

工學碩士 學位論文

人工知能 技法을 利用한 ARM프로세서 基盤의 指紋認識  
信號處理보드 設計에 관한 研究

A Study on Design of ARM Processor based Fingerprint Recognition Signal  
Processing Board using Artificial Intelligence

指導教授 李 尙 培

2003年 2月

韓國海洋大學校 大學院

電子通信工學科 金 東 漢

## 목 차

Abstract .....	i
제 1 장 서론 .....	1
제 2 장 신경회로망 .....	3
2.1 신경회로망 이론 .....	3
2.2 역전파에 의한 학습 .....	6
제 3 장 지문인식 알고리즘 .....	12
3.1 전처리 .....	12
3.2 특징 추출 .....	15
3.3 지문영상의 매칭(Matching) .....	18
제 4 장 지문인식 신호처리 보드 설계 및 실험 .....	20
4.1 CPU 부분 .....	20
4.2 센서(Sensor) 부분 .....	28
4.3 지문인식모듈 구성도 .....	38
4.4 FAR과 FRR .....	38
4.5 실험 결과 .....	39
제 5 장 결 론 .....	49
참고문헌 .....	50
A. 메인 회로도 .....	52
B. 센서 회로도 .....	53
부 록	
C. 지문인식 신호처리보드의 실제 크기 .....	54
D. 지문인식 신호처리보드의 Gerber .....	55

## Abstract

Fingerprint is a unique feature to an individual. It stays with a person throughout his or her life. This makes the fingerprint the most reliable kind of personal identification because it can't be forgotten, misplaced, or stolen.

Now, social requirements of personal identification techniques are expanding in a number of new application areas. Especially fingerprint recognition technologies among biometrics technologies has more spot lighting because fingerprint authorization is the most profitable and convenient method of Biometric Authentics System(BAS).

Generally, BAS is classified into PC(personal computer) BAS and Stand-alone BAS. PC-BAS is only designed for using into computer and Stand-alone BAS(S-BAS) represents independent system of using microprocessor.

In this paper, neural network was used(one for core and another for delta)in feature extraction and the processor of designed Fingerprint identification Signal processing Board(FSB) used SA(StrongARM), it was used into network, set-top box, PDA(Personal Digital Assistant), PCS(Personal Communications Services), etc. and in the most BAS, fingerprint sensor uses optics sensor but in this paper, fingerprint sensor used cmos E-Field type.

After test, designed S-BAS proved that it is possible to implement independently without PC.

## 제 1 장 서 론

현대 정보화 사회에서는 초고속 통신망의 발전으로 인터넷과 정보기술의 발전을 가져왔다. 기업 및 단체, 나아가 국가적 차원의 핵심적인 정보관리수단으로만 이용되었던 예전과는 달리 이제는 정보 기술의 발달과 정보 시스템의 확충으로 개인도 컴퓨터를 이용해 e-mail, PC뱅킹, 전자상거래 등의 정보활용이 확대되고 있다. 이런 정보 활용 분야 속에는 보안을 유지해야 하는 정보들도 인터넷 환경에서 주고받고 있는 실정이다. 이에 따라 컴퓨터와 관련한 보안사고가 빈번히 발생하고 있어서, 귀중한 정보를 보호하고자 보안에 대한 필요성이 요구되어지고, 그 중요성 또한 급격히 증가하고 있는 추세다.

이에 해결방법으로 많은 정보 보호기술이 개발되고 있으며, 그 중 하나의 방법으로 신체의 특성을 이용하는 생체인식기술 역시 매우 중요한 기술분야로 대두되고 있다. 최근 9.11 테러이후 전 세계적으로도 자국의 공항이나 항만 같은 출입국심사에 생체인식시스템을 적용하고 있는 추세이다.

누구에게나 고유하며 복제하기 어려운 신체의 특징을 이용해 보안성을 높인다는 아이디어를 바탕으로 가장 보편적으로 사용되는 지문인식에서부터 얼굴, 홍채, 정맥, 음성 등을 이용한 생체인식 기술이 연구되고 있다.

일반적으로 생체인식 시스템은 PC상의 생체인식시스템과 독립형(standard-alone) 생체인식 시스템으로 구분할 수 있다. PC상의 생체인식시스템이라고 하면 컴퓨터 상에서만 보안이 필요한 부분에 생체인식시스템을 구축하는 것을 말하며, 독립형 생체인식시스템이란 말 그대로 PC상에서 구현한 기술을 마이크로프로세서를 이용해 하나의 독립된 시스템으로 만든 것을 말한다.

생체인식산업이 PC상의 보안시스템 구축에서 이제는 생체인식기술을 이용한 독립형 인증 시스템 구축 및 이에 대한 라이선스 비즈니스 형태로 발전되고 있으며, 향후에는 이동 통신을 이용한 무선 인증 방식으로 더욱 발전될 전망이다.

따라서, 본 논문에서는 생체인식기술의 한 분야인 지문 인식 기술중 개인별로 고유한 융선의 흐름과 특징점 등의 신체적 특징을 인공지능 기법중 하나인 신경회로망을 이용해서 지문인식 알고리즘을 구성했고, 우리가 흔히 접할 수 있는 네트워크 장비, 셋톱 박스, 휴대폰, PDA의 프로세서로 사용되는 ARM(Advanced RISC Machine)프로세서(Processor)의 일종인 SA(StrongARM) 프로세서를 이용하여 독립

형 지문인식시스템을 구현하였다.

본 논문의 구성은 2장에서 본 논문에서 활용하고자하는 인공지능 기법인 신경 회로망을 설명하고, 3장에서는 신경회로망을 이용한 지문인식 알고리즘을 바탕으로 지문인식 프로그램을 소개하고, 4장에서는 지문인식 신호처리보드의 설계 및 실험 결과를 소개하였으며, 마지막으로 5장에서는 결론을 나타내었다.

## 제 2 장 신경회로망

신경회로망은 두뇌에서의 신경세포처럼 행동하는 요소를 포함한다. 신경회로망의 연구는 본래 생물학적인 뇌의 기능을 이해하려고 하는 노력에서부터 시작되었다. 일반적으로 신경회로망에 의해서 해결될 수 있는 대부분의 문제들은 또한 대체적인 방법으로 해결될 수 있다. 특정한 문제를 해결하는데 신경회로망을 사용할 것인가의 문제는 프로젝트를 책임지고 있는 응용 엔지니어나 설계자의 판단 문제이다. 가격, 구동속도, 신뢰도, 유지의 용이함, 초기 개발의 용이함, 개발과 수정의 용이함 같은 중요한 분야들에서의 장점이 있다는 것을 보여줄 수 있기 때문에 신경회로망의 사용은 여러 분야들에서 적합하게 되었다.

### 2.1 신경회로망 이론

인간의 두뇌는 수많은 뉴론(Neuron)들이 서로 거미줄처럼 연결되어 있는 신경회로망(Neural Networks)구조를 이루고 있다고 알려져 있다. 이러한 인간 두뇌의 신경회로망 조직을 모델링(Modeling) 한 것이 신경회로망 모델이며, 뇌(Brain)에 존재하는 생물학적 신경세포와 그것들의 연결관계를 단순화시키고 수학적으로 모델링함으로써 뇌가 나타내는 지능적 형태를 구현하고자 하는 것이 신경회로망이다.

신경회로망이란 생명체의 신경조직을 모방하여 모델화한 정보처리 시스템으로 단순 소자들의 병렬, 분산된 연결구조를 가지고 있으며, 외부로부터 들어오는 입력에 의하여 동적상태응답(Dynamic State Response)을 일으킴으로써 필요한 출력을 형성시키는 것이다.

일반적으로 신경회로망에서는 지식의 내부적인 표현이 신경회로망 전체에 걸쳐 분산되어 있다. 기존의 기억장치의 경우 기억장치 일부의 파손은 그곳에 기억된 자료의 완전한 손실을 가져오지만 신경회로망의 경우는 신경회로망의 일부가 파손되더라도 그 정도가 약해질 뿐 작동은 가능하며 자료의 일부에 오류가 있더라도 그것으로 인해 최종 결과에 미치는 영향은 그리 크지 않다. 또한, 신경회로망에서 학습 능력이라고 하는 것은 신경회로망의 정보 저장에 있어서 특정한 응용 목적에 따라 뉴론간의 연결강도(Connection Weight)를 변화시키는 과정을 학습(Learning)이라 하는데, 이것은 동일한 정보를 기억하는데 있어서도 사람의 IQ 또는 학습 방법에 따라 학습 시간이 다르듯이 신경회로망의 학습에 있어서도 어떠한 신경회로망 모델

또는 학습 방법을 선택하느냐에 따라 학습시간이 빨라지기도 하고 느려지기도 하며, 심지어는 영원히 학습이 되지 않을 수도 있다는 사실을 염두에 두어야 한다. 이러한 특징을 가진 신경회로망은 최근 들어 컴퓨터 분야는 물론이고 공학, 자연과학, 의학 등 매우 다양한 분야에서 심도있게 연구되고 있다.

신경회로망의 구조는 입력층(Input Layer)과 출력층(Output Layer) 그리고 은닉층(Hidden Layer)의 세부분으로 구성된다. 입력층은 시스템의 외부로부터 입력을 받아들이는 층으로 사람의 감각기관에 존재하는 신경세포들에 해당하고, 출력층은 시스템의 외부로 출력을 내보내는 층으로 뇌의 명령을 운동기관에 전달하는 신경세포에 해당하며, 은닉층은 감각기관들로부터 전달된 자료들을 통해 반응을 결정하는 뇌에 해당된다. 각층의 신경세포는 정보전달정도를 나타내는 연결강도라는 시냅스(Synapse)로 연결되어 있다<sup>[1]</sup>.

신경회로망을 응용하는데 있어서 고려할 사항은 첫째, 신경회로망 모델 둘째, 신경회로망 구조 셋째, 신경회로망 학습방법 넷째, 신경회로망 학습패턴 추출방법 다섯째, 실제 시스템의 구현방법이다.

신경회로망의 응용에 있어서 가장 먼저 고려해야 할 사항은 다양한 신경회로망 모델 중에서 응용 목적에 적합한 신경회로망 모델을 선택하는 것이다. 신경회로망 모델의 선택이 잘못되면 원하는 결과를 얻지 못할 수도 있기 때문이다.

신경회로망 모델을 선택한 다음에는 그 구조를 설계하여야 한다. 구조의 설계라 함은 뉴런 수는 몇 개로 하며, 뉴런들을 어떤 형태로 배치하고 어떻게 연결시킬 것인지, 입력과 출력은 어떻게 할 것인지를 결정하는 것이다.

신경회로망의 구성이 완료되면 원하는 일을 할 수 있도록 신경회로망을 학습시켜야 한다. 학습방법으로는 사람이 새로운 학문을 습득하는 방법에는 학교에 가서 선생으로부터 배우거나 자습서를 이용하여 혼자 독학하는 방법이 있듯이 신경회로망의 학습방법에도 지도학습(Supervised Learning)과 자율학습(Unsupervised Learning)방법이 있다. 이들 학습방법은 신경회로망 모델에 따라 적합한 방법을 선택하여야 한다.

학습방법이 결정되면 신경회로망을 학습시킬 학습패턴을 선정하여야 한다. 여기서 학습패턴이라 함은 지도학습의 경우에는 입력패턴과 원하는 출력패턴의 짝을 의미하며, 자율학습의 경우에는 신경회로망이 자율적으로 결과를 출력하므로 입력패턴만을 의미한다.

또한, 학습패턴의 특징추출이라 함은 패턴 중 특히, 응용목적에 크게 영향을 미치는 중요한 부분만을 학습패턴으로 사용함으로써 정보량을 줄이는 것이다.

최종적으로는 실제 신경회로망을 어떻게 구현할 것인가를 결정하여야 한다. 신경회로망을 구현하는 방법에는 VLSI로 구현하는 방법, 광학적으로 구현하는 방법, 디지털 컴퓨터로 구현하는 방법 등이 있다.

## 2.2 역전파에 의한 학습

역전파 학습 알고리즘(Back-Propagation Learning Algorithm)은 역사적으로 신경회로망의 발전에 매우 중요한 역할을 한 알고리즘 중의 하나이다.

신경회로망은 뉴론과 시냅스로 구성된 생체의 신경 조직을 단순화시켜 모방한 모델을 기본으로 하고 있으며, 한 절점에서의 뉴론은 인접한 많은 뉴론으로부터 입력 신호를 받아서 하나의 결론을 출력하게 되고 시냅스는 뉴론의 출력을 받아서 가중치를 곱한 후 다른 뉴론으로 보내게 된다. 이러한 소자들은 입력층과 출력층 그리고 은닉층으로 나뉘어져 전체 신경망을 구성한다<sup>[2]</sup>.

은닉층에서는 학습을 통해 습득된 지식을 이용하여 주어진 입력에 대해 관련된 정보를 인출하게 되는데, 신경회로망의 학습은 신경회로망에 입력패턴이 주어질 때 신경망의 동작결과인 출력패턴이 나왔을 때 이것을 입력패턴에 대한 목적패턴과 같아지도록 스스로 자신의 연결강도를 조절하는 반복과정이다. 다시 말해 신경회로망으로 하여금 일련의 자료들을 경험하도록 하여 신경회로망이 주어진 규칙에 따라 스스로 자신의 연결강도를 조절하는 과정이다.

