Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

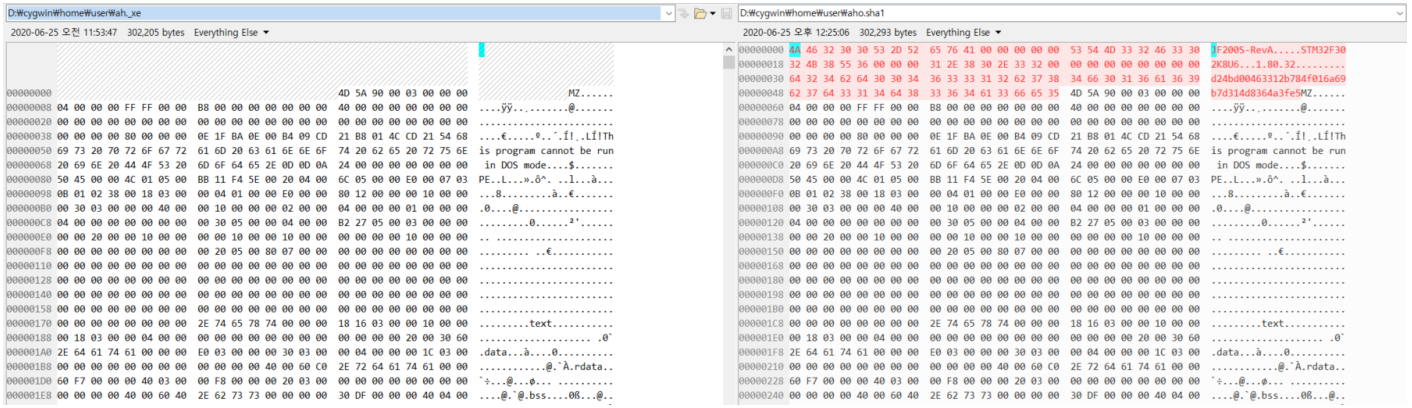
“ah.exe” version 2.16

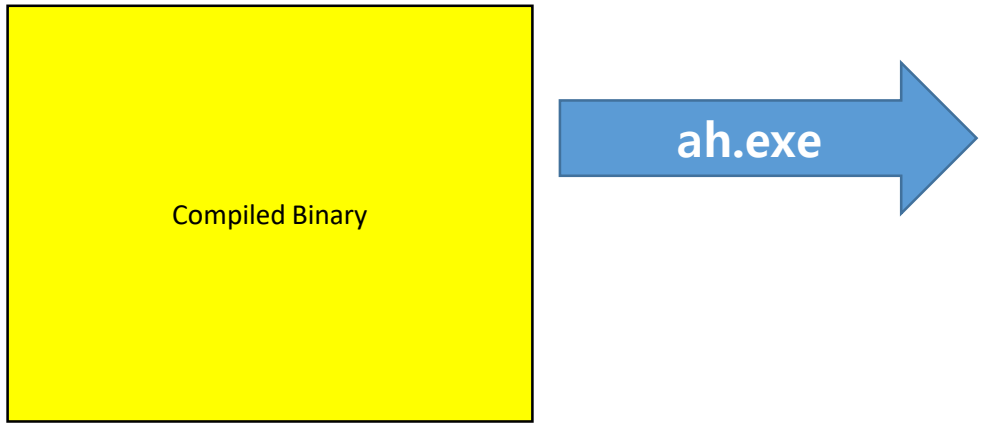
Guide

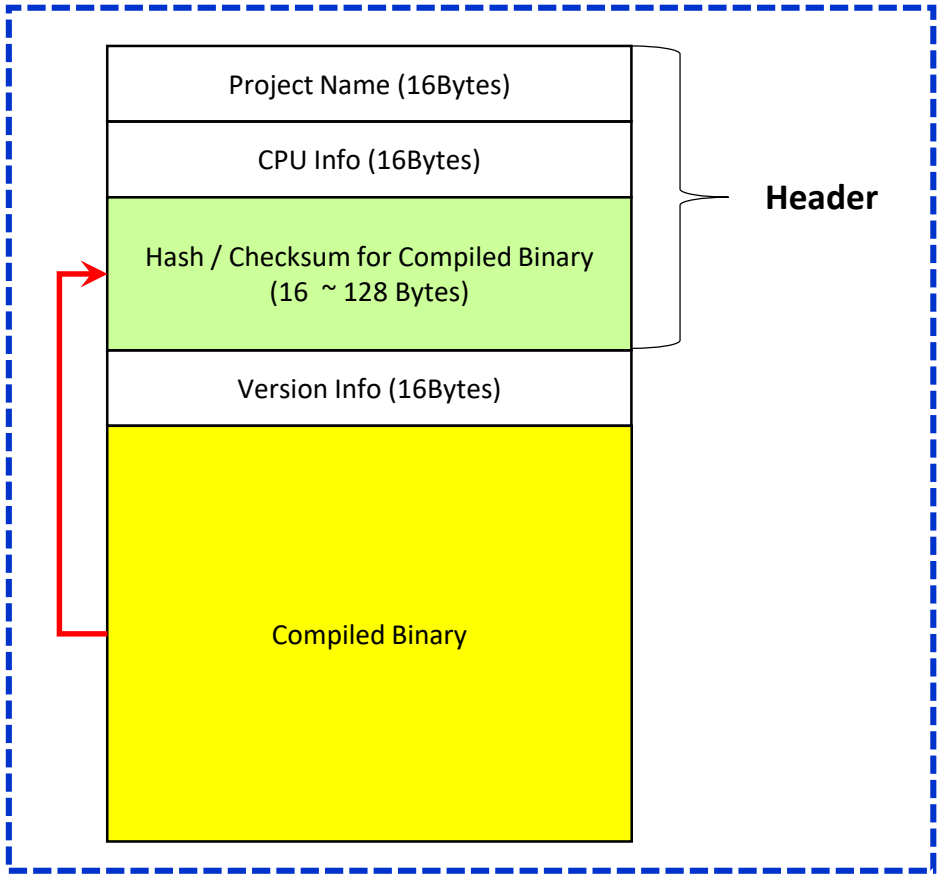
2020. 10. 14

■ Signed firmware 생성하기

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo



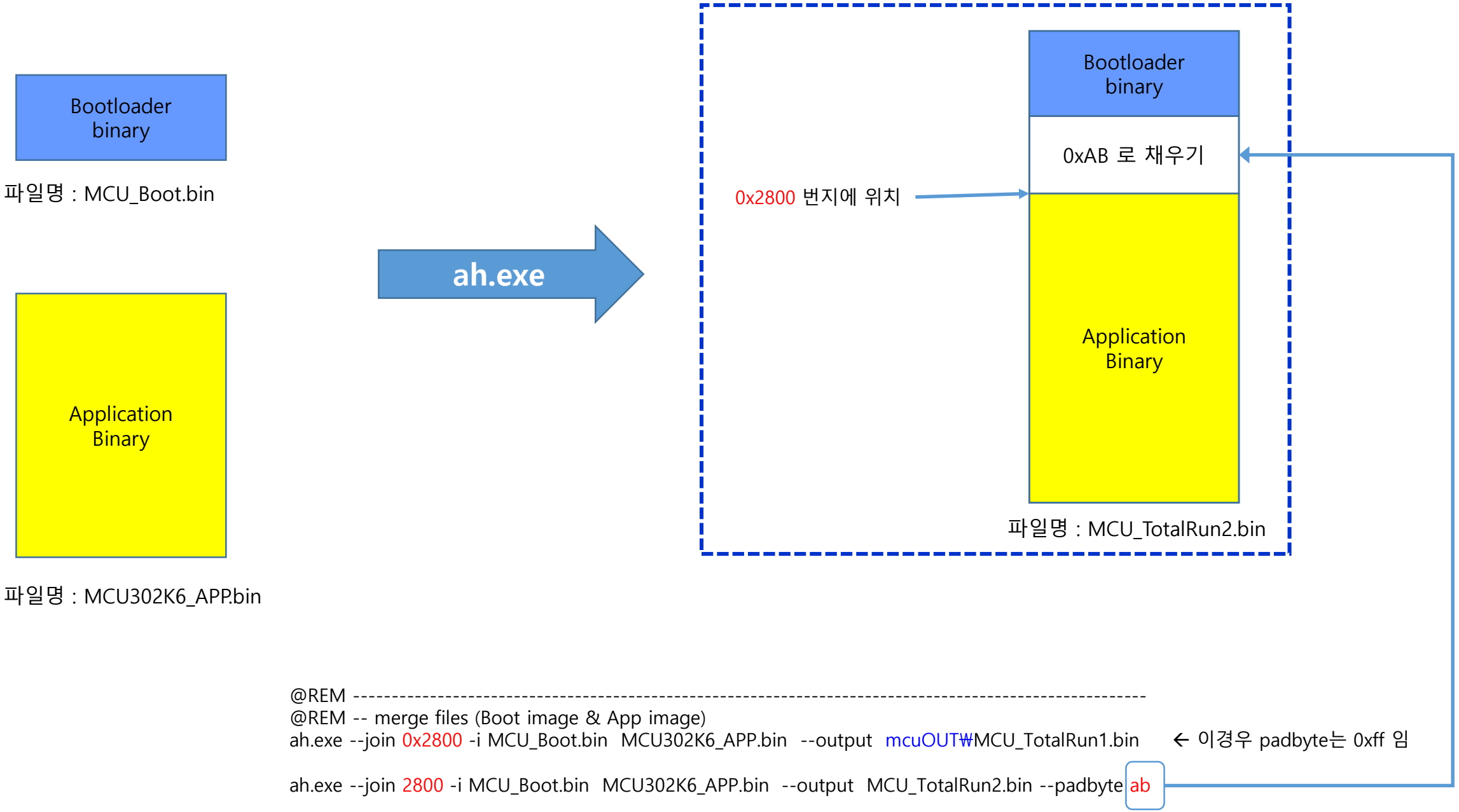





```
@REM ----------------------------------------------------------------------------------------------------
@REM --- Attach Header to binary for making the signed firmware ----
@REM
@REM ah.exe -b M200S-RevA  -m STM32F302K8U6 --cinfo MD5      -i ah.exe  -o aho.R10   -v E4.1.2
ah.exe -b M200S-RevA  -m STM32F302K8U6 --cinfo sha1      --input ah.exe  --output aho.sha1   -v 10.83
@REM ah.exe -b M200S-RevA   -m STM32F302K8U6 --cinfo sha224   --input ah.exe  --output  OUT₩aho.R30   -v M1.8.3
