

Webprocessors for Measuring, Control and Synchronization

P. Klan

Institute of Computer Science
Prague
Czech Republic

J. Smid

SKS Enterprises
Finksburg, MD
USA

Abstract – *The Internet protocol (TCP/IP and HTTP) allows us to exploit new type of devices and algorithms that are suitable for building large and small sensor as well as knowledge networks. Even more importantly, the ease of communication will allow us to build truly dialog-based networks which we see as the next generation networks and artificial intelligence. Networks have many components. One of the basic components is what we call webprocessors. We describe a recently developed device that we named CTRLV4 and its smaller version that we call Miniserver. CTRLV4 is a server that can be controlled by parameters send to it via the GET method.*

CTRLV4 evolved from previous versions interfacing ordinary small computers (window based pc's). Because this device is of a small size and understands HTTP protocol, it is suitable for building large Internet protocol based networks. These networks can be built either as sensor networks or networks that can control other devices.

In this paper, we review some other devices, similar to CTRLV4, suitable for building cost conscious units and networks. Cost is one of the main factors because this type of networks should be ubiquitous.

We present main features of the CTRLV4 PIC18F452 based micro-controller and typical network applications. These applications are suitable as an instructional, research and engineering tool. The applications we will discuss are temperature and motion monitoring and synchronization of non-linear oscillators

Keywords: *webprocessors, network environment, synchronization, measurement, control*

1 Introduction

Measurement and regulation are traditional engineering disciplines. Many laboratories work with a variety of models of physical processes. These models are used for educational purposes [6], [1], [2]. Both our learning and research of traditional engineering disciplines proved more effective when using these models. In predicting virtual worlds of the near future [12] it is likely that most people will communicate with many artifacts, such as personal and public web-processors, web-sensors and IP-cams, web-robots, personal info-bots and databases. Intensive system

communication with artifacts (agents, robots, devices) is the necessary ingredient for a flexible system. This will likely be true about engineering systems as well. Perhaps engineering systems will exploit more special protocols.

2. Communication Modes

Networks consist of nodes and links among them. The nodes are either webprocessors or websensors. We define them as follows:

Websensor (WS) definition:

sensors + (micro)processor + http(tcp/ip) interface

Webprocessor (WP) definition:

http(tcp/ip)interface+ processor

WP does not have to be equipped with any sensors, and WS does not have to have any processing power except for serving values of its sensors. What are the main communication modes of WP and WS? There are three main modes: polling, ftp, bi-directional socket communication.

Polling:

In the simplest scenario ws's are polled by one or more wp's or by a human user. This means that a string is send to the server. This string looks in the PHP or Java environment as follows:

```
“GET name.php?v=value HTTP/1.0 Host:131.118.161.28”
```

The v variable value typically comes from a user or agent, for example from the mouse, a form, or a microphone. This string will be correctly interpreted by a server that understands PHP. PHP is the standard but it is not a part of basic HTTP.

```
” GET /page.html HTTP/1.0\r\n Host:ctrlv4.cs.cas.cz \r\n”
```

This string is standard. This type of string is universal in sense that all correctly implemented web server will understand this query and will respond. In particular, this string will evoke a page page.html to be send back to the client.