이러한 과정을 나타내면 그림 1과 같다.



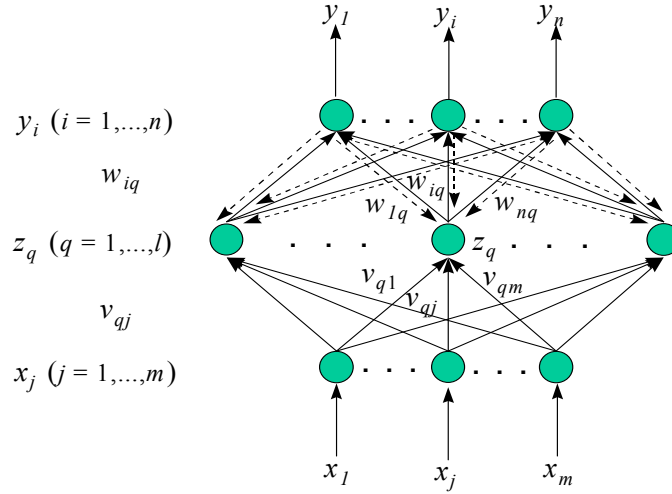


그림 1 3층 구조의 신경회로망

그림 1에서  $m$ 은 입력층의 노드(node) 수,  $l$ 은 은닉층의 노드 수,  $n$ 은 출력층의 노드 수이며, 직선은 전방향 전파(Forward Propagation)를 나타내고 점선은 역방향 전파(Backward Propagation)를 보여주고 있다.

전방향 전파 과정은 주어진 입력패턴  $x$ 에 대해 은닉층의  $q$ 번째 뉴런의 입력으로 받게되는 값으로부터 시작한다.

식 (1)에 의해 은닉층으로의 입력이 계산된다.

$$\text{net}_q = \sum_{j=1}^m v_{qj} x_j \quad (1)$$

은닉층의 출력는 식 (2)와 같다.

$$z_q = f(\text{net}_q) = f\left(\sum_{j=1}^m v_{qj} x_j\right) \quad (2)$$

식 (2)의  $f(\cdot)$ 을 활성화함수(Activation Function)이라 한다.

출력층으로의 입력은 식 (3)과 같고

$$\text{net}_i = \sum_{q=1}^l w_{iq} z_q = \sum_{q=1}^l w_{iq} f\left(\sum_{j=1}^m v_{qj} x_j\right) \quad (3)$$

신경회로망의 출력으로서는 식 (4)와 같다.

$$y_i = f(\text{net}_i) = f\left(\sum_{q=1}^l w_{iq} z_q\right) = f\left(\sum_{q=1}^l w_{iq} f\left(\sum_{j=1}^m v_{qj} x_j\right)\right) \quad (4)$$

식 (1)에서 식 (4)는 주어진 입력으로부터 뉴런층을 통하는 전방향으로 전파되는 과정을 나타낸 것이다. 역방향 전파 과정은 주어진 오차에 대한 역전파 과정이다.

우선 학습에 사용되는 평가함수(Cost Function)는 식 (5)와 같이 정의한다.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^n [d_i - f(\text{net}_i)]^2 \quad (5)$$

$$= \frac{1}{2} \sum_{i=1}^n \left[ d_i - f\left(\sum_{q=1}^l w_{iq} z_q\right) \right]^2$$

주어진 경사 하강법(Gradient-Decent Method)에 의해 은닉층과 출력층 사이의 연결강도를 식 (6)에 의해 갱신한다.

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial w_{iq}} \quad (6)$$

식 (6)에서 사용되는 갱신 규칙  $\partial E / \partial w_{iq}$ 은 식 (7)로 풀이할 수 있다.

$$\Delta w_{iq} = -\eta \left[ \frac{\partial E}{\partial y_i} \right] \left[ \frac{\partial y_i}{\partial \text{net}_i} \right] \left[ \frac{\partial \text{net}_i}{\partial w_{iq}} \right] \quad (7)$$

$$= \eta [d_i - y_i] [f'(\text{net}_i)] [z_q] \triangleq \eta \delta_{oi} z_q$$

여기에 사용된  $\delta_{oi}$ 은 출력층의  $i$ -번째 노드에 대한 오차신호(Error Signal)이고, 식 (8)에 의해 정의된다.

$$\delta_{oi} \triangleq -\frac{\partial E}{\partial \text{net}_i} = -\left[ \frac{\partial E}{\partial y_i} \right] \left[ \frac{\partial y_i}{\partial \text{net}_i} \right] \quad (8)$$

$$= [d_i - y_i] [f'(\text{net}_i)]$$

$$f'(\text{net}_i) : \partial f(\text{net}_i) / \partial \text{net}_i$$

입력층과 은닉층 사이의 연결강도는 식 (9)에 의해 갱신된다.

$$\Delta v_{qj} = -\eta \left[ \frac{\partial E}{\partial v_{qj}} \right] = -\eta \left[ \frac{\partial E}{\partial \text{net}_q} \right] \left[ \frac{\partial \text{net}_q}{\partial v_{qj}} \right] \quad (9)$$

$$= -\eta \left[ \frac{\partial E}{\partial z_q} \right] \left[ \frac{\partial z_q}{\partial \text{net}_q} \right] \left[ \frac{\partial \text{net}_q}{\partial v_{qj}} \right]$$

식 (7)에 의해 식 (9)는 식 (10)으로 유도된다.

$$\Delta v_{qj} = \eta \sum_{i=1}^n [(d_i - y_i) f'(\text{net}_i) w_{iq}] f'(\text{net}_q) x_j \quad (10)$$

식 (10)을 정리하면 식 (11)과 같이 쓸 수 있다.

$$\Delta v_{qj} = \eta \sum_{i=1}^n [d_{oi} w_{iq}] f'(\text{net}_q) x_j = \eta \delta_{hq} x_j \quad (11)$$

식 (11)에서  $\delta_{hq}$ 은 은닉층으로부터 만들어지는 오차신호이고 식 (12)와 같이 정의

된다.

$$\begin{aligned} \delta_{iq} &\triangleq -\frac{\partial E}{\partial \text{net}_q} = -\left[\frac{\partial E}{\partial z_q}\right]\left[\frac{\partial z_q}{\partial \text{net}_q}\right] \\ &= f'(\text{net}_q) \sum_{i=1}^n \delta_{oi} w_{iq} \end{aligned} \quad (12)$$

이상과 같은 갱신방법을 “일반화된 델타 학습 규칙” 이라 부른다. 또한, 연결강도의 갱신은 학습의 가속화를 위해 모멘텀 항(Momentum Coefficient)을 첨가한 경사 하강법에 의해 학습되어진다.

$$\Delta v_{qi}(t+1) = \alpha \Delta v_{qi}(t) - \eta \frac{\partial E}{\partial v_{qi}} \quad (13)$$

$$\Delta w_{iq}(t+1) = \alpha \Delta w_{iq}(t) - \eta \frac{\partial E}{\partial w_{iq}} \quad (14)$$

$\alpha$  : 모멘텀 항

$\eta$  : 학습율

위 식을 그림으로 나타내면 그림 2와 같다.

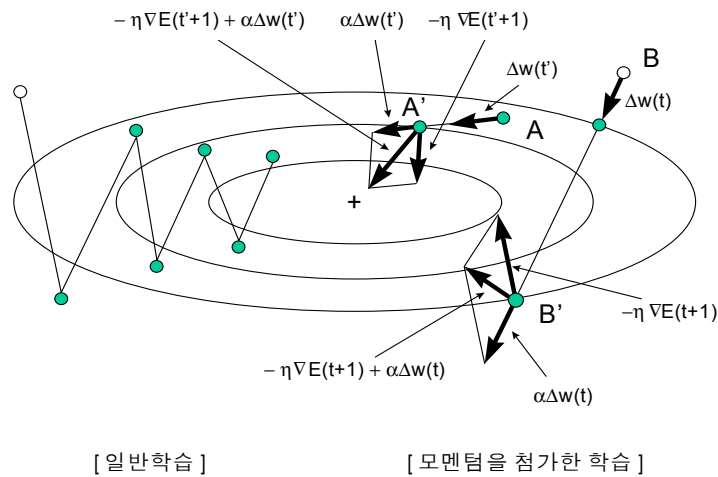


그림 2 2차평면에 대한 에리 경사 하강법

위 그림 2에서 왼쪽 부분은 모멘텀 항이 없을 때의 경로를 나타내는 것이고, 오른쪽 부분은 모멘텀 항을 추가했을 때에 나타나는 경로를 나타낸 그림이다. 위에

서 보는 바와 같이 모멘텀 향이 있을 때의 학습속도가 상대적으로 빠르다는 것을 알 수 있다. 다음은 본 논문에서 구성한 신경회로망을 이용한 특이점 추출을 그림 3으로 나타낸 것이다.

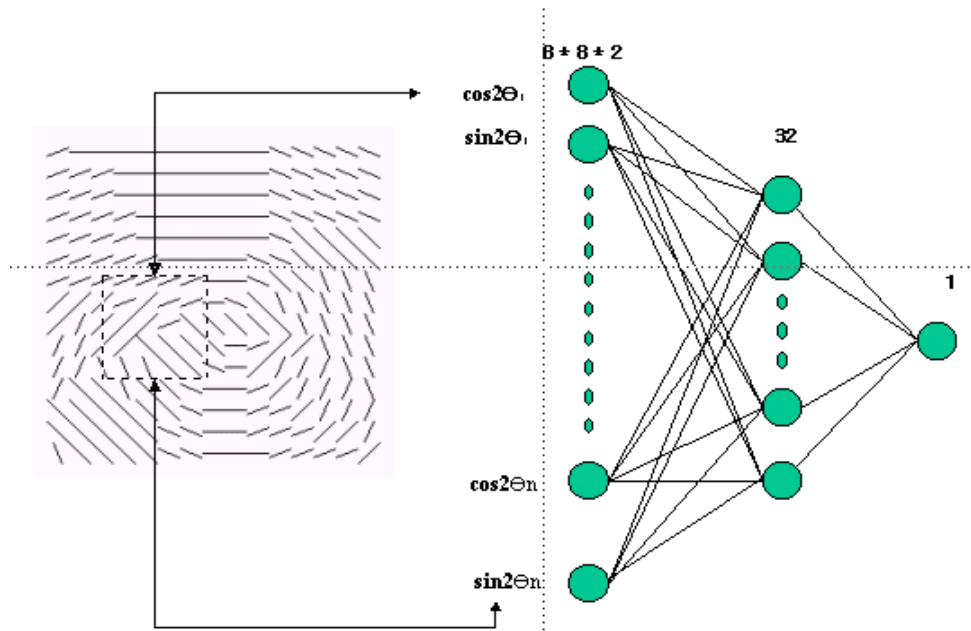


그림 3 특이점 추출을 위한 신경회로망

## 제 3 장 지문인식 알고리즘

본 논문에서는 지문인식 알고리즘 구현을 위해 지문의 특이점(Singular Point)을 추출하는 부분에 인공지능 기법을 사용하여 효율적인 특이점 추출을 수행하였다.

### 3.1 전처리(Preprocessing)

전처리 과정은 입력된 지문영상에서 평활화, 이진화, 세션화 처리를 하게 된다.

#### 3.1.1 평활화(Smoothing)

입력된 지문 이미지는 노이즈와 용선의 끊어짐, 용선의 겹침 등의 오차가 발생한다. 이를 보정하기 위해 두 단계의 필터처리를 해주는데 첫 번째로 256레벨에서의 지문 이미지의 잡음제거와 두 번째는 전체 지문 이미지에서 256레벨이 균일하게 조정하는 것이다.

##### 1) 저대역 통과 필터(Low Pass Filter)

최초 입력으로 들어오는 지문 이미지는 256단계의 그레이 레벨이다. 이 이미지는 많은 노이즈가 섞여 있는데, 가장 많이 섞여 있는 노이즈는 고주파 노이즈와 지문 입력단계에서의 오류이다. 이런 고주파 노이즈와 지문 입력단계 시 오류를 바로 잡기 위해 용선의 방향성을 이용한 저대역 통과 필터를 사용하여 교정하였다 [3][4][5][7].

일반적으로 3가지 공간 컨볼루션 마스크를 사용하는데, 본 논문에서는

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \frac{1}{9} \text{ 를 사용했다.}$$

##### 2) 히스토그램 평활화(Histogram Equalization)

지문입력 시 여러 가지 주변환경과 입력할 때의 압력에 따라 256레벨의 분포가 다르게 나타날 수 있기 때문에, 지문이미지의 명암 값의 분포를 균일하게 하는 작업이다. 히스토그램 평활화는 일정한 분포를 가진 히스토그램을 생성하고, 다음과 같은 3단계로 이루어진다.

- ① 히스토그램을 이용한 명도 값의 빈도 수를 계산한다.
- ② 구한 빈도 수를 이용하여 축적 히스토그램 값을 구하고 정규화 시킨다.

