

MULTITASKING WITH SMALL AVR'S

Strawberries and Chocolate on a budget of Peanuts.

ATMEL/CIRCUIT CELLAR CONTEST ENTRY A3650

ABSTRACT

Multitasking and embedded systems go together like strawberries and chocolate. Even the simplest systems usually have a basic form of multitasking in that interrupts are used to alter the flow of a single background program (often called a "foreground-background" system, herein called fbRTOS.)

As programs get more complicated a simple fbRTOS scheme usually starts to get difficult to program and maintain. Some sort of real-time operating system (RTOS) is needed in order to program in a more structured and maintainable manner.

A full-blown preemptive multi-stack (pmRTOS) is commonly used with large embedded systems projects, and there are quite a few commercial vendors of such operating systems. However, any pmRTOS has RAM requirements that put it beyond the bounds of simple microcomputers with limited RAM. This project describes and demonstrates a cooperative single-stack RTOS (csRTOS) that will run on the smallest AVR microcomputers that have RAM. The demo code runs on an ATMEL AVR 90S4433 that has only 128 bytes of RAM. The csRTOS can be easily modified to work with other AVR processors.

A pmRTOS can switch tasks in response to interrupts that make a higher priority task ready to run (or, rather, ready to resume). With a cooperative RTOS, task switching only occurs when a task makes an operating system service request, and it is therefore necessary to insure that such requests are made often enough to insure that the system realtime response requirements are met. Experience shows that this is often very easy to do.

The csRTOS project was developed using the free WinAVR compiler, and has been ported to and tested with the Codevision and Imagecraft compilers. A small amount of assembly language code is required, and it varies slightly from compiler to compiler. The project documentation contains instructions for porting to your fAVRoite compiler.

The demonstration project runs on the ATMEL STK-200 evaluation board, but could be very easily ported to one of the later evaluation boards such as the STK-500 or Butterfly. The basic functionality of the project was designed to show the use of the csRTOS, with several tasks and one semaphore controlling access to a shared resource. The demo is a useful example for users who wish to use the csRTOS in their own projects. It would also be a good example to use for instructional or tutorial purposes. The complete csRTOS and demonstration project code and documentation will be posted on the Circuit Cellar web site for unrestricted use by anyone.