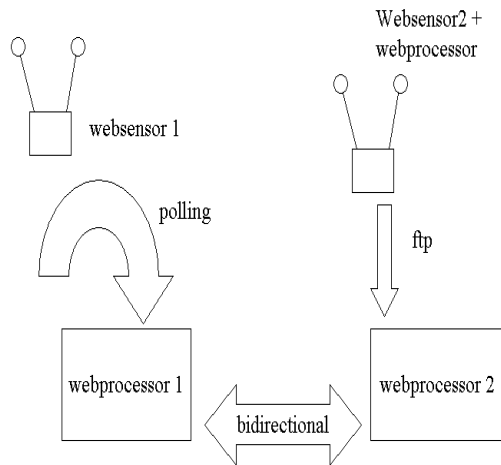


Figure 1 Communication Modes

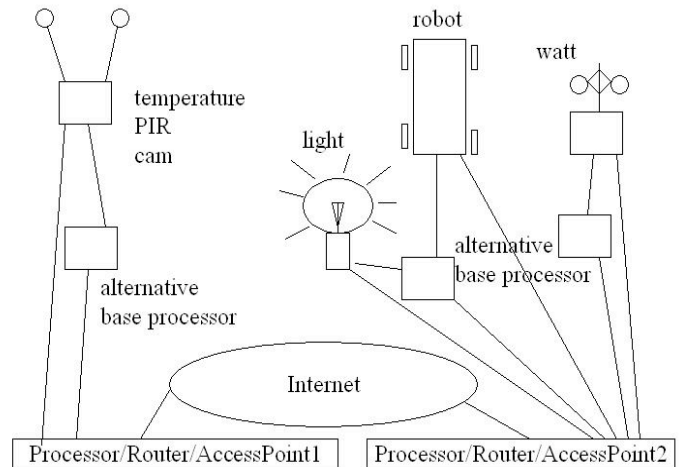


Figure 2 Communication Nodes 1

Ftp:

A node asks to be connected to a ftp server that is able to listen. This feature is implemented in a number of small commercial IP cams. The FTP is usually easier to connect with compare to HTTP when traffic is heavy. The FTP unlike HTTP requires a dialog which may or may not be convenient. The FTP implementation is open source and available in many commercial products (please see section on hardware).

Bi-directional communication:

Webprocessors manage more complicated information than command-invoking fixed programs. These types of processors need to be able to listen and to ask as well. This requires multiple threads or modules running all the time. This is also true for data communication and for language communication. This type of of communication requires a strong computer, such as a Stargate Linux single board.

3. Network Nodes

Miniservers and CTRLV4 with thermometer and PIR represent typical nodes that serve data.

CTRLV4 is designed to control other micro-controllers such as Watt regulator. Even more complex webprocessors that can be programmed is represented by Stargate [14].

4. Hardware for Internet Interfacing

The basic unit for network-based experiments consists of one processor providing either data, command target or a communication link to other network nodes. The other network nodes can be a temperature sensors, one or more infrared (PIR) sensors, an IP cam, a cam, a controller or any other device able to communicate in one of the three basic modes.

Why is this type of hardware appropriate? Assuming that we need sensors and mini-servers at many locations, they should be as ubiquitous as possible and therefore the cost is one of the issues. In order to provide ubiquitous units, developing or integrating units from existing components at a very low cost is essential. Some manufacturers offer inexpensive miniservers, for example Ethernet based on the ATMEL chip, PIC based kit by CCS [10]. Java based controllers TINI are produced by Maxim [11]. TINI has the ability to connect to other network nodes using TCP/UDP sockets and to actively write information to other local and remote web-entities. Red-I distributed by Parallax [15] has been used for some of our robotic experiments.

A next step up is a single board computer such as Stargate [14] by Intel running Linux. There is no limit in terms of computing power that can be used when we move to more powerful hardware. A general overview of on-line robots is provided by [20].

There are a number of kits offering excellent robots in terms of battery power and accessories. Example of those are robot Garcia with Stargate that allows remote re-programming. Garcia can be equipped with CMUcam. A recent success

story is Roomba Create [16] that allows to create an army of inexpensive monitoring devices for house and office.

5. Miniserver and CTRLV4 evolution

First, we have developed a micro-controller that we called CTRL[4]. CTRL was able to communicate via serial line with the standard PC and non-commercial and commercial software such as popular MATLAB. CTRL allowed us to control real-time processes. The current version of CTRL that we call CTRLV4 is based on PIC18f452 by Microchip [17]. CTRLV4 is a custom made miniwebserver that represents the simplest node of our networks. CTRLV4 consists of the standard full-duplex Ethernet microcontroller RTL8019as [18] that contains all necessary components to transmit and receive Ethernet packets (based on the hardware layer addressing system: MAC which stands for Multimedia Access Control addressing). The PIC18f452 microcontroller implements TCP/IP stack and HTTP protocol. In the simplest model, Miniwebserver creates web pages with sensor data. Slightly modified implementation of HTTP also allows us to send data as a part of the standard GET command. These parameters are used to control several digital output of the PIC board. In a higher model analog outputs are also implemented and allows us to control additional devices such as the Watt regulator.

