工學碩士 學位論文

ART2 適用 임베디드 音聲認識 시스템의 設計 및 具現에 관한 硏究

A study on the Design and Implementation of
ART2 Application Embdded Speech Recognition System

指導敎授　李　尙　培

2004年　2月

韓國海洋大學校 大學院

電子通信工學科　柳 洪 錫

工學碩士 學位論文

ART2 適用 임베디드 音聲認識 시스템의 設計 및 具現에 관한 硏究

A study on the Design and Implementation of
ART2 Application Embedded Speech Recognition System

指導敎授 李 尙 培

2004年 2月

韓國海洋大學校 大學院

電子通信工學科 柳 洪 錫

本 論文을 柳洪錫의

工學碩士 學位論文으로 認准함.

委 員 長　　朴 東 國　　印

委　　員　　林 宰 弘　　印

委　　員　　李 尙 培　　印

2004年 2月

韓國海洋大學校 大學院

電子通信工學科

목 차

Abstract

Speech recognition called machine receives human's language and achieves suitable action according to this language. This technology is used on industry whole, and is specially applied to information industry field, digital communication, electronic, multi media etc. Apply to mobile robot that is electric motion wheel chair system at LAB through this technology. So developed to give more convenience for hand and feet uncomfortable disabled person.

In this study, consider that the plant is electric motion wheel chair. So during speech recognition, is used DTW(Dynamic Time Warping) that relative correct recognition rate is fine being speaker dependent type. But consider that real-time have to get small memory and fast processing speed. So introduced VQ(Vector Quantization) used in data compression algorithm of speaker independent. In accordance with, secure fast recognition and small memory. But discovered that recognition rate is fallen by using VQ. So, in after treatment algorithm for correct recognition rate enhancement ART2(Adaptive Reason Theory 2) algorithm application on about 5% correct recognition rate enhancement bring. To use ART2, must be applied error range. Er ror range is applied result that extract 1 order distance in 2 order distance is more than 20 by each distance to apply DTW. Like this, bring fast processing and high correct recognition rate apply ART2.

Because it is moved object, must implement by embedded system. So It choose chip of TMS320C32 that processing a lot of computation

complexity relatively fast and implement embedded system. Memory can store a lot of data in speech considering, therefore possessed 128kbyte's RAM memory and 64kbyte's ROM memory. Input of speech use 16bits stereo audio codec, secure relative correct data through high resolution. The mobile robot uses 80C196KC. and output PWM generating power through HSO that chip had and designed to run motor.

제 1 장 서 론

인간이 기계랑 의사소통을 한다. 이것은 산업의 발달과 더불어 대두되어진 HCI(Human Computer Interface)의 한 형태이다. 여기에는 각종 생체인식, 즉 얼굴인식, 지문인식, 홍채인식 등 여러 가지 수단이 이용되고 있으며, 그 중 한 방법이 음성을 이용한 인식이다. 음성 인식 기술은 기계가 인간의 언어를 받아들여, 그 언어에 맞게 적절한 행동을 수행하는 것을 말한다. 이런 기술은 각종 산업 전반에 사용되어지고 있으며, 특히 정보 산업 분야, 디지털 통신 분야, 가전분야, 멀티미디어 등에 적용되어지고 있다.

음성인식 방법으로는 패턴의 매칭을 거리에 의해서 인식하는 DTW (Dynamic Time Warping)[1],~,[4]와 통계적으로 인식하는 방법인 HMM(Hidden Markov Model)[1],[5],[6], 뇌의 구조를 모델링한 방법인 NN(Neural Network)[1],[7] 등이 있으며, 인식 방법에 따라 한사람에게 적용되는 화자종속형[1]과 여러 사람에게 적용되는 화자독립형[1]으로 나눈다. 본 논문에서는 이동로봇(전동휠체어) 시스템에 음성인식 기술을 사용하여 손발이 불편한 장애인에게 좀더 편리성을 주고자 이 시스템을 개발하게 되었다. 그리고 전동휠체어가 장애인들에게 적용된다는 것을 감안해서 화자종속형으로 인식률이 비교적 좋은 DTW를 전반부 인식 알고리즘으로 사용하고 있고, 여기에 더욱 더 높은 인식을 만들기 위해 이전에 학습되어진 패턴을 유지하면서 새로운 패턴을 학습하는데 필요한 유연성을 잃지 않도록 설계되어지는 자율 신경망으로 마치 분류기처럼 사용되어 지는 ART2(Adaptive Reason Theory 2)[8] 알고리즘을 후반부 인식 알고리즘으로 적용했다.