③ 정규화된 축적 히스토그램을 그레이 스케일 사상 함수로 이용하여 그레이 레벨 값을 매핑한다.

축적 히스토그램의 정규화는 식 (15)와 같다.

$$k_i = \frac{g_{\max}}{n_i} H(i) \quad (15)$$

여기서  $n_i$ 는 지문 영상에서의 픽셀의 총 개수이고,  $g_{\max}$ 는 명도의 최대값이고,  $H(i)$ 는 위에서 구한 축적 히스토그램이다<sup>[3][4][5]</sup>.

### 3) 지문영상의 방향성 추출(Direction Orientation Extracting)

지문이미지의 방향성을 추출하기 위한 방법으로 Mehre 방법, Ratha 방법, 창틀 마스크 방법, Sobel 연산자 방법 등이 있는데, 본 논문에서는 시간 대 성능비가 우수한 Sobel 연산자 방법을 사용했다. 용선의 방향성 추출과정은 지문영상을 16×16 블록으로 나누고, 각 블록에서 각 화소에 대한 방향을 Sobel 마스크에 의해 구하여 8방향으로 양자화 시킨 후 가장 많이 추출되는 방향을 각 블록의 대표방향으로 결정한다<sup>[3][8]</sup>.

방향성 계산을 위해 Sobel 마스크의 크기는 3×3으로 구성했다. 마스크는 다음 그림 4와 같고, 각 마스크에서의 중심화소 크기 S와 방향각  $\theta$ 는 식 (16)~식 (19)과 같다.

a	b	c
d	e	f
g	h	i

그림 4 3×3 Sobel 마스크

$$G_y = (g + 2h + i) - (a + 2b + c) \quad (16)$$

$$G_x = (c + 2f + i) - (a + 2d + g) \quad (17)$$

$$S = \sqrt{G_x^2 + G_y^2} \quad (18)$$

$$\theta = \tan^{-1}(G_y / G_x) \quad (19)$$

### 4) 방향성 평활화

지문의 방향성을 구하면 지문 품질이 좋은 영역에서는 실제 용선흐름과 일치하는 방향성이 생기는 반면에 지문 정보가 손상된 영역에서는 용선 흐름과 무관한 방향성이 생기게 된다. 따라서 지문의 방향성 평활화를 수행해야 한다.

### 3.1.2 이진화(Binarization)

그레이 레벨에서의 이미지 향상을 거친 이미지는 이진화 이미지로 변환되어야 하는데 이때 히스토그램을 토대로 임계값을 정해서 이진화 이미지로 변환한다. 이때 가장 중요한 것이 임계값의 선택문제이다. 실제로 채취된 지문 이미지들은 부분적으로 서로 다른 명암을 가지고 있어 전체적 임계값을 이용하여 이진화를 할 경우 정확한 이진화 처리가 되지 않는다. 그러므로 지역적 임계값(Local Threshold)으로 전체 이미지를 구분하게 된다. 지역적 임계값의 범위는 다음과 같다<sup>[3][4][9]</sup>.

$$2^8 - \text{허용범위} \leq \text{임계값} \leq 2^8 + \text{허용범위} \quad (20)$$

본 논문에서 허용범위를 90으로 잡았을 때, 지문영상의 이진화가 가장 잘 처리되었다.

### 3.1.3 세선화(Thinning)

이진 이미지만으로 특징점을 추출하는 데는 어려움이 있다. 특징점을 구하기에 유리하도록 이진화된 이미지로부터 용선의 폭이 1픽셀인 세선 이미지를 만드는 작업을 해준다. 세선화 작업 후 지문 이미지는 용선의 골격만 남게되며, 특징점의 속성이 잘 나타나게 된다<sup>[6]</sup>.

## 3.2 특징추출

세선화된 지문영상에서 인공지능기법을 이용하여 특이점을 추출하고 단점(Ending Point)과 분기점(Bifurcation)으로 구성된 특징점(Minutiae)을 구한 뒤에 의사 특징점을 후처리과정을 통해 보정하게 된다.

### 3.2.1 인공지능 기법을 이용한 특이점 추출

본 논문에서는 방향성이 추출된 영상에서 블록을 형성하여 각 블록에서의 방향성 특징들을 인공지능 기법의 한 분야인 신경회로망의 입력패턴으로 사용하여 특이점 추출을 수행한다. 각 블록은 8\*8의 영역으로 구성되고, 영역 안의 각 방향성 특

징들은 신경회로망의 입력벡터로 사용된다. 그리고 특이점 영역을 검색하여 특이점이 있는 영역은 1, 없는 영역은 0으로 출력된다. 아래 표 1을 보면 고립점(Isolated Point)과 교차점(Crossing Point)은 후처리과정을 통해 보정되어야 할 의사 특징점에 해당한다<sup>[10]</sup>.

구성된 신경회로망은 입력층이  $8 \times 8 \times 2$ 의 128개의 노드로 구성되어있고, 은닉층은 32개의 노드로 구성되어있고, 출력층은 1개의 노드로 구성되어 있다. 신경회로망의 구조는 그림 5에 나타나 있다.

여러 가지 지문의 특이점의 방향성 특징 벡터 40개를 랜덤하게 추출하여 신경회로망의 학습 데이터로 사용하였다.

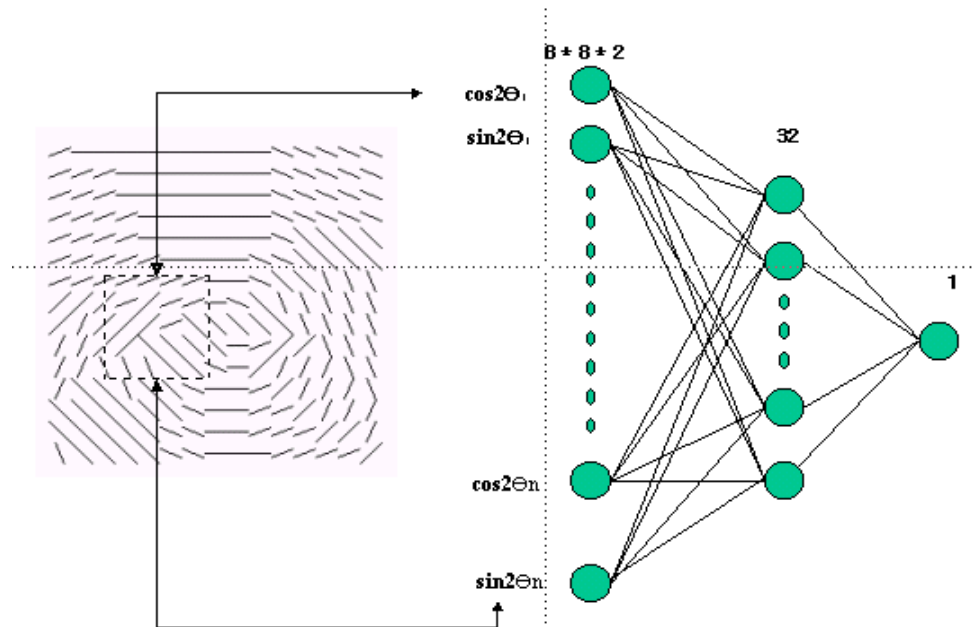


그림 5 특이점 추출을 위한 신경회로망



표 1 특징점

FE	속성
0	점(Isolated Point)
1	단점(Ending Point)
2	융선(Ridge Point)
3	분기점(Bifurcation Point)
4	교차점(Crossing Point)

### 3.2.2 지문영상의 후처리과정(Postprocessing)

특징점 추출 과정에 의해 처음으로 추출된 특징점에는 손상된 지문 이미지와 이미지 처리 오류에 의한 의사 특징점들이 포함되어 있다. 이러한 의사 특징점들은 후처리과정을 통해 제거되어야 한다. 먼저, 융선 사이의 평균 거리 Lamda를 계산한다. 계산된 융선 사이의 평균 거리 Lamda는 잡음 제거 과정에서 사용되는 임계값의 기준으로 하여 알고리즘에 지문의 다양한 변화에 대한 유연성을 준다. 후처리과정의 알고리즘 순서는 다음과 같다<sup>[10]</sup>.

#### 1) 끊어진 융선 연결 알고리즘(Broken Ridge Connect Algorithm)

- ① 평균 융선 거리 Lamda에 의한 마스크구성
- ② 단점이 두 개 이상인 영역을 탐색하여 찾는다.
- ③ 각각의 단점의 방향성을 구한다.
- ④ 각 단점의 좌표사이의 거리를 구한다.
- ⑤ 시작점(Starting Point)을 정한후 3\*3 블록을 이용하여 이웃의 픽셀 중 목적 지점과 가장 가까운 곳을 찾는다.
- ⑥ 찾은 곳으로 이동한 후 ⑤과정에 의해 가장 가까운 목적 지점을 찾은 후 그 곳으로 이동한다.

2) FE(Feature Extraction)가 0인 픽셀을 모두 제거한다.

3) 융선의 길이가 2\*Lamda 보다 짧은 융선을 모두 제거한다.

4) 분기점의 가지의 길이가 Lamda보다 짧고 가지의 끝이 단점일 때 해당 Branch를 제거한다.

5) 분기점의 가지의 길이가 Lamda보다 짧고 가지의 끝이 분기점 일 때 해당

Branch를 제거한다.

### 3.3 지문영상의 매칭(Matching)

지문의 특징인 특이점과 특징점을 이용하여 특징 매칭을 수행할 수 있다. 특이점과 특징점 사이의 상대적 속성이 허용오차 범위 내에서 서로 일치할 때 서로 같은 특징점이라 한다. 즉, 다음의 관계를 모두 만족할 경우 특징점은 일치한다고 판단한다.

1. 특징점과 특이점의 상대적인 위치가 같다. 특징점과 특이점 사이의 거리(Distance)와 각도(Angular)의 일정 범위 내에 특징점이 존재해야 한다.
2. 특징점의 속성(단점, 분기점)이 같아야 한다.
3. 특징점의 각도 차가 각도범위(Angular)에 있어야 한다.

모든 특징점들을 이와 같은 방식으로 서로 비교하여 두 개의 지문 이미지에 대한 유사도(Similarity)를 구하게 된다. 첫 번째, 지문 이미지에서 추출된 특징점의 수를  $m_1$ 이라고 하고 두 번째, 지문 이미지에서 추출된 특징점의 수를  $m_2$ 라고 가정하면, 특징 매칭에 의해서 일치하는 특징점의 수가  $n$ 이라고 할 때 유사도(S)는 식 (21)과 같다<sup>[11]</sup>.

$$S = \sqrt{\frac{n^2}{m_1 \times m_2}} \quad (21)$$

유사도가 특정 임계값 이상이면 두 개의 지문은 서로 일치한다고 판단하며, 그 이하일 경우 서로 다른 지문이라고 판단한다. 임계값은 요구되는 보안성 수준(Security Level)에 따라 유동적이며, 임계값이 낮을수록 신뢰성이 낮으며, 임계값이 높을수록 신뢰성이 높다.

## 제 4 장 지문인식 신호처리 보드 설계 및 실험

본 논문에서 구성한 지문인식 모듈은 크게 2부분으로 구성되었다. 128×128 픽셀 이미지를 입력받는 센서입력부분과 입력받은 이미지를 처리하는 메인부분으로 구성되어 있다. 입력받은 지문 데이터 및 프로그램은 플래쉬 메모리에 저장되어진다.

지문센서는 128×128 픽셀 이미지로 지문영상을 캡처하면, 최적의 지문영상을 위해서 센서 내부적으로 고성능으로 동작하는 최적화 엔진을 내장하여 한번의 지문 입력으로 1초에 10회 이상의 이미지를 스캐닝하게 된다.

### 4.1 CPU부분

본 논문에서의 구성한 지문인식 신호처리보드의 CPU는 최근 영상처리 시스템, 휴대폰, 네트워크 장비, 셋톱박스, PDA 등의 시스템의 핵심부품으로 사용되고 있는 ARM 프로세서의 일종인 SA를 이용해서 구현했다. SA는 ARM사에서 제공하고 있는 ARM V4의 코어를 사용하고 있다<sup>[12][13]</sup>.

#### 4.1.1 SA의 특징

SA의 가장 큰 특성은 저전력과 고성능을 들 수 있다. 저전력 면을 보면 400mW 정도 된다. 즉, 발열량이 매우 적다는 의미가 된다. 이는 특히 임베디드 시스템에서 매우 강력한 특성이 된다. 사용한 CPU 형태는 256 핀 mBGA(mini-BGA)로 매우 작은 패키지 크기를 갖추어서 PDA와 같은 소형화 기기에 유리하다. 단점이라면 PCB에 조립하는 가격이 칩가격에 비해 매우 비싸다는 점이다. SA는 Digital Semiconductor에서 개발한 ARM을 기반으로 한 Intel 프로세서이다. 인스트럭션은 ARM7TDI, ARM9TDMI와 조금 다르지만, 몇 가지 다음과 같은 중요한 차이점이 있다.