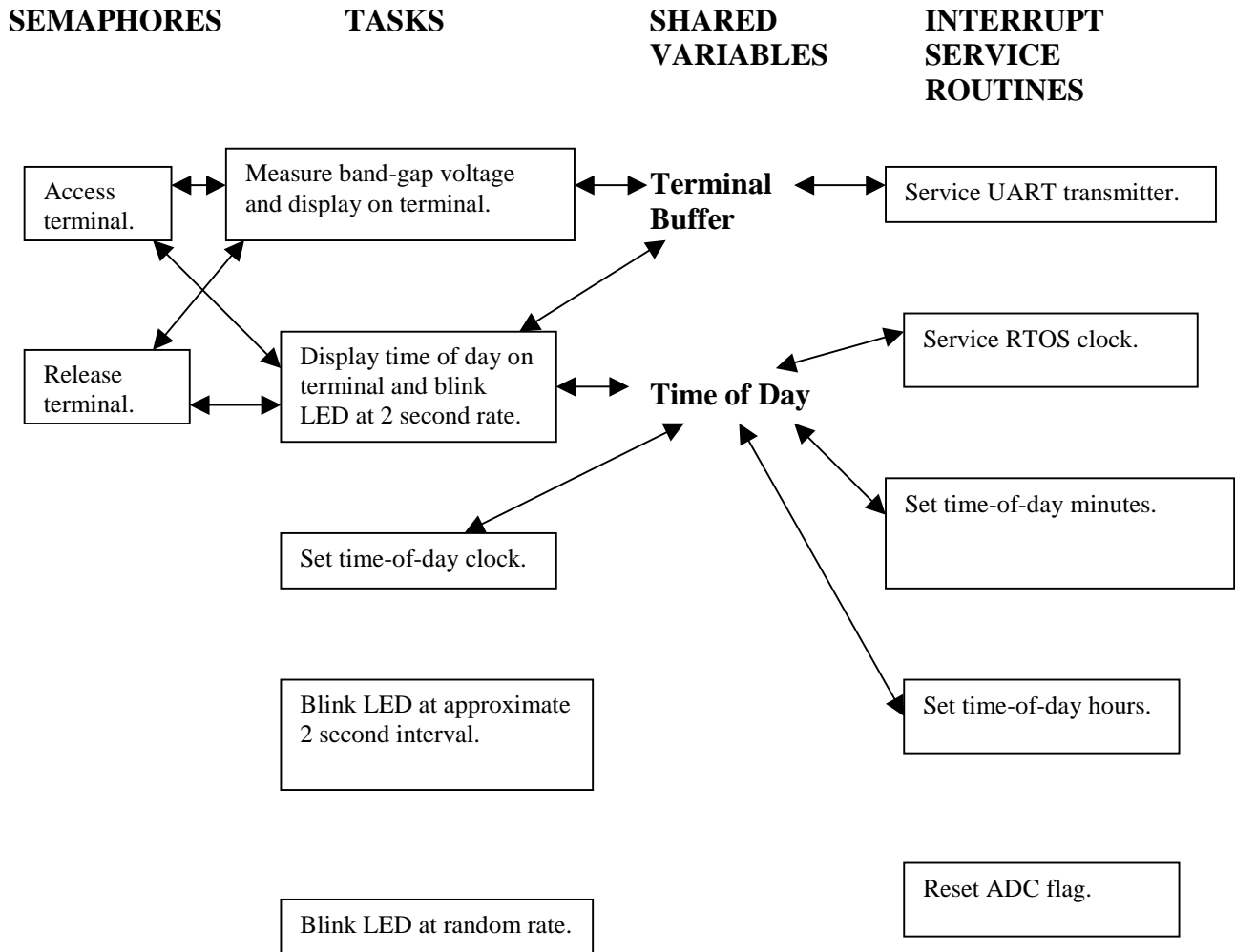
The project demo has the following functionality:

- 1) Keep track of the time of day and blink an LED at a 2 Hz rate using the 4 MHz clock as an "exact" timer (limited only by the crystal oscillator accuracy).
- 2) Blink an LED at approximately a 2 Hz rate, using the basic csRTOS timer ticks that occur at a frequency of approximately 49 Hz. This task is used to demonstrate a very small and simple csRTOS task.

- 3) Blink an LED at a random rate. The code includes a very nice psuedo-random number generator that has good statistical properties and does not require multiply or divide instructions.
- 4) Measure the AT90S4433 band gap voltage at periodic intervals.
- 5) Monitor switch inputs to set the time of day.
- 6) Periodically display the time of day and the bandgap voltage on a serial terminal or terminal emulator (this demonstrates the use of a semaphore for access to the terminal, a shared resource).

