Contest Entry M1685

ReVaLuaTe

Renesas
Valuable
Lua
Terminal

1. Abstract

With the continuously increasing number of processors and associated tools, it would seem that building a new project from scratch should be an easy task even for the uninitiated. While this statement holds a certain amount of truth, the things are not quite as ideal as they appear to be. After the processor is selected (which is not an easy task to begin with), the real work begins. Understanding its architecture, understanding the tool-chain, even understanding the assembly language if required for the project and then putting all these things together is usually a real challenge for just about anyone, especially for the ones with little or no prior experience in the world of microcontrollers. While the challenge in itself is most of the time very interesting and instructive, it also translates into loss of time, and many time loss of money. This, of course, is not desirable in a real world project. It would be so nice to start everything from scratch in a few moments, and then concentrate on writing the application rather than understanding and overcoming the quirks of the toolchain/microcontroller duo. It would be equally nice to minimize the delivery time after the prototype is complete as much as possible, keeping the total number of hardware changes low. Since we're at it, it would also be great if the application could be written in a simple, yet powerful language, preferably also modular and fast. And if we could reduce the auxiliary components required to develop the final application to a minimum. One would probably agree that a low cost solution providing the above requirements is at least interesting. But is it possible to build such a powerful and useful device at a low price?

That said, it is a good time to meet **ReVaLuaTe**. It is a development platform based on the Renesas M30626FHPFP processor (more exactly, based on the excellent SKP16C62P development platform from Renesas). And it's a bit more than your average development platform. Following is a list of features that should prove the previous affirmation without a shadow of the doubt.

First, the platform is fully independent. That is, you don't need a PC to develop and run your programs. You just plug an input device (at the moment a standard PS/2 keyboard) and an output device (at the moment a 320x240 LCD screen) and start writing you code using the build-in editor. The code is written in **LUA,** a high-level programming language with a set of unique features that will be presented

later in the document. Both a **LUA** compiler and an interpreter are present on a **ReVaLuaTe** board. You can test the code as you write it. When it is finished, you can use the same board to run the code, without any modification. With these features, code developing (and even code developing directly at the customer site) is greatly simplified.

ReVaLuaTe is also user friendly and full-featured. Besides the already mentioned PS/2 and LCD connections, **ReVaLuaTe** also supports **MMC/SD** cards with FAT12/FAT16 file systems in read/write mode. New input and output devices can be easily added in software because of the modular architecture. The built-in editor provides all the basic functions required to write a text file (for source editing) and a bit more.

Things become even more interesting though, because the platform is also self-reproducible. That is, given two **ReVaLuaTe** boards and a serial cable between them, downloading code from a board to the other becomes an easy and painless process. You can also exchange files from one board to the other, read memory contents and even open and manipulate "remote" files (files on another board) using the exact same functions that are used to open and manipulate files on the local file systems. "Systems" is not a misprint; there are actually more of them. This will be detailed in a later section.

ReVaLuaTe is also very modular. While you need a keyboard and a LCD to write and test your programs, they are not required when running the final programs. If you combine this with the concepts presented in the previous two paragraphs, you can suddenly imagine a very interesting situation. Imagine a **ReVaLuaTe** board with full configuration (LCD, PS/2, MMC/SD interface). We'll call this board the **full board.** Now, imagine a **ReVaLuaTe** board without LCD, PS/2 and MMC/SD interface. We'll call this the **reduced board.** You can build the two versions on the same PCB, of course. You would generally build a single full board and more reduced boards. You can do all the prototyping on the full board, and afterwards transfer the final code and the required files on the reduced board (remember, the platform is self-reproducible and all that is required for this transfer is a simple serial cable) and actually let the reduced board run the final application, without any hardware modifications. So you can do everything (prototyping and deploying) with **only one board layout!** Supposedly, you'll have to change a thing or two in the application specific I/O hardware section, but the problem is simplified nevertheless. You just

need a pile of these boards, and a single fully equipped one. Develop your code, transfer it to a reduced board, leave it at the customer site, and you're all done. Of course, later you can come back and change the files/memory on the reduced board using the serial connection. Please note that the modularity is also present in the software part, not only in the hardware. In order to make more room for files and user programs, one can choose what parts of the code are to be downloaded from the full board to the reduced board.

ReVaLuaTe is also low-cost. The M16C62 processor, with its excellent performance and capabilities, is the one that makes this possible.

