



REMOTE CONTROLLING ELECTRONIC DEVICES BY INTERNET

PAVLAS, L[ibor]

Abstract: *This paper writes about system developed for controlling electronic devices such as motors, bulbs, sensors and servo motors by remote control system operated by microcontroller Stellaris EKS-LM3S8962, connected to internet. By this system you can securely control electronic devices all over world just by typing right web address in your internet browser.*

Key words: *remote controlling, microcontrollers, Stellaris EKS – LM3S8962, internet*

1. INTRODUCTION

There was a problem, how to be able to operate some electronic devices in our institute, without presence any operator. Necessary condition was to be able easily check status of driven system and easy operating interface. Because of this, I decided, that I need some device to be connected by RJ-45 to internet and to be able to run some operating system for embedded devices with network support.

2. OPERATING SYSTEMS FOR EMBEDDED DEVICES

There is a plenty of operating systems for embedded devices and there is also plenty of easy to use embedded devices (Ganssle & Barr, 2009). So I had to make some research to find out, what device is able to be connected on internet without another hardware making all task harder. Also this device had to be supported by some operating system with internet support.

The operating systems for embedded devices I tested are:

- uC/OS-II
- uC/OS-III
- FreeRTOS
- uIP

First three operating systems runs fine, without any problems, they had a lot of features to run device with low power, but problem is with internet support. You can implement your own internet drivers, but it is not so easy (Labrosse, 1999). FreeRTOS operating system is free for use for all purposes (<http://www.freertos.org/>).

Because of this I decided to use the last one. I fact it is not a real operating system, generally it is mostly just library with functions to run a web server on small embedded devices.

3. HARDWARE TO RUN INTERNET SERVER

Another problem was to select the right hardware, able to run internet server and operate all our devices in our hardware laboratory. We need those features:

- Internet connection to connect our hardware to internet.
- RS 232 port to operate robotic hand handling web camera guarding our project.
- Universal output pins to turn on and off our devices.
- Universal input pins to check status of sensors.
- Realtime debugging features.

Also this device has to be cheap and consumption of electricity should be low. Device should have also some display to show actual status and errors helping to develop our software.

The number of free input and output pins is also important. Because we want to use some already made evaluation kit, there is very often, that on general purpose input and output pins are used for some additional devices, such as speakers, leds, buttons, AD and DA convertors, PWM generators and so. And sometimes it is impossible to disconnect those supporting devices and we cannot use all pins we need (Labrosse, 2002).

Another thing, we have to solve is, what kind of developing software we will use. There is a plenty of developing software not for free and this software is mostly very expensive. So I had to choose evaluation kit which can be programmed by free software.

4. STELLARIS EKS-LM3S8962 EVALUATION KIT

After I did my research, I chose evaluation kit from Texas Instruments called Stellaris EKS-LM3S8962, runned by microprocessor ARM LM3S8962. Features of this kit are:

- Stellaris LM3S8962 MCU with fully-integrated 10/100 embedded Ethernet controller
- OLED graphics display with 128 x 96 pixel resolution
- User LED, navigation switches, and select pushbuttons
- Magnetic speaker
- LM3S8962 I/O available on labeled break-out pads
- Standard ARM® 20-pin JTAG debug connector with input and output modes
- Standalone CAN device board using Stellaris LM3S2110 microcontroller
- Ethernet cable, CAN ribbon cable, USB and JTAG cables
- RS 232

This embedded device is able to be programmed by free development kit by Texas Instruments called Code Composer studio V4 (<http://www.ti.com/tool/eks-lm3s8962>).

5. HARDWARE REALISATION

First of all, I had to find out, how many free general purpose input / output pins we can use. There is 7 x 8 pins, but not all of them are connected to general purpose connector and a lot of them have more functions (AD and DA converters and another functions) (Labrosse, 2009). So I had to decide, what functions I will use, and what functions can be switched off.

I wanted to use LCD display, Ethernet connector, RS-232, buttons, couple of ADC connectors and speaker. The rest of general input output pins are free for external use.

In our laboratory experiments we need to be able turn on the power supply, light, two electromotors in two ways (front and back drive). During experiment we need to measure by two AD converters and two digital inputs. So we need four input pins and six output pins to operate our experiment. All this pins have to be separated from driven electronic devices to protect microcontroller.

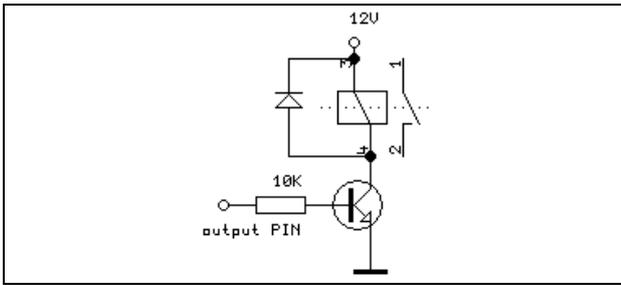


Fig. 1. Output schematics

6. SCHEMATICS

Input and output schematics are simple. For output (Fig. 2.) is used one transistor to amplify signal a relay to separate power part and digital part because of noise interference. So we need six transistors, relays, diodes and resistors to make output.

