AN UNIVERSAL, LINUX-BASED CONTROLLER WITH FREE CONFIGURABLE HARDWARE SYSTEMS

Thomas Wiedemann

University of Applied Science Dresden, List-Platz 1, 01069 Dresden, Germany
email: wiedem@informatik.htw-dresden.de

The paper describes an universal controller system for a wide area of applications, like robotics, automation or measurement. Special characteristics are a main LINUX-based ARM-controller and a second, smaller PSOC-controller with free configurable digital and analogous modules.

Keywords: Distributed controller, embedded LINUX, Programmable hardware

1. INTRODUCTION

A lot of new application areas, like robotics or home automation, require a high degree of flexibility and low efforts for changing the controlling units for new demands. Existing controller modules are often specialized and fixed and can not be adapted quickly to new measuring tasks and hardware in the loop experiments. A second problem is the software development, which is done in general on host computer with cross-compilers. The development cycle includes many uploads of binaries to the controller devices, which slows down the whole development process.

For complex applications there is an need of more than one controller in results of performance and real-time issues. A typical configuration could be found in robotics control, where a powerful controller calculates the strategic behavior and a set of smaller controllers is used for controlling the sensors and activators. The main goal of the presented development, which actual in the state of realization, is to build up a very flexible, distributed multi–level controller system with plug & play capabilities and an high bandwidth of performance.

Additional requirements are

- support of network interfaces, up to WLAN links,
- USB host and client interfaces,
- I²C communication with all submodules,
- free choice of RAM & ROM size and additional Memory-Card interfaces.
- free programmable hardware interfaces (AD&DA, PWM, Interrupts …),
- fast software development cycles and direct testing inside the field envirnoment.

Because the resulting system will be used in a wide range of applications, sometimes also by students and school kids for experiment boards and robotics, a high flexibility must be combined also with a attractive price.

2. THE HARDWARE ARCHITECTURE

The controlling tasks of typical robots can be divided in high-level strategic calculations and a lot of smaller, real-time calculations. In general it is possible to combine both tasks on one processor. But there is a risk of getting real time problems, when the calculation load increases. In result of the low prices of modern processors it seems more useful, to use different modules for the strategic and real time tasks.

In the current system, the main module for strategic tasks is realized with a 200 MIPS ARM920T –processor (see [1]). The board contains a lot of PC-oriented interfaces (see fig. 1) and runs an embedded LINUX operating system. The preferred interfaces to the host computer are the Ethernet or the USB 2.0 interface. In mobile applications this link can be realized wireless also by cheap USB-WLAN-Dongles. The ARM-controller runs the global control program (e.g. the intelligence of a robot) and schedules the actions of the smaller controllers for real-time tasks (see Fig.1).

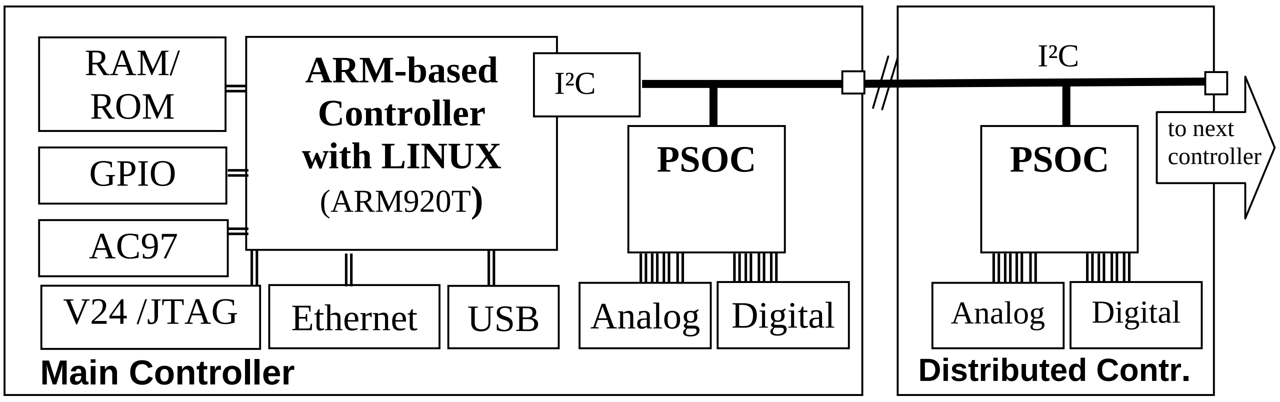

Fig. 1 Controller architecture

The real-time module is realized with the PSOC-processor from CYPRESS. The PSOC controllers (from www.cypress.com) contain a small 8-bit controller and a number of about 30 different analog and digital modules like AD/DA-converters, Multiplexers, PWM, counters and triggers, I²C and telefon line communication interfaces (see [2][3]). The main advantage is the possibility of changing the configuration of the PSOC-modules during run-time. This allows very efficient and fast adaptations to new use cases without any hardware changes.

The I²C interfaces is used for communication with the main controller. The I²C interface allows a large number of connected controllers and is easy to wire. Actually the two I²C-wires and the two power line wires are combined in one cable and are connected on the boards with a standard 4-Pin connector. So with only one click a new controller can be plugged into an existing I²C-bus system.

2. THE SOFTWARE ARCHITECTURE

The software environment is divided in the Linux-environment with all standard UNIX-related network functions and a I²C-command interface to the PSOC controllers. By sending text-messages to the PSOC-controllers the main controller can configure the PSOC-hardware and start specific actions. The results of actions or measurements is also transferred with text messages from the PSOC´s to the main controller. In result the work is divided in complex, but non real time critical programs on the main controller and fast, but smaller tasks on the PSOC-controllers, which give a wide area of possible applications for the customer.

