

Compact flexible router based remote experimenting system

T. Szepe*, Sz. Toth* and Z. Gingl*

* Department of Technical Informatics, University of Szeged, Szeged, Hungary

Abstract - The rapidly changing needs of modern education can be effectively supported by today's small, intelligent devices including tablet PCs, smart phones programmable routers and credit-card sized computers like the Raspberry PI. Since there are many available sensors on the market, it is rather easy to connect real world signal to these devices like temperature, light intensity, acceleration and many more. This way an exciting, cheap, transparent and very efficient software-based real experimentation can be carried out at different levels of education.

Our solution is based on a flexible, easily programmable popular router and a universal sensor-to-USB interface with open source software and hardware plus well documented educational applications. The router provides a computer platform for data acquisition and control while also serves as a remotely accessible device through its wired and wireless ports. In our presentation we shall briefly introduce the system setup and the connection of sensors and actuators. We have developed and tested a simple but efficient temperature control example educational application that can be remotely monitored and controlled via the internet. The transparent hardware and software can be used to attract students' attention, also supports the development of creativity by allowing modifications of the software and hardware configuration.

I. INTRODUCTION

One of the main problems of natural science and engineering education is to find methods that can keep the attention while most of the required and sometimes quite a lot of information is shared. Today information exchange and processing is fast, the students want to quickly understand and learn which can be a problem with many disciplines. Computers and graphical user interface software can be very efficiently used to simulate phenomena, and although they can provide user interaction also, they still appear rather artificial and do not cover real-world effects completely. As in the case of learning computer programming, mechanical and electronic design, many other fields need a lot of practicing, investigation of real systems and real signals. Comparing lecturing and laboratory practicing is often made in the frame of economical efficiency and of course less students can be taught in a larger room, the equipment's and experimental setups are costly in most cases and they are rapidly changing due to today's short time-to-market periods. On the other hand industry, science and teaching need creative participants who not only know but even understand the processes, who can find the required solutions quickly and reliably.

These skills can only be developed efficiently in smaller groups, by a close interaction between the lecturer and students who can even work in a project fashion.

Since there are more and more sensors, sensor interfaces and low cost user-configurable or programmable electronic devices are available, therefore widely usable efficient experimentation can be supported at different levels of education and in different disciplines including informatics, physics, chemistry, biology as well [1 - 6].

Just to name some examples, real world signals can be monitored and by the Arduino microcontroller kit, by sensors connected to the Beagleboard or Raspberry PI single board computers – the latter is especially developed to support creative mass education – or even by simple interfacing of sensors to the audio port of a notebook computer or to a smart mobile phone. Comparing these solutions to the professional educational kits shows much lower cost and availability, much more transparent experimentation and measurement solutions. Many students can easily have their own device and set of sensors, they can change the experiments, try their own ideas therefore their creativity and practical knowledge can be significantly improved. On the other hand, it is important to teach the proper usage of sensors and devices, to estimate the reliability, accuracy and to show the limitations in the same time.

Note that most of the above mentioned devices are network-ready, so students can even share raw data or processed information, they can learn about sharing data over the wired or wireless internet and other networks. They can see that a complete device can have real (hardware) and virtual (software) parts that can provide the different instrumentation and experimentation functions and they make the whole system together.

Besides low cost single board computers, microcontroller boards, Lego robots other small, very low cost and widely available devices can be efficiently used – like routers. Routers also have open source development platforms, have wired and wireless network interfaces and USB ports. Sensor and actuator to USB interfaces can turn them into a very powerful local computerized instrument or a remote laboratory server.

In a usual educational environment there is some kind of shared internet available, which is usually provided by using a local router. For the proposed system a reprogrammable router with USB connection is required. One of possible choices for an easily reconfigurable router

This work was supported by grant TAMOP-4.2.2/A

is the TP-Link TL-WR1043ND, which supports the popular open source software package called OpenWrt [7]. In this way an existing hardware is used preserving the original functionality of the router, only minimal additional hardware components are needed.

