W7100A Application Note
External Data Memory Interface

Version 1.0.1

For more information, visit our website at http://www.wiznet.co.kr

Table of Contents

1 Introduction

W7100A is a high performance one-chip internet solution with embedded 8051 compatible microcontroller, 64KB SRAM, 255Byte Data flash and Hardwired TCP/IP Core. Users can use as much as 16MB of the external memory if needed and the extended external memory can be accessed only by MOVX command. Figure 1 is the data memory map of W7100A and shows that the 0x01000 ~ 0xFDFFFF domain after 64KB RAM can be used as extended external memory.

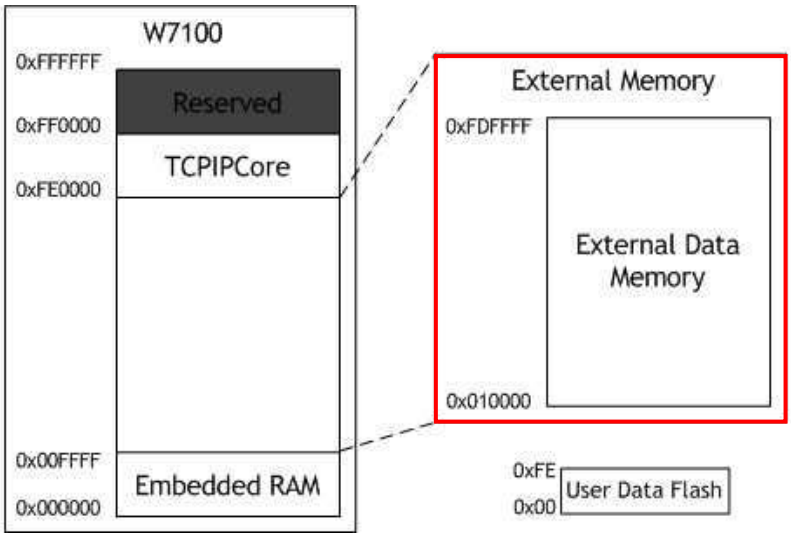

Figure 1. W7100A Data Memory Map

The external memory of W7100A can be accessed by using 4 8bit ports. There are two ways to control the 4 ports from P0 to P3; the selection of how to control is by modifying the EM[2:0] bit of the WCONF register.

1 The Standard mode is similar as the general 8051 external interfaces; it controls the latch enable signal by using the ALE (Address Latch Enable) pin to distinguish the address and data signal of P0. The ALE holding time can be controlled through ALECON register. **Standard 1** mode is when the standard 8051 mode uses a 16bit address, and **Standard 2** mode is when a 24bit address is used.

2 Direct mode is unlike regular 8051 external connection methods; it directly connects the address and data line. **Direct 1** mode is when the direct mode uses a 16bit address, and **Direct 2** mode is when a 24bit address is used.

The traditional 8051 core supports 16bit address domain and the memory can be accessed as much as 64KB. However, the external memory domain is after 64KB or the after 0x010000, so SFR (Special Function Registers) is used in order to access the extended memory of the 8051 core.
Examples of SFR are DPX0, DPX1, and MXAX.

This application note explains how to use the external memory of W7100A.

Section 2 will explain on each mode and how to use the access function and further details will be explained in section 2.1 and 2.2. Section 3 will show examples of each mode reading and writing the external memory. Please refer to document ‘W7100A Datasheet section 2.3 External Data Memory’ for more details on external memory.

2 External Data Memory Access

Like regular 8051 external interface, the external memory of W7100A can be accessed by using 4 modes (Standard1, Standard 2, Direct 1, Direct 2) and the mode can be selected by the EM[2:0] bit settings. Table 1 and 2 shows the bit and port settings for the WCONF register and each mode.

Table 1. W7100A Configuration Register

7	6	5	4	3	2	1	0	Reset
RB	ISPEN	EM2	EM1	EM0	Reserved	FB	BE	0x00

WCONF(0xFF): W7100A configuration register

Note: RB : Reboot after 1 – ISP (APP Entry(0xFFF7 ~ 0xFFFF) RD/WR Enable)
No reboot after 0 **–** ISP

ISPEN : 0 **–** ISP enable, 1 – ISP disable

EM[2:0] : External Memory Mode Setting.