본 논문의 구성은 다음과 같다. 2장에서는 음성의 잡음을 제거하는 방

법인 스펙트럼 차감법과 실음성 구간만을 검출하는 절대 에너지 방식, 그리고 많은 데이터를 보유한 음성을 압축하는 방법 중에 하나인 MFCC (Mel Cepstrum)와 VQ(Vector Quantization)[5],[9]에 대해서, 3장에서는 인식알고리즘인 DTW와 ART2에 대해서 서술한다. 4장에서는 이 알고리즘을 토대로 만든 임베디드형 음성인식보드와 그리고 소프트웨어적으로 어떻게 처리되어지는지, 또 보드를 적용할 플랜트인 이동로봇의 구성에 대한 내용을 서술한다. 5장에서는 이것을 바탕으로 실제 실험과 결과에 대해서 설명을 하고 6장에서 결론을 맺도록 하겠다.

제 2 장 음성 신호 처리

2.1 잡음 처리

음성의 인식률은 잡음이 비교적 없는 환경에서는 높은 성능을 나타낸다. 하지만 우리가 실제 생활하는 환경은 잡음으로 혼재되어 있기 때문에, 성능 저하를 나타낸다. 이러한 문제를 해결하기 위해서는 잡음처리를 해야 된다. 여기에는 여러 가지 잡음 처리 방법이 있다. 그 대표적인 몇 가지 예를 들어 보면 스펙트럼 차감법, 위너 필터링(Winner Filtering), 캡스트럼 정규화법(Cepstrum Normalized Method), RASTA 필터링 방법 등이 있다. 본 논문에서는 잡음이 포함된 음성신호 중 실음성만을 추출하는 방법인 스펙트럼 차감법에 대해 살펴본다[9],[10].

스펙트럼 차감법(Spectral Subtraction Method)은 정적인 배경 잡음이 음성에 첨가되어 있고, 잡음이 새로운 상태로 바뀔 때는 새롭게 배경 잡음의 스펙트럼 크기를 추정할 충분한 시간이 필요하다. 그러나 서서히 변화는 비정적인 잡음인 경우는 잡음 구간 알고리즘 검출이 필요하다. 잡음의 제거는 스펙트럼의 크기만으로 제거한다. 잡음이 섞인 음성신호는 다음과 같다.

$$y(k) = s(k) + d(k) \tag{2.1}$$

여기서 $y(k)$는 잡은 섞인 음성신호, $s(k)$는 음성신호, $d(k)$는 잡음 신호를 나타낸다. 위의 식 (2.1)을 주파수 영역으로 변환하면

$$Y(w) = S(w) + D(w) \tag{2.2}$$

여기서는 식 (2.2)를 다음과 같이 정의 한다.

$$Y(w) = \sum_{k=0}^{N-1} y(k) e^{-jwk}$$

$$y(k) = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(w) e^{jwk} dw$$

식 (2.2)를 이용하여 SPE(Spectral Power Estimation) 방법을 적용하면

$$|Y_m(w)|^2 = |S_m(w)|^2 + |D_m(w)|^2 + 2S_m(w)D_m(w) \tag{2.3}$$

$$\approx \{S_m(w)|^2 + |D_m(w)|^2 \tag{2.4}$$

이 된다. 여기서

$$2S_m(w)D_m(w) = 0$$

이라고 하면, 식 (2.4)는 아래와 같이 나타낼 수 있다.

$$|S_m(w)|^2 \approx |Y_m(w)|^2 - P_d(w) \tag{2.5}$$

$$P_d(w) = |D_m(w)|^2$$

위상에 대한 정보를 나타낸 식은 다음과 같다.

$$S_m(w) = |S_m(w)| \angle Y_m(w) => S_m(n) = F^{-1}\{S(w)\} \tag{2.6}$$

스펙트럼 차감 필터(Spectral Subtraction Filter)의 $H(w)$는 미리 구한 잡음의 스펙트럼 $D_m(w)$의 평균을 사용한다. 그림 2.1은 스펙트럼 차감법에 대한 블록도이다.