The router can handle simultaneous connections from many different sources, including smart phones, tablets, or other pc-s through wifi, or connected to internet it allows remote laboratory experiments. The flexible sensor-to-USB interface is easy to duplicate, hence it is completely open source including schematics and microcode, this means it can be reused to fully customize on purpose. The device uses cheap and widely available components to make, that means even students can reproduce or redesign it for themselves. It does not contain complicated BGA packaged chip like in the famous Rapsberry PI or BeagleBone which requires expensive tool to replicate and provides cheaper alternative to the commercially available counterparts [8, 9].

In the following we demonstrate the use of the popular TP-Link TL-WR1043ND router and the open source hardware and software sensor-to-USB interface in a remotely controlled experiment.

II. IMPLEMENTATION

A remote experimentation system has to be connected to the Internet for a long period of time and it must be capable of providing a static homepage with parameter handling capabilities. Usually it means a desktop PC will run a webcontainer and the rest part of the experimenting setup will be connected to it. Such a configuration is usually not enough cost and power efficient. Fortunately other solution is possible using existing infrastructure. In

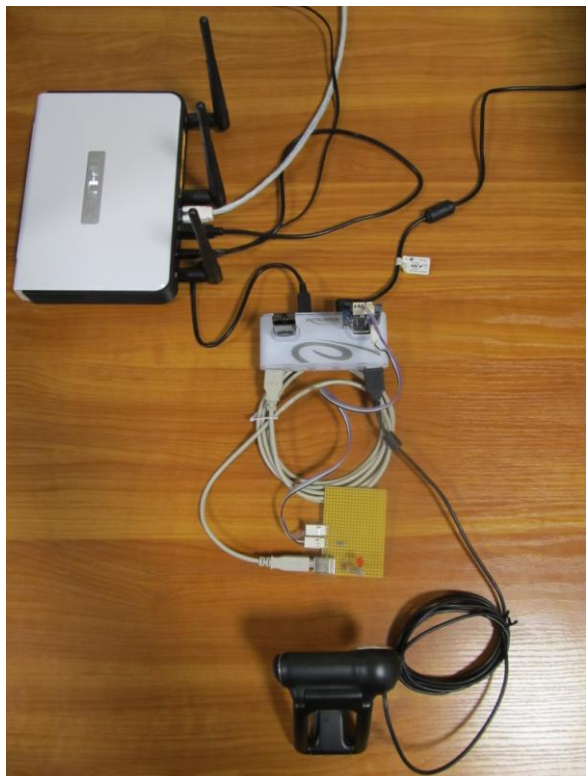


Figure 1. shows the hardware components

many high schools we can find wifi capable routers in or near the class rooms and some of them already equipped with a USB port. Running the proper firmware and software configuration a router can be used as a server device without losing its original capability to operate as an access point.

A. Hardware components

Using a commonly available router with USB port and good price/performance value was a top priority in hardware selection. It is advisable to choose a model supported by OpenWrt [7] or similar actively maintained project like Gargoyle [10]. Our choice for implementation was the TP-Link TL-WR1043ND router with three 802.11n capable wifi links, four Gigabit ethernet ports and a USB port. We used an active USB Hub in order to connect multiple USB devices with more than 3W of cumulated power usage. Built in flash memory was extended with a one gigabyte pendrive and a Logitech Quick Cam Pro 9100 webcam served as a video capture device. The outline of the connected hardware components can be seen in Fig. (1).

The Edaq-24 open source sensor-to-USB interface is based on the C8051F350 microcontroller from Silicon Laboratories [11, 12]. It incorporates a high precision 24-bit analog-to-digital converter, voltage reference, oscillator and rich set of digital peripherals in a single chip. Using a special arrangement of only a few passive analog components and ultralow on-resistance logic level MOSFETs most sensors can be connected without any additional circuitry. For low level signals the on-chip programmable gain differential amplifier provides the required signal scaling. Sensors that can be directly connected include thermistors, photoresistors, photodiodes, accelerometers, Hall-effect magnetic field sensors, Wheatstone bridge pressure sensors, thermocouples and many more. The very high resolution and low noise makes possible to measure extremely small signals, for example a simple Hall-sensor can be used to measure the Earth magnetic field quite accurately.