@REM ah.exe -b M200S-RevA   -m STM32F302K8U6 --cinfo date   --input ah.exe  --output aho.RU1   -v 2.005
@REM ah.exe -b M200S-RevA   -m STM32F302K8U6 --cinfo crc16c  --input ah.exe  --output aho.R40  -v 01.08.13
ah.exe -b M200S-RevA   -m STM32F302K8U6 --cinfo CRC64  --input ah.exe  --output aho.R50  -v 11.88.33
@REM ah.exe -b M200S-RevA   -m STM32F302K8U6 --cinfo adler32  --input ah.exe  --output aho.R60  -v 1083.1234
```

■ Bootloader 와 Application/Kernel 합치기

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

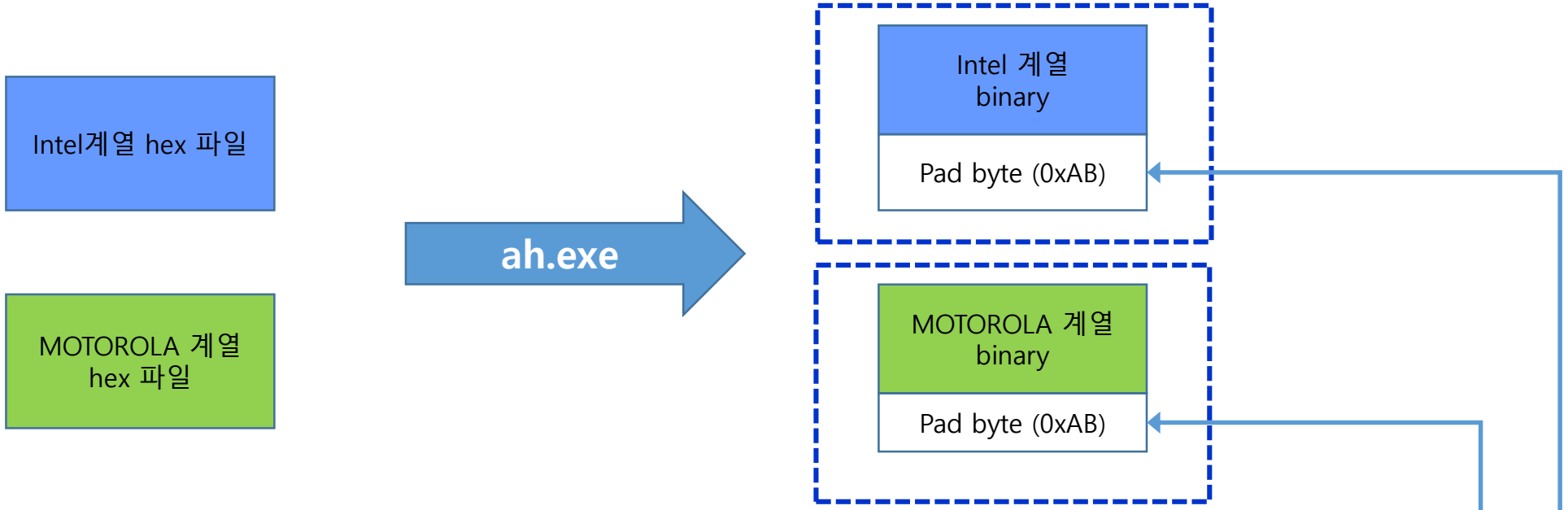



```
@REM ----------------------------------------------------------------------------------------------------
@REM -- merge files (Boot image & App image)
ah.exe --join 0x2800 -i MCU_Boot.bin  MCU302K6_APP.bin  --output  mcuOUT₩MCU_TotalRun1.bin     ← 이경우 padbyte는 0xff 임

ah.exe --join 2800 -i MCU_Boot.bin  MCU302K6_APP.bin  --output  MCU_TotalRun2.bin --padbyte ab
```