- 1) SA는 하바드 구조로 되어 있지만, ARM7은 폰노이만 구조이고, 그리고 ARM9은 하바드 구조다.
- 2) SA는 5 상태 파이프라인 구조이지만, ARM7은 3 상태 파이프라인 구조이고 ARM9은 5 상태 파이프라인 구조다.

3) SA는 ARM 인스트럭션만 지원하지만, ARM7TDMI & ARM9TDMI는 ARM과 Thumb 인스트럭션을 가진다.

4) SA는 ARM7TDMI & ARM9TDMI가 가진 8bit Multiplier보다 빠른 12bit Booth's Multiplier을 가졌다.

5) SA는 JTAG을 기반으로 하는 디버거인 Multi-ICE가진 Embedded Core Debug를 지원하지 않아 디버거를 할 수 없지만, ARM7TDMI & ARM9TDMI는 가능하다.

인텔 SA는 무선휴대 멀티미디어 디바이스를 위한 고성능, 저소비전력의 프로세서이다. 특히 본 논문에서 사용한 SA시리즈는 팜사이즈 디바이스, 스마트폰, 기타 성장하고 있는 무선휴대 멀티미디어 기기들에 적용을 목적으로 100MHz 메모리 버스, SDRAM, SMROM과 같은 다양한 메모리장치, 다양한 I/O 장치들을 손쉽게 설계할 수 있도록 설계되었다. 여기서, 사용한 SA는 신뢰할 수 있는 ARM V4 Architecture를 기반으로 만들어져 있으며, 다른 ARM들과 소프트웨어적으로 호환을 보장하고 있다. SA는 고성능에 향상된 메모리버스와 유연한 적용이 가능한 메모리 컨트롤러를 내장, 특히 그래픽 장비를 덧붙일 때 신뢰할 수 있는 성능을 보장한다.

SA프로세서는 Intel SA-1 코어를 사용하는데 이것은 16KB의 인스트럭션 캐쉬와 8KB의 데이터 캐쉬를 가지고 있으며 MMU를 포함하고 라이더 버퍼는 물론 리드 버퍼도 가지고 있으며 좀 더 향상된 캐싱을 위해 512바이트의 미니 데이터 캐쉬를 별도로 가지고 있다. 다음 표 2는 ARM Core를 비교한 테이블이다<sup>[13]</sup>.

표 2 ARM Core 비교 테이블

	Cache	MMU	Thumb	DSP	Clock
ARM7TDMI	×	×	○	×	133
ARM720T	8K	○	○	×	100
ARM920	16K/16K	○	○	×	250
ARM926EJ	가변적	○	○	○	220-250
ARM1020E	32K/32K	○	○	○	325
SA(StrongARM)	16K/8K	○	×	×	206
Xscale	32K/32K	○	○	○	400

다음 그림 6은 ARM Performance Map을 나타내고 있다<sup>[13]</sup>.

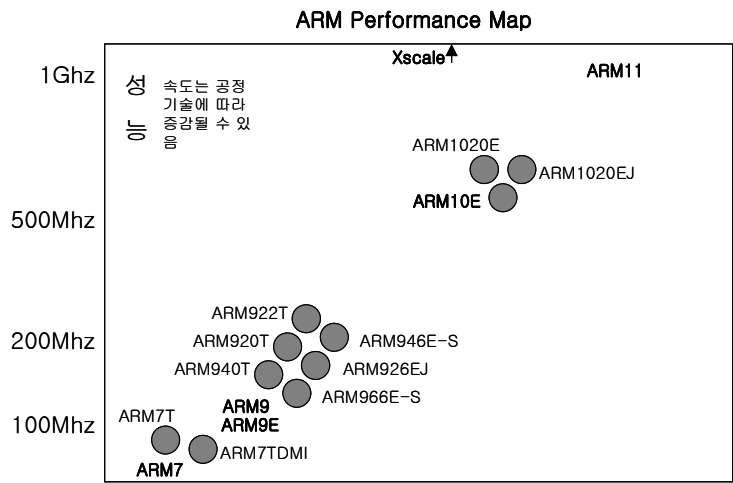


그림 6 ARM Performance Map

다음 그림 7은 SA-1 Core 블록도를 나타낸다.

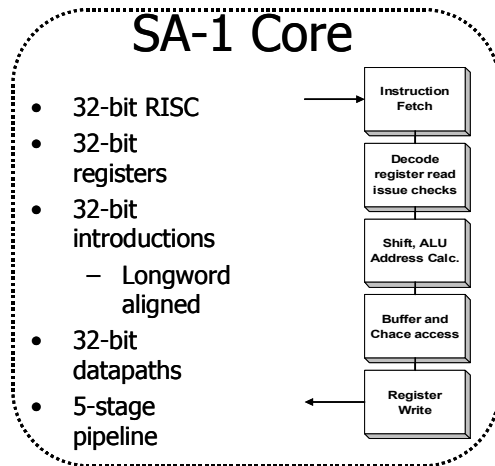


그림 7 SA-1 Core 블록도

#### 4.1.2 SA의메모리 구성

본 논문에서 사용한 SA 메모리부분<sup>[14]</sup>의 메모리 맵을 보면 크게 1) Static 메모리 영역, 2) PCMCIA 메모리 영역, 3) 내부 레지스터 영역, 4) 다이내믹 메모리 영역, 5) 캐쉬 영역으로 나눌 수 있다. 다음 그림 8은 SA의 메모리 맵을 나타내고 있다.

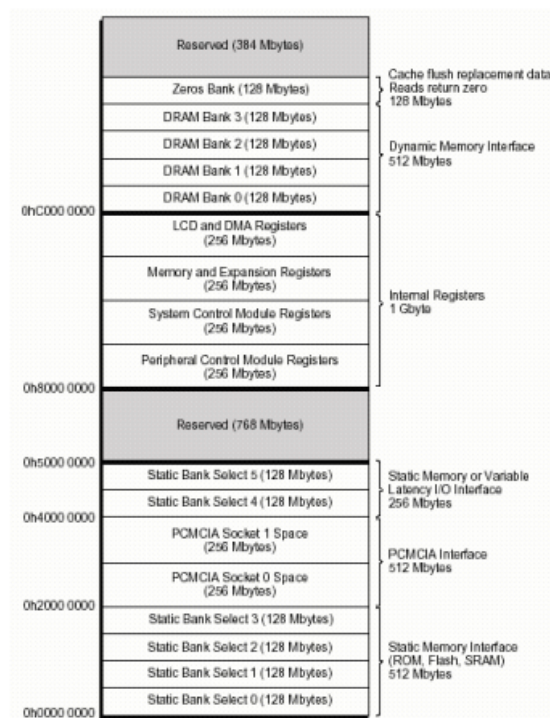


그림 8 메모리 맵

### 1) Static 메모리 영역

ROM, Flash, SRAM 같은 지연 처리가 필요없는 영역과 지연 처리가 필요한 I/O 영역으로 나뉜다. 이 영역은 0x0000 0000 ~ 0x2000 0000, 0x4000 0000 ~ 0x5000 0000으로 두개의 분리된 영역으로 나누어 진다. 0x0000 0000 ~ 0x0800 0000까지는 부팅 처리 소프트웨어가 탑재되는 롬이나 부트 롬 영역에 할당 하게 된다.

각 영역의 크기는 nCS 신호선을 이용하여 처리한다면 128M Byte단위가 된다. SA에 설계되는 외부 인터페이스는 이 공간에 할당한다.

특히 느린 디바이스라면 nCS3 ~ nCS4가 설정되는 0x1800 0000 : nCS3, 0x4000 0000 : nCS4, 0x4800 0000 : nCS5 영역에 설계한다.

## 2) PCMCIA 메모리 영역

PCMCIA는 각 슬롯마다 256MByte단위로 영역이 할당되어 있다.

0x2000 0000 ~ 0x3000 0000 은 소켓 0 의 영역으로 0x3000 0000 ~ 0x4000 0000 은 소켓 1의 영역으로 할당된다.

## 3) 내부 레지스터 영역

SA의 내부에 탑재된 디바이스는 0x8000 0000 ~ 0xC000 0000 영역에 할당되어 있다.

## 4) 다이내믹 메모리 영역 (DRAM 영역)

임베디드 시스템에 사용되는 RAM영역으로 0xC000 0000 ~ 0xE000 0000에 할당되어 있다. 이 공간은 다시 4개로 나뉘어 있는데 각각 128 M Byte 공간을 갖는다.

0xC000 0000 nSDCS0 128MBytes , 0xC800 0000 nSDCS1 128MBytes

0xD000 0000 nSDCS2 128MBytes , 0xD800 0000 nSDCS3 128MBytes

## 5) 캐쉬 영역

이 영역은 캐쉬에 할당된 영역이 아니고 캐쉬를 비우기 위한 것으로 사용될 수 있는 공간이다. 이 영역을 읽기를 하면 무조건 0이 반환된다. 실질적으로 이 영역은 정의만 되어 있는 것이지 사용할 수 있는 공간이 아니다. 이 공간은 0xE000 0000 ~ 0xE800 0000 에 할당되어 있다.

## 6) 예약 영역

그 외 예약영역으로 0xE800 0000 ~ 0xFFFF FFFF 영역은 예약영역이다. 현재 외부에 장치를 붙여 사용 가능한 공간은

0x0000 0000 ~ 0x5000 0000 , 0xC000 0000 ~ 0xE000 0000 까지이다. 그 외의 영역에는 내부적으로 이미 정의된 영역이거나 사용이 불가능한 영역이다. 지문인식 신호처리보드의 경우 외부에 붙는 디바이스는 Flash ROM, SDRAM , RTC, UART 등이 있다. 연결 영역은 Flash ROM는 nCS0 - 0x0000 0000(2M byte 사용). RTC는 nCS4 - 0x4000 0000(1Mbyte 사용), UART : nCS5 - 0x4800 0000(1Mbyte 사용), SDRAM : nSDCS0 - 0xC000 0000(16Mbyte 사용)이다.

### 4.1.3 SA의 메모리 인터페이스

#### 1) 플래쉬 메모리

지문인식 신호처리보드 사용한 플래쉬 메모리의 총용량은 2Mbyte로 구성되어 있다. 플래쉬 메모리의 구조를 표 3에 나타내었다.

표 3 플래쉬 메모리 구조

블록	의미
BLOCK7	로그인 데이터
BLOCK6	지문데이터 및 사용자 레코드
BLOCK5	“
BLOCK4	“
BLOCK3	“
BLOCK2	“
BLOCK1	지문 인식 프로그램 및 응용 프로그램
BLOCK0	부트(Boot) 프로그램

블록 0에는 부트 프로그램(Boot Program)이 들어가는 영역이며, 블록 1에는 지문인식 프로그램 영역으로써, 사용자가 작성한 각종 응용프로그램이 들어가는 영역이다. 블록 2에서 블록 6까지는 응용프로그램에서 사용되는 각종 사용자들의 레코드와 각 사용자에게 대한 지문데이터를 저장한다. 블록 7은 지문인식 모듈에 대한 로그인 정보들을 저장한다.

#### 2) SDRAM(Synchronous Dynamic Random Access Memory) 부분



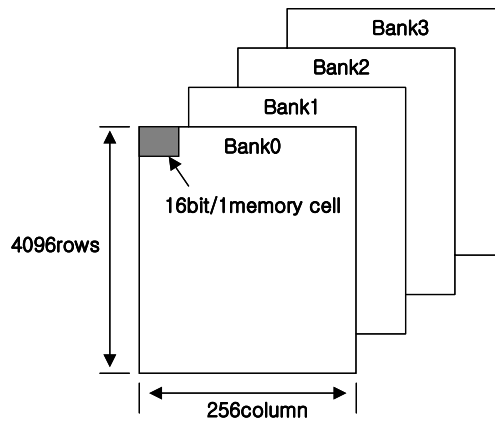


그림 9 SDRAM 메모리 구성

SDRAM 구성은 4096×256개의 메모리 셀로 구성되어 있고, SA의 어드레스 핀(A0~A24)과 연결되어 있다. 주의할 것은 사용되는 메모리가 두 बैं크에서 각각의 16비트씩 32비트 메모리 셀로 구성되므로 메모리셀은 4바이트씩 구성되어 A0와 A1은 사용되지 않는다. 그리고 각각의 बैं크를 접근할 때, 행과 열 주소가 멀티플렉싱 되어진다.

원리를 보면 행 주소를 지정하기 위해서 SA의 [A21~A10]이 SDRAM의 [A11~A0]에 액세스(Access)를 하고, 열 주소를 지정하기 위해서는 SA의 [A9~A2]가 SDRAM의 [A7~A0]에 액세스된다. 즉, 멀티플렉싱이 되는 것이다. 각각의 बैं크를 지정하기 위해서는 SA의 [A23, A22]이 SDRAM의 [BA0,BA1]로 액세스되게 했다. 실제 구성한 지문인식 신호처리보드의 Main 회로도에는 부록의 A를 참고하기 바란다.

## 4.2 센서 부분

그 동안 지문인식 기술은 주로 광학방식이 주류를 이뤄 왔으며 최근 주요 반도체 업체에서 지문 인식용 칩을 개발하면서 반도체 방식 지문인식 시스템의 사용이 늘고 있다. 특히 반도체방식은 광학방식에 비해 지문 모듈의 크기를 소형화 할 수 있고, 가격을 크게 낮출 수 있다는 점 때문에 이를 이용한 시스템 개발이 활기를 띠고 있다.