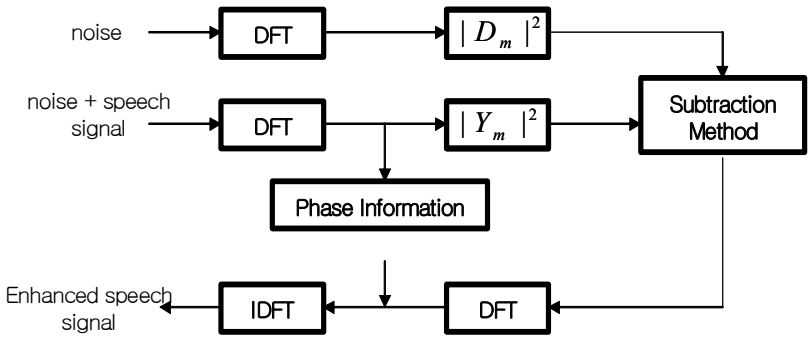


그림 2.1 스펙트럼 차감법

Figure 2.1 Spectrum subtraction method

2.2 음성 데이터 획득

음성의 검출은 음성인식 성능에 큰 영향을 미친다. 이것은 마이크를 통해서 들어온 음성데이터로부터 실제 음성부분만 검출하는 것이다. 여기에는 검출점으로부터 영교차율을 측정해서 교차율이 높으면 유성음을 교차율이 낮으면 무성음으로 판별하는 방식인 영교차율(ZCR : Zero Crossing Rate) 방식과 음성 신호의 구간 당 에너지를 계산해서 일차적으로 유성음 부분만 검출하는 절대 에너지 방식(Short Time Energy)이 있다[1],[9],[10].

본 논문은 실시간으로 음성을 처리하기 때문에 실시간에서 비교적 성능이 떨어지는 영교차율을 적용하지 않았고, 절대에너지 방식을 사용하고 있다. 절대 에너지는 무성음보다 유성음 부분이 크다는 이론을 바탕으로

하고 있다. 식 (2.7)을 통해서 음성 감지를 하고 있으며, 유성음과 무성음을 구별한다.

$$E_i = \sum_{n=0}^{N-1} [S_i(n)]^2 \tag{2.7}$$

본 논문에서는 획득한 음성 데이터를 다음과 같이 처리하고 있다. 프레임 길이를 256으로 설정하고, 데이터의 손실을 방지하기 위해서 프레임을 중복 시키는 방식인 프레임 블록킹을 80으로 적용하고 있다.

2.3 음성데이터의 특징 추출

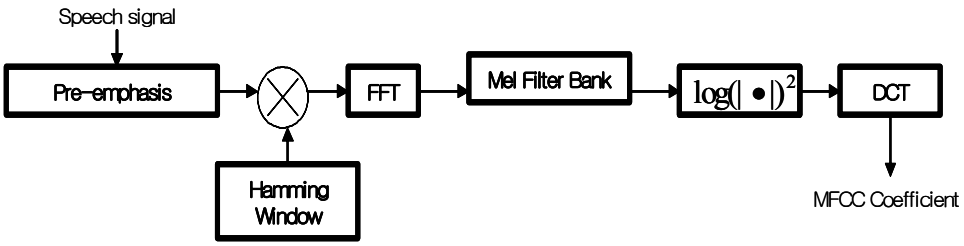


그림 2.2 MFCC의 처리절차

Figure 2.2 Produce of MFCC process

음성의 데이터양은 상당히 크다. 이 큰 데이터양은 알고리즘을 수행하기 위해서는 부적합하므로 음성 데이터를 효율적으로 줄여야 된다. 그렇게 하기 위해서 특징을 추출하는 과정이 필요하다. 특징을 추출하는 알고리즘은 여러 가지가 있다. 그 중 대표적인 두 방식이 있는데 그 중하나는

LPC(Linear Predictive Code) & Cepstrum 방식이고, 또 다른 방식은 MFCC(Mel-Frequency Cepstral Coefficient) 방식이다. 본 논문에서는 MFCC 방식을 사용하고 있다[1],[9]. MFCC는 그림 2.2와 같은 절차를 따르고 있다.