■ hexa to binary 생성하기

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo




```
@REM ----------------------------------------------------------------------------------------------------
@REM=== Convert Intel Hex to Bin ======
ah.exe -i micom_jf8.hex     -o  binary₩MCU_JF8.bin   --intel
ah.exe -i micom_SL.hex     -o  binary₩MCU_SL_max.bin   --intel --length a0000
ah.exe -i micom_917.hex    -o binary₩MCU_917_ff_max.bin    --intel --length a0000
ah.exe -i micom_917.hex    -o binary₩MCU_917_ff_max.bin    --intel --length 640KB
@REM
ah.exe -i micom_917.hex    -o MCU_917_ab_max.bin   --intel --length a0000   --padbyte ab
@REM
@REM=== Convert motorolA Hex to Bin ======
ah.exe -i mahind.hex      -o MCU_mahind.bin    --motorola
ah.exe -i romp.hex        -o MCU_romp_cc_max.bin    --motorola --length 10000 --padbyte cc
ah.exe -i romp.hex        -o MCU_romp_ff_max.bin          --motorola --length 10000
ah.exe -i romp.hex        -o binary₩MCU_romp_7e_max_ALL.bin    --motorola --length 10000   --verbose 1 --padbyte 7e --padarea allarea
@REM
```

■ Hash sum 생성하기

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
@REM ----------------------------------------------------------------------------------------------------
@REM -- Hash ---
ah.exe -i @.@   -o SHA1sum.txt   --checksum SHA1 --verbose datesize
ah.exe -i $.$   -a MD5sum.txt   --checksum MD5 --verbose date
ah.exe -i $.zip   -a MD5sum.txt   --checksum sha256 --verbose size
@REM
@REM ----------------------------------------------------------------------------------------------------
@REM
```

■ make_crc16()

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
unsigned int  crctable[256] =
{
        0x0000, 0x1189, 0x2312, 0x329B, 0x4624, 0x57AD, 0x6536, 0x74BF,
        0x0919, 0x1890, 0x2A0B, 0x3B82, 0x4F3D, 0x5EB4, 0x6C2F, 0x7DA6,
        0x1232, 0x03BB, 0x3120, 0x20A9, 0x5416, 0x459F, 0x7704, 0x668D,
        0x1B2B, 0x0AA2, 0x3839, 0x29B0, 0x5D0F, 0x4C86, 0x7E1D, 0x6F94,
        0x2464, 0x35ED, 0x0776, 0x16FF, 0x6240, 0x73C9, 0x4152, 0x50DB,
        0x2D7D, 0x3CF4, 0x0E6F, 0x1FE6, 0x6B59, 0x7AD0, 0x484B, 0x59C2,
        0x3656, 0x27DF, 0x1544, 0x04CD, 0x7072, 0x61FB, 0x5360, 0x42E9,
        0x3F4F, 0x2EC6, 0x1C5D, 0x0DD4, 0x796B, 0x68E2, 0x5A79, 0x4BF0,
        0x48C8, 0x5941, 0x6BDA, 0x7A53, 0x0EEC, 0x1F65, 0x2DFE, 0x3C77,
        0x41D1, 0x5058, 0x62C3, 0x734A, 0x07F5, 0x167C, 0x24E7, 0x356E,
        0x5AFA, 0x4B73, 0x79E8, 0x6861, 0x1CDE, 0x0D57, 0x3FCC, 0x2E45,
        0x53E3, 0x426A, 0x70F1, 0x6178, 0x15C7, 0x044E, 0x36D5, 0x275C,
        0x6CAC, 0x7D25, 0x4FBE, 0x5E37, 0x2A88, 0x3B01, 0x099A, 0x1813,
        0x65B5, 0x743C, 0x46A7, 0x572E, 0x2391, 0x3218, 0x0083, 0x110A,
        0x7E9E, 0x6F17, 0x5D8C, 0x4C05, 0x38BA, 0x2933, 0x1BA8, 0x0A21,
        0x7787, 0x660E, 0x5495, 0x451C, 0x31A3, 0x202A, 0x12B1, 0x0338,
        0x8C48, 0x9DC1, 0xAF5A, 0xBED3, 0xCA6C, 0xDBE5, 0xE97E, 0xF8F7,
        0x8551, 0x94D8, 0xA643, 0xB7CA, 0xC375, 0xD2FC, 0xE067, 0xF1EE,
        0x9E7A, 0x8FF3, 0xBD68, 0xACE1, 0xD85E, 0xC9D7, 0xFB4C, 0xEAC5,
        0x9763, 0x86EA, 0xB471, 0xA5F8, 0xD147, 0xC0CE, 0xF255, 0xE3DC,
        0xA82C, 0xB9A5, 0x8B3E, 0x9AB7, 0xEE08, 0xFF81, 0xCD1A, 0xDC93,
        0xA135, 0xB0BC, 0x8227, 0x93AE, 0xE711, 0xF698, 0xC403, 0xD58A,
        0xBA1E, 0xAB97, 0x990C, 0x8885, 0xFC3A, 0xEDB3, 0xDF28, 0xCEA1,
        0xB307, 0xA28E, 0x9015, 0x819C, 0xF523, 0xE4AA, 0xD631, 0xC7B8,
        0xC480, 0xD509, 0xE792, 0xF61B, 0x82A4, 0x932D, 0xA1B6, 0xB03F,
        0xCD99, 0xDC10, 0xEE8B, 0xFF02, 0x8BBD, 0x9A34, 0xA8AF, 0xB926,
        0xD6B2, 0xC73B, 0xF5A0, 0xE429, 0x9096, 0x811F, 0xB384, 0xA20D,
        0xDFAB, 0xCE22, 0xFCB9, 0xED30, 0x998F, 0x8806, 0xBA9D, 0xAB14,
        0xE0E4, 0xF16D, 0xC3F6, 0xD27F, 0xA6C0, 0xB749, 0x85D2, 0x945B,
        0xE9FD, 0xF874, 0xCAEF, 0xDB66, 0xAFD9, 0xBE50, 0x8CCB, 0x9D42,
        0xF2D6, 0xE35F, 0xD1C4, 0xC04D, 0xB4F2, 0xA57B, 0x97E0, 0x8669,
        0xFBCF, 0xEA46, 0xD8DD, 0xC954, 0xBDEB, 0xAC62, 0x9EF9, 0x8F70
};


uint16_t make_crc16(uint16_t crc_seed, unsigned char *c_ptr, unsigned int len)
{
        uint16_t crc = crc_seed; /* 0xFFFF; initial crc_seed value */
        unsigned int index = 0;

        index = 0;
        while (len--)
        {
                crc = (crc << 8) ^ crctable[( (crc >> 8) ^ c_ptr[index] )];
                index++;
        }
        return (crc);
}
```

■ make_crc16_ccitt()

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
/* CRC16 implementation acording to CCITT standards */