FB : FLASH Busy Flag for ISP. Read only.

BE : Boot Enable (1 – Boot Running / 0 – Apps Running). Read only.

Table 2. External memory access mode

Mode	EM[2:0]	P0[7:0]	P1[7:0]	P2[7:0]	P3[7:0]
Standard 1	001	Addr[7:0] / Data[7:0]	GPIO	Addr[15:8]	GPIO
Standard 2	011	Addr[7:0] / Data[7:0]	GPIO	Addr[15:8]	Addr[23:16]
Direct 1	101	Data[7:0]	Addr[7:0]	Addr[15:8]	GPIO
Direct 2	111	Data[7:0]	Addr[7:0]	Addr[15:8]	Addr[23:16]

The traditional 8051 core uses 16bit address system and the memory can be accessed as much as 64KB. The memory domain after the 64KB is 0x010000 and is also external memory domain; to access this domain, data page pointer (DPX0 and DPX1) is used. For example, to access the external memory 0x010000 using Standard 1 or Direct 1, set DPX0 as 0x01 and fix the top bit.

In this case, the 16bit EXTWTST register is used to control the external memory access speed; the value can be set from 0 to 65535. Please refer to document ‘W7100A Datasheet Section 2.5.8 External Memory Wait States Register’ for more details.

The example code of the Read and Write function for DPX0 settings of the external memory is as below.

Code Example – External memory read function

```
sfr DPX0 = 0x93;     /* Data Page Pointer 0 */
sfr DPX1 = 0x95;     /* Data Page Pointer 1 */

unsigned char ExMEM_READ(unsigned int addr)
{
        unsigned char dat;
        unsigned char tmpEA;

        tmpEA = EA;                 // Interrupt register backup
        EA = 0;                     // Interrupt disable
        DPX0 = 0x01;                // DPX0 for External memory
        dat = *((volatile unsigned char xdata*)(addr));
        DPX0 = 0x00;                // DPX0 clear
        EA = tmpEA;                 // Interrupt register recover

        return dat;                 // Return read data
}
```

Code Example – External memory write function

```
sfr DPX0 = 0x93;     /* Data Page Pointer 0 */
sfr DPX1 = 0x95;     /* Data Page Pointer 1 */

void ExMEM_WRITE(unsigned int addr, unsigned char dat)
{
        unsigned char tmpEA;

        tmpEA = EA;                 // Interrupt register backup
        EA = 0;                     // Interrupt disable
        DPX0 = 0x01;                // DPX0 for External memory
        *((volatile unsigned char xdata*)(addr)) = dat;
        DPX0 = 0x00;                // DPX0 clear
        EA = tmpEA;                 // Interrupt register recover
}
```

2.1 Standard Mode

The Standard mode is similar as the general 8051 external interfaces; it controls the latch enable signal by using the ALE (Address Latch Enable) pin to distinguish the address and data signal of P0. The ALE holding time can be controlled through ALECON register and needs to be modified according to the memory. The remaining ports beside the address and data signal can be used for GPIO. Please refer to document 'W7100A Datasheet Section 2.5.7 Address Latch Enable Register' for more details on ALECON register.

> Note :
> Please be aware that the RD/WR signal for W7100A will work only if an OR gate connected to ALE is done when the Read / Write signal connects to the memory nWE, nOE (Write Enable, Output Enable).
> Please take note and design your hardware according to this fact.

In Standard 1 mode, the WCONF register value is **0x48**; use (0100 1000) to set the EM[2:0]bit to be (001)b. In Standard 2 mode, use **0x58** (0101 1000)b to set the EM[2:0]bit to be (011)b. Then, use the example code above from Section 2 to use Read and Write functions to access the external memory.

The block diagrams of the W7100A external memory in Standard mode are as below.

