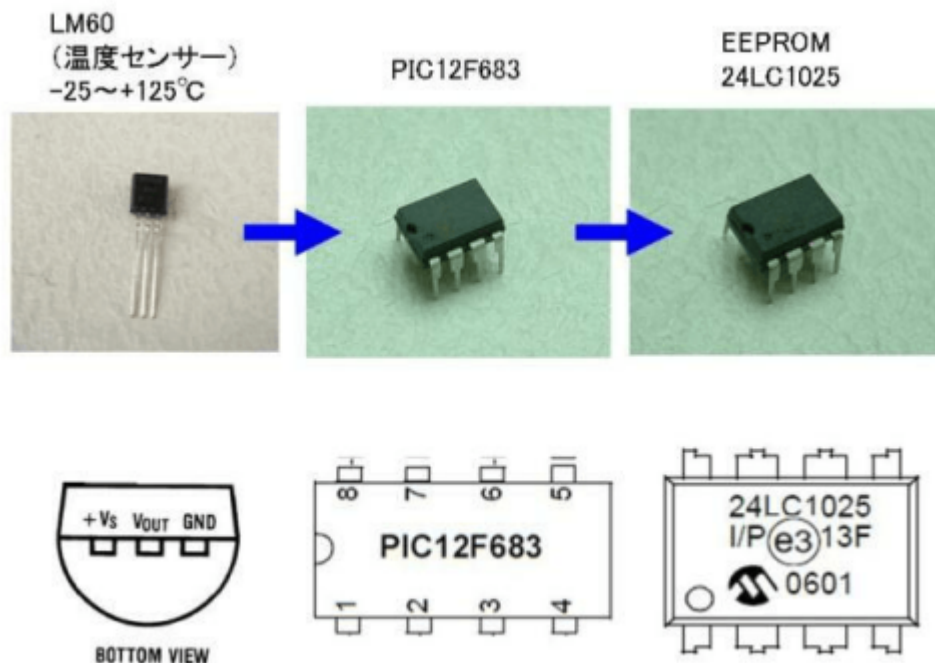


EEPROM/Reader&Writer(SDカード)(PIC18F2550)

概要

最近ではEEPROM(Electrically Erasable and Programmable Read Only Memory)の容量の大きなものが、安価で販売されています。 2008年10月時点で、秋月電子でマイクロチップ社製の1MビットEEPROM 24LC1025が450円で販売

このEEPROMと8ピンのPIC12F683を組み合わせると、かなりコンパクトなデータロガーが出来ます。例えば、次の図のように温度センサーを接続した、温度データロガーを作成することにより、【1秒間隔の記録】約18時間(131072バイト÷2バイト÷60÷60分)【1分間隔の記録】約45日間(131072バイト÷2バイト÷60分÷24時間)【1時間間隔の記録】約7年間(131072バイト÷2バイト÷24時間÷365日)の温度データを記録することが出来ます。



しかしEEPROMにデータを蓄積しただけでは意味がありませんので、この蓄積されたデータを何とかしてパソコンに持って行って、グラフ表示等させて分析する必要があります。

