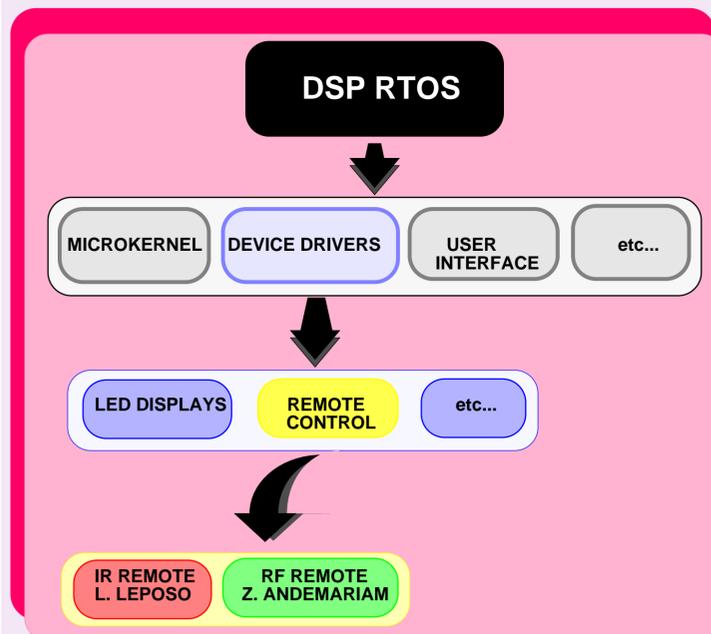


Object-based design of device drivers for Real-Time Operating Systems: A Remote controlled Embedded System



SERTS SOFTWARE ENGINEERING FOR REAL-TIME SYSTEMS

Z. Andemariam, L. Leposo / D. Stewart



INFRARED REMOTE CONTROL:
This involved gaining an understanding of Ir signal processing and communication. Our remote encodes a command then puts it on a 40KHz carrier frequency. Once received, the signal is processed to give this encoded logic output.

There are two standards for remote control signal encoding, the RC5 and RECS80 protocols. The RC5 uses a uniform duration for all bits transmitted, however the transitions/edges occur in the middle of that duration. The RECS80, on the other hand, uses pulse length modulation to encrypt short remote control commands. A logic LOW is represented as a "ZERO" of period T followed by "ONE" of period 2T. The logic HIGH is a "ZERO" of period T followed by a "ONE" of period 4T. Our Ir remote control uses the RECS80 with a T of 0.5ms.

OBJECTIVE:
To investigate the built-in support for peripheral devices in Real Time Operating Systems (RTOS) running on Digital Signal Processors (DSPs).

MOTIVATION:
There are few good and inexpensive RTOSes for DSPs.

Non-Real Time operating systems such as Unix and Windows have two standard formats for driver design that cater, separately, for block and character devices. This standard, however, can't be adapted for RTOSes partly because it results in a lot of CPU overhead due to things such as Context Switching. Port-Based design of drivers provides a single standard for RTOSes in addition to minimizing the overhead associated with device interfacing.

Most of the development of drivers and applications for such purposes are done in an ad-hoc and time consuming manner because little has been done to support reconfigurable interfacing of current RTOSes and external devices (hardware). An RTOS that supports portable reconfigurable applications and drivers will be of great use to the real-time systems community. Some of the immediate benefits will be, a reduction in development time due to rapid prototyping, cost savings that result from this and reusability of code.

APPROACH
No DSP RTOS is available for our work so we will work on the Chimera RTOS that runs on a VMEbus until Echidna is available.

The fundamental approach used in achieving our objectives is to focus on one device- the remote control and use it to gain an insight into the issues that arise with device driver and application development.

The driver and applications for the remote control are then modeled as a PBO (Port-Based Object).

REFERENCES
Design of Dynamically Reconfigurable Real-Time Software Using Port-Based Objects.
D. Stewart, R.A. Volpe, P.K. Khosla
EEE Trans. on Software Engineering, v.23, n.12, December 1997, pp. 759-776

An I/O Device Driver Model and Framework for Embedded Systems
D.B. Stewart
Proc. of IEEE Workshop on Middleware for Distributed Real-Time Systems and Software (MDRTSS), San Francisco, December 1997.

RF REMOTE CONTROL.
This involved gaining an understanding of RF signal processing and communication. Our remote encodes a command then puts it on a 49MHz carrier frequency. Once received, the signal is processed to give this output on a digitizing scope.

Since there was no protocol available for this signal we sought to decode it by time analysis. Each command is encoded as a start sequence followed a unique series of data bits which end with a stop pattern. All commands share the same start and stop sequences. These sequences consist of two LOW-HIGH sequences each with a duration of 4ms (3ms for the LOW and 1ms for the HIGH). The data constitutes a series of LOW-HIGH each with a period of 2ms (1ms for the LOW and 1ms for the HIGH).