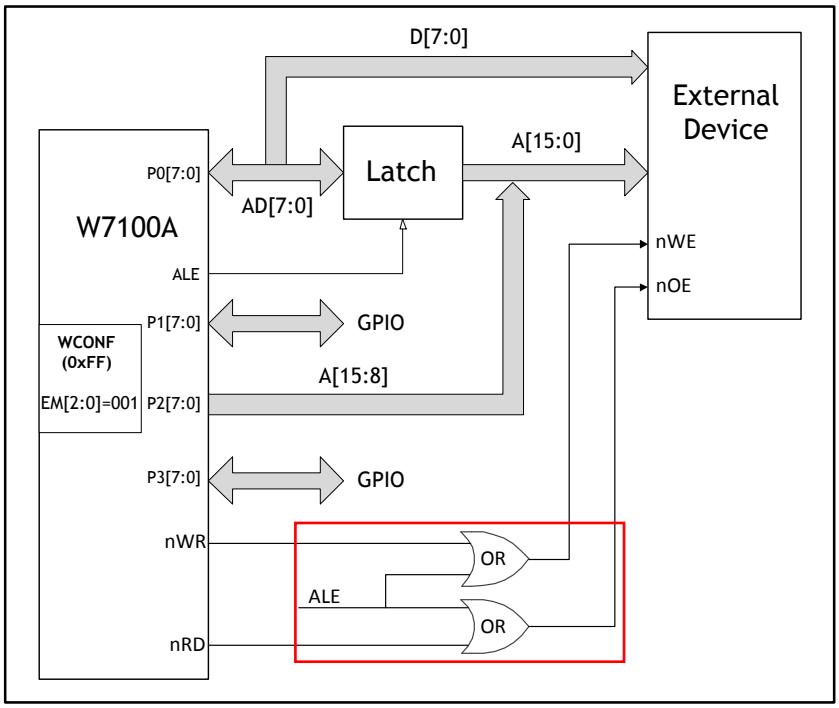

Figure 2. Standard 8051 External Pin Access Mode, EM[2:0] = (001)b

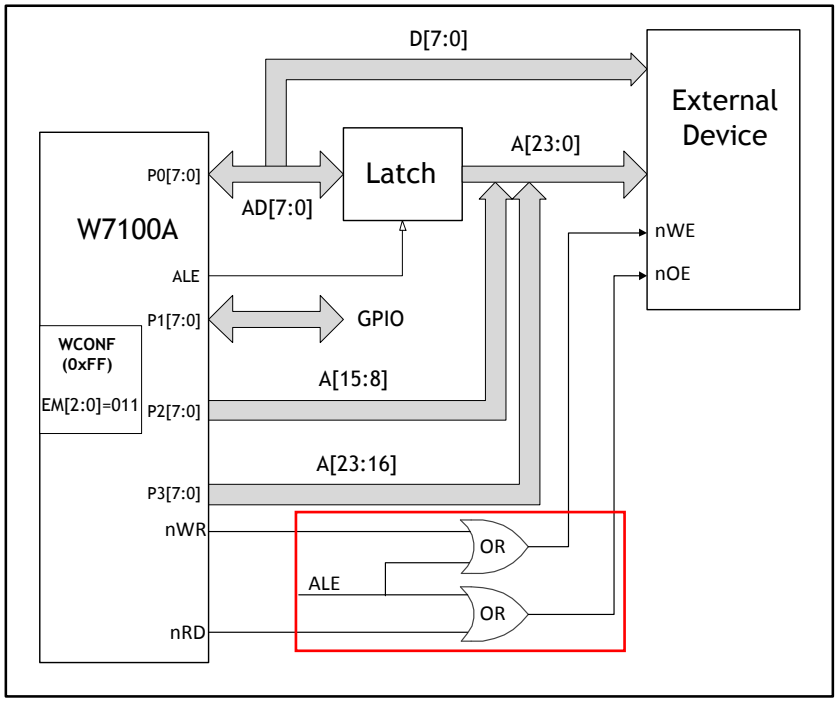

Figure 3. Standard 8051 External Pin Access Mode, EM[2:0] = (011)b

2.2 Direct Mode

Direct mode is when the user directly connects all addresses and data pins. Unlike 8051 latch mode, there are no additional control signals, but more ports are used to directly connect to the external memory. The remaining ports beside the address and data signal can be used for GPIO

In Direct 1 mode, the WCONF register value is **0x68**; use (0120 1000) to set the EM[2:0]bit to be (101)b. In Direct 2 mode, use **0x78** (0111 1000)b to set the EM[2:0]bit to be (111)b. Then, use the example code above from Section 2 to use Read and Write functions to access the external memory.

