

# 홍채 인식 시스템의 응용 프레임워크 구현

진진호\*, 김재훈\*\*, 한경희\*\*\*

\*아주대학교 지식정보보안학과

\*\*아주대학교 정보통신전문대학원

\*\*\*아이락 글로벌

e-mail:aquaxys@ajou.ac.kr

## Application Framework Implementation of Iris Recognition System

Jin-Ho Chun\*, Jai-Hoon Kim\*\*, Kyung-Hee Han\*\*\*

\*Department of Knowledge Information Security, Ajou University

\*\*Graduate School of Information &amp; Communication, Ajou University

\*\*\*I-LOCK Global

### 요 약

홍채 인식 시스템의 응용 프레임워크는 홍채 인식기 모듈을 컴퓨터에 연결할 때 사용되는 USB 호스트 컨트롤러 드라이버와 응용 프로그램 사이에서 둘을 제어하는 스탠다드 라이브러리를 제공한다. 본 논문에서는 스탠다드 라이브러리 설계와 구현을 설명하고 있다. 스탠다드 라이브러리는 표준 호출 규약을 지키므로 응용 프로그램은 다양한 언어로 작성할 수 있다. 또한, 기존의 폴링 방식으로 구현된 스탠다드 라이브러리를 콜백을 이용하여 구현함으로써 네트워크를 통해 많은 사람들이 동시에 홍채 인식기를 사용할 때 리소스 사용을 경감할 수 있는 장점이 있다.

## 1. 서론

인터넷을 이용한 여러 가지 서비스들이 점차 늘어남에 따라, 익명성을 특징으로 가지는 인터넷이라는 공간에서 생기는 신원의 위장이나 도용 등은 해마다 증가하고 있다. 따라서 인터넷에서 생기는 사이버 범죄로부터 자신의 정보와 재산을 보호하고 이를 사전에 예방할 수 있는, 생체 정보를 이용한 생체 인식분야의 필요성은 계속 부각되고 있다. 특히, 홍채 인식은 현재 활성화 되어 있는 지문 인식에 비교하여 위-변조가 불가능하며, 그동안 단점이었던 홍채 인식기의 가격이 점차 낮아지면서, 앞으로의 생체 인식 기술 분야에서 널리 사용될 것이라 예상된다.[1]

본 연구에서는 홍채 인식 기술을 구현하는 홍채 인식 모듈의 통신을 연결해주는 USB 호스트 컨트롤러 드라이버와 응용 프로그램 사이를 제어하는 스탠다드 라이브러리를 설계하고 구현하였다. 구현한 스탠다드 라이브러리는 어떤 언어에서도 연동 및 개발 가능하며, 기존의 폴링 방식이 아닌 콜백 방식으로 구현하여 성능이 개선되었다.

본 논문의 구성은 다음과 같다. 2장에서는 스탠다드 라이브러리에 대한 전반적인 내용을 살펴본다. 3장에서는 스탠다드 라이브러리의 구조를 설계하고, 4장에서는 3장에서 설계한 기능을 구현한다. 마지막으로 5장에서는 결론 및 향후 연구과제에 대해 논한다.

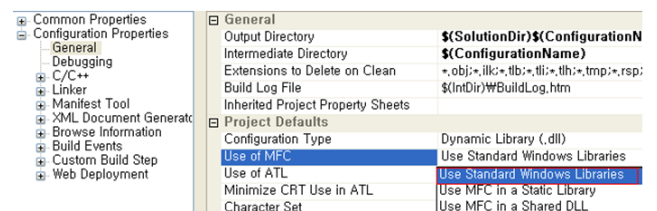
## 2. 관련연구

### 2.1 스탠다드 라이브러리

#### 2.1.1 라이브러리 설정

스탠다드 라이브러리를 구현하기 위해선, 라이브러리를 구현하는 언어와 라이브러리를 사용하는 이종언어간의 사용하는 형식을 통일해 주어야 한다.

c++로 라이브러리를 만들 시에 몇 가지 형태의 라이브러리를 만들 수가 있다. MFC 라이브러리를 사용할 경우 이를 링크할 수 있는데, 라이브러리를 c++로 구현을 하고 응용 프로그램은 비주얼베이직으로 구현한다고 가정할 때, 비주얼베이직은 MFC 라이브러리를 사용하지 않으므로 이를 사용하지 않는 스탠다드 라이브러리로 설정한다.