const unsigned short crc16_tab_ccitt[256]= {
                0x0000,0x1021,0x2042,0x3063,0x4084,0x50a5,0x60c6,0x70e7,0x8108,0x9129,0xa14a,0xb16b,0xc18c,0xd1ad,0xe1ce,0xf1ef,
                0x1231,0x0210,0x3273,0x2252,0x52b5,0x4294,0x72f7,0x62d6,0x9339,0x8318,0xb37b,0xa35a,0xd3bd,0xc39c,0xf3ff,0xe3de,
                0x2462,0x3443,0x0420,0x1401,0x64e6,0x74c7,0x44a4,0x5485,0xa56a,0xb54b,0x8528,0x9509,0xe5ee,0xf5cf,0xc5ac,0xd58d,
                0x3653,0x2672,0x1611,0x0630,0x76d7,0x66f6,0x5695,0x46b4,0xb75b,0xa77a,0x9719,0x8738,0xf7df,0xe7fe,0xd79d,0xc7bc,
                0x48c4,0x58e5,0x6886,0x78a7,0x0840,0x1861,0x2802,0x3823,0xc9cc,0xd9ed,0xe98e,0xf9af,0x8948,0x9969,0xa90a,0xb92b,
                0x5af5,0x4ad4,0x7ab7,0x6a96,0x1a71,0x0a50,0x3a33,0x2a12,0xdbfd,0xcbdc,0xfbbf,0xeb9e,0x9b79,0x8b58,0xbb3b,0xab1a,
                0x6ca6,0x7c87,0x4ce4,0x5cc5,0x2c22,0x3c03,0x0c60,0x1c41,0xedae,0xfd8f,0xcdec,0xddcd,0xad2a,0xbd0b,0x8d68,0x9d49,
                0x7e97,0x6eb6,0x5ed5,0x4ef4,0x3e13,0x2e32,0x1e51,0x0e70,0xff9f,0xefbe,0xdfdd,0xcffc,0xbf1b,0xaf3a,0x9f59,0x8f78,
                0x9188,0x81a9,0xb1ca,0xa1eb,0xd10c,0xc12d,0xf14e,0xe16f,0x1080,0x00a1,0x30c2,0x20e3,0x5004,0x4025,0x7046,0x6067,
                0x83b9,0x9398,0xa3fb,0xb3da,0xc33d,0xd31c,0xe37f,0xf35e,0x02b1,0x1290,0x22f3,0x32d2,0x4235,0x5214,0x6277,0x7256,
                0xb5ea,0xa5cb,0x95a8,0x8589,0xf56e,0xe54f,0xd52c,0xc50d,0x34e2,0x24c3,0x14a0,0x0481,0x7466,0x6447,0x5424,0x4405,
                0xa7db,0xb7fa,0x8799,0x97b8,0xe75f,0xf77e,0xc71d,0xd73c,0x26d3,0x36f2,0x0691,0x16b0,0x6657,0x7676,0x4615,0x5634,
                0xd94c,0xc96d,0xf90e,0xe92f,0x99c8,0x89e9,0xb98a,0xa9ab,0x5844,0x4865,0x7806,0x6827,0x18c0,0x08e1,0x3882,0x28a3,
                0xcb7d,0xdb5c,0xeb3f,0xfb1e,0x8bf9,0x9bd8,0xabbb,0xbb9a,0x4a75,0x5a54,0x6a37,0x7a16,0x0af1,0x1ad0,0x2ab3,0x3a92,
                0xfd2e,0xed0f,0xdd6c,0xcd4d,0xbdaa,0xad8b,0x9de8,0x8dc9,0x7c26,0x6c07,0x5c64,0x4c45,0x3ca2,0x2c83,0x1ce0,0x0cc1,
                0xef1f,0xff3e,0xcf5d,0xdf7c,0xaf9b,0xbfba,0x8fd9,0x9ff8,0x6e17,0x7e36,0x4e55,0x5e74,0x2e93,0x3eb2,0x0ed1,0x1ef0
};

unsigned short make_crc16_ccitt(unsigned short crc, const void *buf, int len) // 0x0; initial crc value
{
                int counter;

                for( counter = 0; counter < len; counter++)
                {
                                crc = (crc<<8) ^ crc16_tab_ccitt[((crc>>8) ^ *(char *)buf++)&0x00FF];
                }

                return crc;
}
```

■ make_crc16_KSC()

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
unsigned int KSCcrc16Tbl[256] = {
        0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
        0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1,0xCE81, 0x0E40,
        0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
        0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
        0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
        0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
        0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
        0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
        0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
        0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
        0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
        0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
        0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
        0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
        0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
        0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
        0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
        0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
        0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
        0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
        0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
        0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
        0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
        0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
        0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
        0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
        0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
        0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
        0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
        0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
        0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
        0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};

//U16 SCRC = 0xFFFF;
uint16_t make_ksc_crc16(uint16_t crc_seed, unsigned char *c_ptr, unsigned int len)
{
        uint16_t crc = crc_seed; // initial crc_seed value "0xFFFF
        unsigned int index = 0;

        index = 0;
        while (len--)
        {

                crc = ( (crc>>8) ^ KSCcrc16Tbl[ (crc^c_ptr[index]) & 0xFF ] );
                index++;
        }
        return (crc);
}
```

■ make_crc32() – 방법1

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
typedef unsigned char uint8_t;  /* 1-byte  (8-bits)  */
typedef unsigned int uint32_t;

const uint32_t crc32_tab[] = {
                0x00000000, 0x77073096, 0xee0e612c, 0x990951ba, 0x076dc419, 0x706af48f,  0xe963a535, 0x9e6495a3, 0x0edb8832, 0x79dcb8a4, 0xe0d5e91e, 0x97d2d988,
                0x09b64c2b, 0x7eb17cbd, 0xe7b82d07, 0x90bf1d91, 0x1db71064, 0x6ab020f2,  0xf3b97148, 0x84be41de, 0x1adad47d, 0x6ddde4eb, 0xf4d4b551, 0x83d385c7,
                0x136c9856, 0x646ba8c0, 0xfd62f97a, 0x8a65c9ec, 0x14015c4f, 0x63066cd9,  0xfa0f3d63, 0x8d080df5, 0x3b6e20c8, 0x4c69105e, 0xd56041e4, 0xa2677172,
                0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b, 0x35b5a8fa, 0x42b2986c,  0xdbbbc9d6, 0xacbcf940, 0x32d86ce3, 0x45df5c75, 0xdcd60dcf, 0xabd13d59,
                0x26d930ac, 0x51de003a, 0xc8d75180, 0xbfd06116, 0x21b4f4b5, 0x56b3c423,  0xcfba9599, 0xb8bda50f, 0x2802b89e, 0x5f058808, 0xc60cd9b2, 0xb10be924,
                0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d, 0x76dc4190, 0x01db7106,  0x98d220bc, 0xefd5102a, 0x71b18589, 0x06b6b51f, 0x9fbfe4a5, 0xe8b8d433,
                0x7807c9a2, 0x0f00f934, 0x9609a88e, 0xe10e9818, 0x7f6a0dbb, 0x086d3d2d,  0x91646c97, 0xe6635c01, 0x6b6b51f4, 0x1c6c6162, 0x856530d8, 0xf262004e,
                0x6c0695ed, 0x1b01a57b, 0x8208f4c1, 0xf50fc457, 0x65b0d9c6, 0x12b7e950,  0x8bbeb8ea, 0xfcb9887c, 0x62dd1ddf, 0x15da2d49, 0x8cd37cf3, 0xfbd44c65,
                0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2, 0x4adfa541, 0x3dd895d7,  0xa4d1c46d, 0xd3d6f4fb, 0x4369e96a, 0x346ed9fc, 0xad678846, 0xda60b8d0,
                0x44042d73, 0x33031de5, 0xaa0a4c5f, 0xdd0d7cc9, 0x5005713c, 0x270241aa,  0xbe0b1010, 0xc90c2086, 0x5768b525, 0x206f85b3, 0xb966d409, 0xce61e49f,
                0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4, 0x59b33d17, 0x2eb40d81,  0xb7bd5c3b, 0xc0ba6cad, 0xedb88320, 0x9abfb3b6, 0x03b6e20c, 0x74b1d29a,
                0xead54739, 0x9dd277af, 0x04db2615, 0x73dc1683, 0xe3630b12, 0x94643b84,  0x0d6d6a3e, 0x7a6a5aa8, 0xe40ecf0b, 0x9309ff9d, 0x0a00ae27, 0x7d079eb1,
                0xf00f9344, 0x8708a3d2, 0x1e01f268, 0x6906c2fe, 0xf762575d, 0x806567cb,  0x196c3671, 0x6e6b06e7, 0xfed41b76, 0x89d32be0, 0x10da7a5a, 0x67dd4acc,
                0xf9b9df6f, 0x8ebeeff9, 0x17b7be43, 0x60b08ed5, 0xd6d6a3e8, 0xa1d1937e,  0x38d8c2c4, 0x4fdff252, 0xd1bb67f1, 0xa6bc5767, 0x3fb506dd, 0x48b2364b,
                0xd80d2bda, 0xaf0a1b4c, 0x36034af6, 0x41047a60, 0xdf60efc3, 0xa867df55,  0x316e8eef, 0x4669be79, 0xcb61b38c, 0xbc66831a, 0x256fd2a0, 0x5268e236,
                0xcc0c7795, 0xbb0b4703, 0x220216b9, 0x5505262f, 0xc5ba3bbe, 0xb2bd0b28,  0x2bb45a92, 0x5cb36a04, 0xc2d7ffa7, 0xb5d0cf31, 0x2cd99e8b, 0x5bdeae1d,
                0x9b64c2b0, 0xec63f226, 0x756aa39c, 0x026d930a, 0x9c0906a9, 0xeb0e363f,  0x72076785, 0x05005713, 0x95bf4a82, 0xe2b87a14, 0x7bb12bae, 0x0cb61b38,
                0x92d28e9b, 0xe5d5be0d, 0x7cdcefb7, 0x0bdbdf21, 0x86d3d2d4, 0xf1d4e242,  0x68ddb3f8, 0x1fda836e, 0x81be16cd, 0xf6b9265b, 0x6fb077e1, 0x18b74777,
                0x88085ae6, 0xff0f6a70, 0x66063bca, 0x11010b5c, 0x8f659eff, 0xf862ae69,  0x616bffd3, 0x166ccf45, 0xa00ae278, 0xd70dd2ee, 0x4e048354, 0x3903b3c2,
                0xa7672661, 0xd06016f7, 0x4969474d, 0x3e6e77db, 0xaed16a4a, 0xd9d65adc,  0x40df0b66, 0x37d83bf0, 0xa9bcae53, 0xdebb9ec5, 0x47b2cf7f, 0x30b5ffe9,
                0xbdbdf21c, 0xcabac28a, 0x53b39330, 0x24b4a3a6, 0xbad03605, 0xcdd70693,  0x54de5729, 0x23d967bf, 0xb3667a2e, 0xc4614ab8, 0x5d681b02, 0x2a6f2b94,
                0xb40bbe37, 0xc30c8ea1, 0x5a05df1b, 0x2d02ef8d
};