### 4.2.1 지문센서와 지문 이미지 획득 방법

현재까지 사용되고 있는 지문입력방식을 보면 크게 3가지 정도로 요약할 수 있다.

1) 지문 원지에 의한 입력방법

전통적으로 이용해오던 방식으로 지문 원지에 잉크를 이용하여 지문입력을 받은 다음 이를 스캐너 등으로 입력하는 방법으로 주민등록증 등을 만들 때 볼 수 있다. 실시간으로 지문을 획득할 수 없으므로 자동화된 지문인식 시스템에서는 쓰이지 않는다. 일반적으로 지문 원지로 입력을 받을 때는 입력영상을 최대한 크게 하기 위해 지문을 회전하면서 채취하게 되는데 이를 회전지문이라고도 한다.

2) 광학방식

광학방식 지문입력기는 기존의 잉크와 원지를 이용한 방식의 불편함과 자동화된 지문인식시스템을 위하여 개발되었다. 광학방식 지문입력기의 개략적인 구조는 아래 그림 11과 같다. 광원에서 발생된 입력광이 프리즘을 통해 지문영상을 맺게 되고, 이를 렌즈를 통하여 영상획득장치로 입력받게 된다<sup>[15]</sup>.

광학방식 지문입력기는 그림 11에서 보듯이 렌즈의 위치와 각도 등이 정밀하게 일치하여야 한다. 광학방식 지문입력기에서는 프리즘에 맺힌 상을 카메라와 같은 구조로 촬영하듯이 영상을 입력받음으로서 실제 사람의 손이 입력부에 단았는지 확인하기 위한 별도의 장치를 필요로 한다. 일반적으로 적외선 센서, 인체저항 센서 등을 이용하여 생체 여부를 판단한다.

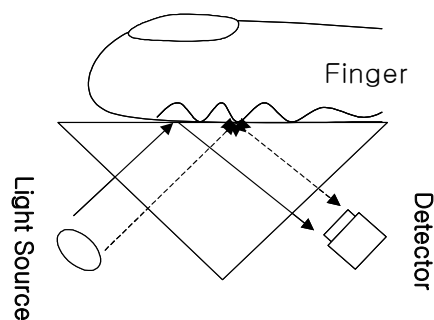


그림 11 광학방식

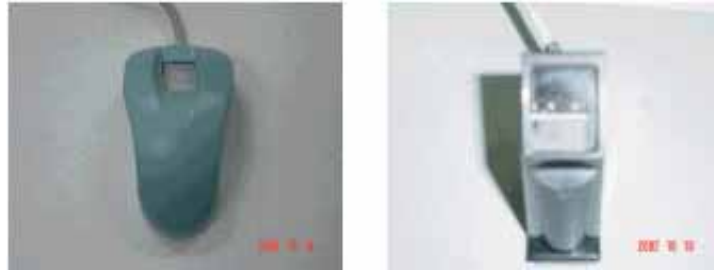


그림 12 광학방식 센서

### 3) 반도체 방식

비교적 최근에 개발된 기술로써, 소형화 및 구조의 간편화로 지문인식 기술의 적용분야가 획기적으로 넓어지게 되는 계기가 되는 방식이다. 또한, 반도체의 특성이 대량생산의 용이성 및 생산원가의 절감 등을 이룰 수 있게 되었다. 아래 그림 13과 같이 반도체 방식 지문입력기는 그 구조가 아주 간단하다. 특별한 구조없이 반도체 표면에 직접 손을 접촉함으로써 지문영상을 획득한다. 반도체 방식은 센서 어레이의 종류에 따라 정전용량, 압력, 열감지등의 방식으로 구분되며, 손가락의 접촉방식에 따라 터치 방식과 슬라이딩 방식이 있다 [15].

반도체 방식의 센서는 고품질 대량 생산이 가능해짐으로써 전체적인 시스템의 저가격화가 이루어지고, 우표크기 만한 소형화가 이루어짐으로써 현재 세계적으로 PC 기반과 임베디드시스템, 마우스, 그리고 자동차 키(Key)에 반도체방식을 적용해 오고 있다.

따라서, 본 논문에서는 현재 가장 각광받고 반도체방식의 지문센서를 이용하여 지문인식 신호처리보드를 구성하였다.

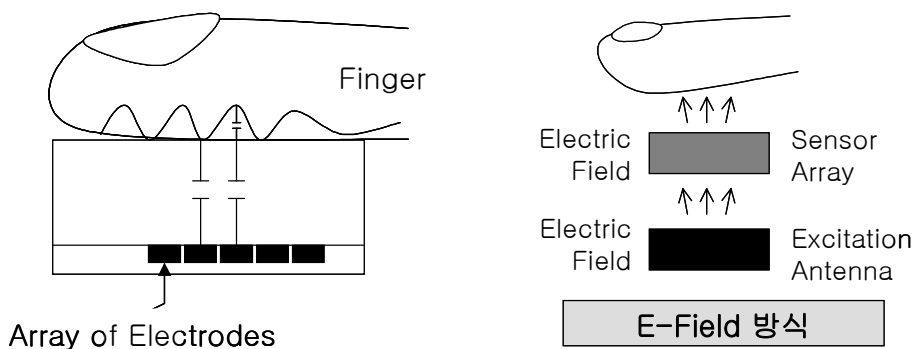


그림 13 반도체 방식 센서 구조



그림 14 Scan 방식



그림 15 Touch 방식

본 논문에서 사용한 E-FIELDTM 센서는 살아있는 사람의 내피층에서 데이터를 검출해서 사용하여 기존 반도체 방식의 최대 단점인 정전기, 습도 등에 대한 내구성을 높인 센서이다. 센서의 외부는 충격에 강한 특수 패키징되어 마모성 테스트(최대 1,320,000 번), 충격 테스트(41,000 psi)에서 강한 내구성을 보여준다. 예로 센서 표면은 7Mohs(Mohs:광물의 경도를 나타내는 단위)의 경도와 세라믹 코팅이 되어 있어 센서가 노출되어있어도, 마모나 장시간 사용에 견디도록 설계 되어있다. 여기서 7Mohs면 단단한 유리와 같은 강도를 나타낸다. 그림 16은 지문에서 Core의 위치를 나타낸 것이다. 일반적으로 손톱의 시작위치 바로 아래에 놓여 있으므로 센서에 입력시 이 부분이 반드시 센싱이 되게 해야한다.

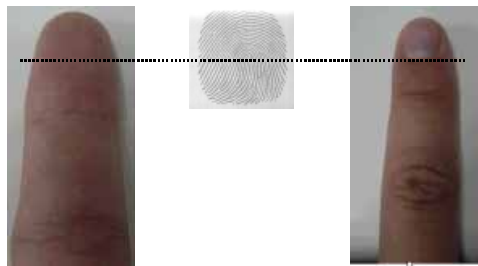


그림 16 Core Alignment

다음 표 4는 두 방식을 간략히 비교한 자료이다<sup>[17]</sup>.

표 4 두 가지 방식의 센서 비교 테이블

	반도체 방식	광학 방식
가격	고가이지만, 대량생산을 통해 가격을 줄일 수 있다.	카메라의 가격이 고가이고, 시스템 구조의 제한에 의해 가격절감이 어렵다.
감지	사람의 생체특성을 이용한 방식이므로 광학방식에 비해 지문의 복제여부를 판단하기가 용이하다.	단순히 프리즘에 닿은 영역을 카메라로 읽는 방식이므로 지문의 복제여부를 판단하기가 거의 불가능하다.
충격	외부에 직접 노출되어 있어 충격에 약하지만, 이를위해 센서의 표면을 특수코팅하여 방충성을 높인다.	정밀하게 위치가 정해졌기 때문에 외부 충격에 의해 광원, 프리즘, 카메라의 위치가 조금이라도 바뀌면 성능이 낮아지고, 프리즘에 상처가 나거나 먼지가 내부에 삽입될 경우에는 카메라의 영상의 잡음이 생긴다.
활용 범위	사이즈 작고 저전력 소자이므로 휴대폰, 노트북, PDA 및 도어락등에 다양하게 적용할 수 있다.	모듈의 크기가 크기 때문에 활용 범위가 다소 제한적이다.

#### 4.2.2 센서의 특성 및 구동 원리

본 논문에서 사용한 센서는 최근 각광받고 있는 반도체 방식의 센서로서 JEDEC 표준 PLCC(Plastic Leaded Chip Carrier) 형태의 68핀으로 구성되어 있으며, 센서의 매트릭스 크기는 가로세로 각각 13mm×13mm이고, 128×128(총 16384개의 픽셀)로 구성된다.

매트릭스는 0-7까지의 8개의 어레이로 구성되어 있고, 매트릭스 주변에는 3mm 두께의 Drive Ring으로 구성되어 있다. Drive Ring에 손가락이 접촉되어야만 센서가 동작하게 된다. 또, 센서는 비동기식의 RS-232통신과 USB통신을 지원한다.

본 논문에서는 비동기식중 RS-232통신을 이용해서 구성하였다. 하나의 Sub-Array는 16픽셀 행과 128픽셀의 열(총2048 element)로 구성되어 있고, 오실레이터에서 16MHz의 크리스탈로부터 입력받으면, 센서 내부의 PLL(Phase Lock Loop)에 의해 32MHz 동작된다.

다음 그림 17은 센서 인터페이스를 나타낸 것이다.

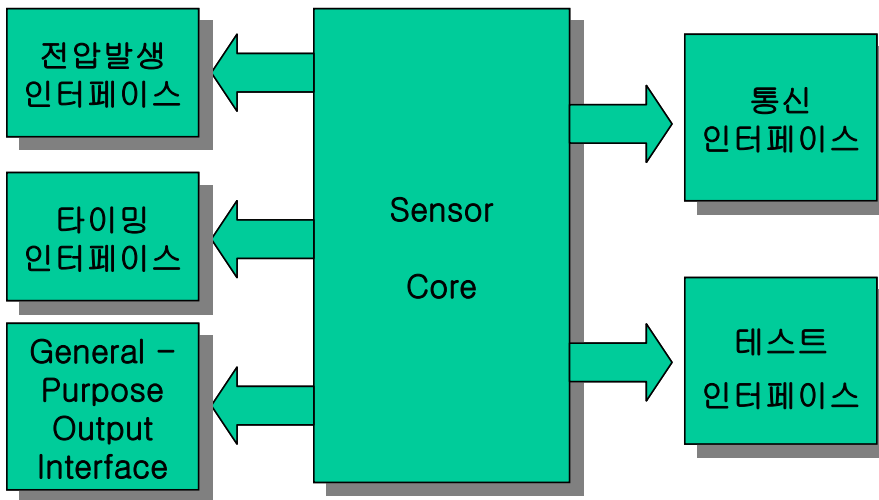


그림 17 센서 인터페이스

1) 시리얼 인터페이스(Serial Interface)에서는 8비트 데이터를 사용한다. 다음 그림 18은 시리얼 패킷 포맷을 나타낸 것이다.

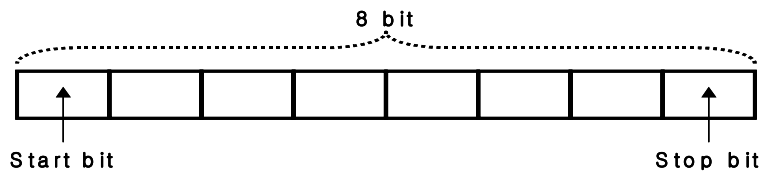


그림 18 시리얼 패킷 포맷

그리고, Baudrate Select 핀이 3개 있는데 여기서 Baudrate를 선택해서 사용할 수 있다.

표 5 Baudrate

BaudSelect1-3	RATE
000	187.5Kbps
001	230.4Kbps
010	375.0Kbps
011	750.5Kbps
100	2Mbps
101	115.2Kbps
110	460.8Kbps
111	921.6Kbps

2) 테스트 인터페이스(Test Interface)에서는 부가적인 제어와 아날로그와 디지털 요소의 시각적인 테스트를 제공한다.

3) 타이밍 인터페이스(Timing Interface)에서는 외부의 16MHz 크리스탈이 내부의 PLL(Phase Lock Loop)에 의해 32MHz로 동작한다.

4) 전압발생기(Excitation Interface)에서는 이미지 추출을 위해 센서 어레이에 파워를 공급한다.

5) GP Output Interface(General-Purpose Output Interface)은 Led나 릴레이(Relay)와 같은 외부 장치(Devices)를 제어하는 기능을 제공한다.

다음 그림 19와 그림 20은 각각 센서 영역(Dimensions) 및 센서의 구성을 나타내고 있다.