(그림 1) 스탠다드 라이브러리 설정

#### 2.1.2 WIN32 API 사용

비주얼베이직은 c런타임 라이브러리를 사용하지 않는다. 따라서 자료구조의 복사나 문자열을 복사할 때 memcpy나 strcpy등을 사용하지 않고, win32 API인 rtlmovememory를 이용하여 c런타임 라이브러리가 포함되지 않도록 구현한다.

2.1.3 호출 규약

비주얼 베이직은 win32 API 호출규약인 stdcall을 사용한다. 이에 비해 c++은 사용하는 함수에 따라 cdecl(c 언어 표준 호출 규약)을 사용한다. 따라서 이종언어간의 API를 정의할 때는 stdcall을 명시함으로써, 언어 간의 충돌을 막을 수 있다.

2.2 콜백 함수

콜백 함수는 응용 프로그램 내에 구현된 코드로서[2], 응용 프로그램의 비동기 처리 시나리오에서 사용자는 특정 명령의 결과가 나올 때까지, 다른 프로세스를 계속해서 실행하고자 한다. 따라서 응용 프로그램은 작업 처리 시간이 소모되는 명령을 배경 스레드에 위임하여 사용자가 결과를 받을 때까지 기다릴 수 있다. 이런 종료의 시나리오 은 콜백 함수를 이용하는 것이 가장 효과적이다.[3]

2.3 매개 변수

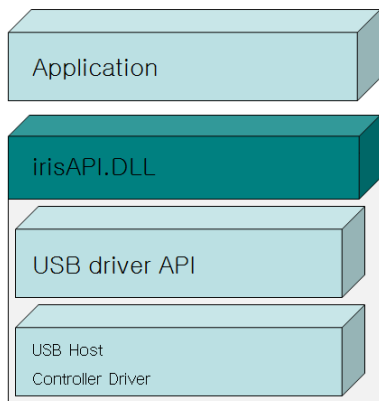
본 논문에서 구현된 표준 라이브러리는 기존의 폴링 방식이 아닌 콜백 방식으로 구현하였다. c++로 응용 프로그램과 라이브러리를 구현할 때에는, 응용 프로그램에서 API를 호출할 때 콜백 함수의 포인터를 라이브러리로 전달하는 방식을 사용하지만, 비주얼 베이직으로 응용 프로그램을 구현할 경우에는 이를 LONG형의 주소 값으로 전달해야 라이브러리와 충돌이 생기지 않는다. 따라서 매개 변수의 통일이 필요하다.

C 방식의 선언
typedef void(*FUNCP)(ReturnData *); void IrisRegister(FUNCP pCallback, char *ID)
C 방식의 호출
IrisRegister(pCallback, ID)
비주얼 베이직의 호출
Call IrisRegister(AddressOf pCallback, ID)

<표 1> 이종언어 간의 매개 변수 통일

3. 표준 라이브러리(irisAPI.dll)의 설계와 구조

3.1 표준 라이브러리의 설계[5]



