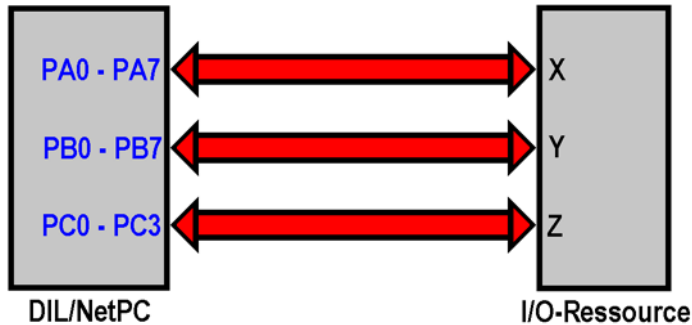


How to program the DNP/1110 Parallel I/O Port (PIO)

- **1. Step:** There are 3 ports with 20 bits: Port A with the bits PA0 to PA7, Port B with PB0 to PB7 and Port C with PC0 to PC3. Design your interface between your external device and the DNP/1110 PIO bits.



- **2. Step:** Write your own application code. Use the following code for a sample to access the Port A direct from a C program, which runs in the Linux user space.

```
// Counter Demo for DIL/NetPC DNP/1110-3V
// Written by MHA - 08.02.2002
// mmap function

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/mman.h>

typedef unsigned char  u8;
typedef unsigned short u16;
typedef unsigned int   u32;

void *pMapmemory(off_t phy_addr, size_t phy_lenght);

// look at SA-1110 Developer's Manual page 9-10

#define GPLR (*(pGPIO_BASE + 0x00 / 4)) /* GPLR @ 0x90040000 */
#define GPDR (*(pGPIO_BASE + 0x04 / 4)) /* GPDR @ 0x90040004 */
#define GPSR (*(pGPIO_BASE + 0x08 / 4)) /* GPSR @ 0x90040008 */
#define GPCR (*(pGPIO_BASE + 0x0C / 4)) /* GPCR @ 0x9004000C */

int main (void)
{
    volatile u32 *pGPIO_BASE;
    int iCnt;

    // check user identity

    if (geteuid() != 0) {
        printf("No root access rights !\n");
        exit(1);
    }
}
```

```

// map SA-1110 GPIO register space at 0x9004000 / size 4096 bytes

pGPIO_BASE = pMapmemory(0x90040000, 0x00001000);

if (pGPIO_BASE == NULL) {
    printf("Can't map memory at 0x90040000 !\n");
    exit(1);
}

printf (" Start Binary Counter for Port A...\n");
printf (" Current Counter Value= 0");

// Set DIL/NetPC PIO Port A = Output

GPDR |= 0xFF00;          // GPDR = 0xFFFFFXX

// Run Counter until User Break by CTRL-C...

for(;;) {

    // Write 8-bit Binary Counter Value to Port A...

    for (iCnt= 0; (iCnt < 256); iCnt++) {

        // Clear Port A

        GPCR = ((iCnt^0xff) << 8);    // GPCR = 0x0000XX00

        // Set Port A

        GPSR = (iCnt << 8);          // GPSR = 0x0000XX00

        printf ("\r Current Counter Value= %3d", iCnt);
        fflush(stdout);
        usleep (100000);
    }
}
return(0);
}

void *pMapmemory(off_t phy_addr, size_t phy_lenght)
{
#define MAP_PAGESIZE 4096UL

    int iFd;
    void *pMem;

    if ((phy_addr % MAP_PAGESIZE) != 0) {
        fprintf(stderr,"physical address error!\n");
        return(NULL);
    }

    if ((phy_lenght % MAP_PAGESIZE) != 0) {
        fprintf(stderr,"physical lenght error!\n");
        return(NULL);
    }
}

```

```
    }

    // open mem device for read/write

    iFd = open("/dev/mem", O_RDWR | O_SYNC);
    if (iFd < 0) {
        fprintf(stderr, "open of /dev/mem fail !\n");
        return(NULL);
    }

    // get pointer to DNP1110 memory

    pMem = mmap(NULL,
                phy_lenght,
                (PROT_READ | PROT_WRITE),
                MAP_SHARED,
                iFd,
                phy_addr);

    if ((pMem == MAP_FAILED) || (pMem == NULL)) {
        fprintf(stderr, "mmap of /dev/mem fail !\n");
        return(NULL);
    }

    // close mem device

    if (close(iFd) != 0)
        fprintf(stderr, "close of /dev/mem fail !\n");

    return(pMem);
}
```