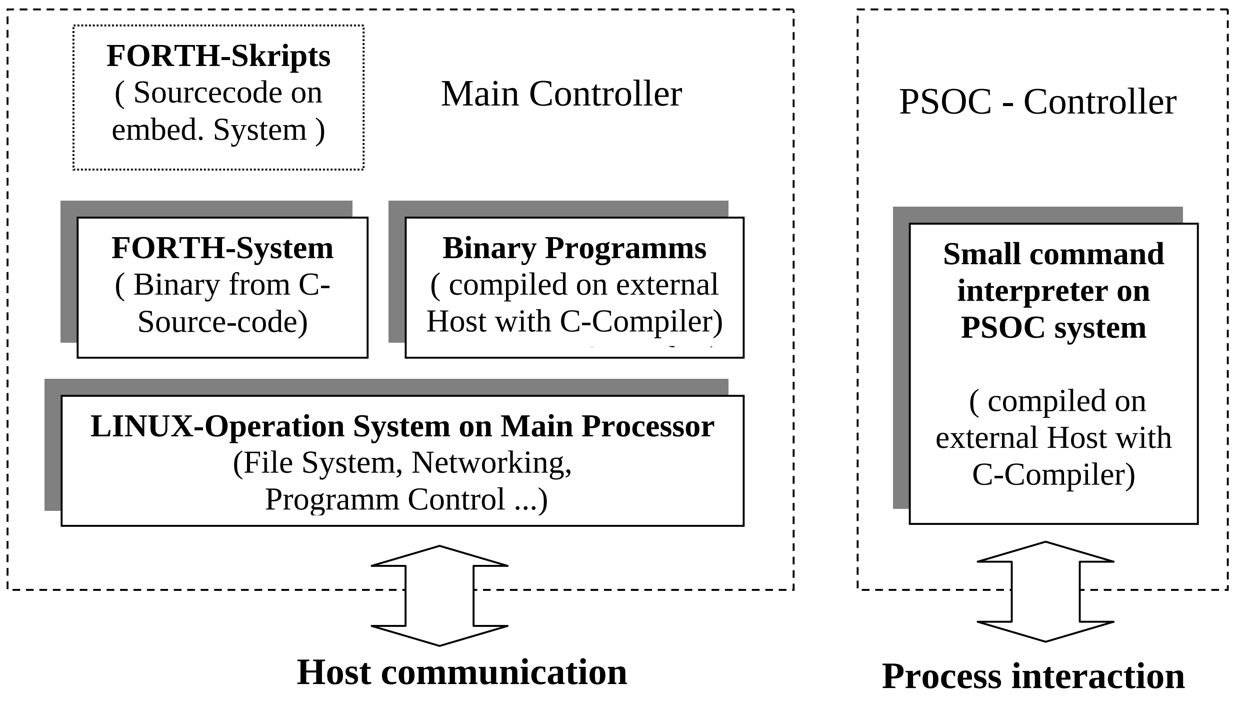
FORTH-Skripts
(Sourcecode on embed. System)
Main Controller
PSOC - Controller
FORTH-System
(Binary from C-Source-code)
Binary Programms
(compiled on external Host with C-Compiler)
Small command interpreter on PSOC system
(compiled on external Host with C-Compiler)
LINUX-Operation System on Main Processor
(File System, Networking, Programm Control ...)
Host communication
Process interaction

Fig. 2 Software architecture

The software on the Main controller can be realized as standard binary programs, which are developed as C-programs and compiled on an external host computer. The Linux operating system supports a file system on a RAM-disk for storing data and configuration files. By uploading or changing the configuration files on the controller itself, the binary program can be adapted in a limited way to changing requirements.

For more flexible tasks the controller contains a ready for run FORTH system, which was developed by the author. FORTH is a very specific programming system, which supports both interpreter and compiler modes. The FORTH-interpreter loads external text scripts from the LINUX file system and executes them directly. This mode is also very useful for testing purposes and on-line control over the host-link.

After an algorithm is tested successful, it can be compiled by the FORTH system for faster execution speed.

The software on the PSOC-controller contains a small command interpreter. It check checks the I²C interface for new commands from the main controller and executes them. The commands are low level text messages and similar to old known PEEK/POKE memory manipulation. In the case of interrupts or other hardware events the results are sent back by the software to the main controller.

On the main controller some of this low level commands are packed together and build much more complex commands. This allows a high level programming language, which could be adapted very close to the needs of the customers.

The FORTH-System allows a In-System development (not only a In-circuit testing and Debugging). New algorithms can be tested without the time-consuming Cross-Compiling and Uploading cycle. Smaller functions can be called directly from the Host-Computer. Supported by a Wire-less-LAN-link, this feature can help in the development and application of mobile robots, where the actual state of the robot can be checked and small changes of the main strategy can be done without stopping or returning the robot. Compared to the traditional development with a lot of try & error – cycles, this direct interaction with the controller in the field is very valuable.

3 CONCLUSIONS

Like in modern software architectures, a layered structure of hardware controllers for complex applications is very useful and can be realized now in result of the lower prices of modern processors and better performance parameters. Additional advantages of the better performance are the support of very flexible operating systems like LINUX and the opportunity of using different control tools and development tools like the FORTH environment. Both layers increase the flexibility and development speed of complex applications.

4. REFERENCES

[1] ARM9 -Design Resources and Data Sheets, – http://www.arm.com/products/CPUs/ARM920T.html- last accessed Sept. 1. 2005, ARM Ltd., United Kingdom

[2] Robert Ashby. Designer's Guide to the Cypress Psoc. Elsevier Publishing, USA 2005

[3] PSOC - Design Resources and Data Sheets - http://www.cypress.com/psocexpress – last accessed Sept. 1. 2005, © Cypress Semiconductor Corp. 2005 , USA