

Source code of "rom_dump.cpp"

```

/*****
Program name:   rom_dump
Module name:    rom_dump.cpp
Description:    The program converts "PA.txt" and makes "SST 27SF512-*.bin" for
                dumping NEC uPD777 program ROM and pattern ROM contents.
                Input files : "PA.txt" which includes dump patterns by texts
                Output files : "SST 27SF512-3.bin", "SST 27SF512-2.bin",
                                "SST 27SF512-1.bin" byte packed binary files
                                to be programmed by EEPROM programmer

Usage:         rom_dump <enter>
Version:       1.0
Date:         June, 2, 2021
Programmer:    Tetsuji Oguchi
(C) Oguchi R&D 2021
*****/

#include <stdio.h>
#include <string.h>
#include <process.h>
// #pragma warning(disable: 4996)

// #define RELEASE

static char   infile[] = "PA.txt", out3file[] = "SST 27SF512-3.bin";
static char   out2file[] = "SST 27SF512-2.bin", out1file[] = "SST 27SF512-1.bin";
static char   ibuf[40], obuf[3][40];
static int    i, j[8] = {0x80, 0x40, 0x20, 0x10, 0x08, 0x04, 0x02, 0x01}, k;
static int    block, length;

FILE   *patifp, *bin3ofp, *bin2ofp, *bin1ofp;

int main()
{
    printf("NEC uPD777 program ROM & pattern ROM dump program\n");
    printf("                (C) Oguchi R&D 2021\n\n");

    // Check & specify file I/O
    if (fopen_s(&patifp, "PA.txt", "r"))
    {
        printf("Input file PA.txt doesn't exist\n");
        exit(0);
    }
    if (fopen_s(&bin3ofp, "SST 27SF512-3.bin", "wb"))
    {
        printf("Output file SST 27SF512-3.bin open error...\n");
        exit(0);
    }
    if (fopen_s(&bin2ofp, "SST 27SF512-2.bin", "wb"))
    {
        printf("Output file SST 27SF512-2.bin open error...\n");
        exit(0);
    }
    if (fopen_s(&bin1ofp, "SST 27SF512-1.bin", "wb"))
    {
        printf("Output file SST 27SF512-1.bin open error...\n");
        exit(0);
    }
    // Start dump
    fgets(ibuf, 40, patifp);

```

```

while (!feof(patifp))
{
    if (ibuf[0] != '-')
    {
        i = 0;
        k = 0;
        length = 8;
        block = 0;
        while (block < 3)
        {
            while (i < length)
            {
                if (ibuf[(block * 8) + i] == '1') k += j[i];
                i++;
            }
            switch (block)
            {
                case 0: fputc(k, bin3ofp); break;
                case 1: fputc(k, bin2ofp); break;
                case 2: fputc(k, bin1ofp); break;
            }
            i = 0;
            k = 0;
            block++;
        }
    }
    fgets(ibuf, 40, patifp);
}

```

```

// End dump
printf("NEC uPD777 program ROM & pattern ROM dump completed\n");

```

```

fclose(patifp);
fclose(bin3ofp);
fclose(bin2ofp);
fclose(bin1ofp);
}

```