

```
// FeliCaLibSample13.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。  
// コミュニケートスルーコマンド (p.175)
```

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

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

```
int main(void)  
{  
    fprintf(stdout, "コミュニケートスルーコマンド¥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;  
    }  
  
    /* 変数の定義と初期化 */  
    input_structure_dumb input_dumb;  
    unsigned char card_command_packet_data[] = {0x00, 0xff, 0xff, 0x00, 0x0f};  
    input_dumb.time_out = 0x1388; // time out: 500msec  
    input_dumb.retry_count = 0x0003; // retry count: 3  
    input_dumb.card_command_packet_data = card_command_packet_data;  
    input_dumb.card_command_packet_length = 0x06;  
  
    output_structure_dumb output_dumb;  
    unsigned char card_response_packet_data[256];  
    unsigned char card_response_packet_length[1];  
    output_dumb.card_response_packet_data = card_response_packet_data;  
    output_dumb.card_response_packet_length = card_response_packet_length;  
  
    /* パケットデータのダンプ */  
    if (!dumb(&input_dumb, &output_dumb)) {  
        fprintf(stderr, "カードが見つからないのでダンプできません。¥n¥n");  
        error_routine();  
        return EXIT_FAILURE;  
    }  
}
```

```

/* パケットデータの表示 */
print_vector("packet data:", card_response_packet_data, card_response_packet_length[0]-1);
fprintf(stdout, "packet length: %d\n", card_response_packet_length[0]);

/* 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 error_routine(void)
{
    enumeration_felica_error_type felica_error_type;
    enumeration_rw_error_type rw_error_type;
    get_last_error_types(&felica_error_type, &rw_error_type);
    printf("felica_error_type: %d¥n", felica_error_type);
    printf("rw_error_type: %d¥n", rw_error_type);

    close_reader_writer();
    dispose_library();
}

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");
}

```