Analog outputs, assembler programmed, no hidden features Web processor CTRL V4 includes:

- 4 analog inputs {0-10 V}
- 2 analog outputs {0-10V, 50 mA}
- 4 logic inputs and outputs
- Serial data bus line Dallas.

Logical outputs are adjusted to allow for direct controlling 12V relay. The entire device is placed on the small board which enables easy linking the other physical devices, Fig. 3. On one side, there is an Ethernet connector RJ 45 together with the signal connector RJ 12. Ethernet connector is used for connecting with the Internet and signal connector is used for linking the device with the serial data bus line Dallas (thermometers, PIR sensors and so on). There is no need to install any controllers while linking with the local computer network. The IP address can be re-set any time. CTRLV4 is power by 15 V source (min. 300 mA). CTRLV4 is a good device for engineering applications.

Mini-server is a stripped down version of CTRLV4. It works via a Dallas bus line which is used for attaching Dallas thermometers. It can also work as a web base relay, for example we can turn a light on/off. Mini-server is the backbone for our temperature/PIR network.

This device is convenient for home use because it is very small (5x5 cm), requiring only Ethernet access and no permanently running PC.

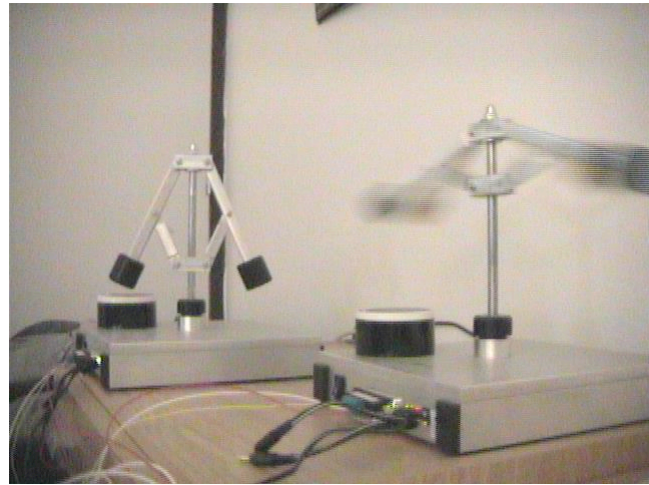


Figure 3 CTRLV4, two Watt regulators

6. Monitoring Applications

Monitoring of temperature and movement and of similar quantities in a network environment represents several challenges. Hardware should be easy to use and inexpensive and so allowing massive use by ordinary people.

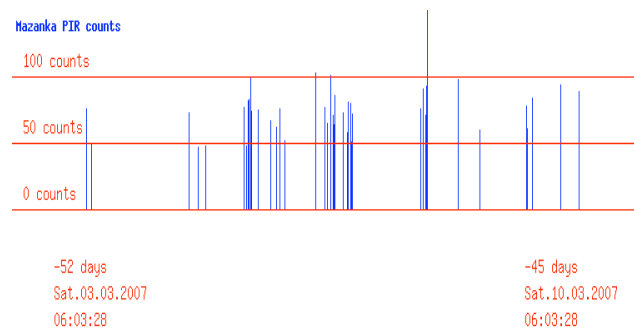


Figure 4 PIR counts

With regards to software and algorithmic factors, a smart structure of webprocessors associated with databases are needed. No matter the content stored in the databases, a new query will need to be satisfied and the answer to this query will need to be constructed from several available distributed databases. In our experiments we have created a modest system of several databases that provide the current and historical data of temperatures and motion counts, historical plots. One of the features that will be of increasing importance is a language like query, in which the user or agent can ask a language query that will be translated into a system specific program.