The block diagrams of the W7100A external memory in Direct mode are as below.

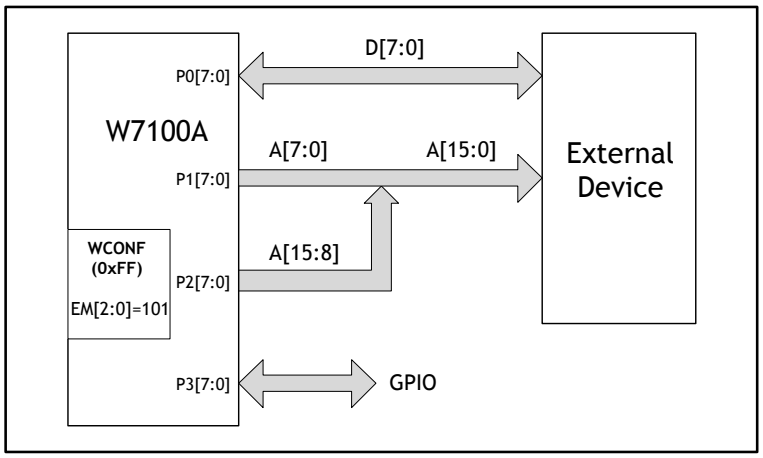

Figure 4. Direct 8051 External Pin Access Mode, EM[2:0] = (101)b

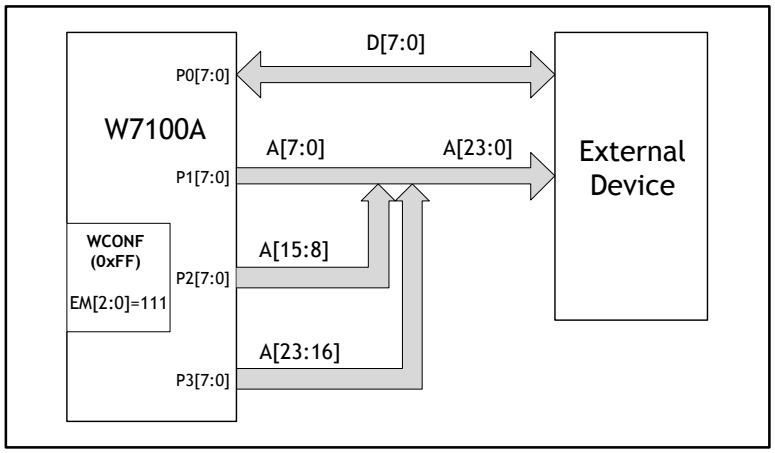

Figure 5. Direct 8051 External Pin Access Mode, EM[2:0] = (111)b

3 Example Demonstration

This section will show example codes for checking and verifying the value (Kbyte units) written and read in the 32KB external SRAM (IS62LV256AL). Since 32KB of memory was used, Chip enable (CE or CS) signal was connected to the unused Addr[15]. Therefore, if Addr[15] is set as 0, the CE signal of the external memory will be enabled and the Read / Write is possible.

Example codes are for testing in Standard 1 mode and Direct 1 mode.

1. Write

Write 0x01 for the domain of 1Kbyte from the external memory's base address (0x010000), and write 0x02 for the domain of 2Kbyte. Write a value increased by 1 for each Kbyte domain.

2. Read and Verify

Read and verify the value used.

Note:

The program codes used in this document are all pseudo codes.

Example codes are based on KEIL µvision V4.10 compiler.