The device has digital output pins also, with the option of pulse width modulation (PWM) outputs to drive LEDs, small loads or even motors and large current actuators if external transistors are used.

The device connects to its host computer via a popular USB-UART chip (FT232RL). The chip was picked for its reliable operation and widely available drivers. Note the most Linux systems even do not require installation of drivers.

For users who do not have the possibility to build their own Edaq-24 sensor-to-USB interface some lower performance options are available like the Arduino Uno and Nano [13] or alternative setups can be developed with limited measurement capabilities using a sound card [14 - 16]. These solutions may require additional circuitry and even external amplifiers to allow the use of various sensors.

B. Software components

All software needed for the experimenting system can be categorized into three groups. Starting from the bottom

to the up first category is the C code running on the micro controller, which handles the measuring procedure with the ADC, the PWM output and all the basic communication on the USB channel. Second category is the server-client architecture with both part running on the router. Their task is to keep the USB resource available for communication and serve as many queries as possible to the Edaq-24 board. The third category contains the operating system of the router which is a linux kernel based firmware called OpenWrt.

On the Edaq-24 a simple communication protocol was implemented. Basically all valid command have to start with a "@" character and the following character will specify the command type. If necessary a third byte has to be sent as a parameter. To set the output to binary mode "@t" has to be sent. If we want to measure the second channel "@C2" is the command and a simple measurement will be done after "@s". In the last case the measurement can be read out as a byte sequence. The source code for the Edaq-24 and other custom written source codes and configuration descriptions are available in [17].

A daemon server application was developed to run as a background process on the router to preserve the hardware resource for USB communication. Keeping in mind that many queries are possible in a short period of time, it is not a good idea to open a writer and a reader stream every time when communication needed, therefore the server process will open them only once and keep them alive. Communication is possible with the server process through a small portion of shared memory, where the implemented protocol is similar as the one on the Edaq-24 board with some extensions. Additional command characters are used to make direct messages possible with the router like terminate the server process with "@X". Thread safety for the shared memory is guaranteed with a semaphore that ensures the given command was processed by the server thread. Passive waiting for a new command is achieved by signaling using OS call and whenever possible the server thread waits in sleep mode. The shared memory is initialized by the server with its own process id which is later used by the client thread in the signaling process.

OpenWrt firmware was configured using the detailed description found in [7]. To reproduce a working configuration some additional install steps are required. Most importantly a pendrive should be mounted automatically at the booting time of the router where most of the installed software should be deployed. In order to access them as the router own packages, all deployed files should be linked to the bin folder of the root file-system. In this way the router file-system will be extended. In order to run the web interface lighttpd and perl script interpreter should be installed. The webcam requires mjpg-streamer and the USB communication may require libusb and libftdi libraries. All important configuration steps can be found in [17].

III. DEMO APPLICATION

To demonstrate one possible use of the proposed setup, a demo application was designed with which basic

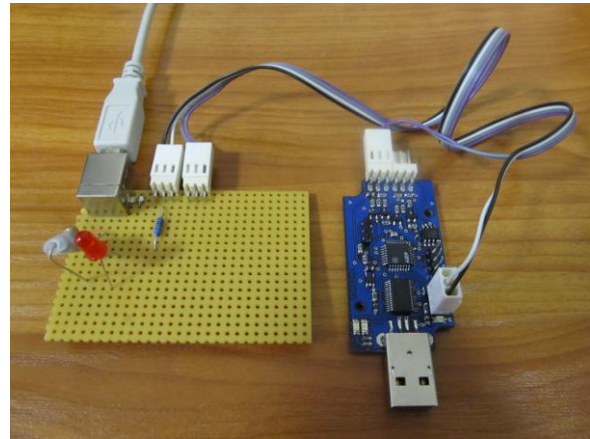


Figure 2. shows the demo board with the Edaq-24

control algorithms can be examined. The aims for the controllers are to reach and maintain a specific temperature by applying a certain voltage across a heater resistor.