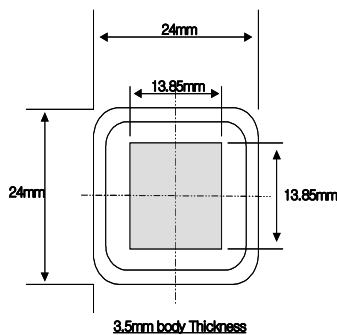


그림 19 센서 영역

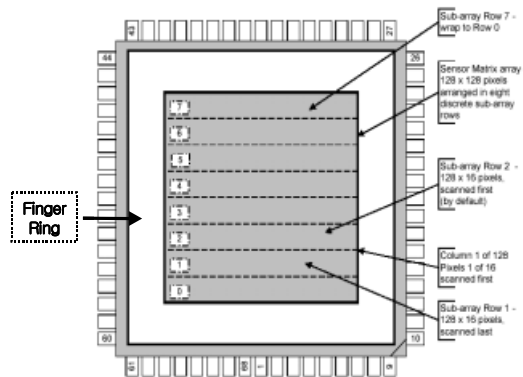


그림 20 센서 구성도

다음 그림 21은 센서 매트릭스 부분에 대해 나타낸 것이다.

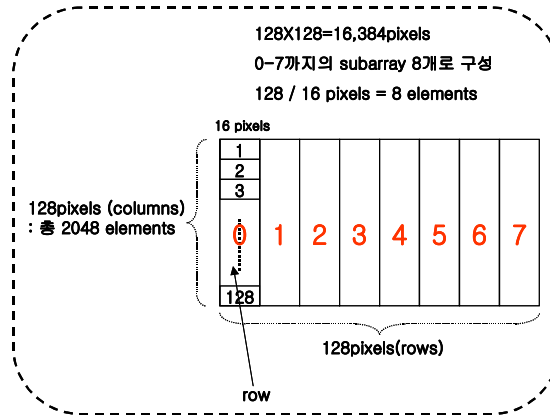


그림 21 센서 매트릭스 구성도

센서 어레이 동작을 보면 다음과 같은 순서로 스캔이 된다.

Row2→Row3→Row4→Row5→Row6→Row7→Row0→Row1

세부적으로 보면 Row2에서의 스캔 동작을 그림 22에 나타내었다.

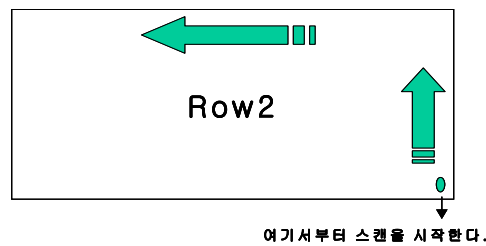


그림 22 센서의 세부적 동작도

다음 그림 23은 센서의 전체 동작 메카니즘을 나타낸 것이다.



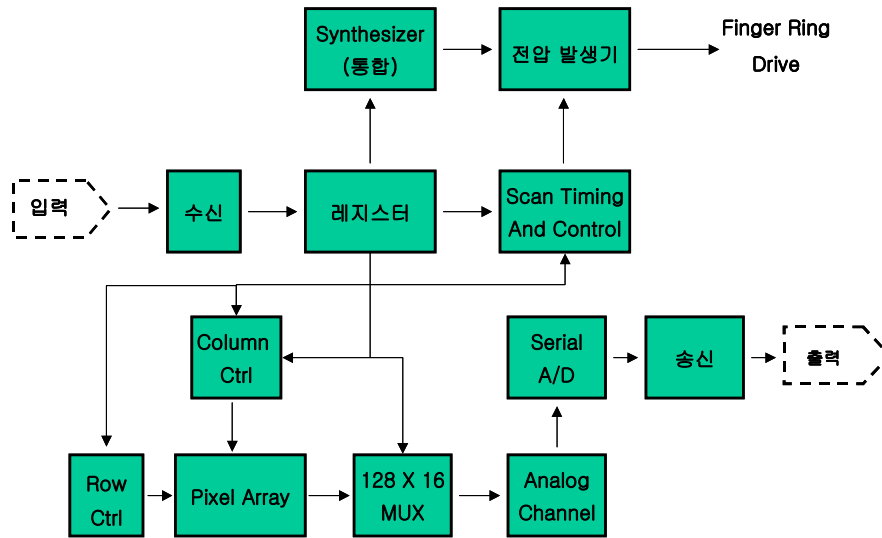


그림 23 센서 전체 동작 매카니즘

위 그림 23에서 센서에는 총 32개의 레지스터가 있는데, 정상적인 지문의 입력시 좋은 이미지를 얻기 위해 센서의 컨트롤과 이미징에 있어서 다음 표 6과 같은 셋팅 값을 가지고 있다.

표 6 센서 레지스터 셋팅 값

Register	Value(Hexadecimal)	Register	Value(Hexadecimal)
80h	00h	8Eh	0Ch
81h	01h	8Fh	72h
82h	00h	90h	12h
83h	00h	91h	0Bh
84h	00h	92h	1Fh
85h	00h	93h	00h
86h	00h	94h	00h
87h	00h	95h	00h
88h	0Ah	96h	00h
89h	38h	97h	14h
8Ah	64h	98h	1Ch
8Bh	0Ah	99h	29h
8Ch	04h	9Ah	05h
8Dh	34h		

실제 구성한 센서 회로도도 는 부록 B를 참고하기 바란다.

### 4.3 지문인식 모듈 구성도

아래 그림 24는 실제 구성한 지문인식 신호처리보드를 나타낸다.



그림 24 실제 설계한 지문인식 신호처리보드 모듈

### 4.4 FAA와 FRR

지문인식 기술에서 성능을 평가하는 기준은 크게 3가지로 나눌 수 있다. 그 첫째는, 접근이 허락되지 않은 사람을 통과시키는 오류, 즉 타인 수락율(FAR-False Acceptance Ratio)이다. 이 비율이 1%라면 지문인식기를 통과한 100명중 1명은 사실상 출입이 금지된 사람이 된다는 것이다. 즉, 타인 수락율이 높을수록 시스템에서는 치명적이 된다. 일반적인 보안을 목적으로 하는 시스템에서 지문인식 기술의 우위를 가늠하는 대표적인 척도로 대개 이 FAR이 꼽힌다.

두 번째는, 출입이 허락된 사람을 통과시키지 않는 오류, 즉 본인 거부율이다. 이 오류율이 높다면 당연히 출입해야 할 사람인데도 출입불가가 되는 것이다. 따라서 현실에서 상용화되기 위해서는 이 비율을 최소화시키는 것이 곧 시스템의 보안성을 높여 어느 정도 안정적인가를 신뢰할 수 있게 하는 기준이 된다. 마지막으로, 속도를 들 수 있다. 검증의 정확성도 중요하지만, 실생활에 지문인식이 사용되려면 지문인식 속도도 빨라야 한다. 다음 그림 25에서 오른쪽(본인)의 왼쪽 낮은 부분이 본인에 대해 거부하는 영역. 즉, FRR이고, 왼쪽(타인)의 오른쪽 낮은 부분이 타인에 대해 수락하게 되는 영역. 즉, FAR이다<sup>[16]</sup>.

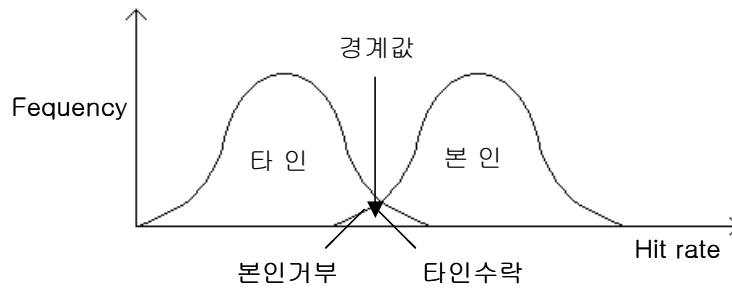


그림 25 FAR 및 FRR

생체보안시스템에서 가장 이상적인 형태는 FRR 및 FAR을 포함한 전체 오류율이 0%이지만, 시스템 성능과 상관없이 사용자의 미숙함 등으로 인해 FRR이 발생하기도 한다.

#### 4.5 실험결과

본 논문에서 설계한 독립형 지문인식 신호처리보드의 구현 테스트를 위해 두 가지 방법을 사용했다. 먼저 PC상에서 지문인식 알고리즘을 구성해서 테스트를 했고, 마지막으로 테스트된 알고리즘을 독립형 지문인식 신호처리보드를 통해 컨트롤 보드와 시리얼 통신을 통해 신뢰성 및 구현 테스트를 했다.

##### 4.5.1 PC상에서의 알고리즘 테스트

본 논문에서 사용한 지문 이미지는 광학방식의 지문입력기를 사용해서 10명당 10개씩 지문을 입력받아 총 100개 지문을 얻어 성능 테스트를 했다. 입력시 raw파일 형태의 지문이미지를 128×128 픽셀 이미지로 재구성해서 사용했으며, 실험환경은 표 7과 같다.

표 7 성능 테스트를 위한 실험환경

	시스템 구성
시스템 사양	Intel celeron 133MHz
개발 환경	Window98
개발 툴	Visual C++ 6.0
Test 지문 개수	성인남자 10명에 대한 10개의 지문 (10×10=100개의 지문)
재구성된 지문이미지 크기	128×128 픽셀
지문영상의 해상도	500DPI(Dot Per Inch)

1) 다음 그림 26 ~ 그림 28은 전처리 과정을 거친 지문이미지이다

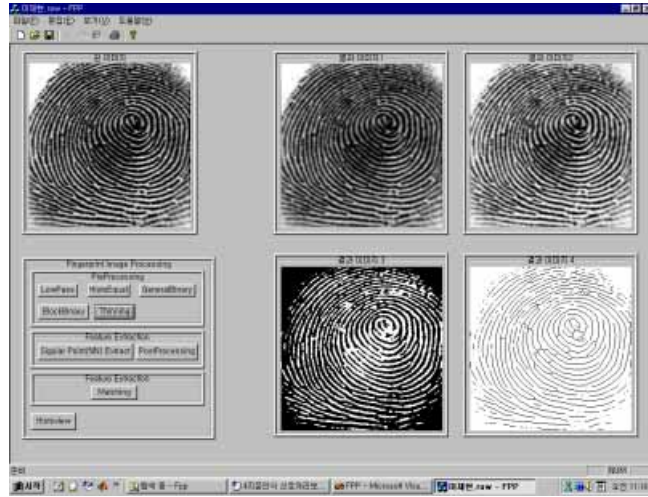


그림 26 전처리 과정을 거친 지문이미지

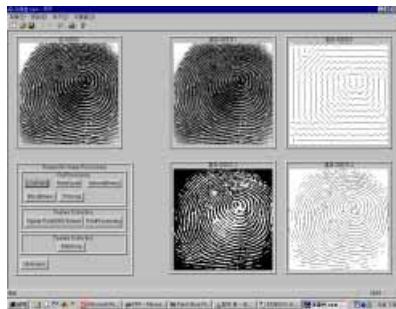


그림 27 방향성 이미지

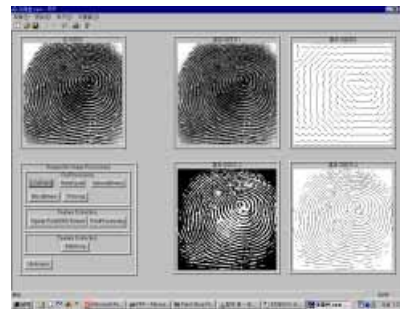


그림 28 평활화 이미지

2) 그림 29는 특징추출 과정을 거친 지문 이미지의 그림이다.

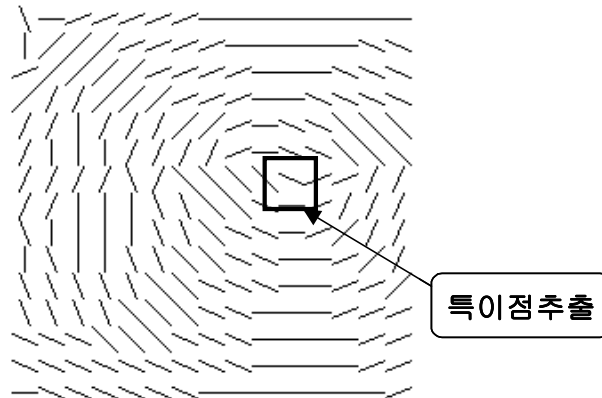


그림 29 특징 추출 이미지

3) 그림 30은 매칭(Matiching) 과정을 거친 지문 이미지이다.

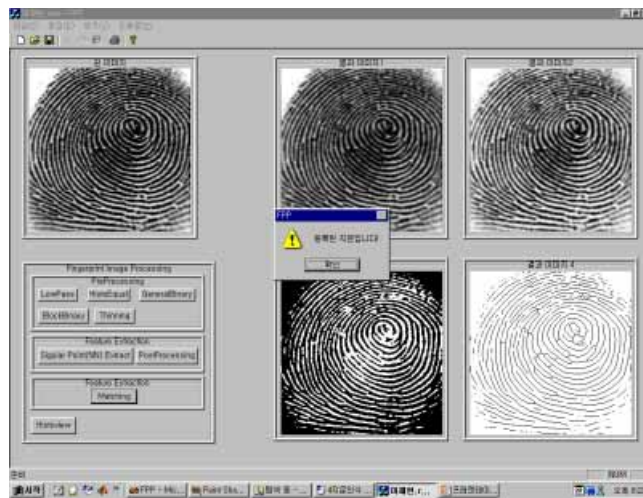


그림 30 매칭된 이미지

#### 4.5.2 독립형 지문인식 신호처리보드의 구현 테스트

PC상에서 구현한 지문인식 알고리즘을 바탕으로 본 논문에서 구현한 독립형 지문인식 신호처리보드에 적용하였다. 실험환경은 표 8과 같다.