uint32_t make_crc32(uint32_t crc, const void *buf, size_t size)
{
                const uint8_t *p;
                p = buf;
                crc = crc ^ ~0U;
                while (size--)
                                crc = crc32_tab[(crc ^ *p++) & 0xFF] ^ (crc >> 8);

                return crc ^ ~0U;
}
```

■ make_crc32() – 방법2

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
#define CRC32_POLYNOMIAL                                (0xEDB88320)
#define CRC32_TAB_SIZE                                  256

void makeCRCtable(unsigned long *table, unsigned long id)
{
        unsigned long i, j, k;

        for(i = 0; i < CRC32_TAB_SIZE; ++i)
        {
                k = i;
                for(j = 0; j < 8; ++j)
                {
                        if (k & 1) k = (k >> 1) ^ id;
                        else k >>= 1;
                }
                table[i] = k;
        }
}

unsigned long calcCRC32(const unsigned char *mem, signed long size, unsigned long CRC)
{
        unsigned long table[CRC32_TAB_SIZE];

        CRC = ~CRC;
        makeCRCtable(table, CRC32_POLYNOMIAL);

        while(size--)
                CRC = table[(CRC ^ *(mem++)) & 0xFF] ^ (CRC >> 8);

        return ~CRC;
}
```

■ make_crc64() – POLY : 0xad93d23594c935a9

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
typedef unsigned long long uint64_t;                    /* 8-bytes (64-bits) */
static const uint64_t crc64_tab[256] = {
    0x0000000000000000ULL, 0x7ad870c830358979ULL,    0xf5b0e190606b12f2ULL, 0x8f689158505e9b8bULL,    0xc038e5739841b68fULL, 0xbae095bba8743ff6ULL,    0x358804e3f82aa47dULL, 0x4f50742bc81f2d04ULL,
    0xab28ecb46814fe75ULL, 0xd1f09c7c5821770cULL,    0x5e980d24087fec87ULL, 0x24407dec384a65feULL,    0x6b1009c7f05548faULL, 0x11c8790fc060c183ULL,    0x9ea0e857903e5a08ULL, 0xe478989fa00bd371ULL,
    0x7d08ff3b88be6f81ULL, 0x07d08ff3b88be6f8ULL,    0x88b81eabe8d57d73ULL, 0xf2606e63d8e0f40aULL,    0xbd301a4810ffd90eULL, 0xc7e86a8020ca5077ULL,    0x4880fbd87094cbfcULL, 0x32588b1040a14285ULL,
    0xd620138fe0aa91f4ULL, 0xacf86347d09f188dULL,    0x2390f21f80c18306ULL, 0x594882d7b0f40a7fULL,    0x1618f6fc78eb277bULL, 0x6cc0863448deae02ULL,    0xe3a8176c18803589ULL, 0x997067a428b5bcf0ULL,
    0xfa11fe77117cdf02ULL, 0x80c98ebf2149567bULL,    0x0fa11fe77117cdf0ULL, 0x75796f2f41224489ULL,    0x3a291b04893d698dULL, 0x40f16bccb908e0f4ULL,    0xcf99fa94e9567b7fULL, 0xb5418a5cd963f206ULL,
    0x513912c379682177ULL, 0x2be1620b495da80eULL,    0xa489f35319033385ULL, 0xde51839b2936bafcULL,    0x9101f7b0e12997f8ULL, 0xebd98778d11c1e81ULL,    0x64b116208142850aULL, 0x1e6966e8b1770c73ULL,
    0x8719014c99c2b083ULL, 0xfdc17184a9f739faULL,    0x72a9e0dcf9a9a271ULL, 0x08719014c99c2b08ULL,    0x4721e43f0183060cULL, 0x3df994f731b68f75ULL,    0xb29105af61e814feULL, 0xc849756751dd9d87ULL,
    0x2c31edf8f1d64ef6ULL, 0x56e99d30c1e3c78fULL,    0xd9810c6891bd5c04ULL, 0xa3597ca0a188d57dULL,    0xec09088b6997f879ULL, 0x96d1784359a27100ULL,    0x19b9e91b09fcea8bULL, 0x636199d339c963f2ULL,
    0xdf7adabd7a6e2d6fULL, 0xa5a2aa754a5ba416ULL,    0x2aca3b2d1a053f9dULL, 0x50124be52a30b6e4ULL,    0x1f423fcee22f9be0ULL, 0x659a4f06d21a1299ULL,    0xeaf2de5e82448912ULL, 0x902aae96b271006bULL,
    0x74523609127ad31aULL, 0x0e8a46c1224f5a63ULL,    0x81e2d7997211c1e8ULL, 0xfb3aa75142244891ULL,    0xb46ad37a8a3b6595ULL, 0xceb2a3b2ba0eececULL,    0x41da32eaea507767ULL, 0x3b024222da65fe1eULL,
    0xa2722586f2d042eeULL, 0xd8aa554ec2e5cb97ULL,    0x57c2c41692bb501cULL, 0x2d1ab4dea28ed965ULL,    0x624ac0f56a91f461ULL, 0x1892b03d5aa47d18ULL,    0x97fa21650afae693ULL, 0xed2251ad3acf6feaULL,
    0x095ac9329ac4bc9bULL, 0x7382b9faaaf135e2ULL,    0xfcea28a2faafae69ULL, 0x8632586aca9a2710ULL,    0xc9622c4102850a14ULL, 0xb3ba5c8932b0836dULL,    0x3cd2cdd162ee18e6ULL, 0x460abd1952db919fULL,