そこで今回は、EEPROMのデータをSDカードにファイルとして保存したり、逆にSDカード上のファイルのデータをEEPROMに書き込むユニットを製作しました。

動作原理

<EEPROM⇔SDC書き込み処理>

1. SDカード上のファイル(eep2sdc.txt)をリライトモードでオープンする。
2. EEPROMから16バイト分のデータを読み出す。
3. 読み出したデータをLCDに表示する。
4. 読み出したデータをファイルに書き込む。

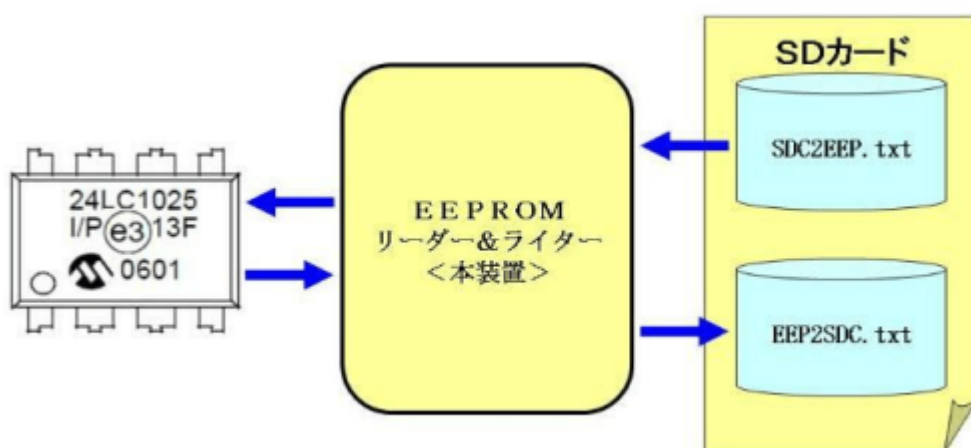
5. EEPROMの最後まで、2.~4.を繰り返す。

<SD⇔EEPROM書き込み処理>

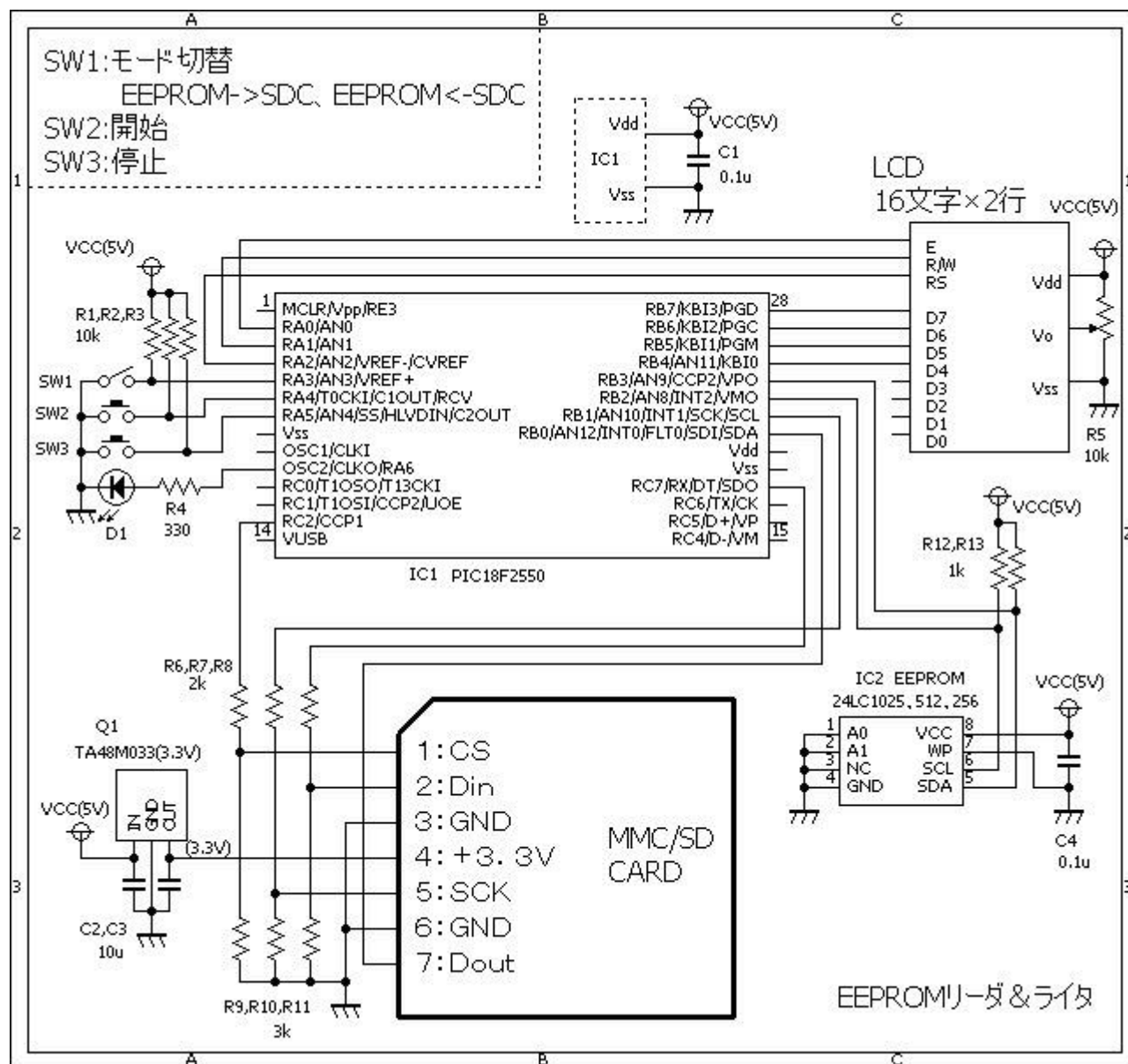
1. SDカード上のファイル(sdc2eep.txt)をリードモードでオープンする。
2. ファイルから16バイト分のデータを読み出す。
3. 読み出したデータをLCDに表示する。
4. 読み出したデータをEEPROMに書き込む。
5. ファイルの最後まで、2.~4.を繰り返す。