Code Example – 32KB External SRAM Read / Write Test Application

```
#define             MEMSIZE                     0x7fff      // 32KB

/* Mode select */
//#define           STANDARD1
#define             DIRECT1

void ExMEM_TEST(void)
{
        unsigned char xdata read;       // Read data
        unsigned char xdata d;          // Write data for test
        unsigned int xdata rcount, ecount;
        unsigned int xdata i;

        // WCONF: W7100 Configuration Register
        // ALECON: Address Latch Enable Register
#ifdef STANDARD1
        WCONF |= 0x48;              // Standard 1 mode, (0100 1000)b
        ALECON = 0xff;              // default
#endif
#ifdef DIRECT1
```

```
        WCONF |= 0x68;              // Direct 1 mode, (0110 1000)b
        ALECON = 0;
#endif

        // External memory WR/RD timing control (16-bits), default : 0xffff
        EXTWTST0 = 0x10;
        EXTWTST1 = 0x00;

        /* 1. Write */
        d = 0x01;
        for(i = 0x0000; i <= MEMSIZE; i++)
        {
                ExMEM_WRITE(i, d);      // Data write
                if (((i+1) % 0x0400) == 0)
                {
                        d++;
                }
        }

        /* 2. Read and compare */
        d = 0x01;
        rcount = 0;     // Read count
        ecount = 0;     // Error count

        for(i = 0x0000; i <= MEMSIZE; i++)
        {
                if((i % 0x10) == 0x00) printf("[0x01%.4x] ", i);
                read = ExMEM_READ(i);       // Data read
                if(read == d)
                {
                        printf("%.2bx ", read);
                }
                else
                {
                        printf("%.2bx ", read);
                        ecount++;
                }

                if((i % 0x10) == 0x0F) printf("\r\n");
                if (((i+1) % 0x0400) == 0)
                {
                        d++;
                }
                rcount++;
        }
        printf("[ExMEM R/W Test result] Written byte count = %u,
                                                Error count = %u\r\n", rcount, ecount);
}
```

The results of the example code through the terminal are as below.