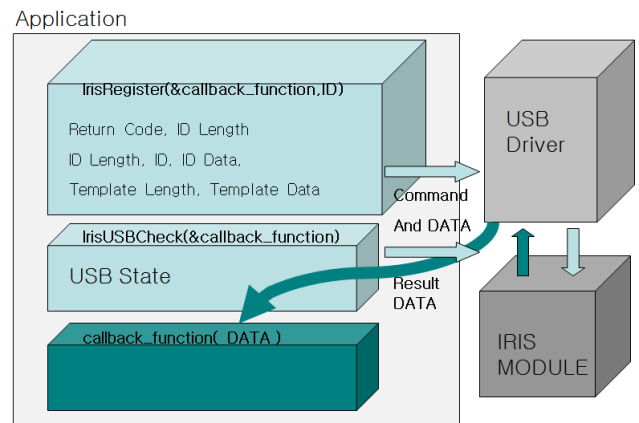
(그림 2) 라이브러리 아키텍처

본 연구에서 사용하는 홍채 인식기는 USB 드라이버를 통해 컴퓨터로 쉽게 연결할 수 있다. (그림 2[4]는 윈도우 2000, 윈도우 xp이상의 환경에서 사용하는 드라이버의 아키텍처를 도식화 하여 나타내었다. 홍채 인식기 모듈과 컴퓨터의 통신은 USB를 통해 통신을 하며, 이를 지원하는 USB 호스트 컨트롤러 드라이버와 홍채

인식 응용 프로그램간의 연결을 맺는 역할을 한다.

스탠다드 라이브러리(irisAPI.dll)는 크게 3가지의 역할을 API로 제공한다. 첫째, 홍채 인식기의 주요 기능을 API로 제공하며, 주요기능으로는 홍채 인식, 홍채 인증, 홍채 템플릿(정보)의 관리 기능이 있다. 둘째, USB 드라이버와 응용 프로그램과의 연결 상태 여부를 판단하여 상태 정보를 응용 프로그램에게 전달하는 API. 셋째, 동적 DLL로딩을 위해 USB드라이버의 연결을 종료하거나 (재) 연결하는 기능을 하는 API가 있다.

(그림 3)은 홍채 인식기(모듈)와 동적 라이브러리(irisAPI.dll), 응용 프로그램간의 데이터 흐름을 도식화 하였다. 동적 라이브러리가 제공하는 API를 이용하여 홍채 등록, 홍채 인증 등을 하는 응용 프로그램을 개발할 수 있다.



(그림 3) 홍채인식기(모듈)와 동적 라이브러리(irisAPI.dll) 어플리케이션 간의 데이터 흐름

3.2 스탠다드 라이브러리의 구조[5]

- 홍채 인식 기능 담당하는 API의 구조

```
typedef struct RxBufferReturnData
{
    unsigned char RxTemplateData[MAX_SIZE];
    int RxTemplateLength;
    unsigned char RxDataIDTotal[MAX_SIZE];
    unsigned char RxDataID[ID_LENGTH_SIZE];
    unsigned char RxDataIDTotalLength;
    unsigned char RxCode;
}ReturnData;
```

- 상태 정보를 응용 프로그램에게 전달하는 API

```
bool g_USBStatus
```

3.3 콜백 함수 포인터 정의(c++)

- 홍채 인식 기능 담당하는 API에서 사용하는 함수 포인터의 정의

```
typedef void(*FUNCP)(ReturnData *);
```

- 상태 정보를 응용 프로그램에게 전달하는 API에서 사용하는 함수 포인터의 정의

```
typedef void(*UsbFUNCP)(bool);
```

#### 4. 스탠다드 라이브러리(irisAPI.dll)의 구현

본 연구에서 응용 프로그램은 c++로 구현하였으며, 따라서 콜백 함수를 구현할 때, c방식의 호출을 사용하였다.(표.1참조) 본문에서는 홍채를 등록하고 인증하는 절차에 대해 구현하고, 프로그램의 실행과정을 본다.

##### 4.1 IrisRegister (홍채 등록)[5]

- 이 함수는 홍채 인식 모듈에 사용자의 홍채를 등록하는 역할을 한다.