ReVaLuaTe has more features than the ones already presented, like the possibility of updating the firmware directly from the SD/MMC card, enhanced (buffered) file system support, a low footprint Flash file system and much more. All of these are to be presented in the later chapters.

The hardware subsystem

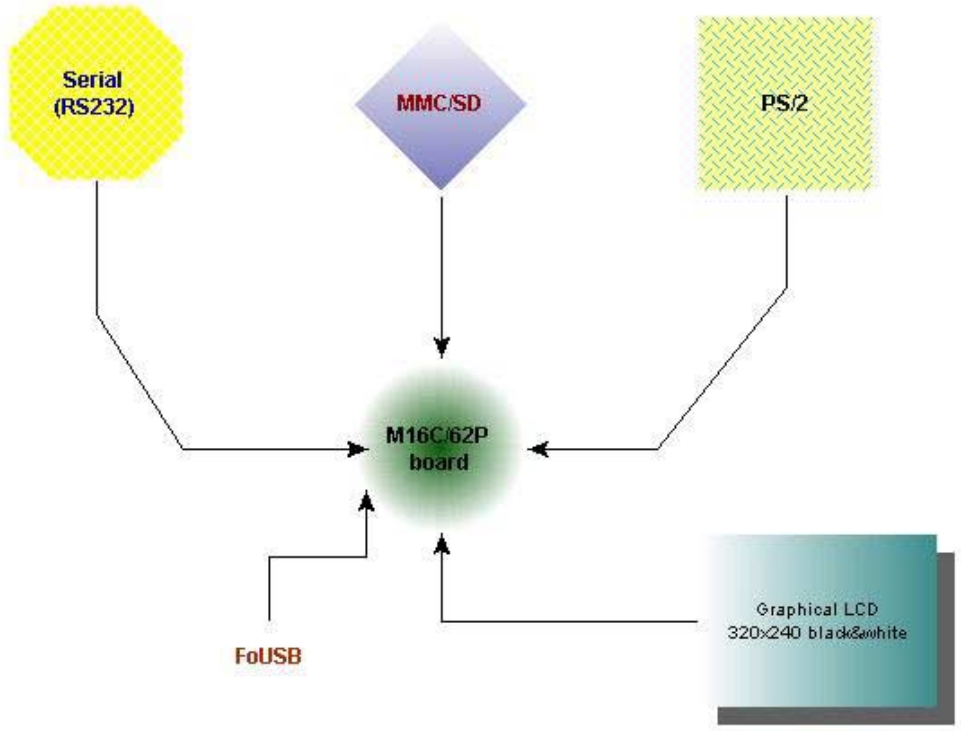

Figure 1 - Hardware subsystem

The ‘heart’ of the system is the M16C/62P evaluation board from Renesas. A few modifications were made to the board in order to make it suitable to this project.