    0x256b24ca6b12f26dULL, 0x5fb354025b277b14ULL,    0xd0dbc55a0b79e09fULL, 0xaa03b5923b4c69e6ULL,    0xe553c1b9f35344e2ULL, 0x9f8bb171c366cd9bULL,    0x10e3202993385610ULL, 0x6a3b50e1a30ddf69ULL,
    0x8e43c87e03060c18ULL, 0xf49bb8b633338561ULL,    0x7bf329ee636d1eeaULL, 0x012b592653589793ULL,    0x4e7b2d0d9b47ba97ULL, 0x34a35dc5ab7233eeULL,    0xbbcbcc9dfb2ca865ULL, 0xc113bc55cb19211cULL,
    0x5863dbf1e3ac9decULL, 0x22bbab39d3991495ULL,    0xadd33a6183c78f1eULL, 0xd70b4aa9b3f20667ULL,    0x985b3e827bed2b63ULL, 0xe2834e4a4bd8a21aULL,    0x6debdf121b863991ULL, 0x1733afda2bb3b0e8ULL,
    0xf34b37458bb86399ULL, 0x8993478dbb8deae0ULL,    0x06fbd6d5ebd3716bULL, 0x7c23a61ddbe6f812ULL,    0x3373d23613f9d516ULL, 0x49aba2fe23cc5c6fULL,    0xc6c333a67392c7e4ULL, 0xbc1b436e43a74e9dULL,
    0x95ac9329ac4bc9b5ULL, 0xef74e3e19c7e40ccULL,    0x601c72b9cc20db47ULL, 0x1ac40271fc15523eULL,    0x5594765a340a7f3aULL, 0x2f4c0692043ff643ULL,    0xa02497ca54616dc8ULL, 0xdafce7026454e4b1ULL,
    0x3e847f9dc45f37c0ULL, 0x445c0f55f46abeb9ULL,    0xcb349e0da4342532ULL, 0xb1eceec59401ac4bULL,    0xfebc9aee5c1e814fULL, 0x8464ea266c2b0836ULL,    0x0b0c7b7e3c7593bdULL, 0x71d40bb60c401ac4ULL,
    0xe8a46c1224f5a634ULL, 0x927c1cda14c02f4dULL,    0x1d148d82449eb4c6ULL, 0x67ccfd4a74ab3dbfULL,    0x289c8961bcb410bbULL, 0x5244f9a98c8199c2ULL,    0xdd2c68f1dcdf0249ULL, 0xa7f41839ecea8b30ULL,
    0x438c80a64ce15841ULL, 0x3954f06e7cd4d138ULL,    0xb63c61362c8a4ab3ULL, 0xcce411fe1cbfc3caULL,    0x83b465d5d4a0eeceULL, 0xf96c151de49567b7ULL,    0x76048445b4cbfc3cULL, 0x0cdcf48d84fe7545ULL,
    0x6fbd6d5ebd3716b7ULL, 0x15651d968d029fceULL,    0x9a0d8ccedd5c0445ULL, 0xe0d5fc06ed698d3cULL,    0xaf85882d2576a038ULL, 0xd55df8e515432941ULL,    0x5a3569bd451db2caULL, 0x20ed197575283bb3ULL,
    0xc49581ead523e8c2ULL, 0xbe4df122e51661bbULL,    0x3125607ab548fa30ULL, 0x4bfd10b2857d7349ULL,    0x04ad64994d625e4dULL, 0x7e7514517d57d734ULL,    0xf11d85092d094cbfULL, 0x8bc5f5c11d3cc5c6ULL,
    0x12b5926535897936ULL, 0x686de2ad05bcf04fULL,    0xe70573f555e26bc4ULL, 0x9ddd033d65d7e2bdULL,    0xd28d7716adc8cfb9ULL, 0xa85507de9dfd46c0ULL,    0x273d9686cda3dd4bULL, 0x5de5e64efd965432ULL,
    0xb99d7ed15d9d8743ULL, 0xc3450e196da80e3aULL,    0x4c2d9f413df695b1ULL, 0x36f5ef890dc31cc8ULL,    0x79a59ba2c5dc31ccULL, 0x037deb6af5e9b8b5ULL,    0x8c157a32a5b7233eULL, 0xf6cd0afa9582aa47ULL,
    0x4ad64994d625e4daULL, 0x300e395ce6106da3ULL,    0xbf66a804b64ef628ULL, 0xc5bed8cc867b7f51ULL,    0x8aeeace74e645255ULL, 0xf036dc2f7e51db2cULL,    0x7f5e4d772e0f40a7ULL, 0x05863dbf1e3ac9deULL,
    0xe1fea520be311aafULL, 0x9b26d5e88e0493d6ULL,    0x144e44b0de5a085dULL, 0x6e963478ee6f8124ULL,    0x21c640532670ac20ULL, 0x5b1e309b16452559ULL,    0xd476a1c3461bbed2ULL, 0xaeaed10b762e37abULL,
    0x37deb6af5e9b8b5bULL, 0x4d06c6676eae0222ULL,    0xc26e573f3ef099a9ULL, 0xb8b627f70ec510d0ULL,    0xf7e653dcc6da3dd4ULL, 0x8d3e2314f6efb4adULL,    0x0256b24ca6b12f26ULL, 0x788ec2849684a65fULL,
    0x9cf65a1b368f752eULL, 0xe62e2ad306bafc57ULL,    0x6946bb8b56e467dcULL, 0x139ecb4366d1eea5ULL,    0x5ccebf68aecec3a1ULL, 0x2616cfa09efb4ad8ULL,    0xa97e5ef8cea5d153ULL, 0xd3a62e30fe90582aULL,
    0xb0c7b7e3c7593bd8ULL, 0xca1fc72bf76cb2a1ULL,    0x45775673a732292aULL, 0x3faf26bb9707a053ULL,    0x70ff52905f188d57ULL, 0x0a2722586f2d042eULL,    0x854fb3003f739fa5ULL, 0xff97c3c80f4616dcULL,
    0x1bef5b57af4dc5adULL, 0x61372b9f9f784cd4ULL,    0xee5fbac7cf26d75fULL, 0x9487ca0fff135e26ULL,    0xdbd7be24370c7322ULL, 0xa10fceec0739fa5bULL,    0x2e675fb4576761d0ULL, 0x54bf2f7c6752e8a9ULL,
    0xcdcf48d84fe75459ULL, 0xb71738107fd2dd20ULL,    0x387fa9482f8c46abULL, 0x42a7d9801fb9cfd2ULL,    0x0df7adabd7a6e2d6ULL, 0x772fdd63e7936bafULL,    0xf8474c3bb7cdf024ULL, 0x829f3cf387f8795dULL,
    0x66e7a46c27f3aa2cULL, 0x1c3fd4a417c62355ULL,    0x935745fc4798b8deULL, 0xe98f353477ad31a7ULL,    0xa6df411fbfb21ca3ULL, 0xdc0731d78f8795daULL,    0x536fa08fdfd90e51ULL, 0x29b7d047efec8728ULL,
};