- 함수 호출

```
void IrisRegister(FUCNP pCallback, char *ID)
```

- 콜백 함수로 결과 데이터 수신

```
void CallbackFunction(ReturnData *Data)
```

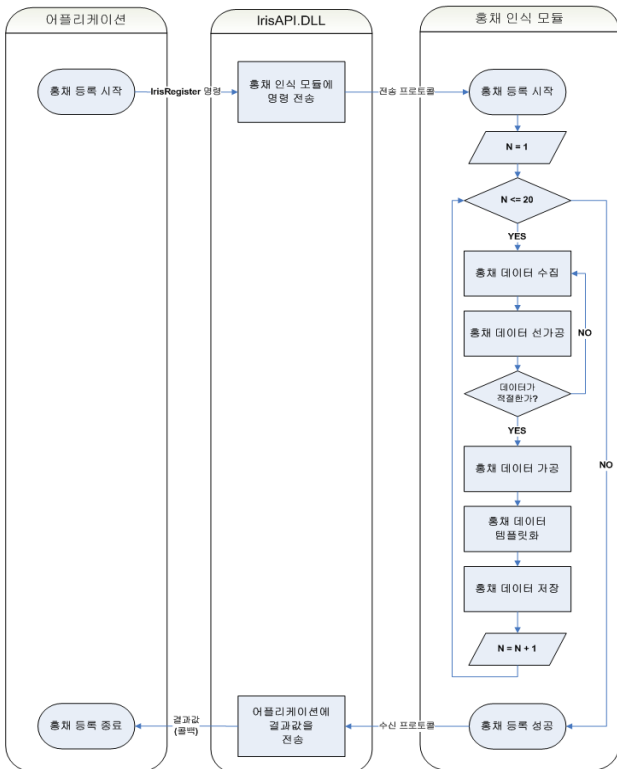
- 매개변수

- . pCallback : 콜백 함수의 함수 포인터를 DLL에 전송
- . ID : 홍채를 등록할 사용자의 ID를 DLL에 전송

- 예제 코드

```
IrisRegister(CallBackFuntion,ID);
:
CallBackFunction(ReturnData Data)
{
    ReturnData.RxCode = Data.RxCode;
    PutMessage(RxCode);
    //등록 성공의 경우 RxCode는 0xf1
}
```

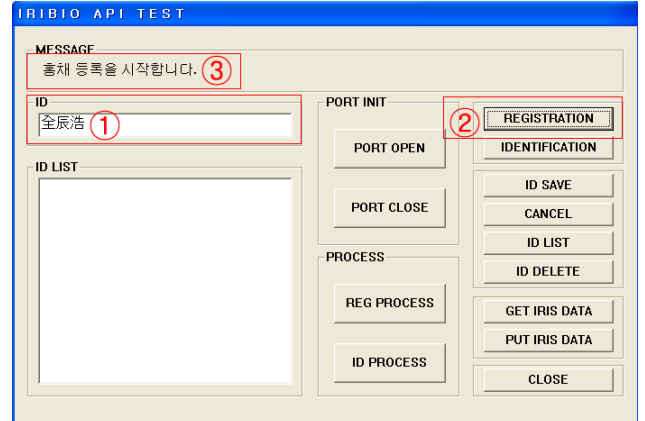
- IrisRegister(홍채 등록)의 과정은 다음과 같다.



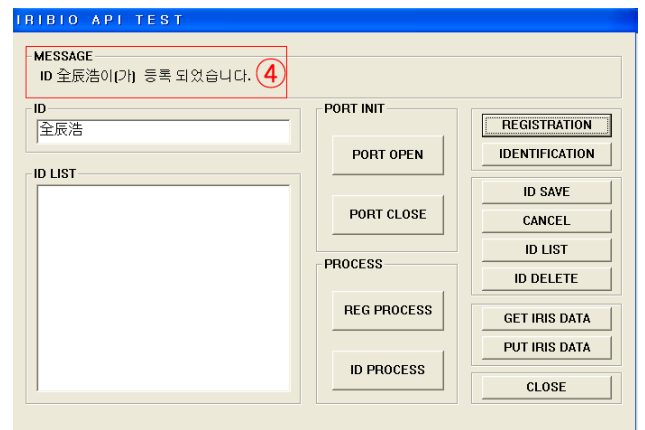
(그림 4) IrisRegister(홍채 등록)의 과정

- 어플리케이션에서 홍채 등록 과정은 다음과 같다.