The software subsystem

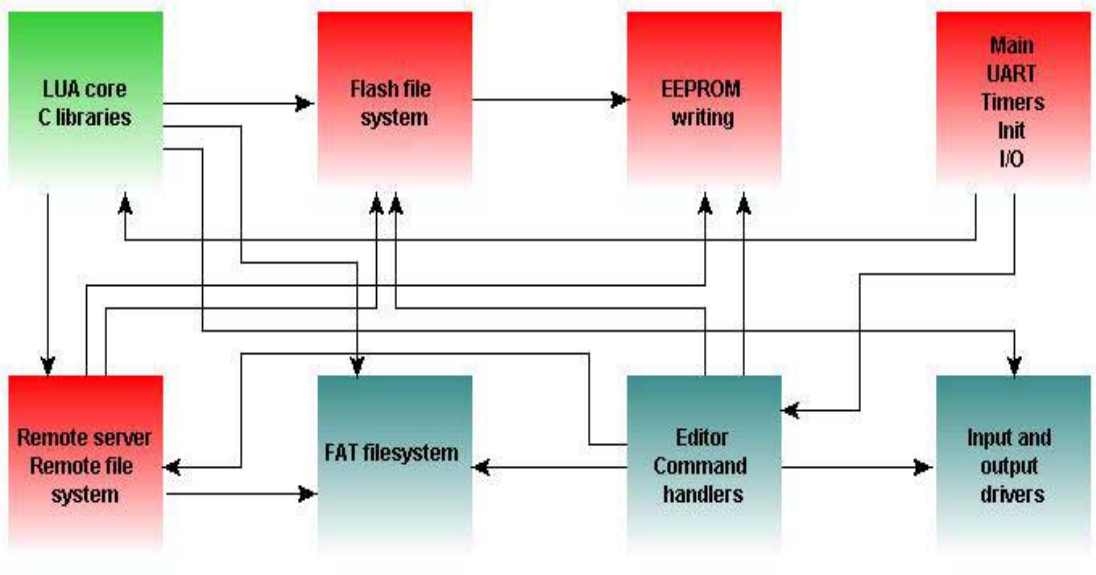

Figure 2 - Software subsystem

The hardware

The main board

This will look pretty familiar, as it is the SKP16C/62P schematic with a few modifications ☺ The character LCD is out of the schematic, as it is not used. Also, all the resistor from R1 to R6 (the one dealing with the LEDs and buttons on-board) were removed, as the corresponding I/O lines are used for other purposes. BYTE is connected to VCC to configure an 8-bit external memory data bus. That's about all to be said here. The figure below is for reference only, for a full size picture look in the 'images' directory of the project's archive (sch_mainboard.jpg).

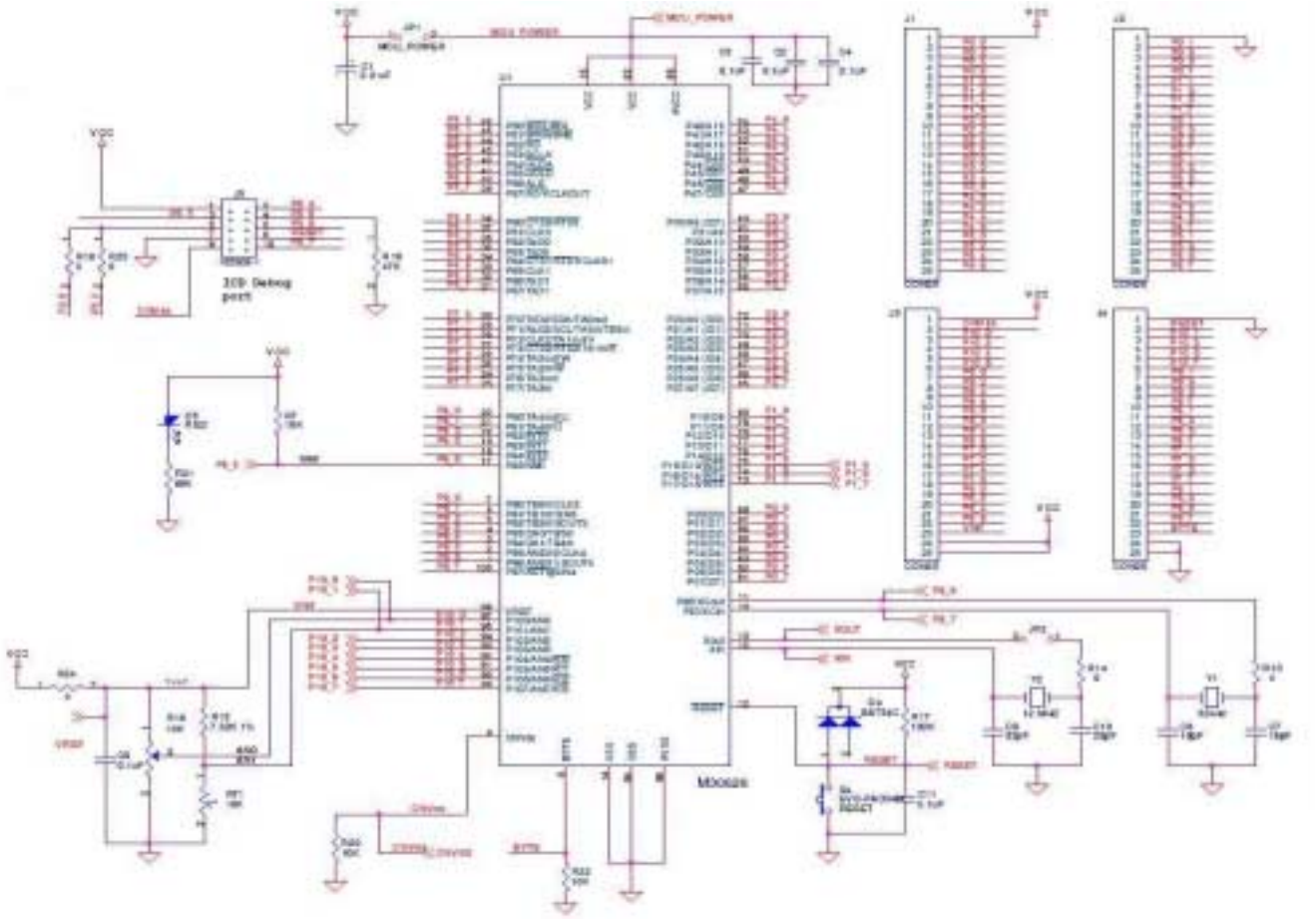

Figure 3 - The main board schematic (sch_mainboard.jpg)

