

```

// FelicaLibSample06.cpp : コンソール アプリケーションのエントリ ポイントを定義します。
// ポーリングとカード情報の取得 (Pollingコマンド)

#include "stdafx.h"
#include <stdio>
#include <stdlib>

#include "felica.h"

void print_vector(char* title, unsigned char* vector, int length);

int main(void)
{
    fprintf(stdout, "ポーリングとカード情報の取得 (Pollingコマンド) %n%n");

    /* ライブラリの初期化 */
    if (!initialize_library()) {
        fprintf(stderr, "ライブラリの初期化に失敗しました。 %n%n");
        return EXIT_FAILURE;
    }

    /* リーダ・ライタの自動認識とオープン */
    if (!open_reader_writer_auto()) {
        fprintf(stderr, "リーダ・ライタのオープンに失敗しました。 %n%n");
        return EXIT_FAILURE;
    }

    /* ポーリングとカード情報の取得 (変数と定数設定) */
    // 第1引数とする構造体、ポーリングをするために必要な情報
    structure_polling polling;
    unsigned char system_code[2] = {0x00, 0x00};
    polling.system_code = system_code;
    polling.time_slot = 0x00;

    // 第2引数とする自然数、カードの枚数
    unsigned char number_of_cards = 0;

    // 第3引数とする構造体、ポーリングの時に取得したカードの情報
    structure_card_information card_information;
    unsigned char card_idm[8]; // 製造IDブロックの製造ID (IDm)
    unsigned char card_pmm[8]; // 製造IDブロックの製造パラメータ (PMm)
    card_information.card_idm = card_idm;
    card_information.card_pmm = card_pmm;

    /* ポーリングとカード情報の取得 (Pollingコマンド) */
    if (!polling_and_get_card_information(&polling, &number_of_cards, &card_information)) {
        fprintf(stderr, "Felicaカードが見つかりません。 %n%n");
        return EXIT_FAILURE;
    }

    /* 取得データの表示 */
    fprintf(stdout, "カードの枚数: %d%n", number_of_cards);
}

```

```

print_vector("製造ID (IDm):", card_idm, sizeof(card_idm)); // 製造ID (IDm)
print_vector("製造パラメータ (PMm):", card_pmm, sizeof(card_pmm)); // 製造パラメータ (PMm)

/* リーダ・ライタのクローズ */
if (!close_reader_writer()) {
    fprintf(stderr, "リーダ・ライタのクローズに失敗しました。¥n¥n");
    return EXIT_FAILURE;
}

/* ライブラリの解放 */
if (!dispose_library()) {
    fprintf(stderr, "ライブラリの解放に失敗しました。¥n¥n");
    return EXIT_FAILURE;
}

fprintf(stdout, "¥nプログラムの実行を終了します。¥n¥n");

return EXIT_SUCCESS; // 正常終了を示す戻り値
}

void print_vector(char* title, unsigned char* vector, int length)
{
    if (title != NULL) {
        fprintf(stdout, "%s ", title);
    }

    int i;
    for (i = 0; i < length - 1; i++) {
        fprintf(stdout, "%02x ", vector[i]);
    }
    fprintf(stdout, "%02x", vector[length - 1]);
    fprintf(stdout, "¥n");
}

```