<メイン処理>

1. 内臓モジュール⇔LCD⇔MMC等を初期化する。
2. 開始スイッチが押されるのを待つ。
3. モードが、EEPROM⇔SDCであれば⇔<EEPROM⇔SDC書き込み処理>を呼び出す。
4. モードが、SDC⇔EEPROMであれば⇔<SDC⇔EEPROM書き込み処理>を呼び出す。
5. 2.に戻る。



回路図



ソースコード

EepromSdc.c

```
//*****
*
*/
/*
  リーダ&ライター(カード対応)』
    ■ 概要
  のデータをカードにファイルとして保存する。
  eep→"eep2sdc.txt"
  カードにあるファイルをeepへ書き込む。
  sdc→"sdc2eep.txt"
*/
//*****
*
```

```

#define          LED                      PORTA.F6

#define          SW_MODE                  PORTA.F3    //EEPROM↔SDC
#define          SW_START                 PORTA.F4    //開始
#define          SW_STOP                  PORTA.F5    //停止

#define          ON                       1
#define          OFF                      0

#define          CR                       0x0d
#define          LF                       0x0a

#define          ACK                      1
#define          NO_ACK                   0

#define          DATA_SIZE               16

//*****
*

static unsigned   short   dataBuf[DATA_SIZE];

//*****
*

void    EEPROM_24LC1025_Init()
{
    Soft_I2C_Config(&PORTB, 3, 2);
}

//*****
*

void    EEPROM_24LC1025_Page_Read(unsigned long addr, unsigned short
*buf, unsigned short len)
{
    unsigned   short   cnt;
    //
    Soft_I2C_Start();
    if ((addr & 0x10000) == 0)
        Soft_I2C_Write(0xA0);
    else
        Soft_I2C_Write(0xA8);
    Soft_I2C_Write((addr >> 8) & 0xFF);
    Soft_I2C_Write(addr & 0xFF);
    Soft_I2C_Start();
    if ((addr & 0x10000) == 0)
        Soft_I2C_Write(0xA1);
    else
        Soft_I2C_Write(0xA9);
    for (cnt = 0; cnt < (len - 1); cnt++) {