2.3.1 프리엠파시스

프리엠파시스(Pre-emphasis)는 음성의 저주파 성분을 약화시키고 고주파 성분만을 강조시켜 음성신호의 DC성분을 제거하는 방식이다. 여기서 DC성분은 주로 마이크에서 많이 발생한다. 그 방식은 다음과 같다[1],[3],[9].

$$\overline{S}(n) = S(n) - aS(n-1), \quad 0.9 < a < 1.0 \tag{2.8}$$

본 연구에서는 a의 계수를 0.95로 사용하고 있다.

프리엠파시스 처리된 신호 $\overline{S}(n)$을 N개의 샘플로 구성된 프레임들로 분할 한 후 인접한 프레임과 중첩시키는 프레임 블록킹(Frame Blocking) 방식을 취한다. 본 연구에서 한 프레임의 크기는 256이며, 중첩 프레임은 80으로 하고 있다.

2.3.2 해밍 윈도우

프리엠파시스와 프레임 블록킹을 통해서 프레임별로 잘라내고 중첩되어진 음성데이터는 각 프레임의 시작과 끝에서 신호 불연속을 최소화 시켜야 된다. 그래서 본 논문에서는 양 끝단 부분에 비교적 노이즈가 적은 해밍윈도우를 사용하고 있다. 해밍윈도우(Hamming Window)는 다음과 같

이 표현된다[9].

$$w(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right) \quad (0 < n < N-1) \tag{2.9}$$

2.3.3 MFCC

잡음에 강한 특징 벡터로는 멜 캡스럼과 루스 캡스트럼 계수가 있다. 기존의 연구 결과에 의하면 멜 캡스트럼이 가장 좋은 인식률을 나타내는 것을 보였다. 본 논문에서도 멜 캡스트럼을 사용하고 있다. 멜 캡스트럼은 인간의 청각 특성을 이용한 것으로, 멜(Mel) 이라는 단위를 사용하고 있다. 멜은 톤 신호의 인지된 피치 또는 주파수 측정치를 나타내는 단위이다[1],[5],[9]. 멜 캡스트럼의 원리는 인간의 청각 시스템이 피치를 선형적으로 인지하지 못하는 것처럼 톤 신호의 물리적인 주파수에 선형적으로 대응하지 않는다는 이론을 바탕으로 하고 있다. 그래서 Stevens와 Volkman은 1000Hz에 대해서 1000mel로 선정하고, 인지된 피치가 기준 주파수에 두 배가 되도록 2000mel로 표기했다. 이 대응 관계를 통해서 실제 물리적인 주파수와 인지된 주파수 사이에 관계를 구했다. 그 관계는 다음과 같다. 1KHz 이하에서는 선형적으로 1KHz 이상에서는 대수적 즉 로그스케일(Log Scale)로 대응되는 관계를 사용하고 있다. 멜과 주파수사이의 대응 관계는 다음 식과 같이 근사적으로 표현할 수 있다.

$$F_{mel} = 2595 \ \log_{10}(1 + \frac{F_{Hz}}{700}) \tag{2.10}$$

여기서 F_{mel}는 각각 근사식에 의해 구해진 인지된 주파수이고, F_{Hz}는

실제 주파수를 나타낸다. 멜 캡스트럼은 다음과 같은 절차로 수행한다. DFT(Discrete Fourier Transform) 또는 FFT(Fast Fourier Transform) 크기를 멜과 주파수의 대응관계에 따라 주파수 축에서 와핑(Warping)하여 이의 대수 값을 역 DFT/FFT하여 8에서 14차 사이에서 계수를 구한다. 그림 2.3에서 나타난 20개의 삼각 대역 통과 필터를 이용하여 임계 대역 필터를 통과한 로그 에너지 출력을 X_k라 하면 M개의 캡스트럼 계수는 다음 식으로 표현된다.

$$C_n = \frac{1}{20}\sum_{k=1}^{20} X_k \cos\left[\mathrm{n}\left(\mathrm{k}-\frac{1}{2}\right)\frac{\pi}{20}\right] \tag{2.11}$$

여기서 C_0는 음성 프레임의 평균 에너지이며, 초기 값이다.