A. Demo hardware

The demo heater board role is to give a realistic example of a delayed controlled heating system similar to a building with a thermostatically controlled heater. Thus the main component of the board is a thermistor coupled to a heater resistor as shown in Fig. (2) connected to the Edaq-24 board.

The resistance of the cheap thermistor sensor will correlate with the temperature of the environment. The accuracy of the temperature readings will be under 1°C with the used 10Kohm model in room temperature, which makes it an adequate sensor for the task. Note that, the 24bit ADC built into the micro-controller makes it possible to use high precision sensors without preamplification, thus the thermistor is directly connected to the input of the data acquisition board through an inner pull-up resistor.

The heater resistor requires much higher current than the PWM output could provide, so a transistor will switch the 5V presented on a USB port. The separation of the switched voltage through the heater resistor is a practical concern, because the micro-controller on the data acquisition board already less then a hundred milliamps. Making it more attractive an LED is switched on simultaneously with the heater to make it visible for the experimenter. Remote users can watch the blinking using a regular USB webcam.

Note that using a second PWM output to control an electric fan could be used to alter the heat loss similar to the dropping of the outside temperature raise the heat loss of buildings in winter. With basic electrical knowledge the proposed heater board can be built with a few cheap components even on a breadboard with the possibility to further upgrade with new features, like air ventilation.

Temperature measurement application:

To make the measurement please hit the Execute button!

Current temperature is: °C

Execute

On-Off control test application:

Please enter the target temperature:

Note that, between 10.0 and 40.0 values are accepted, otherwise the value is considered 25.0

Also note that, the control sequence will continue to run until 10000 cycles finished, or any other command will terminate the controller.

Execute

PID control test application:

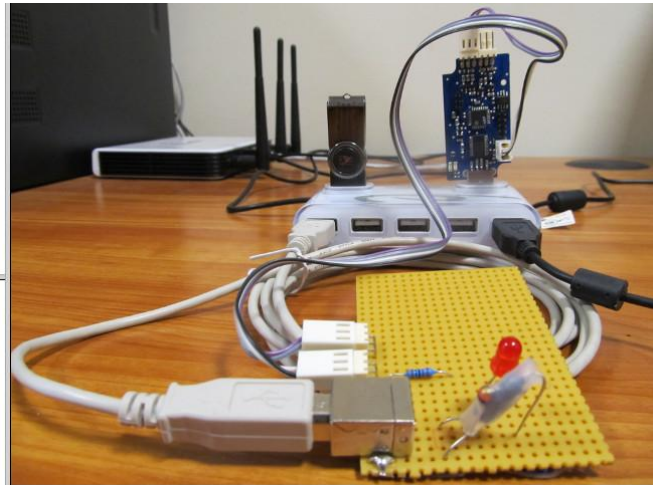
Please enter the target temperature:

Please enter the PID controller parameters: P: I: D:

Note that, all notifications are valid as for the on-off control.

Also note that the output of the controller is between 0 and 255, so choose the parameters carefully.

Execute



Data downloader and message based application:

Download the latest finished control dataset as a CSV file [here](#).

Send simple command to the Edaq board.

Please enter new command: and parameter:

Result of last command:

Note that if a command does not have parameter, then leave it empty.

Execute

Figure 3. Shows the screenshot of the remote application

B. Demo software

A web based user interface was developed to illustrate the difference between the on-off or also known as bang-bang controller and the variations of the PID controller.