uint64_t make_crc64(uint64_t crc, const unsigned char *s, uint64_t l)
{
        uint64_t j;
        for (j = 0; j < l; j++)  {
                uint8_t byte = s[j];
                crc = crc64_tab[(uint8_t)crc ^ byte] ^ (crc >> 8);
        }
        return crc;
}
```

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
typedef unsigned long long uint64_t;                    /* 8-bytes (64-bits) */
static const uint64_t crc64_tab_isc[256] = {
        0x0000000000000000ULL, 0x42F0E1EBA9EA3693ULL, 0x85E1C3D753D46D26ULL,    0xC711223CFA3E5BB5ULL, 0x493366450E42ECDFULL, 0x0BC387AEA7A8DA4CULL,
        0xCCD2A5925D9681F9ULL, 0x8E224479F47CB76AULL, 0x9266CC8A1C85D9BEULL,    0xD0962D61B56FEF2DULL, 0x17870F5D4F51B498ULL, 0x5577EEB6E6BB820BULL,
        0xDB55AACF12C73561ULL, 0x99A54B24BB2D03F2ULL, 0x5EB4691841135847ULL,    0x1C4488F3E8F96ED4ULL, 0x663D78FF90E185EFULL, 0x24CD9914390BB37CULL,
        0xE3DCBB28C335E8C9ULL, 0xA12C5AC36ADFDE5AULL, 0x2F0E1EBA9EA36930ULL,    0x6DFEFF5137495FA3ULL, 0xAAEFDD6DCD770416ULL, 0xE81F3C86649D3285ULL,
        0xF45BB4758C645C51ULL, 0xB6AB559E258E6AC2ULL, 0x71BA77A2DFB03177ULL,    0x334A9649765A07E4ULL, 0xBD68D2308226B08EULL, 0xFF9833DB2BCC861DULL,
        0x388911E7D1F2DDA8ULL, 0x7A79F00C7818EB3BULL, 0xCC7AF1FF21C30BDEULL,    0x8E8A101488293D4DULL, 0x499B3228721766F8ULL, 0x0B6BD3C3DBFD506BULL,
        0x854997BA2F81E701ULL, 0xC7B97651866BD192ULL, 0x00A8546D7C558A27ULL,    0x4258B586D5BFBCB4ULL, 0x5E1C3D753D46D260ULL, 0x1CECDC9E94ACE4F3ULL,
        0xDBFDFEA26E92BF46ULL, 0x990D1F49C77889D5ULL, 0x172F5B3033043EBFULL,    0x55DFBADB9AEE082CULL, 0x92CE98E760D05399ULL, 0xD03E790CC93A650AULL,
        0xAA478900B1228E31ULL, 0xE8B768EB18C8B8A2ULL, 0x2FA64AD7E2F6E317ULL,    0x6D56AB3C4B1CD584ULL, 0xE374EF45BF6062EEULL, 0xA1840EAE168A547DULL,
        0x66952C92ECB40FC8ULL, 0x2465CD79455E395BULL, 0x3821458AADA7578FULL,    0x7AD1A461044D611CULL, 0xBDC0865DFE733AA9ULL, 0xFF3067B657990C3AULL,
        0x711223CFA3E5BB50ULL, 0x33E2C2240A0F8DC3ULL, 0xF4F3E018F031D676ULL,    0xB60301F359DBE0E5ULL, 0xDA050215EA6C212FULL, 0x98F5E3FE438617BCULL,
        0x5FE4C1C2B9B84C09ULL, 0x1D14202910527A9AULL, 0x93366450E42ECDF0ULL,    0xD1C685BB4DC4FB63ULL, 0x16D7A787B7FAA0D6ULL, 0x5427466C1E109645ULL,
        0x4863CE9FF6E9F891ULL, 0x0A932F745F03CE02ULL, 0xCD820D48A53D95B7ULL,    0x8F72ECA30CD7A324ULL, 0x0150A8DAF8AB144EULL, 0x43A04931514122DDULL,
        0x84B16B0DAB7F7968ULL, 0xC6418AE602954FFBULL, 0xBC387AEA7A8DA4C0ULL,    0xFEC89B01D3679253ULL, 0x39D9B93D2959C9E6ULL, 0x7B2958D680B3FF75ULL,
        0xF50B1CAF74CF481FULL, 0xB7FBFD44DD257E8CULL, 0x70EADF78271B2539ULL,    0x321A3E938EF113AAULL, 0x2E5EB66066087D7EULL, 0x6CAE578BCFE24BEDULL,
        0xABBF75B735DC1058ULL, 0xE94F945C9C3626CBULL, 0x676DD025684A91A1ULL,    0x259D31CEC1A0A732ULL, 0xE28C13F23B9EFC87ULL, 0xA07CF2199274CA14ULL,
        0x167FF3EACBAF2AF1ULL, 0x548F120162451C62ULL, 0x939E303D987B47D7ULL,    0xD16ED1D631917144ULL, 0x5F4C95AFC5EDC62EULL, 0x1DBC74446C07F0BDULL,
        0xDAAD56789639AB08ULL, 0x985DB7933FD39D9BULL, 0x84193F60D72AF34FULL,    0xC6E9DE8B7EC0C5DCULL, 0x01F8FCB784FE9E69ULL, 0x43081D5C2D14A8FAULL,
        0xCD2A5925D9681F90ULL, 0x8FDAB8CE70822903ULL, 0x48CB9AF28ABC72B6ULL,    0x0A3B7B1923564425ULL, 0x70428B155B4EAF1EULL, 0x32B26AFEF2A4998DULL,
        0xF5A348C2089AC238ULL, 0xB753A929A170F4ABULL, 0x3971ED50550C43C1ULL,    0x7B810CBBFCE67552ULL, 0xBC902E8706D82EE7ULL, 0xFE60CF6CAF321874ULL,
        0xE224479F47CB76A0ULL, 0xA0D4A674EE214033ULL, 0x67C58448141F1B86ULL,    0x253565A3BDF52D15ULL, 0xAB1721DA49899A7FULL, 0xE9E7C031E063ACECULL,
        0x2EF6E20D1A5DF759ULL, 0x6C0603E6B3B7C1CAULL, 0xF6FAE5C07D3274CDULL,    0xB40A042BD4D8425EULL, 0x731B26172EE619EBULL, 0x31EBC7FC870C2F78ULL,
        0xBFC9838573709812ULL, 0xFD39626EDA9AAE81ULL, 0x3A28405220A4F534ULL,    0x78D8A1B9894EC3A7ULL, 0x649C294A61B7AD73ULL, 0x266CC8A1C85D9BE0ULL,
        0xE17DEA9D3263C055ULL, 0xA38D0B769B89F6C6ULL, 0x2DAF4F0F6FF541ACULL,    0x6F5FAEE4C61F773FULL, 0xA84E8CD83C212C8AULL, 0xEABE6D3395CB1A19ULL,
        0x90C79D3FEDD3F122ULL, 0xD2377CD44439C7B1ULL, 0x15265EE8BE079C04ULL,    0x57D6BF0317EDAA97ULL, 0xD9F4FB7AE3911DFDULL, 0x9B041A914A7B2B6EULL,
        0x5C1538ADB04570DBULL, 0x1EE5D94619AF4648ULL, 0x02A151B5F156289CULL,    0x4051B05E58BC1E0FULL, 0x87409262A28245BAULL, 0xC5B073890B687329ULL,
        0x4B9237F0FF14C443ULL, 0x0962D61B56FEF2D0ULL, 0xCE73F427ACC0A965ULL,    0x8C8315CC052A9FF6ULL, 0x3A80143F5CF17F13ULL, 0x7870F5D4F51B4980ULL,
        0xBF61D7E80F251235ULL, 0xFD913603A6CF24A6ULL, 0x73B3727A52B393CCULL,    0x31439391FB59A55FULL, 0xF652B1AD0167FEEAULL, 0xB4A25046A88DC879ULL,
        0xA8E6D8B54074A6ADULL, 0xEA16395EE99E903EULL, 0x2D071B6213A0CB8BULL,    0x6FF7FA89BA4AFD18ULL, 0xE1D5BEF04E364A72ULL, 0xA3255F1BE7DC7CE1ULL,
        0x64347D271DE22754ULL, 0x26C49CCCB40811C7ULL, 0x5CBD6CC0CC10FAFCULL,    0x1E4D8D2B65FACC6FULL, 0xD95CAF179FC497DAULL, 0x9BAC4EFC362EA149ULL,
        0x158E0A85C2521623ULL, 0x577EEB6E6BB820B0ULL, 0x906FC95291867B05ULL,    0xD29F28B9386C4D96ULL, 0xCEDBA04AD0952342ULL, 0x8C2B41A1797F15D1ULL,
        0x4B3A639D83414E64ULL, 0x09CA82762AAB78F7ULL, 0x87E8C60FDED7CF9DULL,    0xC51827E4773DF90EULL, 0x020905D88D03A2BBULL, 0x40F9E43324E99428ULL,
        0x2CFFE7D5975E55E2ULL, 0x6E0F063E3EB46371ULL, 0xA91E2402C48A38C4ULL,    0xEBEEC5E96D600E57ULL, 0x65CC8190991CB93DULL, 0x273C607B30F68FAEULL,
        0xE02D4247CAC8D41BULL, 0xA2DDA3AC6322E288ULL, 0xBE992B5F8BDB8C5CULL,    0xFC69CAB42231BACFULL, 0x3B78E888D80FE17AULL, 0x7988096371E5D7E9ULL,
        0xF7AA4D1A85996083ULL, 0xB55AACF12C735610ULL, 0x724B8ECDD64D0DA5ULL,    0x30BB6F267FA73B36ULL, 0x4AC29F2A07BFD00DULL, 0x08327EC1AE55E69EULL,
        0xCF235CFD546BBD2BULL, 0x8DD3BD16FD818BB8ULL, 0x03F1F96F09FD3CD2ULL,    0x41011884A0170A41ULL, 0x86103AB85A2951F4ULL, 0xC4E0DB53F3C36767ULL,
        0xD8A453A01B3A09B3ULL, 0x9A54B24BB2D03F20ULL, 0x5D45907748EE6495ULL,    0x1FB5719CE1045206ULL, 0x919735E51578E56CULL, 0xD367D40EBC92D3FFULL,
        0x1476F63246AC884AULL, 0x568617D9EF46BED9ULL, 0xE085162AB69D5E3CULL,    0xA275F7C11F7768AFULL, 0x6564D5FDE549331AULL, 0x279434164CA30589ULL,
        0xA9B6706FB8DFB2E3ULL, 0xEB46918411358470ULL, 0x2C57B3B8EB0BDFC5ULL,    0x6EA7525342E1E956ULL, 0x72E3DAA0AA188782ULL, 0x30133B4B03F2B111ULL,
        0xF7021977F9CCEAA4ULL, 0xB5F2F89C5026DC37ULL, 0x3BD0BCE5A45A6B5DULL,    0x79205D0E0DB05DCEULL, 0xBE317F32F78E067BULL, 0xFCC19ED95E6430E8ULL,
        0x86B86ED5267CDBD3ULL, 0xC4488F3E8F96ED40ULL, 0x0359AD0275A8B6F5ULL,    0x41A94CE9DC428066ULL, 0xCF8B0890283E370CULL, 0x8D7BE97B81D4019FULL,
        0x4A6ACB477BEA5A2AULL, 0x089A2AACD2006CB9ULL, 0x14DEA25F3AF9026DULL,    0x562E43B4931334FEULL, 0x913F6188692D6F4BULL, 0xD3CF8063C0C759D8ULL,
        0x5DEDC41A34BBEEB2ULL, 0x1F1D25F19D51D821ULL, 0xD80C07CD676F8394ULL,    0x9AFCE626CE85B507ULL
};
```

