Lantronix Xport tutorial

Contents:

Connecting to Xport:

Lantronix Xport has 2 interfaces you can connect to.
- RS232
- Ethernet.

Connecting to Xport by RS232:
- lunch hyper terminal
- in "connect using" field chose port in Your computer connected to XPortServer
- chose parameters. If You didn't change it on port server parameters are:
  Bits per second: 9600
  Data bits : 8
  Parity : none
  Stop Bits : 1
  Flow control : none
- additional settings in Hyper terminal (if You want text in second terminal look good):
  file->prosperities. Chose tab "settings".
  In option backspace keys sends chose ctrl+H space ctrl+H
  get into "ASCI Setup". Check send line ends with line feeds.

Connecting to Xport by Ethernet:
- lunch hyper terminal
- in "connect using" field chose TCP/IP (win sock)
- in "Host address" write IP address of XportServer.
- in "Port number" write port number used by XPortServer
- additional settings in Hyper terminal (if You want text in second terminal look good):
  file->properties. Chose tab "settings".
  In option backspace keys sends chose ctrl+H space ctrl+H
  get into "ASCI Setup". Check send line ends with line feeds.

You can connect by serial and Ethernet in the same time and You will see that data from serial is forwarded into Ethernet and data from Ethernet into serial.

Configuring device:

You can configure Xport by:

- Xport installer
- Telnet
- RS232

Configuration by Xport installer:

-lunch Xport Installer. Press “search” Button. If Xport has assigned IP, you will see it on the list.

- If in Your network is DHCP server, Xport will get IP address, but it is not always comfortable. For example if You want connect to Xport, You need to check its IP address in Xport Installer
- If you cant find Xport press assign button. Write MAC Address (is written on label of Xport) and IP You want to assign.

Configuration by telnet:

-lunch telnet xxx.xxx.xxx.xxx 9999

where:

xxx.xxx.xxx.xxx – ip address of Xport

-to enter setup mode press enter

Configuration by RS232:

This configuration method is very useful when something is wrong and Ethernet interface is not working.

Connect by hyper terminal to Xport:
chose this settings to your COM port in PC:

- Bits per second : 9600
- Data bits : 8
- Parity : none
- Stop Bits : 1
- Flow control : none

Press and hold “x” key on your hyper terminal keyboard and reset device. Wait until you get message on screen of hyper terminal. Press Enter, to get into setup mode.

Using WEB server:

Xport support an internal web server that may be utilized by the web programmer for storage and retrieval of documents, images, and Java applets.
Web browsers, request information from web servers by using a protocol known as Hypertext Transfer Protocol or HTTP. HTTP server support GET and HEAD methods. It does not support the POST method, or any "server side" required processing. (No CGI support.)

XPort Server contain 512KB of flash memory. The flash memory area is divided into 64KB pages, which are available for "web sites". These areas are called WEB1 – WEB6.

When a client makes a GET or HEAD request for a file, the HTTP server process will first look in WEB1 for the file. If the file is not found in WEB1, the process will look in WEB2, then WEB3 and so on. When the file is located, it is sent to the client.
In areas WEB1 – WEB6 Xport store WEB pages. Pages are compressed inside *.cob file. Because of 64KB flash pages site *.cob can not be larger than 64KB.

Creating COB files:
In order to store your files on the Device Server's flash card, they must be in the correct format. Lantronix provides a utility called "**web2cob.exe**" to create the required the archive file for the "web site".

Syntax: Web2cob.exe /d <directory> /o <output_file_name>

Where:
"directory" is the directory that contains the files to be archived
"output_file_name" is the name of the archive to be created
Example:
Web2cob /d myfiles /o web3.cob

Putting COB files by Xport Installer:
- lunch Xport Installer.
- chose appreciate Xport Server (if You have more Xport Servers in network)
- press upgrade button.
- Chose WEBX slot. Browse *.cob file.
- Xport Installer wil load page. And that's all.

Putting COB files by TFTP:
To load a .COB file into the Device Server, you will need a tftp client program. Several tftp programs are available that run on Microsoft Windows platforms. For our example we will use the tftp client built into Windows 2000, and Windows NT.

To load the .COB file into the Device Server, follow this process:
From a command or DOS prompt on Windows 2000 or NT
C:> tftp -i xxx.xxx.xxx.xxx PUT \zzz\yyy.cob WEBX
where:

xxx.xxx.xxx.xxx:	Xport's IP address.
\zzz\yyy.cob:	path and filename of web archive.
X:	number of WEB slot on Xport (1-6)

Customizing firmware:

Before You start, makesure, that You have all necessary tools.

