Converter 12 bits and 8-14 bits parallel camera interface reaching up to 54 MBytes / s. The CPU at field level can process the information and arrange it in the appropriate form. CPU performance is 1.25 DMIPS / MHz (Dhrystone 2.1). The main advantage of the core is the embedded hardware accelerator for parity. It is unloaded through the core when inserting and checking the checksum in the IP, UDP, TCP and ICMP protocols.

Software

The presented system works with LwIP stack, developed by Adam Dunkels at the Swedish Institute of Computer Science. LwIP is an open-source TCP / IP stack that is licensed under a modified BSD license. This stack is written in the programming language "C". Its main task is to maintain a full-scale TCP / IP stack with the reduced use of RAM, preferably in the field of embedded systems. LwIP supports the following protocols: IPv4 and IPv6, ICMP, IGMP, UDP, TCP, DNS, SNMP, DHCP, PPP, ARP.

Three programmable APIs (API *application programming interface*) are included in LwIP – Raw API, Netconn API и BSD Socket API.

The system works without OS / RTOS, which is a major drawback when multitasking is necessary. The data is processed and sent as standard html, which allows direct monitoring of the required data on the controller input. Data visualization of ADC uses Server Side Includes (SSI).

SSI is a method used to dynamically include dynamic data in HTML code. This is done by placing a specific tag inside the HTML code of the web page. The tag should have the following format: <!--#tag--> The following tag is used inside the HTML code for the ADC conversion page: <!--#t-->. When there is a request for the ADC webpage (which has a ".shtml" extension), the server will parse the webpage. When the tag is found, it will be replaced by the ADC conversion value.

• *Functionality and communication at server level*

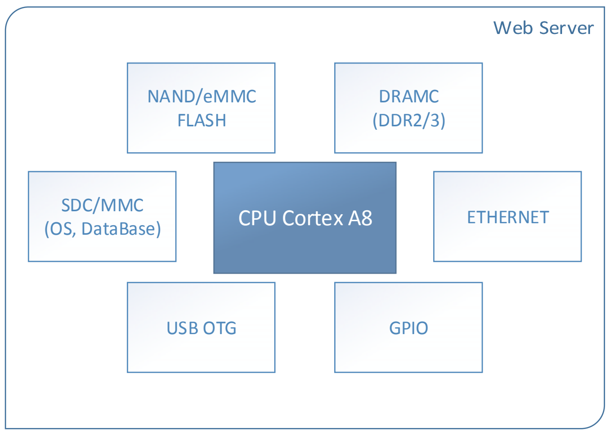

Fig.3. Server architecture

Software in the server side of the system

The powerful processor adapted for special applications and additional peripherals is provided by the operating system LINUX (distribution - DEBIAN 7). The hardware is accessed indirectly in the presence of an operating system. Memory is controlled automatically and communication is considered at a higher level. The focus is entirely on getting the data from the microcontroller to field level, saving it in a database and giving access to consumers through appropriate human-machine interface HMI.

Apache and MySQL are installed on the server. Apache HTTP Server is an open source and allows data collection and user access to stored data. Data storage is implemented in the MySQL database, also open-source.

The basic functionality for downloading data from the microcontroller is implemented by using the high-level language PHP. The first step to setting up communication between server and microcontroller is to check whether the microcontroller is in the network and corresponds to an IP address. The function which checks the communication channel is ping (). Source code is given in Table 1. The function returns diagnostic information about the performance of the device. It indicates whether the device is still available and responds to the query. If it responds, the function calculates its response time [ms] and detects problems in the communication channel. After establishing the communication, the flow of data is recorded periodically with a query to the database, implemented in PHP. SQL server arranges the information in an appropriately structured table. Each record is given a time stamp and an ID.

TABLE 1. CODE OF THE FUNCTION PING ()

```
function ping($host, $port, $timeout)
 {
 $timeB = microtime(true);
 $fP = fSockOpen($host, $port, $errno, $errstr,
$timeout);
 if (!$fP)
{
   return "down";
}
 $timeA = microtime(true);
 return round((($timeA - $timeB) * 1000), 0)." ms";
}
```

HTML, CSS, and JS framework - Bootstrap 3 are used for designing the user interface. The components are predefined to already developed sites - text boxes, tables, bar graphs, etc. They are known to the user. During system development the focus is on functionality, since it is crucial to the performance and HMI interface. User access to the architecture of the application is built on MVC (Model-View-Controller), and is implemented using modern Web browser (Fig. 4).