And this is what it looks like on the prototype board, connected via side connectors:

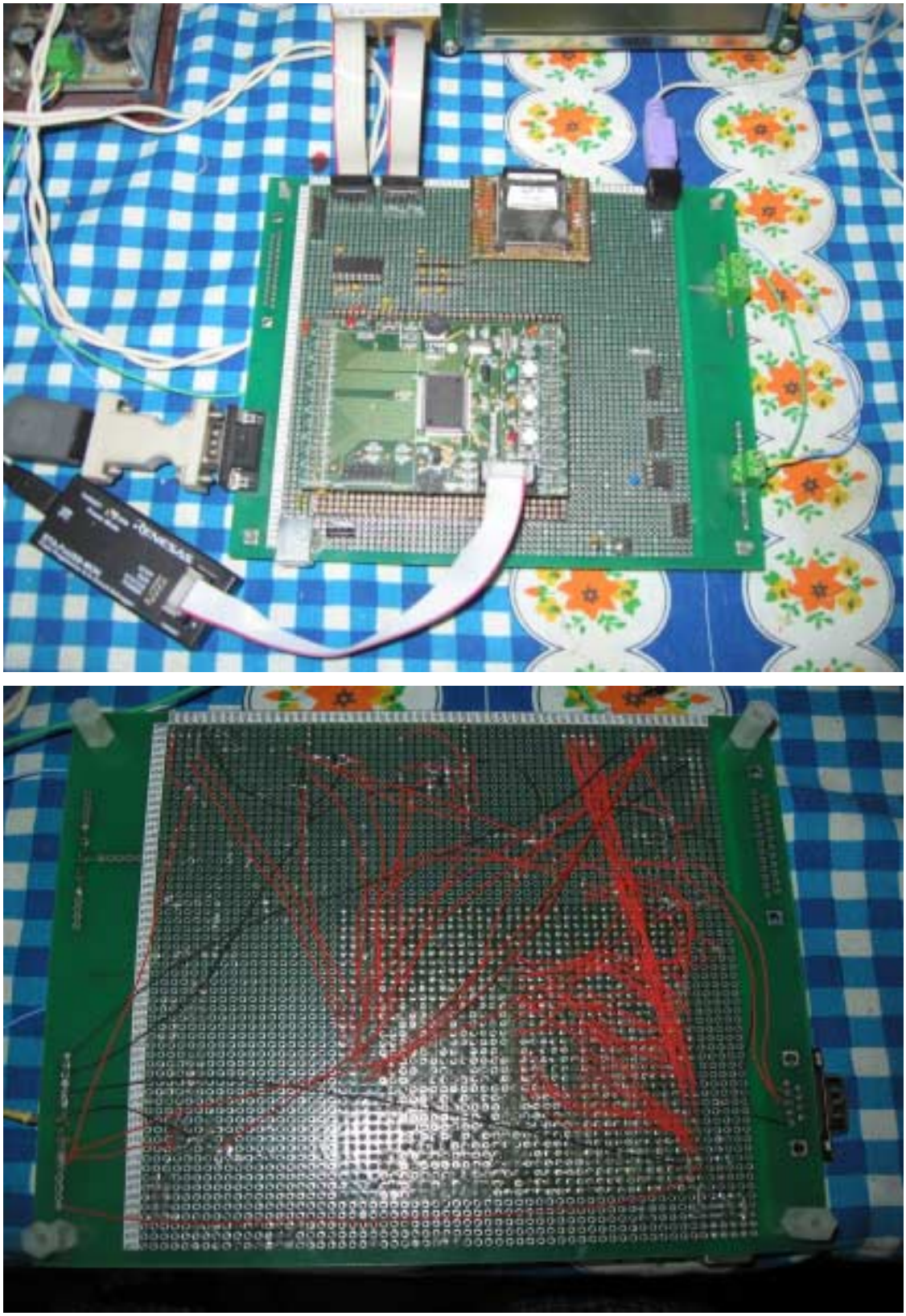

Figure 4 - Main board pictures (pict_mainboard.jpg, pict_wires.jpg)