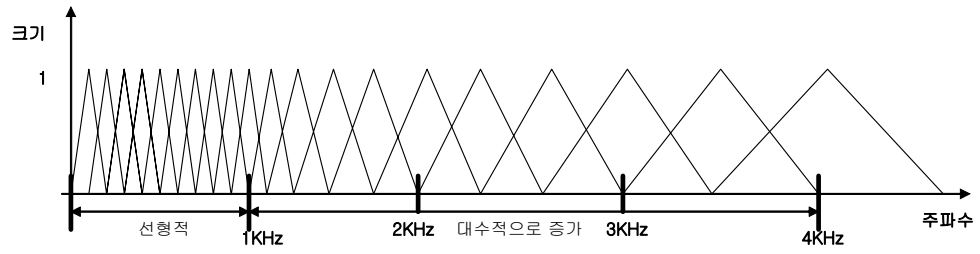


그림 2.3 멜 캡스트럼 대역(삼각 대역 필터)

Figure 2.3 Bandwidth of Mel-Cepstrum(Bandwidth of triangle filter)

최종적으로 MFCC의 절차를 조합해보면 프리엠파시스를 통해서 고주파 성분이 강화된 음성은 해밍 윈도우를 씌워서 신호의 불연속 일치를 약화시킨다. 그리고 FFT를 사용해서 주파수 영역으로 바꾼다. 시간영역에서

주파수 영역으로 바뀐 음성 데이터는 20개의 로그형 mel-filter banks를 거친다. 이 때 사용되어진 필터는 그림 2.3과 같은 필터를 사용한다. 필터를 거친 음성은 로그화를 통하고, DCT(Distance Cosine Transform)를 통해서 한 프레임 당 12개의 계수 값을 생성하게 된다. 이 때 DCT는 식(2.11)에 나타나 있다.

2.4 벡터 양자화

벡터 양자화(Vector Quantization)는 입력 값의 차원이 너무 크거나 그 값의 범위가 매우 큰 경우, 대표 패턴이 저장된 코드북으로부터 이에 대응되는 양자화 값으로 차원 수를 줄이거나 범위를 줄이는 방법이다. 본 논문에서는 MFCC 계수를 이용하여 양자화 테이블을 생성한다. 양자화 테이블의 생성에는 K-means 알고리즘과 이진 트리 알고리즘이 사용된다. 본 논문에서는 이진 트리 알고리즘을 사용하고 있다[11],[12].

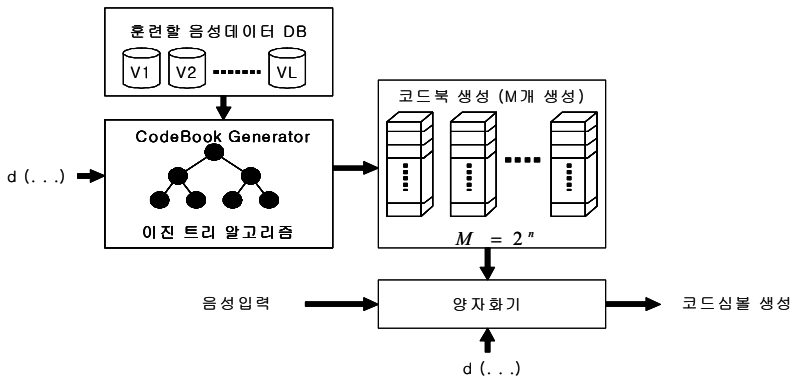


그림 2.4 벡터양자화의 처리

Figure 2.4 Processing of vector quantization

먼저 벡터 양자화의 계통도는 그림 2.4와 같다. 본 논문에서는 벡터 양자화의 크기인 M은 512로 설정 하였으며, 에러비율은 0.01, 분할 파라미터(e : Splitting Parameter)값은 0.001로 설정했다. 벡터 양자화의 프로그램 순서는 3가지로 구분한다. 특징추출이 저장되어 있는 수십-수백개의 파일을 한 개의 파일로 만드는 과정인 Vector-gather와 훈련할 데이터를 이용해서 코드북 크기(M=256-1024)에 따라 벡터 양자화 코드북을 생성하는 Clustering the Training Vector 과정, 그리고 입력되는 음성신호로부터 추출한 계수와의 거리가 가장 가까운 양자화 테이블의 심볼을 생성하는 Vector Classification으로 분류한다.