Necesarry tools:
- Software for transferring *.rom file into XportServer. installer can do it. By Ethernet and by Serial. If something is wrong. For example if new software is broken. It is still posible to load new software, but only by RS232.
- C++ compiler, linker. I use Turbo C++ 3.0 On this computer this software is in c:\tc directory.
- Editor. For example notepad, but much more comfortable is using special editor with C++ syntax like for example edit plus.

Compilation:
Examples in directory DemoX you can compile by entering this directory and writing make all. But on my software something was wrong and I have written script file. I did it by modifying makefile. Makefile is instruction for make.exe how to build software. But make.exe was not able to execute dos commands on windows XP. And that's why I needed to create bat file.

```
cls
REM main_x.obj: main.c
tcc -1 -O1 -Z -c -ms -a- -I..\inc -I..\ -DMODS_XPT -omain_x main.c

REM setpar_x.obj: setpar.c ..\stdf\stdpar.h
tcc -1 -O1 -Z -c -ms -a- -I..\inc -I..\ -DMODS_XPT -osetpar_x setpar.c

REM demo.obj: demo.c
tcc -1 -O1 -Z -c -ms -a- -I..\inc -I..\ demo.c

REM tools.obj: tools.c
tcc -1 -O1 -Z -c -ms -a- -I..\inc -I..\ tools.c

REM root.obj:
copy  ..\lib\root.obj .

REM xpt.exe:
tlink @xpt.lk

e2i xpt X1 nix

del *.obj
del *.exe
del *.map
```

Deleting object file prevent linking old version of this files with new versions of another files.

Multitasking
The XPort's round robin multitasking is controlled with four interrupts in the following priorities:
- 1. Serial interfaces
- 2. Timer
- 3. Network interface
- 4. Standard Priority

1 is the highest priority. That means that e.g. the network event can interrupt the standard event.
CoBox' multitasking occurs only when you call the **nice ()** function.
It is very important to insert this function in any longer loop, otherwise the watchdog will reset the device after 1 second. nice () is also called from some other functions.
That's why some C/C++ programs written to Xport looks quite different.
For example:

- Input operations. You can not wait until all word/command will be in buffer, because user can be slow and it can take him more then 1s. Watchdog will reboot Xport after 1s. Below is example of calculator.

Example of source code:

In CPK there is a lot of examples. On the beginning the easiest way to write something is to modify existing demo file. Just chose the most appreciate demo and modify it.
In each demo project, there are several source files.

- **Main.c** is a common block of all demos. Main.c contains all needed procedures for initializing so-called "process" that is a main feature of CoBox's operation system. The WebProcess() is launched from main.c for those demos requiring web services.
- **Setpar.c** is where changes to setup menu are made.
- **Tools.c** is a collection of nice to have utilities, and is common across all demos. By default these utilities are undefined with #ifdef statements.
- **Demo.c** is where most of the functionality changes take place. In writing own modified firmware modifying only demo.c is enough on beginning.

My software example is calculator. You send data by RS232 and receive results by this interface.
I had modified only one file – demo.c. I needed alse to write script to make compilation and linking.