< Standard 1 mode >

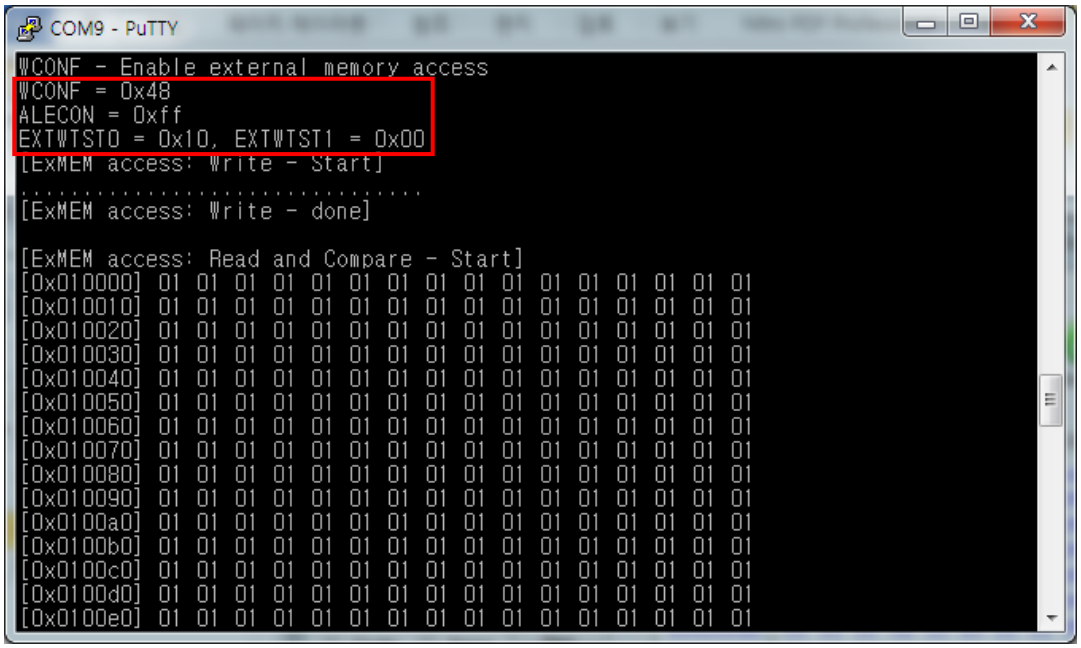

Figure 6. Application Example Results-1 : Standard 1 mode

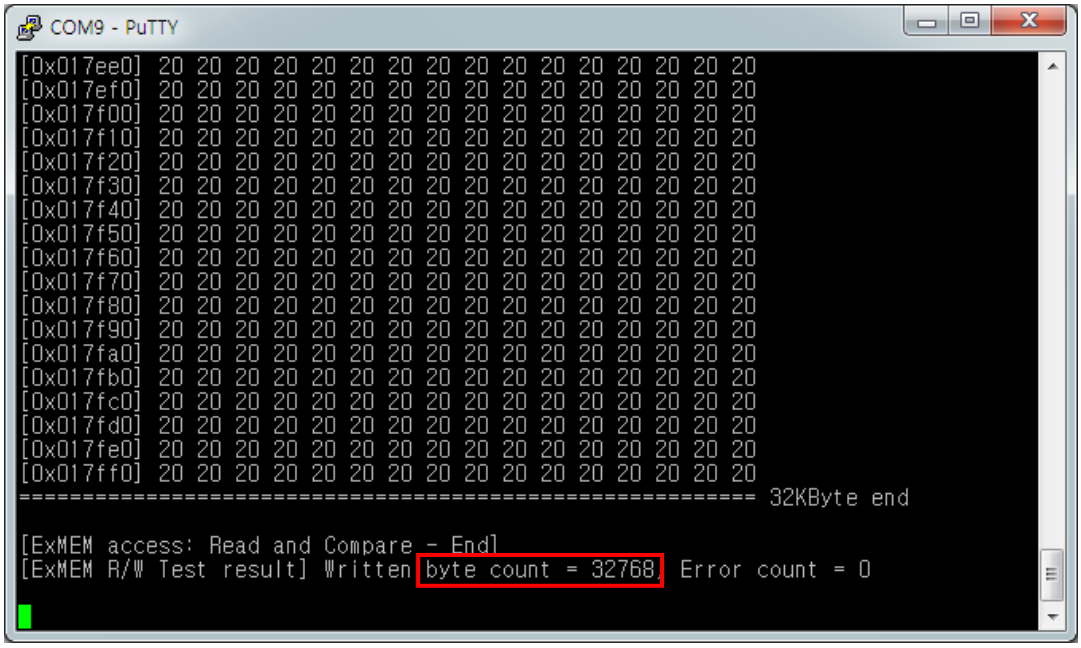

Figure 7. Application Example Results-2 : Standard 1 mode

< Direct 1 mode >

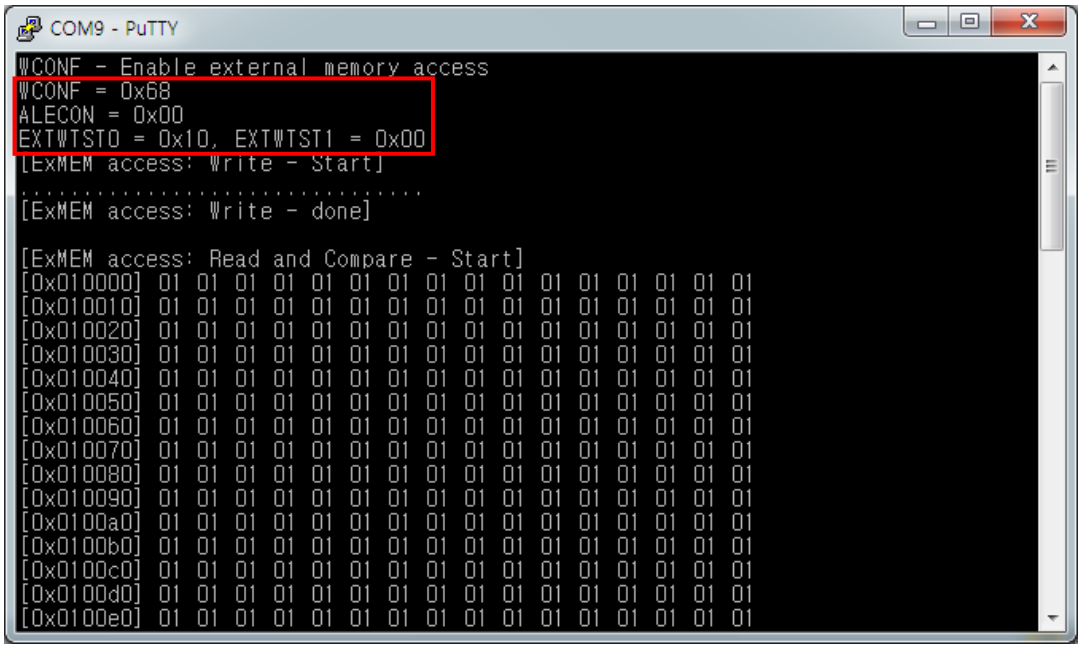

Figure 8. Application Example Results-3 : Direct 1 mode