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
#define CRC64_POLY_ECMA_182		0xc96c5795d7870f42ULL

uint64_t isc_crc64_init()
{
		uint64_t crc;
		crc = 0xffffffffffffffffULL;

		return crc;
}

uint64_t isc_crc64_update(uint64_t crc, const void *data, size_t len)
{
		const unsigned char *p = data;
		int i;

		if( (NULL==data) )  return;

		while (len-- > 0U)
		{
				i = ((int) (crc >> 56) ^ *p++) & 0xff;
				crc = crc64_tab_isc[i] ^ (crc << 8);
		}
		return crc;
}

uint64_t isc_crc64_final(uint64_t crc)
{
		uint64_t calcCRC;
		calcCRC = crc ^ 0xffffffffffffffffULL;

		return calcCRC;
}
```

■ make_adler32()

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
typedef unsigned char byte;

#define BASE                                    65521L /* largest prime smaller than 65536 */
#define NMAX                                    5552
/* NMAX is the largest n such that 255n(n+1)/2 + (n+1)(BASE-1) <= 2^32-1 */

#define DO1(buf,i)  {s1 += buf[i]; s2 += s1;}
#define DO2(buf,i)  {DO1(buf,i); DO1(buf,i+1);}
#define DO4(buf,i)  {DO2(buf,i); DO2(buf,i+2);}
#define DO8(buf,i)  {DO4(buf,i); DO4(buf,i+4);}
#define DO16(buf)   {DO8(buf,0); DO8(buf,8);}

unsigned int make_adler32(unsigned int adler, const byte *buf, unsigned int len)
{
                unsigned int s1 = adler & 0xffff;
                unsigned int s2 = (adler >> 16) & 0xffff;
                int k;

                if( NULL==buf ) return 1L;

                while (len > 0)
                {
                                k = len < NMAX ? len : NMAX;
                                len -= k;
                                while (k >= 16)
                                {
                                                DO16(buf);
                                                buf += 16;
                                                k -= 16;
                                }
                                if (k != 0) do {
                                                s1 += *buf++;
                                                s2 += s1;
                                } while (--k);
                                s1 %= BASE;
                                s2 %= BASE;
                }
                return (s2 << 16) | s1;
}
```

■ joaat_hash

Email : tp.joo@daum.net
https://cafe.naver.com/topjoo

```
unsigned int joaat_hash(unsigned int joaat_hash, unsigned char *key, size_t len)
{
    unsigned int hash = joaat_hash; // initial value must be 0x00
    size_t i;

    for (i = 0; i < len; i++)
    {
        hash += key[i];
        hash += (hash << 10);
        hash ^= (hash >> 6);
    }
    hash += (hash << 3);
    hash ^= (hash >> 11);
    hash += (hash << 15);
    return hash;
}
```

사용 예)

```
#define BUFSIZE                 1024*2
Int CalaJOAAT(void)
{
.......
                unsigned long long          TotDLen = 0LL;
                unsigned int                calcJoaat = 0;

                while ( LenRead = fread(data_buf, sizeof(unsigned char), BUFSIZE, inpfile) )
                {
                        TotDLen += LenRead;
                        calcJoaat = joaat_hash( calcJoaat, data_buf, LenRead );
                }
........
}
```