For Input (Fig. 2.), there isn't any need of relay, for separating parts, power and digital, there can be only optical separation. Because, there is used only optical separation, input signal can has faster changes (higher frequency). For Input we need just four resistors and two optical separations.

For connection internet a RS-232, there isn't any need of additional electronic devices, so there is only those two schematics need.

7. SOFTWARE REALISATION

Program driving this system was made in Code composer studio V4 in C language. The most important library is uIP for microprocessor ARM LM3S8962.

My task was to develop simple web page, where the user can check all states of driven system (read inputs and signals from AD converters). This simple web page is stored in microprocessors memory and when some user connects on public IP of this system, he gets prewritten page, modified by actual state of driven system.

To protect system from hackers attack, there is password protection, protecting some output pins and it is up to final user, what part of system should be protected by password and what part can be readen and modifiable by public. Because of this, part of system can used in public system to show (for example) our laboratory results.

In our final version of our system, there is only one pin, guarded by password. This pin supposed to turn on power supply. Because of this, if the operator with acces rights to turn on our experiment, turn on the power supply, the others (friends, students etc.) can try to use our system and check feedback of our simple demonstration example. If anybody from public users will feel, that there can be some error, causing problems, he can turn off all system to protect electronic devices, including robotic hand.

In system, used in real laboratory experiments, this public acces can be seriously dangerous, but there isn't any problem to change small part of source code to change acces rights and nobody without authorized rights can check our experiment.

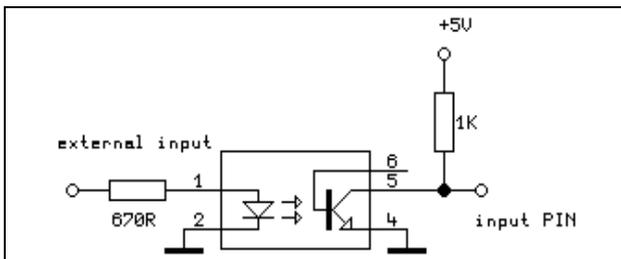


Fig. 2. Input schematics

8. FUTURE IMPROVEMENTS

The biggest disadvantage of this system is the protection against hacking attack. The simplest way, how to protect system is to set password. But this isn't the safest way. Another improvement will be to set a list of allowed IP addresses, from which any user can read status of our system. This can divide users to three parts, first part is users from right IP, with password, those are administrators. After there are users from right IP without password, they can be just users, reading and watching status of our system and last part of visitors are the one from invalid IP addresses, they are attackers and they will not get any part of web page.

Another important safety improvement will be improvement against DOS attack (<http://www.us-cert.gov/cas/tips/ST04-015.html>). Now, system response is slow, about 1 second. So attacker can make system not responding just by sending packets every second and this is my future work, making it much more resistant.

9. CONCLUSION

During our laboratory experiment, we build remote internet driven system, which can be used without help of any another computers, or complicated electronic devices. By this system, we are able to operate our laboratory electronic devices without any human presence in this laboratory. The user with acces rights is able to use our system all over the world, just with internet access and internet browser.

There is not a problem to fastly reconfigure system and drive absolutely different laboratory experiment.

By our system, the user can do both, analog and digital measurement, operate simple electronic devices, such as motors, lights, turn on also power supply sources, but also some more sophisticated devices, such a servo driven robotic hand or operate camera connected on internet, guarding all laboratory experiments.

Because, there is also abuse protection with rights modification, we can simply decide, what part of our experiment can be used by anybody and what part is password protected.

10. ACKNOWLEDGEMENTS

Many thanks to my colleagues at University of West Bohemia: Petr Kratochvil, Stepan Jenicek and Richard Lipka. Many thanks go also to my advisor Doc. Ing. Vlastimil Vavricka, CSc., for his continuous support in the Ph.D. program.

11. REFERENCES

- Ganssle, J. & Barr, M. (2009). *Embedded Systems Dictionary*, CMP Books, 1578201209, Gilroy CA, USA
- Labrosse, J.J. (1999). *Embedded Systems Building Blocks, Second Edition: Complete and Ready-to-Use Modules in C*, CMP Books Distribution Center, 978-0879306045, Gilroy CA, USA
- Labrosse, J.J. (2002). *uC/OS-II The Real-Time Kernel, second edition*, CMP Books Distribution Center, 1-57820-103-9, Gilroy CA, USA
- Labrosse, J.J. (2009). *uC/OS-III The Real-Time Kernel*, Micrium press, 978-0-9823375-3-0, Weston FL, USA
- ***(2011) <http://www.freertos.org/> - The FreeRTOS Project, Accessed on: 2011-09-20
- ***(2011) <http://www.ti.com/tool/eks-lm3s8962> - EKS-LM3S8962 Evaluation Kit Accessed on: 2011-10-15
- ***(2011) <http://www.us-cert.gov/cas/tips/ST04-015.html> - Understanding Denial-of-Service Attacks Accessed on: 2011-10-13