```

```
        buf[cnt] = Soft_I2C_Read(ACK);    //acknowledge
    }
    buf[cnt] = Soft_I2C_Read(NO_ACK);    //not acknowledge
    Soft_I2C_Stop();
}

//*****
*

void    EEPROM_24LC1025_Page_Write(unsigned long addr, unsigned short
*buf, unsigned short len)
{
    unsigned    short    cnt;
    //
    Soft_I2C_Start();
    if ((addr & 0x10000) == 0)
        Soft_I2C_Write(0xA0);
    else
        Soft_I2C_Write(0xA8);
    Soft_I2C_Write((addr >> 8) & 0xFF);
    Soft_I2C_Write(addr & 0xFF);
    for (cnt = 0; cnt < len; cnt++) {
        Soft_I2C_Write(buf[cnt]);
    }
    Soft_I2C_Stop();
    //    Delay_ms(100);
}

//*****
*

void    eeprom2sdc()
{
    unsigned    long    addr;
    unsigned    short    cnt;
    //□□□のファイルのオープン
    Mmc_Fat_Assign("eep2sdc.txt", 0xA0);
    Mmc_Fat_Rewrite();
    //
    addr = 0;
    for (addr = 0; addr < 0x20000; addr += DATA_SIZE) {
        LED = ON;
        EEPROM_24LC1025_Page_Read(addr, dataBuf, DATA_SIZE);
        LED = OFF;
        for (cnt = 0; cnt < DATA_SIZE; cnt++)
            Lcd_Custom_Chrc(2, cnt + 1, dataBuf[cnt]);
        Mmc_Fat_Write(dataBuf, DATA_SIZE);
        //
        LongToStr(addr + DATA_SIZE, dataBuf);
        Lcd_Custom_Out(1, 11, &dataBuf[5]);
        //
    }
}
```

```
        if (SW_STOP == 0)
            return;
    }
}

//*****
*

void    sdc2eprom()
{
    static    unsigned    long    fsize, addr;
    static    unsigned    int     cnt1, cnt2, round;
    static    unsigned    short   character, fraction;
    //□□□のファイルのオープン
    Mmc_Fat_Assign("sdc2EEP.txt", 0);
    Mmc_Fat_Reset(&fsize);
    //    fsize = Mmc_Fat_Get_File_Size();
    round = fsize / DATA_SIZE;
    fraction = fsize % DATA_SIZE;
    //
    addr = 0;
    for (cnt1 = 0; cnt1 < round; cnt1++) {
        LED = ON;
        for (cnt2 = 0; cnt2 < DATA_SIZE; cnt2++) {
            Mmc_Fat_Read(&character);
