

Personal details

Name:

Laszlo Benke

Address:

Europe, Hungary, Budapest, H1093

E-mail:

benkelaci88@gmail.com

eLinux.org username:

benkelaci

Skype and IRC nickname:

benkelaci

Studies:

Budapest University of Technology and Economics (HUNGARY)
Electrical Engineering (Embedded and PC based systems) M.Sc.

Language:

Hungarian (second: English)

Availability:

From April 21st to August 31th, 25-30 hours/week for the GSoC, with 2 week pause (1 week in June to preparing for state exam, 1 week in August for holiday).

Github:

<https://github.com/benkelaci/gsoc>

HelloWorld preliminary task:

<https://github.com/benkelaci/gsoc/blob/master/helloworld.bin>



Background information

I am currently a post graduate student in Electrical Engineering M.Sc. (embedded systems) at Budapest University of Technology and Economics (BME), in Hungary. I did my B.Sc. degree in the same place (2011).

Projects and thesis themes:

- Thesis (B.Sc.): PowerLED controlling by PSoC (control software – C, PSoC Creator, PSoC Designer and PCB designing – Altium Designer, DMX protocol)
- Melfa robot controlling by body movement (C#, Melfa BASIC, Kinect)

- QB50 project: measure system for face-temperature of the CubeSat satellite (STM32, ARM microcontroller - C, Altium Designer and analog design). In this project mostly I am programming STM32 board. I had to byte-level-program (not with the standard library) the digital protocols (SPI, I2C, UART) to improve to space program. And I made a high reliable DC/DC converter (controlling via STM32 board).
- Home security system: A webcam that connect to a router. In the router run an OpenWRT embedded linux operation system. I made a daemon, which watched the front door (motion sensing) and wait for the correct entrance code (via keypad).
- Thesis (M.Sc.): Integrated HVAC control and protocol converter on the basis of open software. Robotház Kft. Home automation system and PLC system integration (Linux, Qt, C++, PLC-ST). On the other hand there was a big embedded Linux task: upgrade a Qt program, that run a touch screen. I made a Modbus TCP communication module, that can connect to a PLC and other Modbus TCP compatible device. In this subproject I was dealing with cross compiling (e.g.: simulate arm device, toolchain installing, uploading issues etc.).

I have 4 year working experience (part-time job):

- CTDI Kft, Budapest – program developer: 2010-2013
workflow system development and maintain (PHP, HTML, Javascript, PostgreSQL)
- Evosoft Hungary, Budapest – embedded software engineer: 2012-2013
Industrial Sector, Sinamics development (C++), Main project: Relaycard design and program development (C#, C++ and circuit design)
- Robotház Kft, Budapest – PLC developer: 2013- (current job)
C#, PLC, Cloud computing, Linux development

Programming skills: (1-weak knowledge, 5- strong knowledge)

- C/C++ (4), C# (4)
- PLC (4), Linux Development (3)
- Assembly (2)
- PHP (3), JavaScript (2)

Hobbies:

- playing squash, cycling
- reading books (mostly physics and biography, renewable energies)

Motivation

In my career it was difficult to decide for me which orientation would be the best for me: programming or staying my mayor (electrical engineer and circuit design). Nowadays the electrical engineer and the informatics are inseparable. That is why I always stand for improve both skills (via projects, thesis or in the job). I think this organization is the best for me, because I will need my circuit design knowledge and the programming skills as well. Unfortunately I will end the school in this year (June), and this means this year is the last chance to participate the GSoC.

When I had a practice (in school), I enjoyed the working with embedded device (mostly we used Beagleboard). We tried some embedded operation systems (uC-OS II, QNC, embedded Linux) and made embedded webserver. The webserver solution was my favourite, because I upgraded it: I used at home. In winter I could check what temperature was in my room, and I turned on the electrical heating device if needed.

Unfortunately I have not participated in an open-source project. Maybe the Touch screen program development (I described in M.Sc. thesis) was an “open-source project”, but in this situation the main purpose was commercial and the deadline was too short. You can see the project on my github, but the result was only 2 new file. The cross-compile and the simulation took a long-time (1 month). It was poorly documented, but the developer (from BTicino company is Italy) helped me.

Project proposal

Project Title

Digital protocol stimulus generator for BeagleBone, with web pages appearance and controlling. Plus document on Beaglebone community.

Summary

Main purpose is to help developer make communications with devices. If there is few device simulator (stimulus generator) then there is no need to purchase real device and the development time can be reduced.

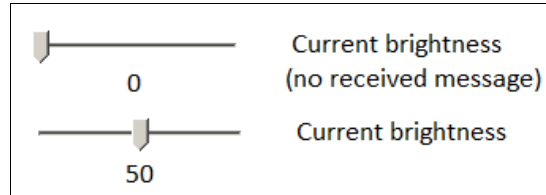
I am going to make these protocols and peripheral device: UART, I2C, SPI, PWM, ADC, USB.

On the other hand I will make a simple time diagram visualizer for certain protocol. This means the signals (e.g.: CLK, MISO, MOSI, SS) store while communications is being and shows in the webpage. The PRU firmware load implementing is my idea to solve this.

Finally: it is important to usable to document the using on http://www.elinux.org/BeagleBone_Community.