1. ID를 입력한다.
2. '등록'버튼을 누른다.
3. 메시지가 메시지 창에 출력되면서, 홍채 인식기에서 홍채를 등록한다. (이에 대한 프로세스. (그림 4 참조))
4. 홍채 등록 완료(그림 6)



(그림 5) 홍채 등록 시작



(그림 6) 홍채 등록 완료

##### 4.2 IrisCertify (홍채 인증)[5]

- 이 함수는 홍채 인식 모듈에 등록된 홍채 템플릿으로 사용자를 인증하는 역할을 한다.

- 함수 호출

```
void IrisCertify(FUCNP pCallback, char *ID)
```

- 콜백 함수로 결과 데이터 수신

```
void CallbackFunction(ReturnData *Data)
```

- 매개변수

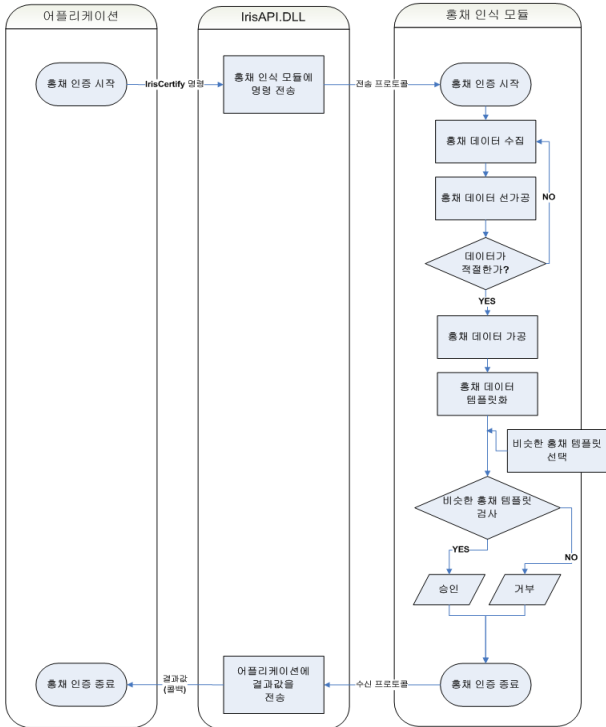
- . pCallback : 콜백 함수의 함수 포인터를 DLL에 전송
- . ID : 홍채를 인증할 사용자의 ID를 DLL에 전송

- 예제 코드

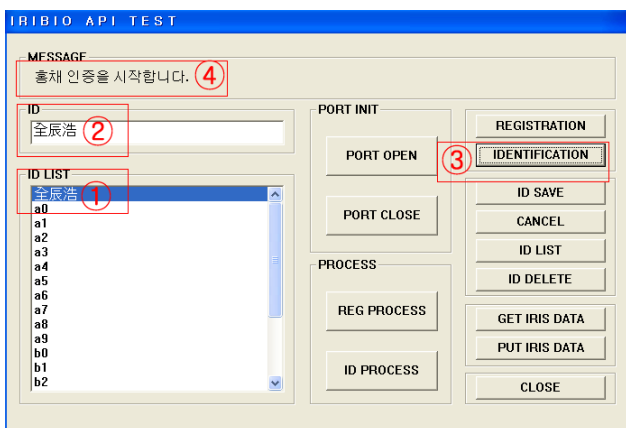
```

IrisCertify(CallBackFuntion, ID);
:
CallBackFunction(ReturnData Data)
{
    ReturnData.RxCode = Data.RxCode;
    PutMessage(RxCode);
    //인증 실패의 경우 RxCode는 0x02
}
    
```

- IrisCertify(홍채 인증)의 과정은 다음과 같다.