            dataBuf[cnt2] = character;
            Lcd_Custom_Chr(2, cnt2 + 1, character);
        }
        LED = OFF;
        EEPROM_24LC1025_Page_Write(addr, dataBuf, DATA_SIZE);
        addr += DATA_SIZE;
        //
        LongToStr(addr, dataBuf);
        Lcd_Custom_Out(1, 11, &dataBuf[5]);
        //
        if (SW_STOP == 0)
            return;
    }
    //
    if (fraction > 0) {
        LED = ON;
        for (cnt2 = 0; cnt2 < fraction; cnt2++) {
            Mmc_Fat_Read(&character);
            dataBuf[cnt2] = character;
        }
        LED = OFF;
        EEPROM_24LC1025_Page_Write(addr, dataBuf, fraction);
        //
        LongToStr(addr + fraction, dataBuf);
        Lcd_Custom_Out(2, 1, dataBuf);
    }
}
```

```
}

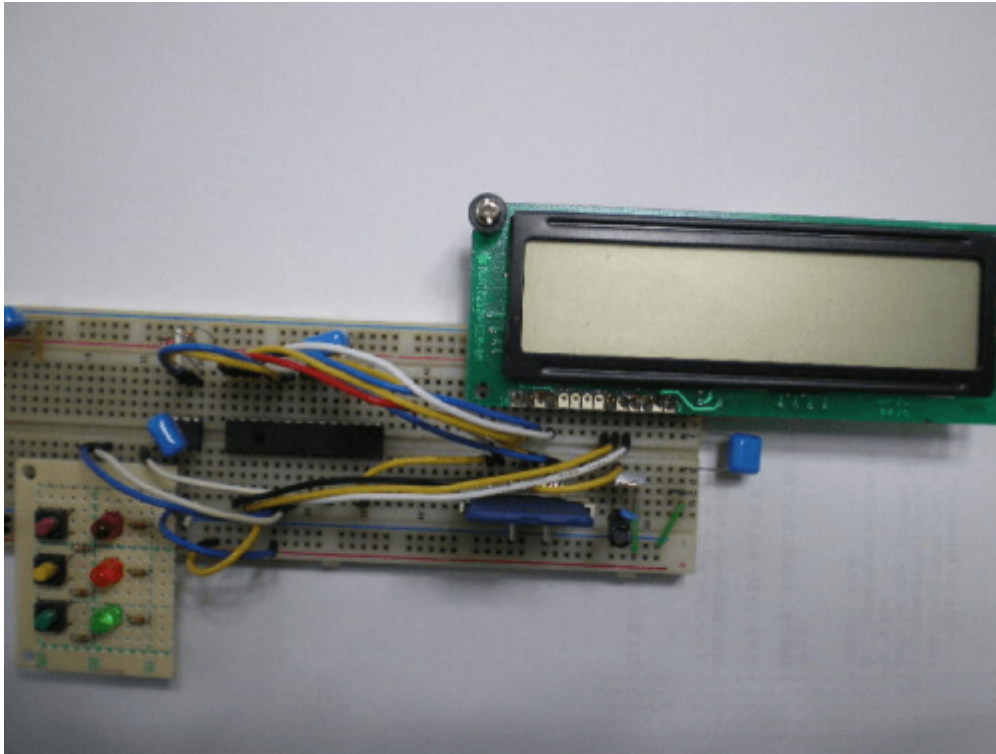
//*****
*

void main()
{
    //変数の定義
    unsigned short cnt;
    //
    OSCCON.IRCF2 = 1;
    OSCCON.IRCF1 = 1;
    OSCCON.IRCF0 = 1;
    OSCCON.SCS1 = 1;
    OSCCON.SCS0 = 1;
    //変換の設定
    ADCON1.PCFG3 = 1;
    ADCON1.PCFG2 = 1;
    ADCON1.PCFG1 = 1;
    ADCON1.PCFG0 = 1;
    //ポートの設定
    TRISA = 0b10111111;
    TRISB = 0b00000000;
    TRISC = 0b00000001;
    //変数の初期化
    //の初期化
    Lcd_Custom_Config(&PORTB,7,6,5,4,&PORTA,2,1,0);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    Lcd_Custom_Cmd(LCD_CLEAR);
    for (cnt = 1; cnt <= 16; cnt++) {
        Lcd_Custom_Chrc(1, cnt, 0xFF);
        LED = ON;
        Delay_ms(50);
        LED = OFF;
        Delay_ms(50);
    }
    for (cnt = 1; cnt <= 16; cnt++) {
        Lcd_Custom_Chrc(2, cnt, 0xFF);
        LED = ON;
        Delay_ms(50);
        LED = OFF;
        Delay_ms(50);
    }
    Lcd_Custom_Cmd(LCD_CLEAR);
    //の初期化
    LED = ON;
    Spi_Init_Advanced(MASTER_OSC_DIV64, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
    if (Mmc_Fat_Init(&PORTC, 2)) {
        Lcd_Custom_Out(1, 1, "MMC error!");
        while (1) {
```