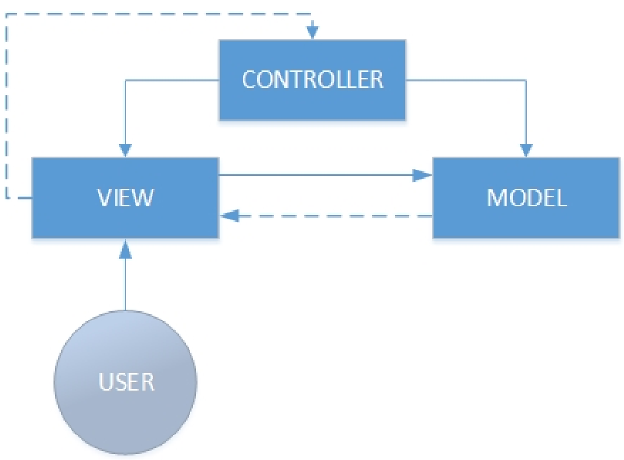

Fig. 4. Simplified diagram of MVC

The Model. The model is at the core of the application. It contains models and data for key parameters to be entered, modified, displayed, etc. The model and the real world are different. The abstract model of the world is the one that is a product of our minds. We perceive it as solid objects, formulas, mathematical symbols, diagrams etc.

*The view***.** The view is the part of the source code which is responsible for displaying the data from the model.

The controller. The controller communicates with the block model and the block view by extracting data from the model. It calls for additional methods on the model, and provides preliminary data processing. Upon completion of operations, the controller presents the results to the visualization block - the view.

III. WORKING CAPACITY AND ENERGY EFFICIENCY OF THE EMBEDDED SYSTEM

Numerous tests are performed to check the working capacity and efficiency of the embedded system. The results are stored in a specially developed database. The various records of parameters and formats indicate that the embedded system is running stably and reliably with over 100 000 records stored in the database. The information is quickly processed and readily accessible from external software products.

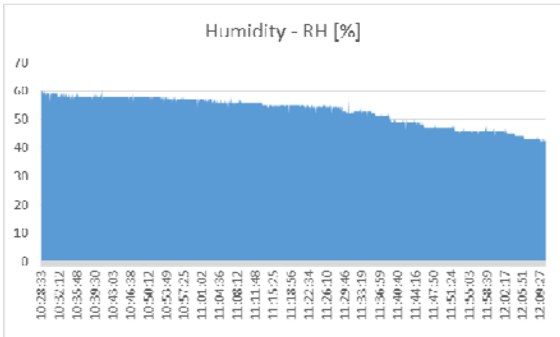

Fig.5. Relative humidity in the region of Rousse

Figure 5 presents data on the relative humidity in the region of Rousse, recorded by the embedded system. The graph shows the capacity for implementation of the information into other systems and specialized data processing databases. The system flexibility is a major factor in its development.

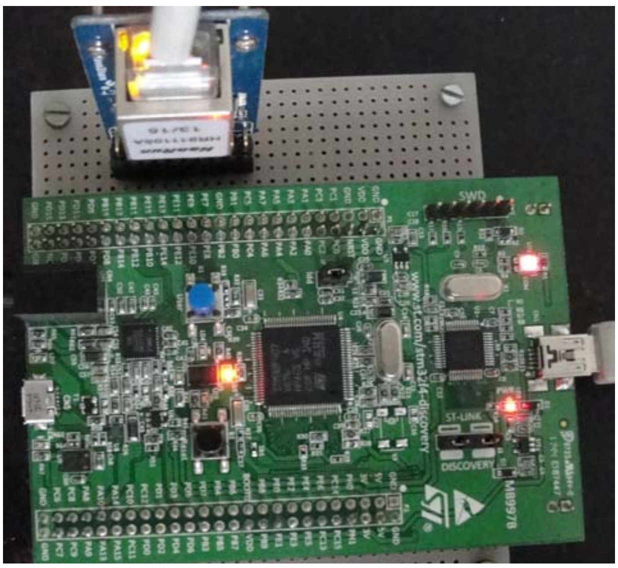

Fig. 6. Prototype of the process level controller

Figure 6 presents the working prototype of process level controller.