(그림 7) IrisCertify(홍채 인증)의 과정



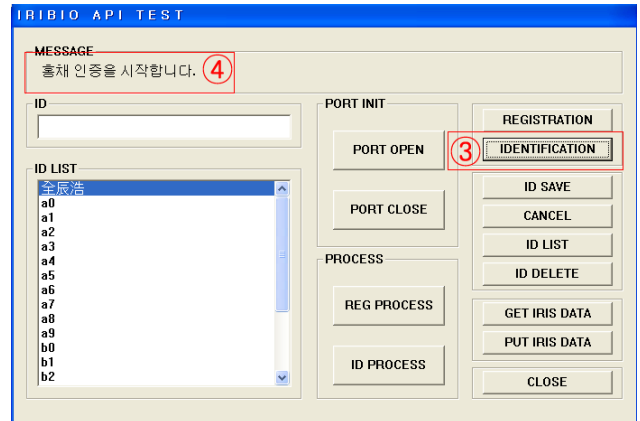
(그림 8) 홍채 인증 시작 (인증할 ID 입력)

- 어플리케이션에서 홍채 등록 과정은 다음과 같다.

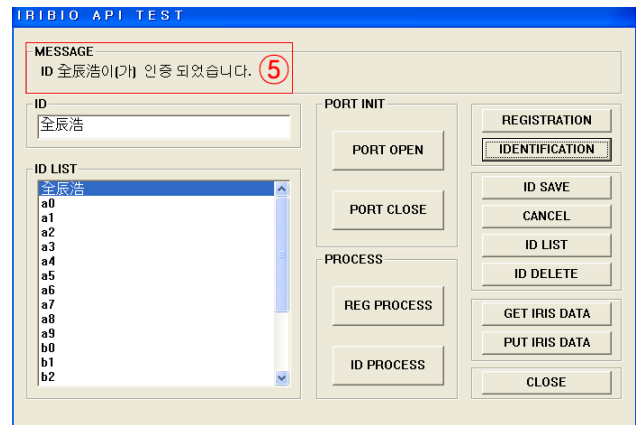
1. ID를 ID리스트에서 선택한다. (ID를 선택 않을 경우, 바로 ③인증)
2. 선택한 ID가 ID창에 표시된다.

3. '인증'버튼을 누른다.

4. 메시지가 메시지 창에 출력되면서, 홍채 인식기에서 홍채를 인증한다. (이에 대한 프로세스는 위의 (그림 7) 참조)



(그림 9) 홍채 인증 시작 (ID 입력 없이 모듈에서 검색하여 인증)



(그림 10) 홍채 인증 완료

## 5. 결론 및 향후 연구과제

본 연구는 홍채 인식 기술을 구현하는 홍채 인식 모듈의 통신을 연결해주는 USB 호스트 컨트롤러 드라이버와 응용 프로그램 사이를 제어하는 스탠다드 라이브러리를 설계하고 구현하였다. 특정 언어만 지원하는 라이브러리가 아니기 때문에 다양한 언어로 개발할 수 있다. 추가적으로 웹에서 이용 가능한 웹서비스로 확장하기 위해 액티브엑스를 이용한 라이브러리(ocx)도 구현하였으며, 향후 이를 사용하여 웹서비스를 할 수 있는 웹 응용 프로그램을 개발하는 연구가 필요하다.

### 참고문헌

- [1] “홍채 인식 기술에 관한 특허 동향”, 정보통신산업진흥원, 2005
- [2] MSDN, “콜백함수” : [http://msdn.microsoft.com/ko-kr/library/d186xcf0\(v=VS.80\).aspx](http://msdn.microsoft.com/ko-kr/library/d186xcf0(v=VS.80).aspx)
- [3] MSDN, “콜백을 사용하여 Windows 응용 프로그램 만들기” : [msdn.microsoft.com/ko-kr/library/hz8wbs6k\(v=VS.80\).aspx](http://msdn.microsoft.com/ko-kr/library/hz8wbs6k(v=VS.80).aspx)
- [4] “D2XX\_Programmer’s\_Guide”, FTDI Ltd, 2010
- [5] “Iris Security System Programmer’s guide”, 아이락 글로벌, 2010