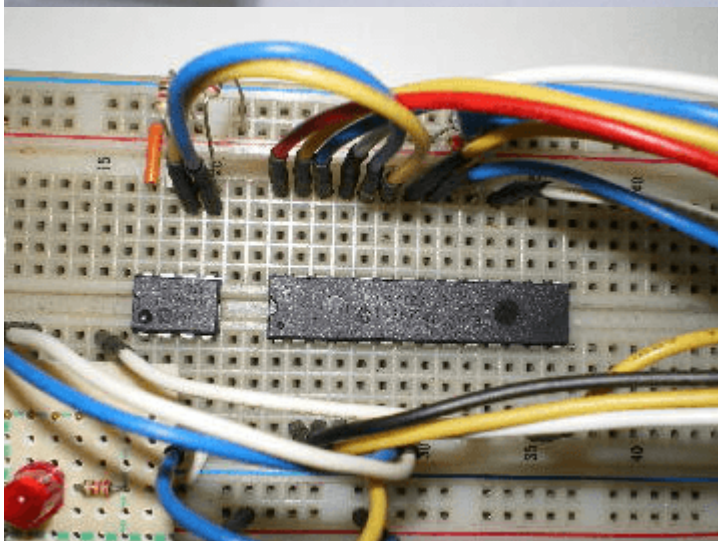
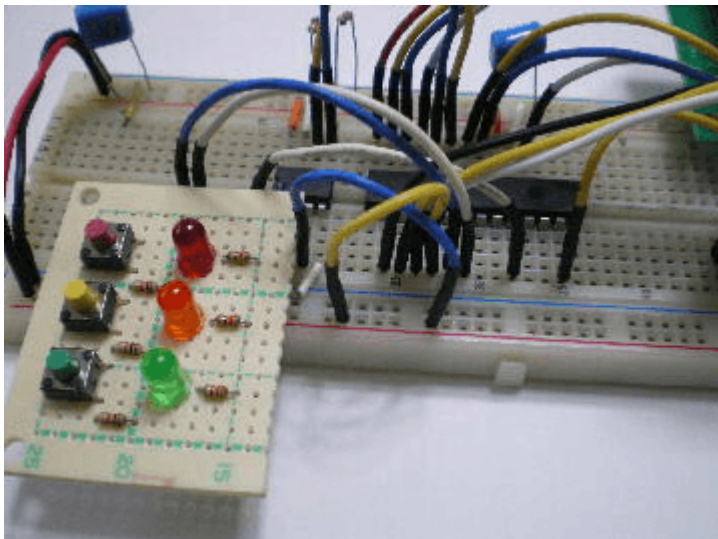
```
        LED = ON;
        Delay_ms(50);
        LED = OFF;
        Delay_ms(50);
    }
}
Spi_Init_Advanced(MASTER_OSC_DIV16, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
LED = OFF;
//
EEPROM_24LC1025_Init();
//
while (1) {
    Lcd_Custom_Out(1, 1, "EEPROM <-> SDC");
    //開始スイッチが押されるのをチェックする。
    while (SW_START == 1) {
        Delay_ms(10);
    }
    Lcd_Custom_Cmd(LCD_CLEAR);
    //
    if (SW_MODE == 1) {
        Lcd_Custom_Out(1, 1, "PROM->SDC");
        eeprom2sdc();
        Delay_ms(1000);
        Lcd_Custom_Cmd(LCD_CLEAR);
    } else {
        Lcd_Custom_Out(1, 1, "PROM<-SDC");
        sdc2eeprom();
        Delay_ms(1000);
        Lcd_Custom_Cmd(LCD_CLEAR);
    }
}
}
} //~!

//*****
*
```

