

## Project #2 – Networking with Serial Ports on XINU/WRT54GL Platform

CSE321: Embedded and Realtime Operating Systems

Fall 2010

### 1. Objectives and overview of the project:

In Project #1 the minimal operating system XINU was enhanced by adding shell commands and by adding a simple device driver (LedLoop). In this project we will explore the serial communication ports of XINU for communication among serial devices. Objectives of the project are to:

- (i) study the software representing various components of the Embedded XINU operating system
- (ii) study the serial port features (UART0 and UART1) offered by XINU/WRT54GL platform
- (iii) establish a communication channel between devices connected to these ports via a cyclic executive loop and
- (iv) build a simple chat application with user login feature and realtime constraints.

### 2. Project Environment:

**2.1 Operating system XINU:** XINU ("Xinu Is Not Unix") is a small, academic operating system developed at Purdue University by Dr. Douglas E. Comer in the early 1980s for the LSI-11 platform; it has now been ported to a variety of platforms. Embedded XINU is an update of this project which attempts to modernize the code base (to ANSI-compliant C) and port the system to a modern architecture (specifically the MIPS architecture).

**2.2 Hardware WRT54GL:** We will use wireless router WRT54GL as a host for embedded the software we will develop for enhancing the features of an embedded operating system. The WRT54G is notable for being the first consumer-level network device that had its firmware source code released to satisfy the obligations of the GNU GPL. This allows programmers to modify the firmware to change or add functionality to the device. WRT54GL features a Broadcom MIPS processor BCM5352 (200-250Mhz), a four port switch, 802.11b and 802.11g wireless LAN support, a Web interface for configuration of the router, 16Mbytes of RAM and 4Mbytes of flash memory. Any modification to the router function itself has to be done by updating the firmware on the flash memory. However we will use the RAM to load the embedded XINU.

**2.3 Test Environment:** A simple embedded XINU network is shown in Figure 1.

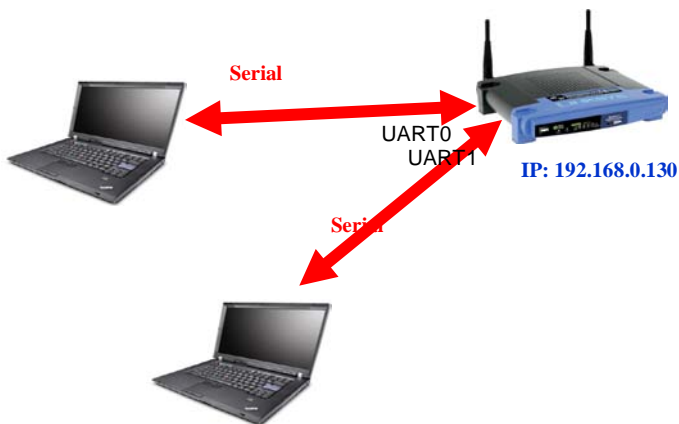


Figure 1: Network to enable serial network with Embedded XINU /WRT54GL platform

### 3. What to do?

**3.1 Study Embedded XINU files :** Download the Embedded XINU bundle (tarball) from <http://xinu.mscs.mu.edu> and untar it. Examine the directory structure and understand the purpose of various directories. Specifically, study these files in the directories:

- **system (system operations): initialize.c, main.c**

The following code from initialize.c will create the main process and shell 0 (TTY0/Console):

```
open(CONSOLE, SERIAL0);
ready(create((void *)main, INITSTK, INITPRIO, "MAIN", 2, 0, NULL), RESCHED_NO);
```

The following code in main (main.c) initializes the Shell 1 to be TTY1/SERIAL1:

```
process main(int argc, char **argv)
{
  /* Associate TTY1 with second serial port. */
  open(TTY1, SERIAL1);
  enable();

  /* Launch one shell process for each TTY device. */
  ready(create((void *)shell, INITSTK, INITPRIO, "SHELL0", 1, CONSOLE), RESCHED_NO);
  ready(create((void *)shell, INITSTK, INITPRIO, "SHELL1", 1, TTY1), RESCHED_NO);
}
```

Thus there is a root process and main process and shell0 and shell1 (corresponding to uart0 and uart1 of the UART).

- **Shell directory: Shell.c file**

Read the data structure associated with a shell process and how a line buffer is handled, tokenized etc.

```
process shell(ushort descrp)
{
  char  buf[SHELL_BUFLLEN];          /* line input buffer    */
  ushort buflen;                    /* length of line input */
}
```

Do you understand how to create a new process now by looking at the above process definitions and process creations?

- **all the files in tty and uart directories.**

### 3.2 Write a simple cyclic executive

This cyclic executive will read Shell1 and write to Shell1 and vice versa.

### 3.3 Write a chat program

Add a simple chat program with login feature that will prefix messages with the originator's name and print it on the device (lap top). Extend the chat program to allow multiple user (more than 2) and allow chatters to join and leave.

## 4. Deadline for the project: Nov 1, 2010 Monday by 5pm.