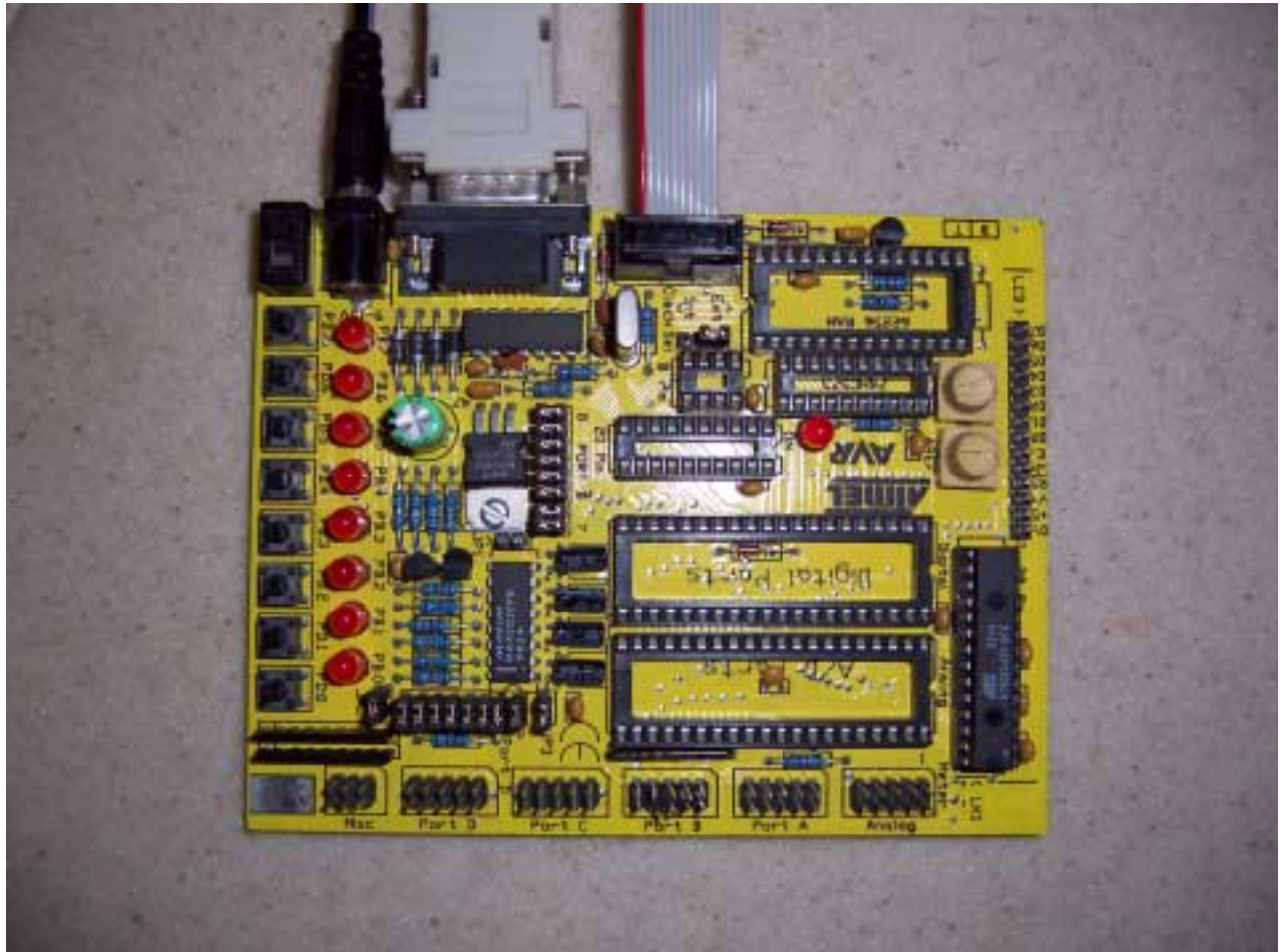
BLOCK DIAGRAM

Each function is encapsulated in a separate task (except for the first, which is coded in an interrupt service routine). The block diagram shows how the tasks are interconnected with semaphores and shared variables.



PROJECT PHOTO

Note that the essence of this contest entry is a very useful real-time operating system software technique for developing embedded systems using small AVR parts. No hardware was built for the project - the demo code was developed using the STK-200 development board, shown below.



SOFTWARE SAMPLE CODE

The functions required by a particular AVR embedded system project are broken down into distinct tasks which are largely independent of other tasks. Here is an example that shows a simple task that blinks an LED. An operating system call is used to control the on and off time of the LED. Other tasks are able to run during the waiting interval.

```
void blinkTask(void) {           // Just about the simplest possible task.
    ostASK_INIT;
    for(;;) {
        PORTB &= !0x01;         // Turn on the LED on PORTB.0
        osWAIT(15);             // Suspend this task for about 0.3 seconds
        PORTB |= 0x01;         // Turn off the LED on PORTB.0
        osWAIT(85);             // Wait about 1.7 secs
    }
}
```

SUMMARY

csRTOS is a cooperative real-time operating system for AVR micros having small RAM. Even the ATtiny13 with only 64 bytes of RAM can use csRTOS. A demonstration program with 5 tasks using the AT90S4433 (with 128 bytes of RAM, most of which are not used in the demo) is shown as an example. Complete source code is included, and may be freely used without restriction.