Example: demo.c

```
/*****************************************************************************/
/* demo.c                                                                    */
/*****************************************************************************/
/* Version  X0.00  demo template                                             */
/*****************************************************************************/
/* History:                                                                  */
/* 1998-11-18  jl  first beta version with all functionality                 */
/* 1999-10-29  as  beautified for template project                           */
/*****************************************************************************/

/*** Included header files ***/
#include "bldFlags.h"
#include <style.h>
#include <globals.h>
#include <kernel.h>
#include <io.h>
#include <ip.h>                          /* defines variables and constants of IP-protocol */
#include <tcp.h>                         /* defines variables and constants of TCP-protocol */
#include <udp.h>
#include "demo.h"

/*** Global variables ***/
int number1=0;
int number2=0;
int result=0;
```

```
short int sign=1;               //sign if input number;
short int n1e=0;                //number 1 exists
short int n2e=0;                //number 2 exists
char op='n';                    //mathematical operator. n mean that operator doesn't exists
char letter='n';                //input character

/*** Prototypes ***/
void give_result( );


/*************************************************************************/
/* Calculator                                                            */
/*************************************************************************/
void demo( )
{
        int m=0;
        printf("\n\r*** XPortServer Calculator ***\n\r" );
        while( 1 )
        {
     nice( );                                   // obligatory nice() function for
multitasking

                if( kbhit( ) )
                {
                        letter=getch( );        //reading from serial port

                        if((op=='n')&&(n1e==0)&&(letter=='-'))
                        {                       //new input number1 is negative
                                sign=-1;        //negative sign
                                putch('-');
                                number1=0;      //reseting number
                                letter='n';     //it will prevent to execute any condition
                        }                       //till next char will be read
                        if((op!='n')&&(number2==0)&&(letter=='-'))
                        {                       //number2 is negative
                                sign=-1;
                                putch('-');
                                letter='n';     //the same idea
                        }

                        if((letter>='0')&&(letter<='9'))
                        {                       //new digit
                                putch(letter);
                                m=sign*(letter-'0');    //digit value. If number is negative new
                                                        //value is also negative
                                if (op=='n')            //mathematical operator not defined, so it
                                                        //mean, that number1 id beaing reading
                                {
                                        if (n1e==0)             //number1 doesn't exists.
                                        {
                                                n1e=1;          //number1 exists
                                                number1=0;      //reseting number1
```

```
                                }
                                number1*=10;                    //shift1 left al decimal digits
                                number1+=m;                     //puting on the and readed digit
                        }
                        else
                        {
                                number2*=10;                    //similar operation but on
                                                                //number2. After mathematical
                                number2+=m;                     //operations number2=0, so it is
                                                                //not necessery to clean it
                        }
                }
        // Demand of rezult by typing "=" or "+"
                if ((n2e==1)&&(op!='n')&&((letter=='+')||(letter=='=')))
                        {
                                sign=1;                                 //cleaning sign;
                                give_result();                          //score is in number1
                                printf("=%d\n\r", number1);
                                if (letter=='=')
                                        op='n';
                                else
                                {                                       //starting adding
                                        printf("%d+",number1);          //rewriting number1
                                        n1e=1;                          //number1 exists
                                        op='+';                         //setting operator
                                }
                                letter='n';       //no more condition in this loop cycle
                        }

                if((letter=='+')||(letter=='-')||(letter=='*')||(letter=='/')&&(op=='n'))
                {
                        if (n1e==0)             //You want to make operation on Your result
                        {                       //from last mathematical operation
                                n1e=1;          //number1 exists
                                printf("%d",number1);
                        }
                        sign=1;                 //Sign of number2 is positive on start
                        n2e=1;                  //You can start input number2. On start is =0
                        op=letter;              //setting operator
                        putch(letter);
                }
            }
        }
}

void give_result()
{
        int score=-9999;                        //result of division by 0
        if (op=='+') score=(number1+number2);
        if (op=='-') score=(number1-number2);
        if (op=='*') score=(number1*number2);
        if ((op=='/')&&(number2 != 0)) score=(number1/number2);
```

```
        op='n';
        number1=score;
        number2=0;
        n2e=0;
        n1e=0;
}
```

I also removed some portion of code from main.c. This code will be never necessary for Xport. In this file (main.c) was added because this example are not only for Xport but for all products with COB environment.