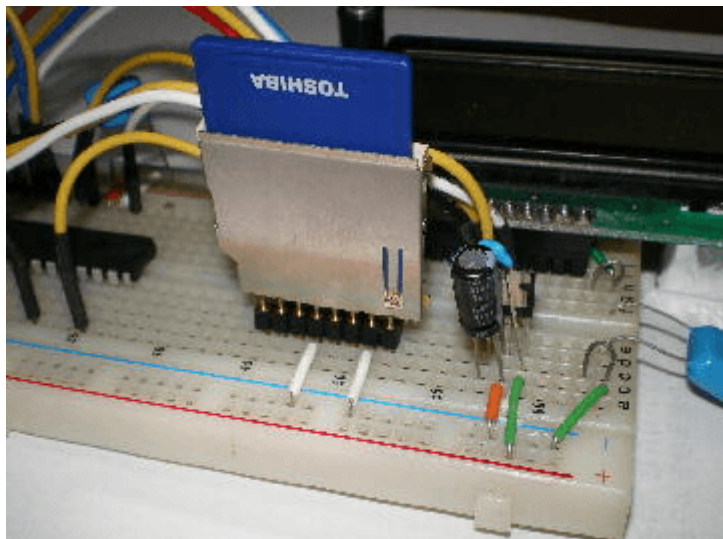
動作確認



左側:スイッチおよびLED部分です。 右側:24LC1025とPIC18F2550部分です。



左側:SDカード部分です。 右側:LCD部分です。起動時の表示画面です。



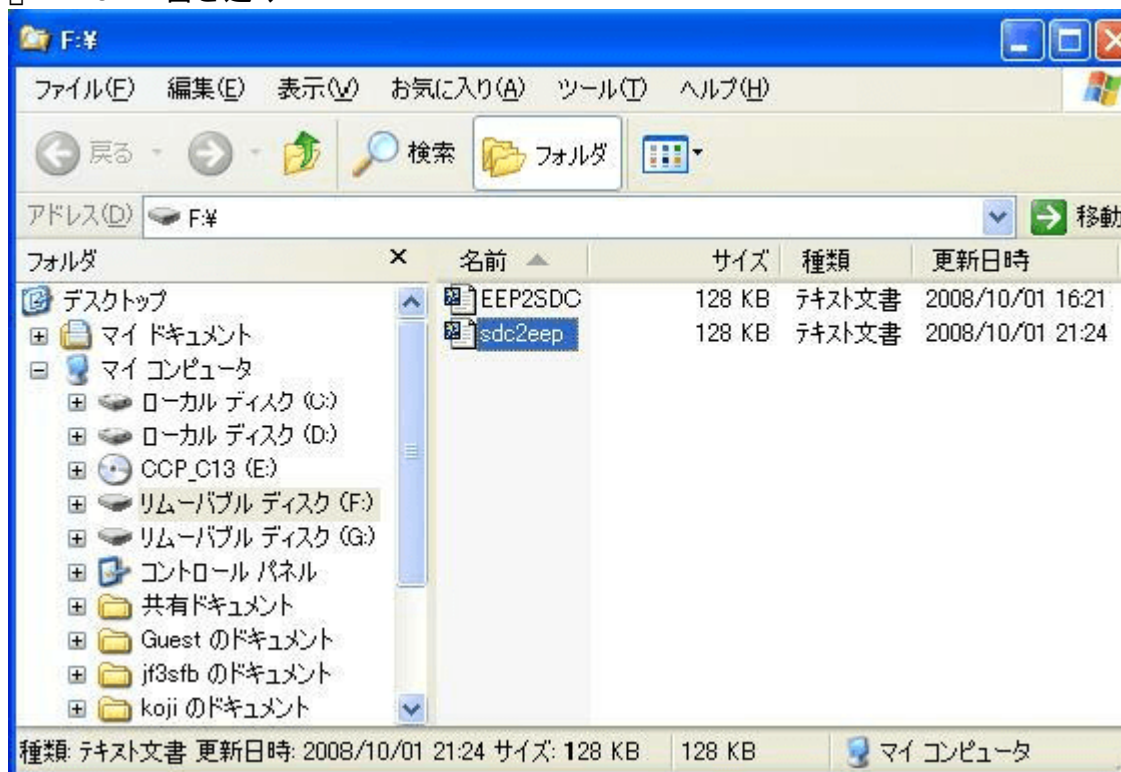
EEPROM⇔SDカードへの書き込み 1行目の右側:書き込んでいるデータのサイズです。2行目:書き込んでいるデータを表示しています。(高速に変化しますので、目で確認することは困難です)



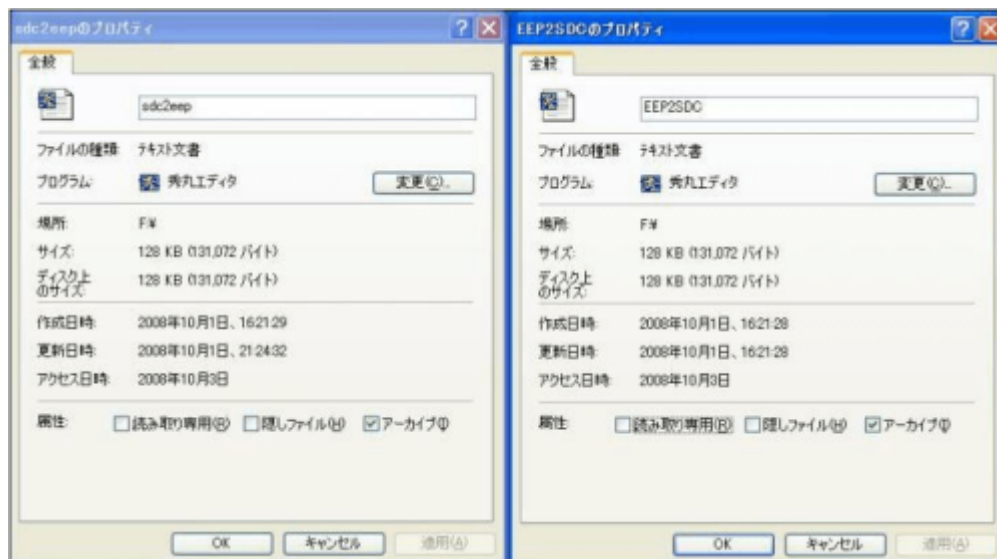
SDカード⇔EEPROMへの書き込み 1行目の右側:書き込んでいるデータのサイズです。 2行目:書き込んでいるデータを表示しています。(高速に変化しますので、目で確認することは困難です)



