

Tony Morelli
CPE 606
HW #9
UDP Tic Tac Toe for PSP

For this assignment, I created a network based game of Tic Tac Toe which is meant to be played against two Playstation Portables using the wifi link. Source code and binaries are included for the PSP and for a PC running linux. Either target can be the client or the server, or you can play against 2 of the same target (play PSP vs PSP). The linux version of the game is mainly meant for debugging, so if you want to run the linux version you MUST rebuild it to contain the ip address of the server PC. By default it contains 192.168.1.40 (the ip address of my PSP).

The source code contains the files main.c TicTacToe.c/h, and TicTacToeMsgs.h. All of the TicTacToe* files are meant to be used on all target platforms untouched. This is all of the TicTacToe logic and shows how easily games can be ported for different targets. Most of the code in main.c is specific for the linux machine or for the PSP (although I could make this even more common between games). TicTacToe.c/h contains logic specific to the TicTacToe game (mainly evaluating the board to see if there is a win). TicTacToeMsgs.h contains defines for message types being passed between the client and the server.

The client and server communicate using UDP. UDP was chosen as the second part of this assignment will be playing the PSP vs the Nintendo DS and currently the DS only supports UDP. With UDP, packets are not guaranteed to reach their destination. Therefore good coding would have some sort of ACK so the sender knows the packet reached its destination. Unfortunately this code does not do that. In all my testing I never dropped a packet, but I know it will happen from time to time. When that happens, the games will be out of synch and the only way to fix it is to restart both applications. If you are about to lose a game of TicTacToe, I would highly suggest powering off your wireless router as the game will have to be restarted. Also, the client and server connect wirelessly through *infrastructure* mode. That is they connect to each other through a wireless access point. This is opposed to *ad-hoc* mode where the client and server connect directly to each other without the use of a wireless access point. The reason this approach was done was also due to limitations of the current Nintendo DS open source libraries.

When the game is started on the PSP, the user has the option of how the game is to be played. Does the player want to be a Server, or a Client. If Server is chosen, the PSP waits for a client to connect to it on the well known port of 9930. If Client is chosen, the PSP prompts the player to enter the IP address of the server. Be sure to enter the correct IP Address as if the PSP cannot connect to a server, the PSP must be rebooted in order to try to reconnect to the server.

Once the client and server are talking to each other, the game of Tic Tac Toe is played until the players get bored. The server is always X and the client is O. The server always moves first. On the PSP side, the player moves a cursor to each of the 9 board positions and press X to place a token. On the linux side, the user is prompted to enter a board position to place a token. The board position is a number from 1 to 9. Positions 1-3 are the top row, left to right, 4-6 are the middle row left to right, and 6-9 are the bottom row left to right.

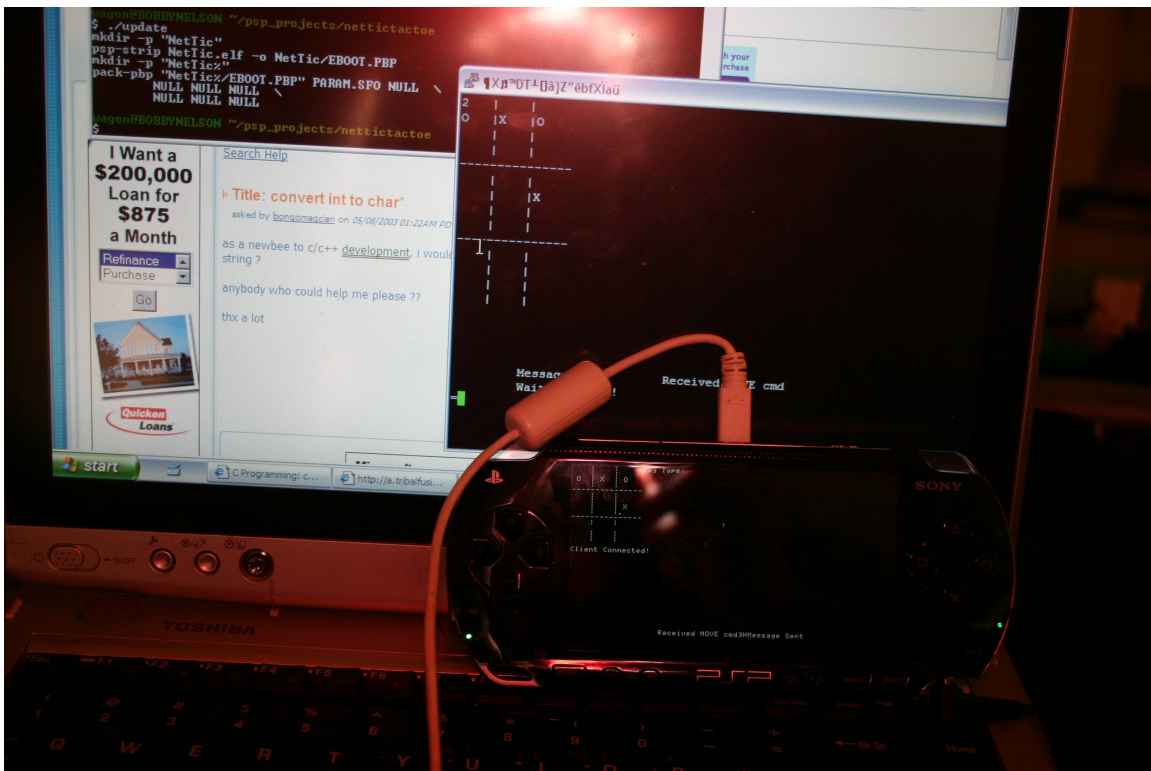
Screen shots follow:



Initial screen – Select Client or Server



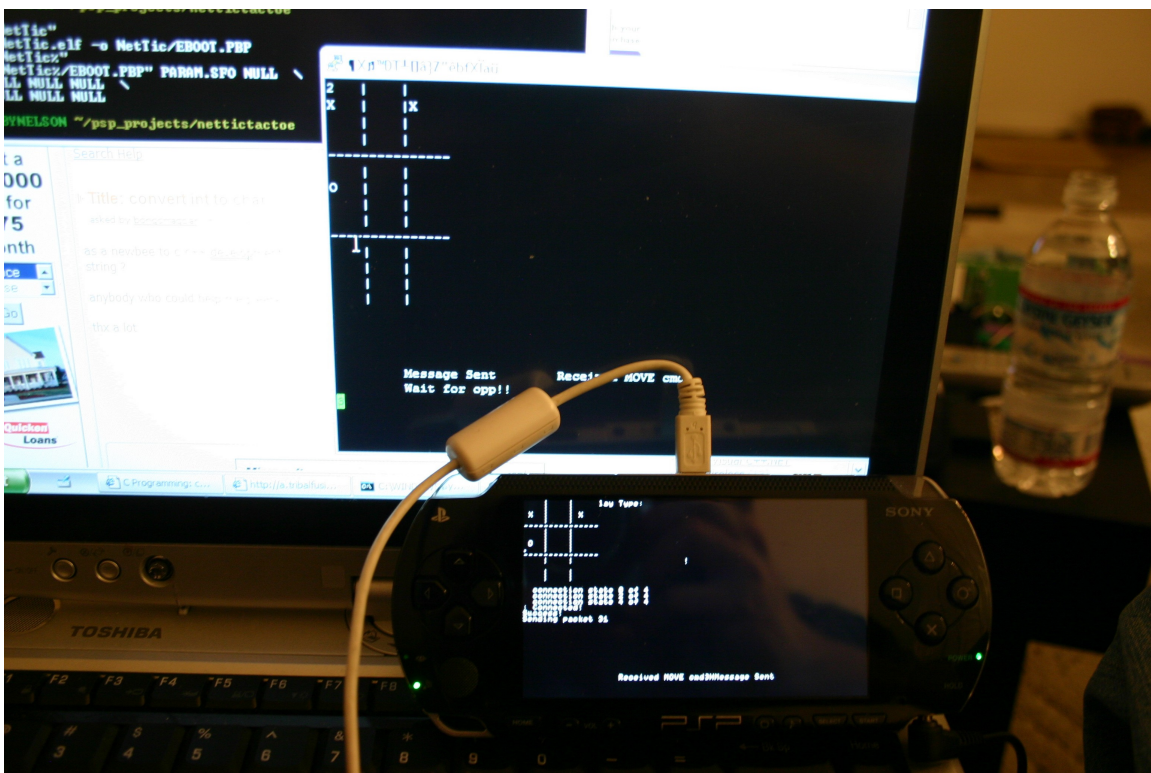
In this case Server is selected. The PSP waits for a client to connect. Notice the green LED on the lower left hand corner of the PSP. This indicates the network is up and talking.



The PSP as the server with the PC client in the background



Running as a client you will be asked to enter the IP of the server



This shows the PC as the server. The USB cable in all these pictures is to upload code to the PSP. All Tic Tac Toe communications is over the WIFI link.