Main components of the web UI are depicted in Fig. (3), which is a simple HTML that uses frameset for the subtasks. Each frame source comes from a separate content provider, which makes the system modular and more robust. All user inputs are HTML forms with different number of text inputs submitted to the server. A PERL script handle the posted data and works as an interface between the HTML code running in the browser and the client application running on the router.

The top-left section contains a simple application with which a remote temperature reading is possible. By clicking on the button the measured temperature will be shown in the text.

On-Off controlling can be started using the middle-left part of the web UI, where a target temperature is required as an input field. By clicking on the button the controller will be initialized to keep the resistor temperature at the given value. Started controllers will keep going until a certain iteration number has been reached or any new message was received by the server thread. When the controller stops, it will write a CSV file containing all temperature measurements target temperature and control

signals with timestamps in milliseconds. The CSV file can be downloaded using the bottom-right section of the UI.

PID controlling application can be found in the bottom-left section. It can be used to test simple proportional, PD, PI or PID controllers by giving the proper parameters. Setting the target temperature is also available in this subframe.

The top-right part contains a video-stream of the setup allowing instantaneous visual feedback of the experiment. Note that in order to see the quick blinking of the LED higher frame-rate may be required than the default when using mjpg-streamer. Also note that higher frame-rate will increase the required bandwidth needed for smooth playback.

Main function of the application in the bottom-right corner is to enable simple communication with the experimenting setup. All implemented commands with or without parameter can be called here. Second role of this part is to offer a link with which the last logged controller data can be downloaded.

To illustrate the difference between the on-off and a proportional controller a temperature measurement experiment was performed. Identical parameters such as the target temperature available time frame and similar starting temperatures were set highlight the characteristic difference between the two controller types. The results with the measured and target temperature and the control

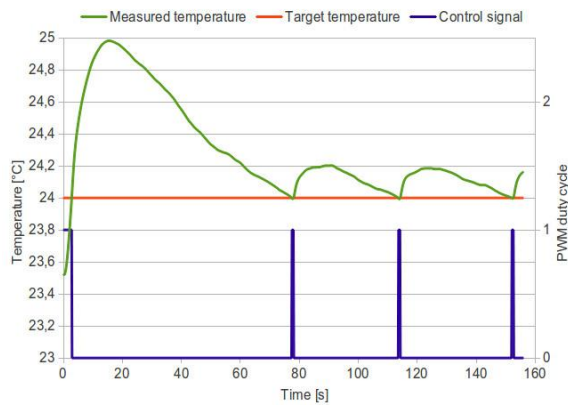


Figure 4. shows the characteristics of the on-off controller

signal are depicted as a function of time in Fig. (4) for the on-off controller and in Fig. (5) for the PID controller. The results clearly showed that the on-off controller had a significantly higher overshoot compared to the PID controller.

IV. RESULTS AND CONCLUSION

Cheap, broadly available hardware and some custom made open source extension boards with open source software aid highly efficient education. They can be used in various real experiments for example teaching many disciplines and showing that the experiments are transparent and scalable. Students can have their own setup, they can realize different instruments using the same hardware just by replacing the software that performs most of the instrumentation functions. The low cost, multi user experimenting equipment significantly reduces the need for alternative real laboratory resources.

In our paper we have shown a simple way to use a popular router and an open source sensor-to-USB interface for real and exciting remotely controlled experiments. The system can be used in various educational fields. Undergraduate or graduate students of engineering can use the remotely controlled laboratory experiment to examine the behavior of a real system even from home at any time, no further laboratory apparatus is needed. The lecturer can even ask for doing homework as well. The transparent system provides scalable education:

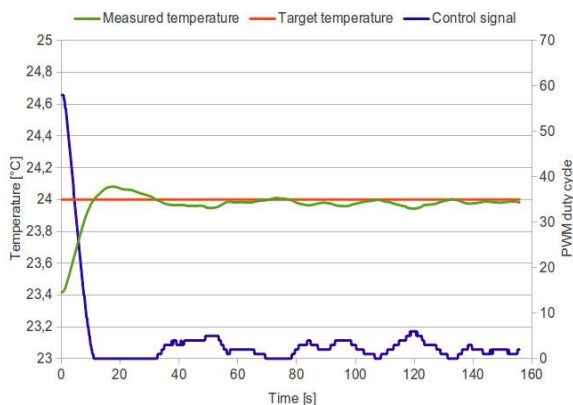


Figure 5. shows the characteristics of the PID controller

students can use already developed experiments; they can be asked to add further sensors; they can port the method to many other real mechanical and electronic systems; they can be involved even in software and hardware modifications and developments. Note also that larger projects can serve as a basis of the students' thesis both at the bachelor or master of science level, their creativity can be significantly improved. Note that the easily available, cheap, transparent computer controlled system can support teacher demonstration and home experimentation in the education of students at age of 12 to 18 years also.

The whole solution is open source therefore it can be used as a basis for many additional measurements and experiments. Note that it can be easily ported to other popular very low cost platforms including the Beaglebone and Raspberry PI single-board computers.

ACKNOWLEDGMENT

The publication is supported by the European Union and co-funded by the European Social Fund. Project number: TAMOP-4.2.2.A-11/1/KONV-2012-0073.

REFERENCES

- [1] Kopasz K, Makra P, Gingl Z. Edaq530: a transparent, open-end and open-source measurement solution in natural science education. *Eur. J. Phys.* 2011, 32:491-504.
- [2] ComLab: Computerised laboratory in science and technology teaching. <http://e-prolab.com/en/index.html>
- [3] Stanislav Holec, Martin Hruška, and Jana Raganová. Integrated science through computer-aided experiments. *Informatics in Education* 2004, 3:219–228.
- [4] F Schauer, F Lustig, and M Ožvoldová. ISES—Internet School Experimental System for computer-based laboratories in physics. *INEER special volume: Innovations 2009 - World Innovations in Engineering Education and Research*, pp:109–118.
- [5] B Murovec and S Kocijancic. Educational data acquisition system with USB interface. *EUROCON Computer as a Tool. The IEEE Region 8*, 2003, 1:490–494.
- [6] R Sallier, R Missler, and A Schütze. VenDASys — A versatile experimentation platform for educational purposes. *Education Engineering (EDUCON)*, 2010, IEEE pp:205–210.
- [7] OpenWrt Linux distribution for embedded devices <https://openwrt.org>
- [8] Vernier LabQuest. <http://www.vernier.com/labquest/>
- [9] Pico Technology: Oscilloscopes and data loggers for education. <http://www.picotech.com/education>.
- [10] Gargoyle Router Management Utility <http://www.gargoyle-router.com>
- [11] Gingl Z. A photoplethysmograph experiment for microcontroller labs. *INTERNATIONAL JOURNAL OF ELECTRICAL ENGINEERING EDUCATION* 2012, 49(1):42-60.
- [12] Gingl Z. Fabricate a high-resolution sensor-to-USB interface. 2011, *EDN* 56 (22): 54-57.
- [13] LIU, Jianhua; GRANT, Christopher; GALLAGHER, Donald. Adding new functions to the remote airfield lighting system. In: *IEEE SoutheastCon 2010 (SoutheastCon)*, Proceedings of the. IEEE, 2010. p. 106-110.
- [14] Chris Butlin. Electronic media: DrDAQ - an educational datalogger and virtual oscilloscope from Pico Technology. *Physics Education*. 2001, 36(1):85–86.
- [15] Hunt, M. B. & Dingley, K. Use of the sound card for datalogging. *Phys. Educ.* 2002, 37:251-253.
- [16] Gingl Z., Mingsz R., Makra P. & Mellár J. Review of sound card photogates. *Eur. J. Phys.* 2011, 32:905-924.
- [17] Remote laboratory applications <http://www.inf.u-szeged.hu/robotics/remote>