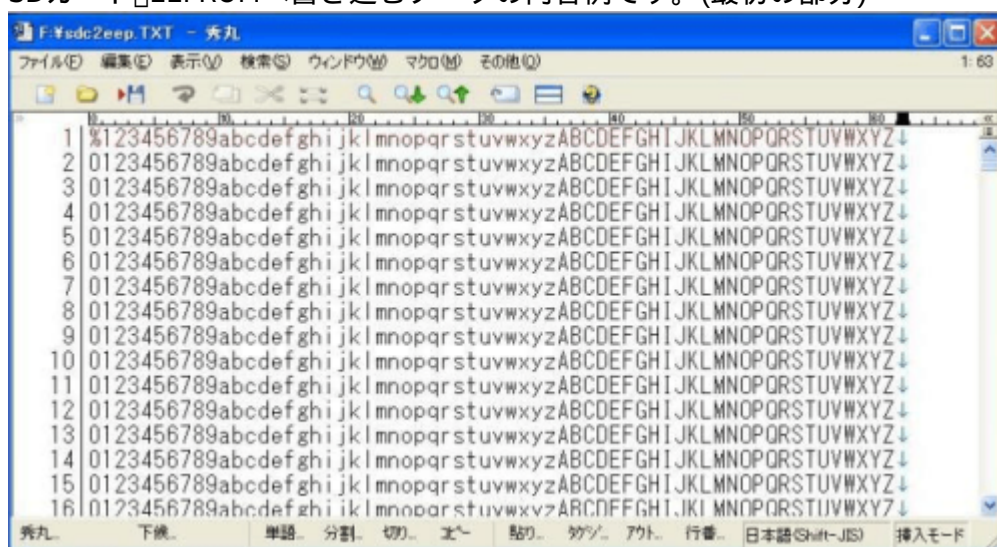
SDカード上のファイルです⇔ EEP2SDC.TXT:EEPROM⇔SDカードへ書き込み SDC2EEP.TXT:SDカード⇔EEPROMへ書き込み



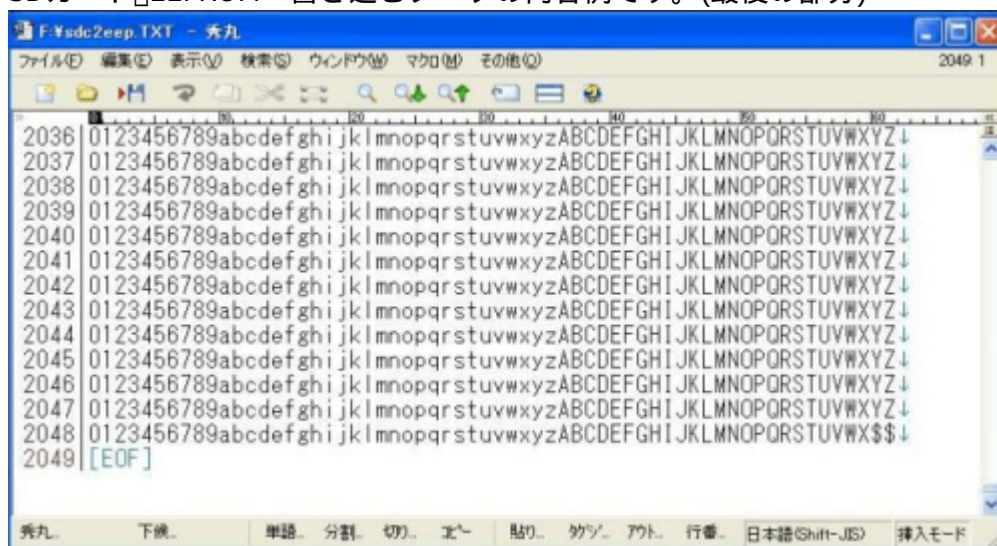
EEP2SDC.TXTとSDC2EEP.TXTのファイルのサイズです。(131072バイト)



SDカードEEPROMへ書き込むデータの内容例です。(最初の部分)



SDカードEEPROMへ書き込むデータの内容例です。(最後の部分)



著作権表示 copyright notice

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。詳細 This page is a

copy of Mr. Inasaki's closed website, and the copyright is held by him.[Details](#)

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:otherpic:176>

Last update: **2020/05/01 19:11**