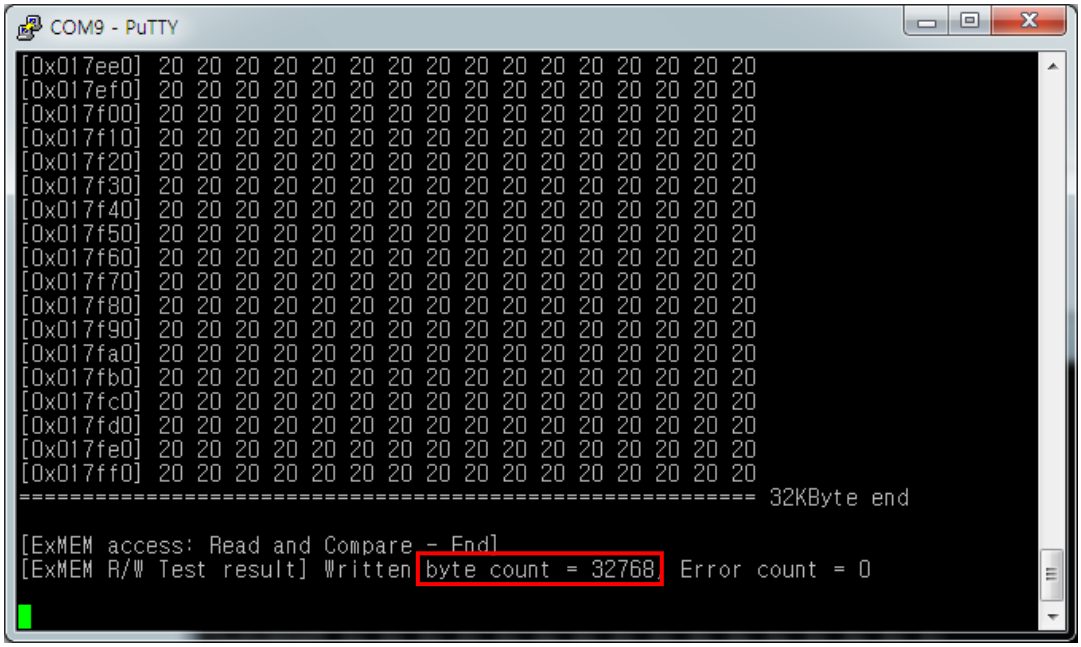

Figure 9. Application Example Results-4 : Direct 1 mode

Document History Information

Version	Date	Descriptions
Ver. 1.0.0	Sep, 2012	Application note released

Technical Support: support@wiznet.co.kr
Sales & Distribution: sales@wiznet.co.kr
For more information, visit our website at http://www.wiznet.co.kr