

```
// FeliCaLibSample10.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。  
// 論理カードのシステムコードのリストを取得
```

```
#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, "論理カードのシステムコードのリストを取得\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, "リーダー・ライタのオープンに失敗しました。 \n\n");  
        return EXIT_FAILURE;  
    }
```

```
    /* 変数と定数設定 */
```

```
    structure_polling polling;  
    unsigned char system_code[2] = {0x00, 0x00};  
    polling.system_code = system_code;  
    polling.time_slot = 0x00;
```

```
    structure_card_information card_information;  
    unsigned char card_idm[8];  
    unsigned char card_pmm[8];  
    card_information.card_idm = card_idm;  
    card_information.card_pmm = card_pmm;
```

```
    input_structure_request_system_code input_request_system_code;  
    input_request_system_code.card_idm = card_idm;
```

```
    output_structure_request_system_code output_request_system_code;  
    unsigned char system_code_list[4];  
    output_request_system_code.system_code_list = system_code_list;
```

```
    /* ポーリングと論理カードのシステムコードのリストを取得 */
```

```
    if (!polling_and_request_system_code(&polling, &input_request_system_code,  
                                         &card_information, &output_request_system_code)) {  
        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, "number of system codes: %d\n",
output_request_system_code.number_of_system_codes);
    print_vector("system code list:", system_code_list,
2*output_request_system_code.number_of_system_codes);

    /* 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");
}

```