표 8 독립형 지문모듈의 실험환경

	시스템 구성
컨트롤 보드 프로세스	AT89C52
테스트 방법	Serial 통신
개발 툴	Keil C (Language)
테스트 지문 개수	10명에 대한 10개의 지문 (10×10=100개의 지문)
재구성된 지문이미지 크기	128×128 픽셀
지문영상의 해상도	500DPI(Dot Per Inch)

##### 1) 테스트 방법

먼저, 지문인식 신호처리보드에 지문을 입력하여 컨트롤 보드와 serial 통신을 통해 데이터의 송/수신을 확인했다. 통신 패킷은 총 7byte로 구성되어 있는데, 패킷 구성을 보면 다음 표 9와 같다.

표 9 시리얼 통신 패킷 구성도

Command	Para1	Para2	ErrorCode	CheckSum
1Byte	2Byte	2Byte	1Byte	1Byte

- ① Command : 독립형 지문인식 신호처리보드를 활성화시키는 명령을 보내는데 사용한다.
- ② Para1 : 데이터 전송에 사용한다.
- ③ Para2 : Para1의 추가 데이터가 있는 경우에 데이터전송에 사용한다.
- ④ ErrorCode : 명령에 대한 독립형 지문인식 신호처리보드의 실행

결과를 나타내는데 사용한다.

⑤ Checksum : 통신 에러를 검출하는데 사용한다.

## 2) 통신 패킷 사용방법

① 모든 명령들은 control board로부터 독립형 지문인식 신호처리 보드로 전송된다.

② 독립형 지문인식 신호처리보드는 Command Packet에서 같은 형식을 사용하여 승인된 각 명령으로부터 응답을 받는다.

Command Packet은 PC로부터 독립형 지문인식 신호처리보드로 보내진다.

ACK Packet은 독립형 지문인식 신호처리보드로 보내진다.

③ 독립형 지문인식 신호처리보드는 ACK의 에러코드 필드안의 명령의 결과를 보낸다. 만약 명령의 활성화가 에러없이 완전하다면, 에러 코드는 '0'이다.

다음 그림 31은 테스트를 위해 구성된 컨트롤 보드와의 인터페이스를 나타낸 블록도이다.

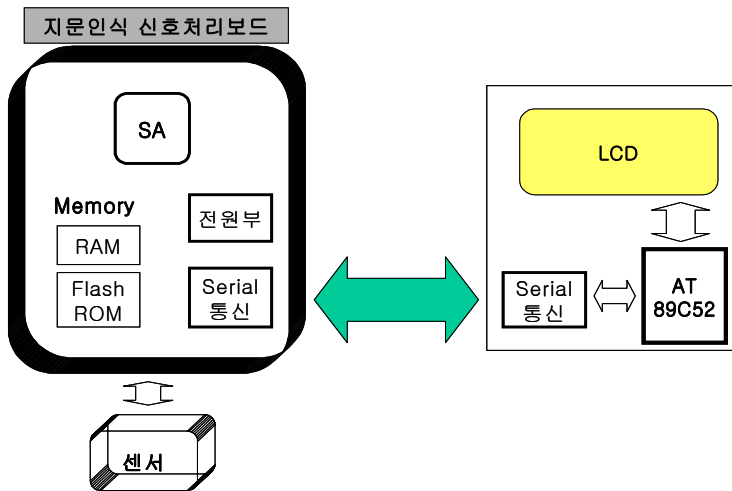


그림 31 구성된 컨트롤 보드와의 인터페이스 블록도



다음 그림 32는 실제 구성한 지문인식 보드와 컨트롤 보드와의 시리얼 구현 테스트이다.

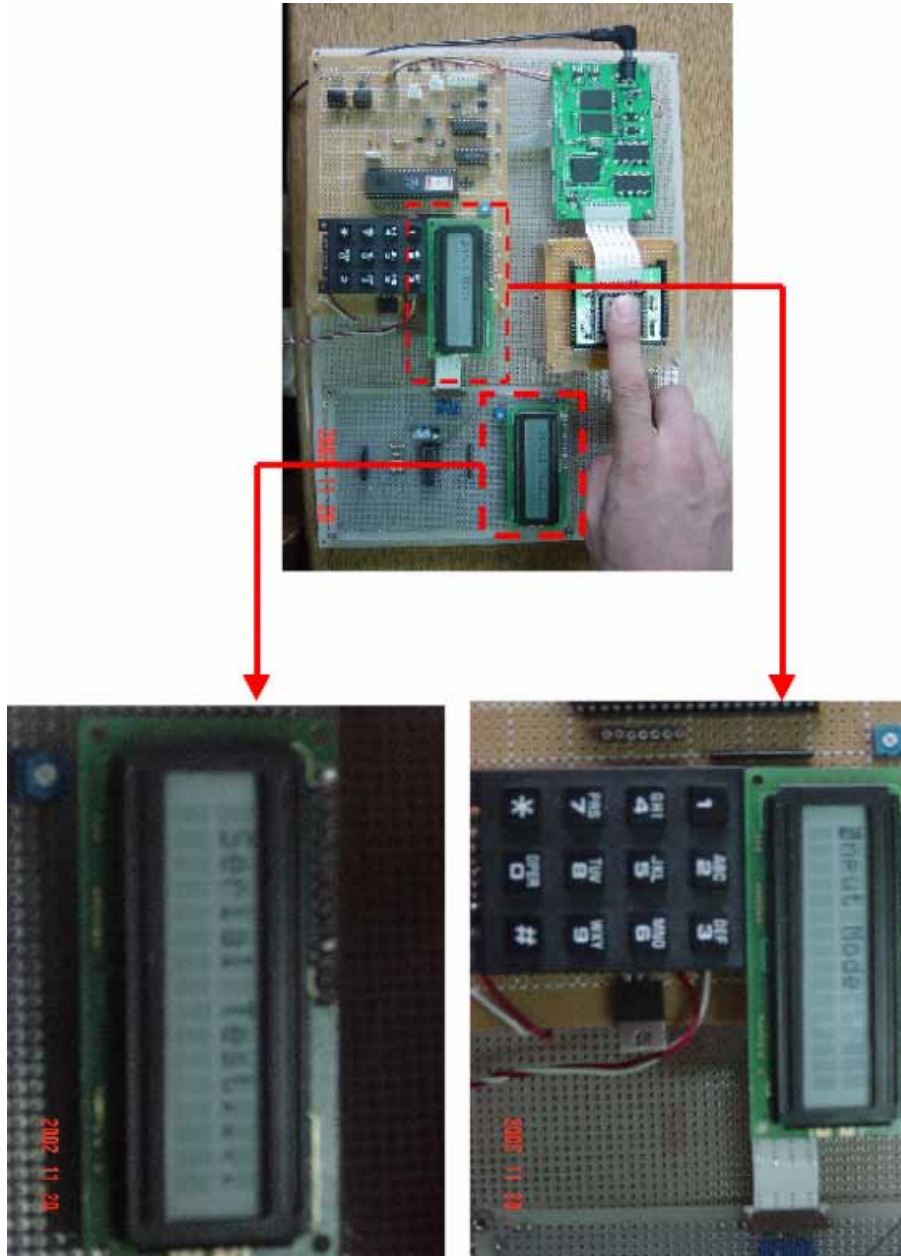


그림 32 지문보드와 컨트롤 보드와의 시리얼 인터페이스 구현 테스트  
다음 그림 33은 지문등록을 나타낸 것이다.

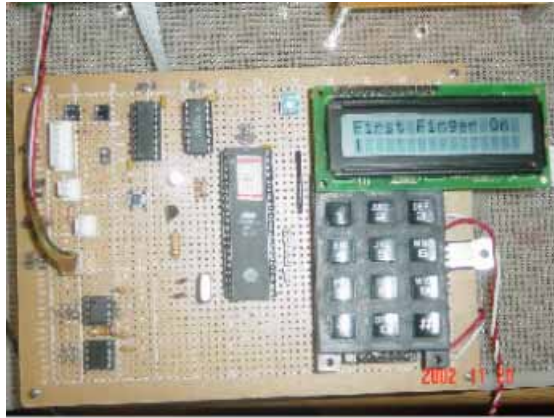


그림 33 지문등록을 나타낸 그림

다음 그림 34은 지문인증을 나타낸 것이다.



그림 34 지문인증을 나타낸 그림

위 그림 33, 그림 34에서 보는 바와 같이 지문인식 신호처리 보드와 컨트롤 보드와의 시리얼 인터페이스 구현 테스트를 한 결과 충분히 독립형 지문인식 신호처리보드로 사용할 수 있음을 알 수 있었다.

## 제 5 장 결 론

본 논문에서는 생체 인증 방식중의 하나인 지문인식에 인공지능 기법인 신경회로망을 이용하여 독립형 지문인식 신호처리보드를 설계, 구현하였다. 지문인식 알고리즘을 PC상에서 테스트 하였고, 설계한 독립형 지문인식 신호처리보드를 테스트하기 위해 독립형 지문인식 신호처리보드와 컨트롤 보드와 Serial 통신을 통해 구현 테스트를 하였다.

먼저, 지문인식 알고리즘 구현에 있어서 일반적인 전처리 과정을 거쳐, 특징추출 시 본 논문에서는 방향성이 추출된 영상에서 블록을 형성하여 각 블록에서의 방향성 특징들을 인공지능 기법의 한 분야인 신경회로망의 입력패턴으로 사용하여 특이점(singular point) 추출을 수행했다. 다음으로, 컨트롤 보드와 Serial 통신을 통해 독립형 지문인식 신호처리보드에 대해 신뢰성을 테스트를 한 결과 충분히 PC없이 독립적으로 동작할 수 있음을 제시하였다.

앞으로의 연구방향은 좀 더 효과적인 지문인식 알고리즘 개발과 다양한 연령층과 성별에서의 지문 이미지의 표본 추출을 통한 알고리즘 성능 테스트 그리고, 활용분야의 확대를 위해 지문인식 신호처리보드의 크기를 최소화하는 방향으로 연구가 계속되어야 할 것이다.