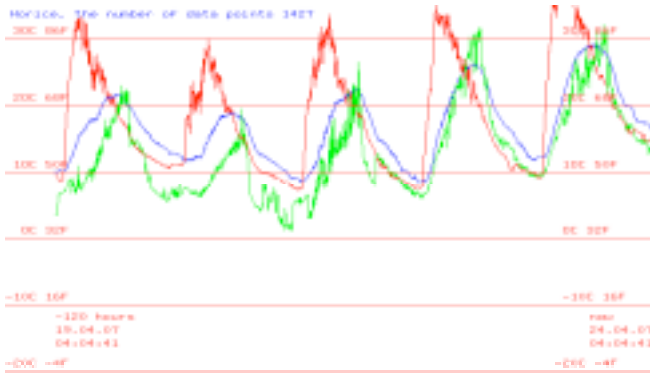


Figure 5 Temperature at Horice

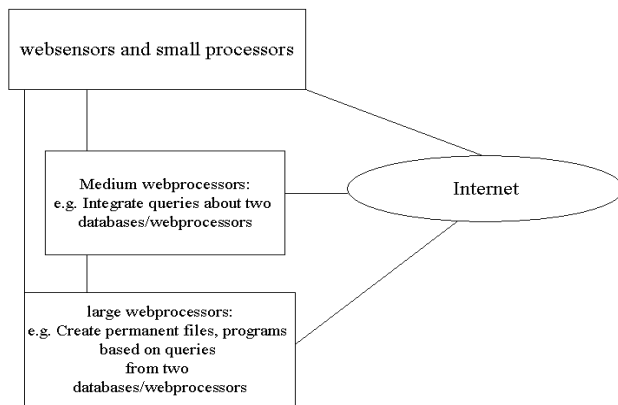


Figure 6 Network Nodes 2

Another planned feature is that network will allow the user or agent to retrieve data from distributed databases that has been not previously linked. For example, the user might be interested in correlation of temperatures of two places and in addition to that to correlate these data with motion detector signal at a third location.

7. Synchronization

One type of synchronization can be represented by the thermostat problem where one or more devices should be set to generate the same and possibly known fixed temperature. This type of problem can be successfully solved using the standard proportional integral (PI) controller. The second type of synchronization deals with oscillators with a jump non-linearity. This type of synchronization was developed by Mirollo and Strogatz as a mathematical model that explained long standing mystery of synchronization of fireflies [8]. Hardware implementation of both classes mentioned above can be realized using Watt regulator. This custom made regulator is controlled by CTRLV4 (fig3). In scenario 1, the number of revolutions is set, the goal is to control both Watts to achieve the set level revolutions. In scenario 2, the number of revolutions increases up to the maximum and then drops to zero. The goal is to

synchronize the movement up and down. The N oscillator system mathematical formulation is as follows: A system of N nonlinear oscillators can be described by their interaction. Each oscillator is characterized by a state variable

$$0 < x_i < 1, i = 1, 2, 3, \dots, N;$$

including special values $x_i = 1, 0$. When $x_i = 1$, the i th oscillator "fires" and x_i jumps back to zero [8]. In the simplest scenario the oscillators interact without any delay: when a given oscillator fires, it pulls all the other oscillators up by an amount ϵ , or pulls them up to firing, whichever is less:

$$\text{if } x_i(t) = 1 \text{ then } x_k(t) = \min(1, x_k(t) + \epsilon)$$

for all k different from i . The interface will play important role in sensor networks. For geometry based models such as oscillators, it is convenient to provide animated and speech based interface. We have developed a JAVA-based interface. This allows an agent to control the number of revolution using the mouse and a small vocabulary. Because of the applet built-in security, the applet passes instruction to a local PHP program and this program sends a message to CTRLV4 via socket communication. The extension of this approach allows for the accumulation of virtual commands (a sequence of elementary commands) while deferring execution to a later time decision.

Appa DNS ping: The number of data points 8263

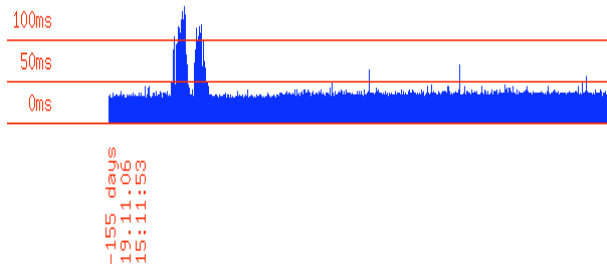


Figure 7 Delays at Appaloosa

8. Sync with Delay

In reality, the signal propagates with a finite speed. Our non-linear oscillators can be visualized as sound (finite speed) based communicating fireflies and sync success is measured by optical check (infinite speed). To cope with a variety of possible delay processes all available algorithms for predicting delays might be consulted. Regression models cover nonlinear trends in delays. Standard Autoregressive Moving Average (ARMA) algorithm is convenient for modeling of oscillating processes. Entropy model (CC) is a sub-model of ARMA that is suitable for exponential processes. The application of predicted delay