Appendix 1: DNP/1110 Pinout - JEDEC 64-pin DIL Connector (1. Part)

<i>Pin</i>	<i>Name</i>	<i>Group</i>	<i>Function</i>
1	PA0	PIO	Parallel I/O, Port A, Bit 0
2	PA1	PIO	Parallel I/O, Port A, Bit 1
3	PA2	PIO	Parallel I/O, Port A, Bit 2
4	PA3	PIO	Parallel I/O, Port A, Bit 3
5	PA4	PIO	Parallel I/O, Port A, Bit 4
6	PA5	PIO	Parallel I/O, Port A, Bit 5
7	PA6	PIO	Parallel I/O, Port A, Bit 6
8	PA7	PIO	Parallel I/O, Port A, Bit 7
9	PB0	PIO	Parallel I/O, Port B, Bit 0
10	PB1	PIO	Parallel I/O, Port B, Bit 1
11	PB2	PIO	Parallel I/O, Port B, Bit 2
12	PB3	PIO	Parallel I/O, Port B, Bit 3
13	PB4	PIO	Parallel I/O, Port B, Bit 4
14	PB5	PIO	Parallel I/O, Port B, Bit 5
15	PB6	PIO	Parallel I/O, Port B, Bit 6
16	PB7	PIO	Parallel I/O, Port B, Bit 7
17	PC0	PIO	Parallel I/O, Port C, Bit 0
18	PC1	PIO	Parallel I/O, Port C, Bit 1
19	PC2	PIO	Parallel I/O, Port C, Bit 2
20	PC3	PIO	Parallel I/O, Port C, Bit 3
21	RXD1	SIO	COM1 Serial Port, RXD Pin
22	TXD1	SIO	COM1 Serial Port, TXD Pin
23	CTS1	SIO	COM1 Serial Port, CTS Pin
24	RTS1	SIO	COM1 Serial Port, RTS Pin
25	DCD1	SIO	COM1 Serial Port, DCD Pin
26	DSR1	SIO	COM1 Serial Port, DSR Pin
27	DTR1	SIO	COM1 Serial Port, DTR Pin
28	RI	SIO	COM1 Serial Port, RI Pin
29	RESIN	RESET	RESET Input
30	TX+	LAN	Ethernet Interface, TX+ Pin
31	TX-	LAN	Ethernet Interface, TX- Pin
32	GND	----	Ground

Table 1: DNP/1110 Pinout - Pin 1 to 32

Appendix 2: DNP/1110 Pinout - JEDEC 64-pin DIL Connector (2. Part)

<i>Pin</i>	<i>Name</i>	<i>Group</i>	<i>Function</i>
33	RX+	LAN	Ethernet Interface, RX+ Pin
34	RX-	LAN	Ethernet Interface, RX- Pin
35	RESOUT	RESET	RESET Output
36	VBAT	PSP	Real Time Clock Battery Input
37	CLKOUT	PSP	Clock Output (Default 3.6864 MHz)
38	TXD2	PSP	COM2 Serial Port, TXD Pin
39	RXD2	PSP	COM2 Serial Port, RXD Pin
40	INT5	PSP	Interrupt Input 5
41	INT4	PSP	Interrupt Input 4
42	INT3	PSP	Interrupt Input 3
43	INT2	PSP	Interrupt Input 2
44	INT1	PSP	Interrupt Input 1
45	CS4	PSP	Chip Select Output 4
46	CS3	PSP	Chip Select Output 3
47	CS2	PSP	Chip Select Output 2
48	CS1	PSP	Chip Select Output 1
49	RDY	PSP	External Ready Input
50	RD	PSP	Read Signal, Expansion Bus
51	WR	PSP	Write Signal, Expansion Bus
52	SA3	PSP	Expansion Bus, Address Bit 3
53	SA2	PSP	Expansion Bus, Address Bit 2
54	SA1	PSP	Expansion Bus, Address Bit 1
55	SA0	PSP	Expansion Bus, Address Bit 0
56	SD7	PSP	Expansion Bus, Data Bit 7
57	SD6	PSP	Expansion Bus, Data Bit 6
58	SD5	PSP	Expansion Bus, Data Bit 5
59	SD4	PSP	Expansion Bus, Data Bit 4
60	SD3	PSP	Expansion Bus, Data Bit 3
61	SD2	PSP	Expansion Bus, Data Bit 2
62	SD1	PSP	Expansion Bus, Data Bit 1
63	SD0	PSP	Expansion Bus, Data Bit 0
64	VCC	----	3.3 Volt Power Input

Table 2: DNP/1110 Pinout - Pin 33 to 64