## 참 고 문 헌

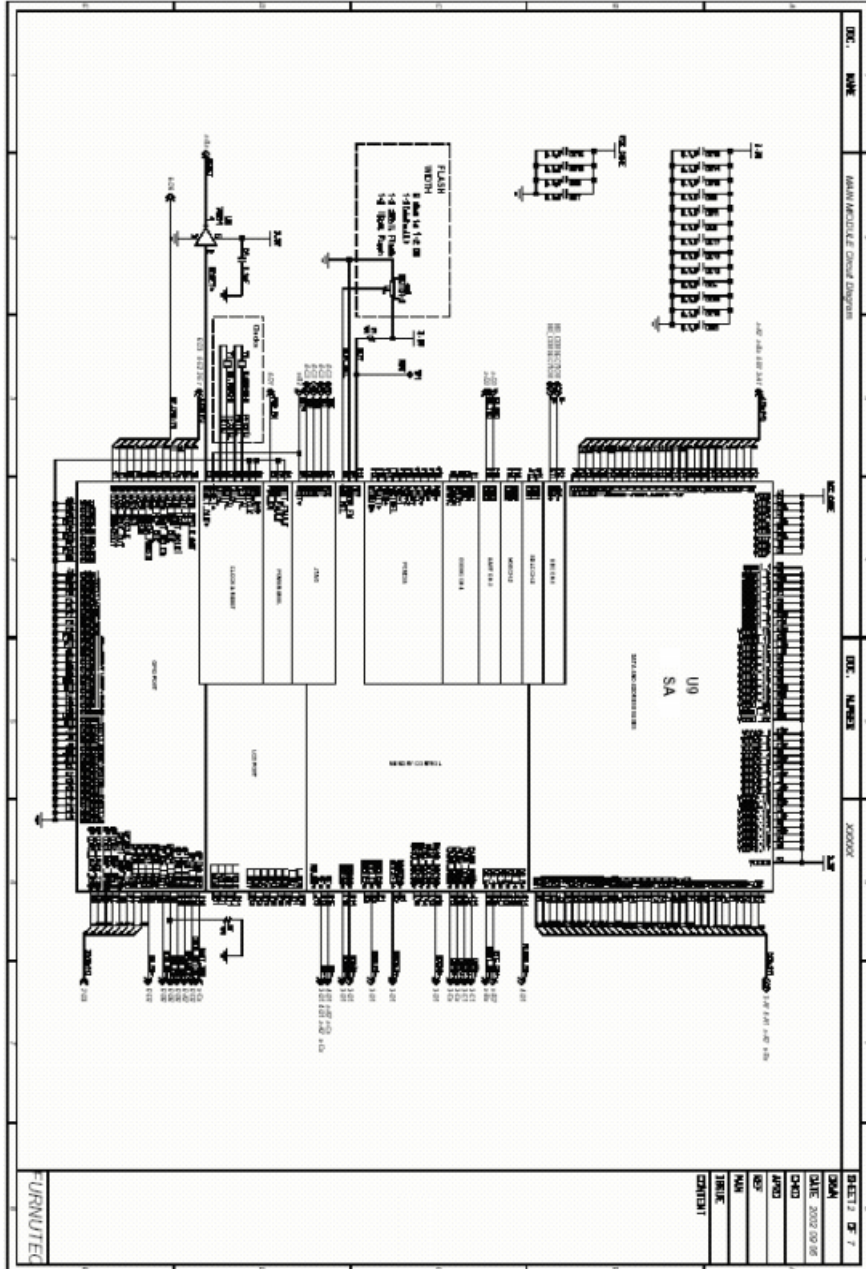
- [1] 이상배, L.C.Jain, R.K.Jain “하이브리드 지능 시스템”, 도서출판 그린, pp.1-13, 1997.
- [2] 이상배, “퍼지-뉴로제어 시스템”, 교학사, pp.111-149, 1999.
- [3] 장동혁, “Visual C++를 이용한 디지털 영상 처리의 구현”, PC어드밴스, 1999.
- [4] 이문호, “Visual C++ 실용 영상 신호처리”, 대영사, 2001.
- [5] L.C.Jain, U.Halici, I.Hayashi, S.B.Lee, S.Tsutsui, “Intelligent Biometric Techniques in Fingerprint and Face Recognition”, CRC Press.
- [6] 李柱尙, “A Study on the Fingerprint Recognition Method Direction using Neural Networks”, 韓國海洋大學校 碩士學位論文, 2001.
- [7] 변오성, 이철희, 문성룡, 임기영, “뉴로-퍼지를 이용한 영상 필터 연구”, 한국퍼지 및 지능 시스템 학회 논문집, 2001.
- [8] 김 현, “RSTI불변 지문 특징량 추출 및 인식과 응용”, 인하대학교 석사 학위논문, 1997.
- [9] 조성원, 김재민 “A Study on the Fingerprint Recognition Preprocessing using adaptive binary method”, 한국퍼지 및 지능 시스템 학회 논문집, Vol.12, No.3, pp.227-230, 2002.
- [10] 이재현, “人工知能 技法을 利用한 指紋 認識 알고리즘 및 시스템 具現에 관한 研究”, 한국해양대학교 박사학위 논문, 2001.
- [11] Joon Ho Hwang, “Design of Authentication Protocols using Biometrics and Implementation of Efficient Fingerprint Identification System”, Pohang University of Science and Technology.
- [12] 신재룡, 이석희, 곽윤식, 유재수, 조기형, “지문영상의 윤곽개선을 위한 신경망 알고리즘”, 정보통신학회지, Vol.6, No.2.
- [13] <http://www.arm.com>, “ARM Architecture”.
- [14] AIJI System, “ARM7 Application Course”, 2002.
- [15] <http://www.intel.com>, “Intel StrongARM Microprocessor Development Manual”.
- [16] 김인기, “접촉발광소자를 이용한 지문인식에 관한 연구”, 선문대학교 석사학위논문, 1999.
- [17] 김승환, “지문인식 시스템에서 본인여부 판정과 오인식 확률 추정”, 1998.

[18] 현종은, “지문인식을 이용한 보안 Cyber 연구 및 구현”, 연세대학교 석사학위  
논문, 2000.

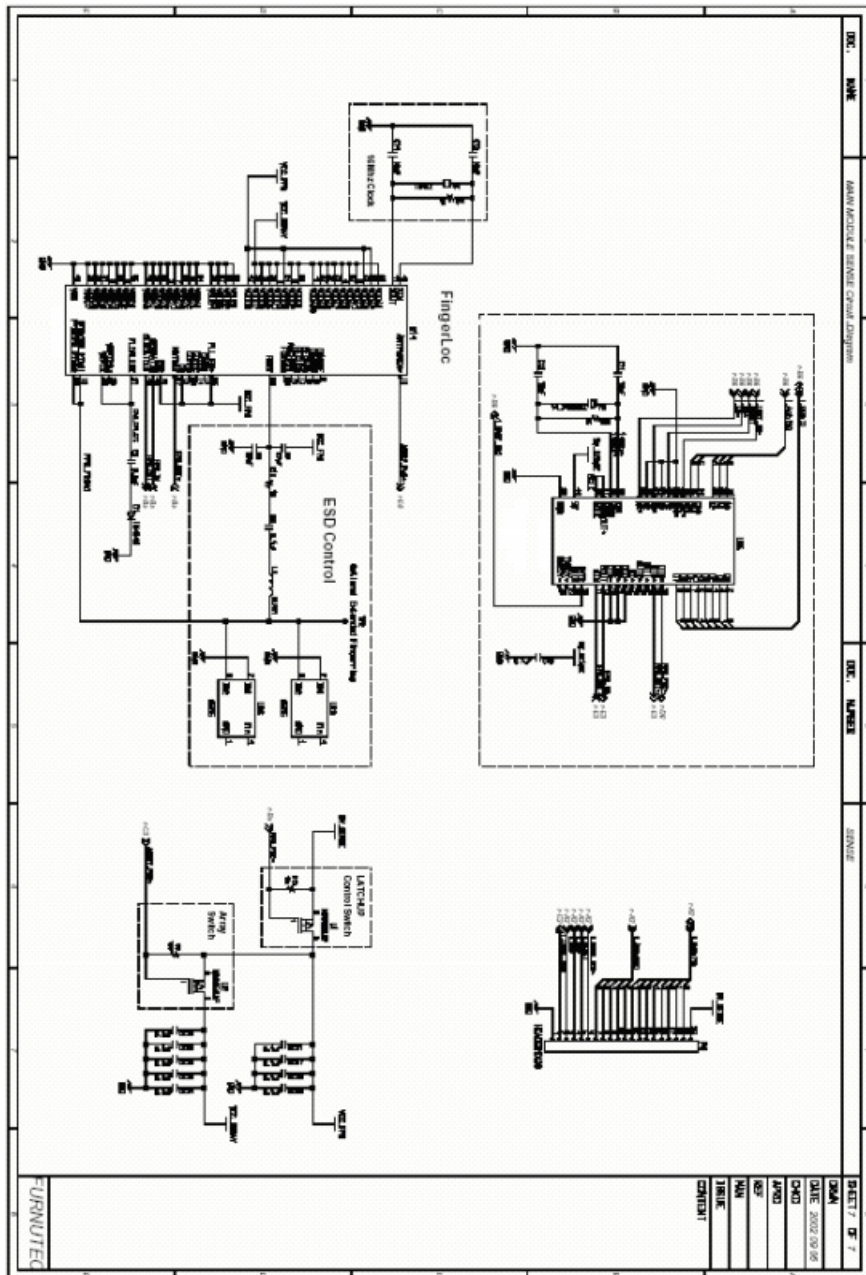
[19] GONZALEZ & WOODS, "Digital Image Processing", Addison Weley  
Longman, 1998.

# 부록(APPENDIX)

## A. 메인 회로도

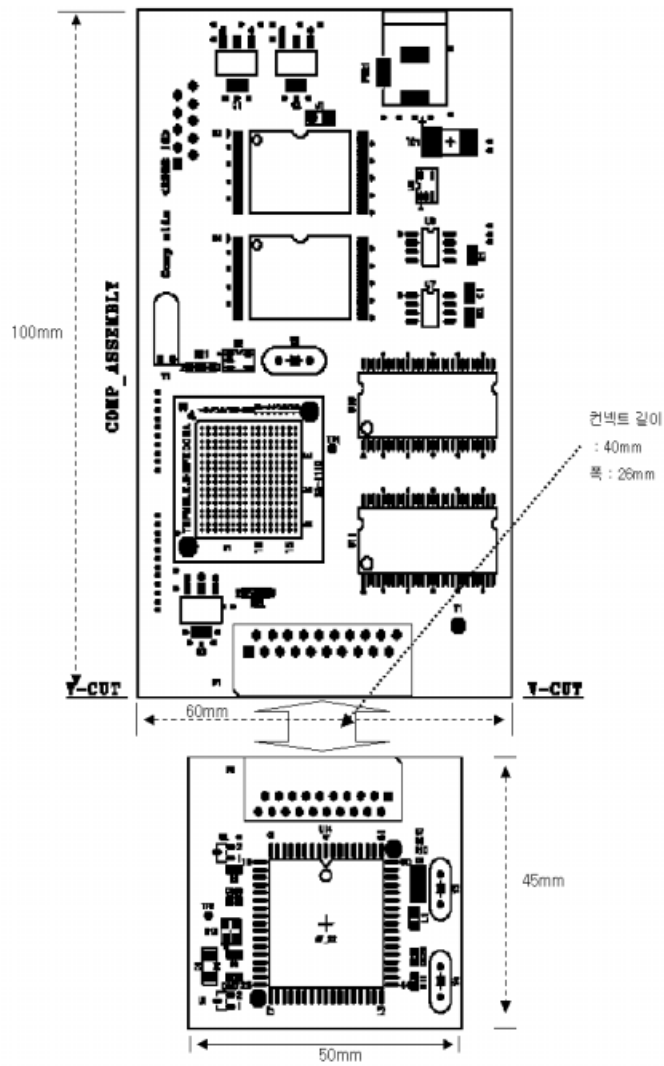


B. 센서 회로도





C. 지문인식 신호처리보드의 실제 크기





## 감사의 글

대학원 생활 2년이 어떻게 지나갔는지 모를 정도로 세월이 무척이나 빠르게 흘러 인생의 반이 지나가고 있는 지금, 뒤돌아보면 후회되고, 아쉬운 일이 무척이나 많습니다. 끝이 아닌 시작이라는 마음으로 절 아는 모든 분들에게 깊은 고마움을 전합니다.

먼저, 논문이 나오기까지 깊은 애정과 관심을 가져주신 지도교수님이신 이상배 교수님께 감사의 말을 전합니다. 그리고 바쁘신 와중에도 논문 심사를 위해 조언과 충고를 해주신 박동국 교수님, 임재홍 교수님께 깊은 감사를 드립니다. 멀리 계시지만, 항상 좋은 말씀을 아끼지 않으신 연구실 선배님이신 탁한호 교수님과 김일 교수님, 그리고 졸업한 여러 선배님들께도 깊은 감사를 드립니다.

2년 동안 생활해 오면서, 팀이란 조직 때문에 힘든 일을 책임지고 할 수밖에 없었던 재현선배, 삭막한 연구실 생활 속에서도 늘 밝은 모습을 보여준 지금은 애기 아빠인 주상선배, 커피 맨 석민 선배, 종윤씨, 멀리서 자기 일처럼 도움을 아끼지 않은 태화, 그리고 학부 때부터 대학원 연구실까지 같이 생활을 한 영탁이와 창규, 연구실 동기들, 후배들께도 고맙다는 말을 전합니다.

짧은 지식이나마 도움을 줄 수 있었던 동우, 우영, 정인, 근녕, 길영이 에게도 고맙다는 말을 전합니다.

힘들 때 항상 옆에서 도움을 준 고향에 있는 동수, 창순, 정호, 태익 우리 5인방 친구들에게 고마움을 전합니다.

마지막으로, 자나깨나 못난 아들 걱정만 하시다가 하늘 나라로 떠나신 보고싶은 어머니, 항상 버팀목이 되어주시는 아버지께 이 논문을 바칩니다.

- 바다가 보이는 아치섬에서 -

지금 생각하면 그때는 무척 슬펐습니다. 그러나 그때의 슬픔을 통해 눈물의 의미와 함께 기쁨의 소중함도 알게 되었으니 이제 나는 지나간 슬픔의 시간에 감사할 수 있습니다.

지금 생각하면 그때 나는 너무 외로웠습니다. 그러나 그때의 외로움을 통해 사람의 귀함과 함께 살아가는 삶의 아름다움을 알게 되었으니 이제 나는 지나간 외로운 시간에 감사할 수 있습니다.

지금 생각하면 그때는 너무 힘들었습니다. 그러나 그때의 어려움을 통해 나는 인내와 성실의 가치를 알았으며 지금은 어떤 어려움이 닥쳐도 두려움 없이 마주치게 되었으니 이제 나는 지나간 힘들었던 시간에 감사할 수 있습니다.

지금 생각하면 그때는 크게 낙심하였습니다. 그러나 그때의 그 절망을 통해 소생의 기쁨과 희망의 불빛을 항상 내안에 간직하게 되었으니 이제 나는 그 좌절의 시기를 감사할 수 있습니다.

지금 생각하면 그때는 참 어리석었습니다. 그러나 그때의 그 어리석음을 통해 지혜의 소중함과 배움의 가치를 알게 되었으니 이제 나는 그 어리석었던 시간을 소중히 간직하며 감사할 수 있습니다.

지금 생각하면 그때는 너무 많은 실수를 저질렀습니다. 그러나 그때의 그 실수를 통해 좀더 신중하고 현명해 질 수 있었으니 이제 나는 그 실수들에 감사할 수 있습니다.