

```
#include <windows.h>
#include <stdio.h>
#include "FTD2XX.H"

const unsigned char start_byte = 0xF0;
const unsigned char stop_byte = 0xFF;

DWORD ParsingPos (FT_HANDLE &a_hPort) {
// функция читает из USB-порта и разбирает
// посылку из 24002 байт;
// стартовый байт равен 0xF0,
// стоповый 0xFF
// Аргументы:
// a_hPort - дескриптор USB-порта
// a_NumPos - порядковый номер посылки (0 ...99)

    FT_STATUS    ftStatus;    // статус USB-порта
    unsigned char out_byte = 0; // байт, считанный из USB-порта

    const int    NCounts      = 2400; // количество отсчётов в посылке
    const int    ByteInCount  = 10;   // количество байт в отсчёте
    const int    LenPos       = 24002; // количество байт в посылке
    unsigned char aiPos[LenPos];      // массив посылки со стартовым и стоповым байтами
    int          ibyte = -1;          // индекс байта в посылке; если ibyte = -1 -
    ожидание посылки
    int          NumPos = 0;          // номер считанной посылки

    int          UPos[NCounts];      // массив с отсчётами напряжения
    int          IPos[NCounts];      // массив с отсчётами тока

    char         cFileName[15]; // перем. для составления имени файла
    // дескрипторы файлов
    FILE         *fpit = NULL,
                *fput = NULL;

    // переменные для чтения из USB-порта
    DWORD        numout,
                RxBuf=0,
                TxBuf=0,
                EventStat;

    // инициализация массива посылки
    memset (aiPos, 0x00, LenPos);

    // бесконечный цикл чтения посылок
    while (1) {

        if (ibase == -1) {

            ftStatus = FT_GetStatus(a_hPort, &RxBuf, &TxBuf, &EventStat);

            // если в приёмный буфер не пустой,
            // происходит чтение 1 байта
            if ( (ftStatus == FT_OK ) && ( RxBuf != 0 ) ) {

                // чтение из порта по одному символу
                ftStatus = FT_Read(a_hPort, &out_byte, 1, &numout);

                if ( ( ftStatus == FT_OK ) && ( numout == 1 ) ) {
                    // цикл ожидания стартового байта посылки
                    if ( out_byte == start_byte ) {

                        // первый байт посылки
                        ibyte++;
                        aiPos[ibase] = out_byte;
                    /** отладка */ printf("start_byte = 0x%X\n", out_byte);
                }
                else {
                    continue;
                }
            }
        }
    } // if
}
```

```
else {

    // чтение посылки
    while (out_byte != stop_byte ) {

        ftStatus = FT_GetStatus(a_hPort, &RxBuf, &TxBuf, &EventStat);

        // если приёмный буфер не пустой,
        // происходит чтение 1 байта
        if ( ( ftStatus == FT_OK ) && ( RxBuf != 0 ) ) {

            ftStatus = FT_Read(a_hPort, &out_byte, 1, &numout);
            if ( numout == 1 ) {
                ibyte++;
                aiPos[ ibyte ] = out_byte;
            }
            else {
                // возвращается код ошибки
                ;
                break;
            }
        }
    }

    /** отладка */ printf(" stop_byte = 0x%X\n", out_byte);
    /** отладка */ printf("Package number = %d: %d bytes was readed\n", NumPos+1, ibyte+1);

    // открытие текущих файлов отсчётов тока и напряжения
    /** отладка */ sprintf(cFileName, "./data/u%02d.txt", NumPos);
    /** отладка */ if( (fput = fopen(cFileName, "w" ) ) == NULL ) return 100;
    /** отладка */ sprintf(cFileName, "./data/i%02d.txt", NumPos);
    /** отладка */ if( (fpit = fopen(cFileName, "w" ) ) == NULL ) return 100;

    // посылка считана
    NumPos++;

    // неполная посылка
    if ( ibyte != ( LenPos - 1 ) ) {
        break;
    }

    // состояние ожидания новой посылки
    ibyte = -1;

    // разбор одного отсчёта
    int bmax = ByteInCount / 2; // индекс середины отсчёта

    // индекс начального отсчёта
    int index = 1;
    for (int count = 0; count < NCounts; count++) {

        // отсчёт напряжения или тока, который
        // формируется из четырёх байт посылки
        int data_UI = 0;

        // формирование отсчёта наяржения или тока
        for (int i = 0; i < 2; i++) {

            // указатель на первый байт отсчёта;
            // вспомогательная переменная для разбора отсчёта
            unsigned char *p_arr = &aiPos[ index + bmax * i ];

            // номер старшего байта переменной
            // voltage (тип int)
            int nb = 3;
            // копирование индикаторного байта напряжения
            unsigned char ind_byte = p_arr[0] << 3;

            for (int b = 1; b < bmax; b++) {
                // сравнение с "0" старшего бита
                if ( ( p_arr[ b ] >> 7 ) == 0x0 ) {
                    // составление одного отсчёта напряжения/тока
                    ind_byte = ind_byte << 1;
                    // запись в старший бит значения из индикаторного бита
                }
            }
        }
    }
}
```

```
        // и присвоение соответствующему байту отсчёта напряжения/тока
        ( ( unsigned char *) &data_UI )[ nb ] = ( ind_byte & ( 1 << 7 ) )
    ) | p_arr[ b ];
        nb--;
    }
    }
    if (i==0) UPos[count] = data_UI; // отсчёт напряжения
    else     IPos[count] = data_UI; // отсчёт тока
}
index += ByteInCount;

} // for count

// инициализация массива посылки
memset (aiPos, 0x00, LenPos);

/** отладка */ for (int count = 0; count < NCounts; count++) {
/** отладка */     fprintf(fput, "%d \n", UPos[count]);
/** отладка */     fprintf(fpit, "%d \n", IPos[count]);
/** отладка */ }

/** отладка */ fflush(fpit);
/** отладка */ fflush(fput);

}
} // while (1)

return 0;
}
```