Energy efficiency of the system

In classical systems of data collection information and its suitability are the only important aspects at the end of the process. A small percentage of process studies address the overall concept of optimal amount of data, reliability of records, fast and secure access to information, energy efficiency and others. Energy consumption is the most important indicator of the performance of the developed system.

The amount of used electrical energy is a key factor in the development of the embedded systems. Consumed energy serves as a basis for structuring the parameters of the renewable power source of the system.

When testing, the developed embedded system is compared to a similar one, on which HTTP and SQL servers are implemented on a personal computer. Comparisons of energy consumption are made for the period of the reported humidity in Fig. 5.

In the studies carried out a server located on a personal computer with a power consumption P = 492W is used as a basis for comparison. Energy consumption is calculated based on its active workload time t = 101min=1.68h.

$$\begin{aligned} &E = P.t = 492.1,68 = 826,56\ Wh \\ &\Rightarrow E = 0,8265\ kWh \end{aligned} \quad (1)$$

The developed ARM-based server has a power $P = 0,32W$. Energy consumption is calculated for the same time $t = 1,68$h.

$$\begin{aligned} &E = P.t = 0,32.1,68 = 0,5376\ Wh \\ &\Rightarrow E = 0,0005\ kWh \end{aligned} \quad (2)$$

The results obtained for the energy consumed show energy savings of more than 1500 times for embedded systems against classical system implemented on a personal computer. Summarized for 24 hours, this indicator is sufficient for the selection of power supply for process level of the system, based on renewable source of energy.

IV. Conclusion

This study proposes architecture of an embedded system based on advanced electronics. The architecture contributes to high accuracy of measurements.

The developed embedded system has the following advantages:

- compact size and mobility, employability and adaptability to the environment in which it is placed;
- high speed, low power consumption;
- reliability and precision of work.

The test system is universal. It is very suitable for educational purposes. It is also applicable when monitoring processes.

Acknowledgements

The present document has been produced with the financial assistance of the European Social Fund under Operational Programme "Human Resources Development". The contents of this document are the sole responsibility of "Angel Kanchev" University of Ruse and can under no circumstances be regarded as reflecting the position of the European Union or the Ministry of Education and Science of Republic of Bulgaria.

Project № BG051PO001-3.3.06-0008 "Supporting Academic Development of Scientific Personnel in Engineering and Information Science and Technologies"

References

[1] Doug Abbott. Linux for Embedded and Real-time Applications, 2003, Newnes.

[2] Dan Eisenreich, Brian DeMuth. Designing Embedded Internet Devicec. 2003, Newness.

[3] LwIP TCP/IP stack demonstration for STM32F4x7 microcontrller, application note, ST.

[4] Борис Цанков. Телекомуникации фиксирани,мобилни и IP. 2006. Нови Знания, София

[5] P. Mitros, K. Afridi, G. Sussman, C. Terman, J. White, L. Fischer, and A. Agarwal, "Teaching Electronic Circuits Online: Lessons from MITx's 6.002x on edX," in International Symposium on Circuits and Systems (ISCAS). Beijing, China: IEEE, May 2013.

[6] "UC Berkeley Joins edX," edX, July 2012. [Online]. Available: https://www.edx.org/press/uc-berkeley-joins-edx[7] J. C. Jensen, E. A. Lee, and S. A. Seshia, "Teaching Embedded Systems the Berkeley Way," in Workshop on Embedded Systems Education (in conjunction with ESWeek), Tampere, Finland, October 2012.

[8] R. Smith. Open dynamics engine. [Online]. Available: http://ode.org

[9] Massachusetts Institute of Technology (MIT), "The iLab Project," Last accessed: March 2013. [Online]. Available: https://wikis.mit.edu/

[10] R. Singh, S. Gulwani, and A. Solar-Lezama, "Automated feedback generation for introductory programming assignments," in Programming Languages Design and Implementation (PLDI), 2013 confluence/display/ILAB2/Home}