Objective:

Understand the handling of a TCP/IP network stack.

Preparation:

Prepare the code of the tasks below.

Task:

3.1. Complete the previous program as follows:

3.2. The first task shall also do the following:
- It also creates and starts tasks six.

3.3. The sixth task shall do the following:
- It implements a TCP server at port 4444 and allows max. 4 simultaneous connections.
- At each connect request arriving it spawns a work task which will handle the request and then awaits new requests.

3.4. The TCP work task shall do the following:
- It sends a welcome message to the client.
- It awaits a query from the client:
  If the query is “2”, it sends the value of the potentiometers as Volt.
  If the query is “3”, it sends the value of the potentiometers in hex representation.
  If the query is “1”, it sends the value of the potentiometers in decimal representation.
  If any other key has been pressed an error message shall be send.

3.5. Verify your server by starting a command window (Start → Run: cmd) and then type:
- telnet 90.0.0.50 4444

Report:

Not requested.

Notes:

5.1. Set the stack of both the TCP-Server-Task and the TCP-Work-Task to 0x10000!

5.2. You have to include the following header files:
- #include <stdlib.h>
  #include <stdio.h>
  #include <intLib.h>
  #include <taskLib.h>
  #include <sysLib.h>
  #include <time.h> (already known)
- #include <sockLib.h>
  #include <inetLib.h>
  #include <string.h> (new)