After removing this code main.c looks:

```
/*****************************************************************************/
/* Main Module :                                                             */
/* Initialize HW, read setup for serial and network interface                */
/* Start customer specific functionality                                     */
/*****************************************************************************/
/* Version  X0.00  demo template                                             */
/*****************************************************************************/
/* History:                                                                  */
/* 1998-11-18  jl  first beta version with all functionality                 */
/* 1999-10-29  as  beautified for template project                           */
/* 2002-10-14  gm  Modified for CPK500                                       */
/* 2003-03-05  pt  Modified for CPK520 XPort additions                       */
/*****************************************************************************/

/* Definitions */
#define  DEF_VARS                /* defines what set variables from GLOBALS.H is used */

/* Included files */
#include "bldFlags.h"
#include <style.h>               /* defines main data types - BYTE, WORD, etc. */
#include <proconf.h>             /* defines some variables of configuration */
#include <globals.h>             /* defines global variables */
#include <hwstruc.h>
#include <io.h>
#include <ip.h>
#include <udp.h>
#include <string.h>
#include <kernel.h>
#include <mbuf.h>
#include "demo.h"                /* defines in "demo.h" true or false DEBUG mode is set */
#if defined (XPORT) || defined (M100)
#ifdef XPORT
#include <\XPort\bitsXP.h>
#endif
#include <\SerFl\ECtypes.h>
#include <\SerFl\serflash.h>
#endif
#include <digio.h>


/* External variables */
```

```
extern void demo(void);
#ifdef XPORT
extern BYTE S_tmpRec7[];                    /* doris - temp copy of setup record 7 */
extern void defaultCP_settings(void);
extern void SaveCPsettings(void);
#endif

/* Global variables */
GLOBAL BYTE InitChar = 0;                   /* Init Char '*' */
GLOBAL WORD ProConf  = PRO_NET;             /* Network support */
GLOBAL WORD TCPSPort = 9999;                /* TCP Setup Port No. */


/* SNMP Variables - must be given! */
/* S_name[] MUST have 12 spaces at the end (or declared 12  bytes larger) */
#ifndef BJB_SNMP
GLOBAL char S_name[] = "Demo Software for CPK Snr xxx-xxx Vy.yy           ";
GLOBAL char S_obj[]  = { 10, 0x2b, 6, 1, 4, 1, 141, 59, 2, 1, 2 };
#endif
static BYTE SetStack[256];              /* defines stack of SETUP process thru COM/Telnet */

#ifdef WEB
#define WEBSSIZE 256
static BYTE ConStack[256];              /* defines stack of SETUP process thru UDP-client */
#endif
void   *sioptr;                         /* defines I/O pointers of TCB for channel 1 */

/* network driver definitions */
#ifdef N0
extern int N0_DRV;
#endif
#ifdef ND
extern int ND_DRV;
#endif

/*****************************************************************************/
GLOBAL void newmain() {
/*****************************************************************************/
   extern void SetProcess( );   /* defines/starts SETUP process thru Telnet */

#ifdef WEB
   extern void WebProcess( );            /* defines/starts mini WEB-server */
   extern void ConfProcess( ); /* defs/starts SETUP process thru TCP-client */
   spawn(WebProcess,kalloc(WEBSSIZE),WEBSSIZE,0,"Web1");   /* WEB SERVER */
   spawn(WebProcess,kalloc(WEBSSIZE),WEBSSIZE,0,"Web2");   /* WEB SERVER */
   spawn(WebProcess,kalloc(WEBSSIZE),WEBSSIZE,0,"Web3");   /* WEB SERVER */
   spawn(ConfProcess, ConStack, sizeof( ConStack ), 0, "Conf" );
#endif

   spawn( SetProcess, SetStack, sizeof( SetStack ), 0, "TConf" );

#pragma warn -eff
#ifdef N0
```

```
  N0_DRV;
#endif
#ifdef ND
  ND_DRV;
#endif
#pragma warn +eff

/* sets LED state */
  BlinkRWord    = 0;                    /* red LED: OFF */
  BlinkGWord[0] = 0xffff;               /* green LED: always ON */
  BlinkGWord[1] = 0;                    /* yellow LED: OFF */

  sdelay(1000);                         /* give tasks tiem to get memory */
  MBufInit();                           /* ready with initialisation, now use rest as buffers */


/* switches pointers of active TCB & CCB to parameters of first serial port  */
  ActCCB = AllCCB;                      /* assigns address AllCCB[0] (COM1) to act CCB */
  ActPro->Chan_Nr = 0;                  /* points to use COM1 (number 0) in active TCB */
  ActPro->CCB_Ptr = AllCCB;             /* changes pointer to active CCB in active TCB */

/* sets all necessary line parameters for first serial port */
  AllCCB[0].V24_mode  = Setup[16];      /* assigns COM1 mode from SETUP to CCB */
  AllCCB[0].V24_speed = Setup[16+1];    /* assigns COM1 speed from SETUP to CCB */
  AllCCB[0].FlowCtl   = Setup[16+2];    /* assigns COM1 flow control to CCB */
  AllCCB[0].Flags     = 0;              /* sets for COM1 Response mode, flushing, etc. */

  InitLocalChan();                      /* ... and initializes interface COM1 */

  sioptr=ActPro->IO_Ptr;
/* THIS STORES AN I/O POINTER; NECESSARY FOR ACCESS*/
demo();                                 /* Start CoBox Demo */
}

void VersionInit(void) {
 extern BYTE fw_type[];
 extern WORD firmwarecheck;

#ifdef FW_TYP
 strcpy(fw_type,FW_TYP);
#endif

#ifdef FW_CHECK
 memcpy(&firmwarecheck,FW_CHECK,2);
 firmwarecheck=ntohs(firmwarecheck);
                        /* put it in right order - ie. 3L instead of L3  = Swap bytes */
#endif

#ifdef XPORT
        if ( EE_Read(S_tmpRec7,896,126) )       /* read setup record 7 (126 bytes) */
        {                                       /* some error */
                defaultCP_settings();           /* factory default values */
                EE_Write(S_tmpRec7, 896, 126);  /* store "new" record 7 data in non-volatile
```

```
                                                                    memory */
        }
                                /* initializae the digital I/O struct based on setup record 7 */
        SaveCPsettings();
        dio_vbit_init();        /* low-level setup of all the digital I/O bits */
#endif

}                               /* main.c ends here */
```