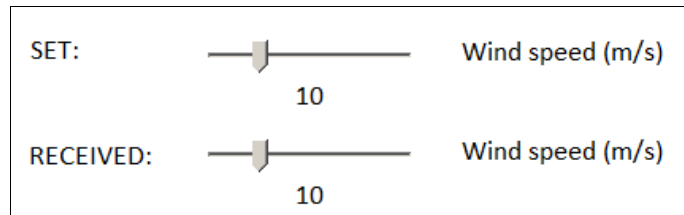
Each protocols and peripheral device belong to 2 example. Both example belongs to a definite hardware device. Fortunately I have got these device.

- UART: the DMX and Modbus protocol based on RS485. The example will be a simple DMX device simulator (brightness control via DMX control packet) and a Modbus (e.g.: wind speed meter). In the DMX case there is no response message, this is only controlling. That is why it is enough to show the result. Like this:



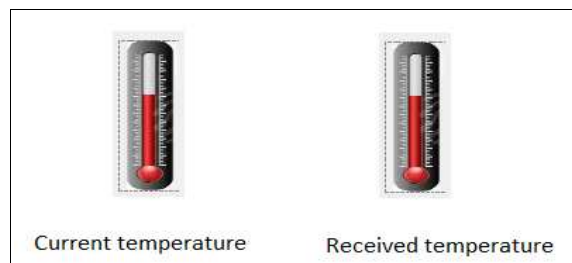
DMX controlling example appearance in the webpage. First example shows if there is no received control message from the user program module yet.

In the Modbus case there will be a simulated value and received value. If the communication is ok, then both result will be the same. Like this:



Modbus controlling example appearance in the webpage. In the first line you can show the set value (by the user) and in the second line you can show the received and converted message.

- I2C: First example: thermometer. You can set the temperature and show the received temperature value (like Modbus case). Like this:



You can set by clicking the thremometer (left) and show the received message)

The second example will be an EEPROM programming. You can set the bytes value (maybe there will be only 4 byte for easy understanding). Like this:

EEPROM content:	<input type="text" value="B"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	
Read content:	<input type="text" value="6"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="button" value="Read"/>
Write content:	<input type="text" value="B"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="button" value="Write"/>

I2C EEPROM handling example appearance in the webpage. In the first line you can show the current EEPROM content in the second line you can show the readed content (in this case the read show the past values) and third line you can show the required (and activated) write-content.

- SPI: Accelerometer and ADC device. The appearance is similar than previous examples.
- PWM: LED brightness controlling and DC motor speed controlling.
- ADC: using TMP35 devices and potentiometer.
- USB: I would like to use Kinect sensor. A first step would be integrating webcam frame grabbing into presenting the captured images on a web page. If there is enough time I will be use motion sensing part.

Timeline

There will be 4 main phase:

Phase 1 (introduction): 22th of April to 18th of May

After accepting my proposal (hopefully) I am going to review all related documents and try to demos in BBB. More detail:

- Familiarizing with the Bonescript I/O interfaces (Node.JS, JavaScript)
- Testing the digital protocols (C), start making a simple signals logger with PRU (assembly level)
- Connect with related person (mostly my mentor) and discuss the detailed idea (to efficiency). I think, because this is an open-source project, then the cooperation is important factor to make a usability project – I know I do not know everything, but at least I know where to look.
- I plan to send a report every week to my mentor.

Phase 2 (coding until mid evaluation): 19th of May to 22nd of June

- Week 1-2: Make user-friendly web based GUI for the UART testing example (Bonescript) like in this proposal. Dealing with SPI, PWM communication.
- Week 3: Pause (prepare to my state exam and take)

- Week 4: GUI for SPI and PWM. Documentation harmonizing.
- Week 5: Simple UART time diagram maker (PRU)

Phase 3 (mid-term evaluation): 23rd of June to 27th of June

- Week 6: Presentation and discuss the results. Repair if needed.

Phase 4 (coding after mid-term evaluation): 27rd of June to 10th of August

- Week 7: Complete UART time diagram maker fixing (PRU)
- Week 8: I2C and ADC communication implementation and web based GUI. USB example.
- Week 9: Pause (holiday).
- Week 10: Testing and testing and fixing. If I have enough time, then improve the time diagram maker (SPI or PWM).
- Week 11: Prepare to finalize project.

I and the community

I will improve the bonescript with new modules. If the newby want to develop program (mostly using digital protocol) then this project can help them. Important part is the documentation, of course. The documentation can be as a tutorial.

I will have a chance to using PRU. My plan contains to optimization and documentation as well.

I had a conversation with Jason Kridner (jkridner@gmail.com). He got me many instructions and I am very thankful for it. His advice was that I must use PRU to make a simple logical analyser. Firstly I would liked to make Matt and Hunyue idea (http://elinux.org/BeagleBoard/GSoC/Ideas#Documentation_for_BeagleBone_and_BeagleBone_Black), but during the conversation I came to realize that I can use bonescript to improve this idea. Thomas Cort (linuxgeek@gmail.com) and Victor MV (v.mayoralv@gmail.com) helped me as well (another e-mail thread).

I heard there is another person who made similar project (his main purpose is to make a complete logical analyser). I do not know if there can be cooperate with he, but I am open any ideas. To help each other.

If I will get stuck on our project: Firstly I am search on the internet and ask in the forums (eLinux, beagleboard). Finally I ask my teacher in our city to help me. He is so helpful and he is well-learned and creative. But I think this community is hard work and creative as well.