then compensates the actual delay. Numerical simulation suggests that that non-linear sync is either proportional to the variance of the delay or sync does not occur.

For data visualized in figure 7 the lowest error of 1.54ms was returned by AR, 1.59ms was returned by ARMA, and 1.69ms was returned by CC. Sync occurred with the error at the level of the predicted error.

9. Summary and Conclusions

The applications of sensor networks will be increasingly more attractive as the devices suitable for networking and the methods of communicating with networks will become easier to use and more intelligent. We have discussed some available hardware and methods for working with networks. In particular, we outlined some experiments developed for our prototype websensors and webprocessors (CTRLV4 and Miniservers). Substantial challenges remain. In order for IPcam, cams, mobile cams and microcontrollers for monitoring and control to be easy to use and intelligent, a more robust artificial intelligence solution is needed.

10. Acknowledgement

Jan Smid was during this research supported by the Department of Computer Science, Bowie State University. Petr Klan was during this research supported by projects of Ministry of Education, Youth and Sports of the Czech Republic 1N04002 and 1P05ME799, institute research plan number Z10300504.

11 References

- [1] Horák P. (2000): Laboratory Experiments for Control Theory Courses: A Survey. *Annual Reviews in Control* 24: 151-162.
- [2] Klán Petr, Hofreiter M., Macháček J., Modrlák O., Smutný L., Vašek V. (2005): Process Models for a New Control Education Laboratory. In: Preprints of the IFAC World Congress, Prague.
- [3]. Klán P. (2004): Instrument for Measurement and Control by PC. *Automa* 11: 34-36.
- [4] Klán P., Darbellay G. (1997): An information-theoretic adaptive method for time series forecasting. *Neural Network World*, Vol. 7, No. 2, 227-238.
- [5] Klán P., Marík J., Valáček P., Görner V. (1990): Adaptive PID controllers with monolithic microcontrollers. Praha, ÚTIA_SAV, 125 pages.
- [6] Lee P.L., Allen R.M., Cole G.R., Shastri S.S (2003): A modular laboratory for process control and process engineering. *Journal of Process Control* 13: 283-289.
- [7] Smid J., Obitko M., Bencur A (2005): Concept and Sensor Network Approach to Computing: The Lexicon Acquisition Component. Springer LNCS/LNAI 3825.
- [8] Mirollo R., Strogatz S.H. (1990): Synchronization of Pulse-Coupled Biological Oscillators. *SIAM J.APPL. MATH*, Vol.50, No.6, pp. 1645-1662.
- [9] Benthem J. (2000): TCP/IP Lean. Web Servers for Embedded Systems. CMP Books.
- [10] <http://www.ccsinfo.com/>
- [11] <http://www.maxim-ic.com/products/microcontrollers/tini/>
- [12]. Toffoli T. (2002): A Knowledge Home. Personal knowledge structuring in a computer world.
- [13] Cao J, William S.C., Lin D., Sun D. (2002). Internet traffic tends toward Poisson and independent as the load increases. *Nonlinear Estimation and Classification*. Springer Verlag, NY.
- [14] <http://www.acroname.com>
- [15] <http://www.parallax.com>
- [16] <http://store.irobot.com/>
- [17] <http://www.microchip.com>
- [18] <http://www.realtek.com.tw>
- [19] A. Rowe, Ch. Rosenberg, I. Nourbakhsh: A Low Cost Embedded Color Vision System. <http://www.cs.cmu.edu/~cmucam/>.
- [20] Golberg K., Siegwart Edts.: Beyond Webcams, An Introduction to Online Robotics. The MIT Press 2002, <http://mitpress.mit.edu/online-robot>