A CAN-Based Real-Time Lab Environment*

Hans A. Hansson, Mikael Sjödin and Harmen van der Velde
Department of Computer Systems
Uppsala University
P.O. Box 325 S-751 05 Uppsala
e-mail: {hansh,mic,frankimo}@docs.uu.se

July 1997

Abstract

We present a CAN-based distributed real-time system lab environment. The hardware platform consists of MC68HC11 microcontroller boards, equipped with i82526 CAN-controllers. We use an off-the-shelf real-time kernel together with a commercial C development environment. In addition to these applications we have developed a set of tools to facilitate development and validation of distributed real-time systems.

In the lab environment we have developed two sample applications for use in our undergraduate education and for evaluation of the lab environment.

Introduction

We present a CAN-based distributed real-time system lab environment consisting of:

- A hardware platform.
- A real-time kernel and software development tools.
- Two applications.

The environment provides a platform for assignments to our undergraduate students. It also gives us a test-bed for testing new ideas in our research of real-time systems design, development and validation.

The Lab Environment

The hardware platform consists of MC68HC11 micro-controller boards (from I+ME in Germany), equipped with i82526 CAN-controllers. We use the TPK real-time kernel from Northern Real-Time Technologies in the UK [NRT], together with C-compiler, Assembler, Linker, and high-level Debugger from IAR Systems in Sweden [IAR]. In addition to these off-the-shelf tools we have developed

- TPKSIM [vV96], a simulation tool facilitating the development of application software executing under the TPK-kernel.
- FPSCALC [vdV97], a general tool for fixed priority schedulability analysis of (possibly distributed) real-time systems.

TPKSIM implements the major system calls of TPK and simulates fixed priority scheduling of the tasks in the system. TPKSIM also implements a simulated real-time clock, allowing simulations to be run in time-sharing systems without interference from other activities in the system. TPKSIM is implemented in C and runs under Linux, Solaris 1, and Solaris 2.

The FPSCALC tool allows us to analyse the real-time behaviour of messages sent on the CAN-bus as well as the application software executing on the micro-controller boards. In fact, we can *á priori* conveniently determine if specified deadlines are met, even for applications distributed over several micro-controller boards and communicating over the CAN-bus. The analysis is substantially facilitated by the use of the TPK-kernel, which has been developed with fixed priority scheduling analysis in mind. FPSCALC is implemented in C and should run in most environments, but has only been tested under Linux, Solaris 1, and Solaris 2.

*This work is supported by a grant from the Volvo Science Foundation.

The Applications

The two applications are simple, albeit realistic, real-time processes controlled by our distributed computer system. The first application is a marble sorter (originating from university of York) in which marbles of two types (glass and metal) roll down towards a selector, which should be controlled such that metal marbles end up in one container and glass marbles in another. A schematic picture of the marble sorter is provided in figure 1.

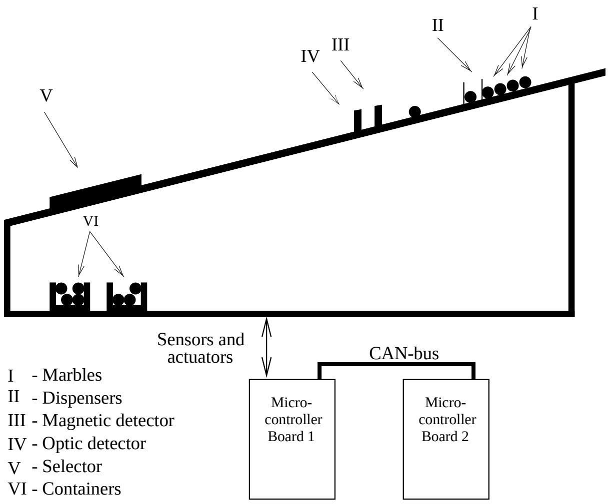

Figure 1: The marble sorter

The marble sorter has been used in undergraduate education. The students are given the assignment to write the control program which sorts the marbles. The assignment is formulated so that the students need to use both microcontrollers when solving the assignment. The students also need to validate the timing behaviour of their proposed solution (i.e. the need to perform a schedulability analysis of the system). The complete assignment specification is available on-line [Sjö97].

The second application is a Fischer-Technik model of a production cell (originating from FZI in Germany), consisting of two conveyer belts, a crane, two robot arms, and an elevating table, which should be controlled such that a flow of metal pieces are transported to and from a press. Figure 2 shows a photo of the complete model. This production cell is a classical "benchmark" in the real-time system validation community. The control program for this cell is sufficiently simple to be formally analysed, yet it provides

Figure 2: The production cell

a realistic example of a safety critical real-time application.

References

[IAR] IAR Systems. URL: `http://www.iar.se/`.

[NRT] Northern Real-Time Technologies Ltd. URL: `http://www.nrtt.demon.co.uk/`.

[Sjö97] Mikael Sjödin. The marble sorter assignment, 1997. URL: `http://www.docs.uu.se/docs/-undergrad/instances/spring97/-rt_dv/marble_sorter/`.

[vdV97] Harmen van der Velde. Description of the FPSCALC program. Technical Report 97/89, Uppsala University, Department of Computer Systems, 1997.

[vV96] Harmen vd Velde. Real-Time FPS simulation with TPKSIM. Technical Report 97/88, Uppsala University, Department of Computer Systems, 1996.