우선 Vector Gather과정은 MFCC를 통해서 나온 여러 개의 데이터를 하나로 묶어서 처리한다. MFCC를 통해서 묶여진 하나의 데이터 덩어리는 Clustering the Training Vector에 따라 분할한다. 이때 사용되어진 알고리즘이 이진 트리 알고리즘이다. 이진 트리 알고리즘은 1개의 벡터 코드북으로부터 분할을 이용하여 M개의 코드북을 작성하는 방법이다. 그 순서는 다음과 같다.

단계 1) 전체 학습 자료의 중심 벡터를 구한다. 즉 전체 벡터들을 차수별로 합산 및 평균을 구해서 새로 생성된 코드북에 넣어 1개의 벡터 코드북을 생성한다.

단계 2) 이전 코드북의 크기가 두 배가 되도록 분할한다. 분할하는 공식은 아래 식과 같다.

$$Y_n = Y_n(1+e) \tag{2.12}$$

$$Y_n = Y_n(1-e) \tag{2.13}$$

e : splitting parameter ($0.01 < e < 0.05$), Y_n : 벡터

단계 3) 학습 자료들을 가장 가까운 코드에 속하는 부공간으로 분류한다. 다시말하면, 훈련할 벡터들을 분할된 코드북에 가까운 곳을 찾아 그곳에 소속시키고, 소속된 벡터들끼리 다시 중심 벡터(Centroid)를 만들어 최상의 코드북을 생성한다. 여기서는 양자화 테이블을 생성하는 알고리즘 중 하나인 K-means 알고리즘을 이용하여 중심 벡터를 구한다.

$$C_n^{(i+1)} = \frac{\Sigma Q(X_m) = C_n^{(i)} X}{\Sigma Q(X_m) = C_n^{(i)} 1} \tag{2.14}$$

X_m : 훈련할 데이터, $Q(X_m)$: 현재 소속 코드북

단계 4) 코드북과 벡터 사이의 거리들을 합산시켜 평균화한 값인 왜곡(Distortion)을 구한다.

$$D_{ave}^{(i)} = \frac{1}{M_k} \sum_{m=1}^{M} \| X_m - Q(X_m) \|^2 \tag{2.15}$$

$D_{ave}^{(i)}$: 평균 왜곡 X_m : 훈련할 데이터

$Q(X_m)$: 현재 소속된 코드북, M_k : 소속된 코드북 총 수

단계 5) 현재의 전체 왜곡이 이전의 전체 왜곡과 차이가 허용범위보다 작은 경우 종료하고, 아니면 단계 3)부터 과정을 되풀이한다. 그러나 원하는 코드북보다 작으면 단계 2)로 가고 그렇지 않으면 끝낸다. 다시 말하면 본 논문에서는 코드북의 크기 M이 512에 도달할 때까지 단계 2)에서 단계 4)를 반복 수행한다는 말이다. 훈련 벡터의 분류는 그림 2.5와 같다.

위의 단계를 거쳐서 분류된 벡터들은 512개의 코드북을 생성한다. 생성된 코드북은 새로운 입력벡터들과의 사이에 최적의 코드북 심볼(index)을 출력한다. 최적의 코드북 심볼은 식 (2.16)과 같이 표현한다.

$$M^* = arg\ \min_{l<m<M} d(v, y_m) \tag{2.16}$$

M^* : 최상의 벡터 인덱스 값 행렬

Y_m : M차원 벡터 코드북 $(1 \leqq m \leqq M)$

v : 특징추출된 벡터 입력 값

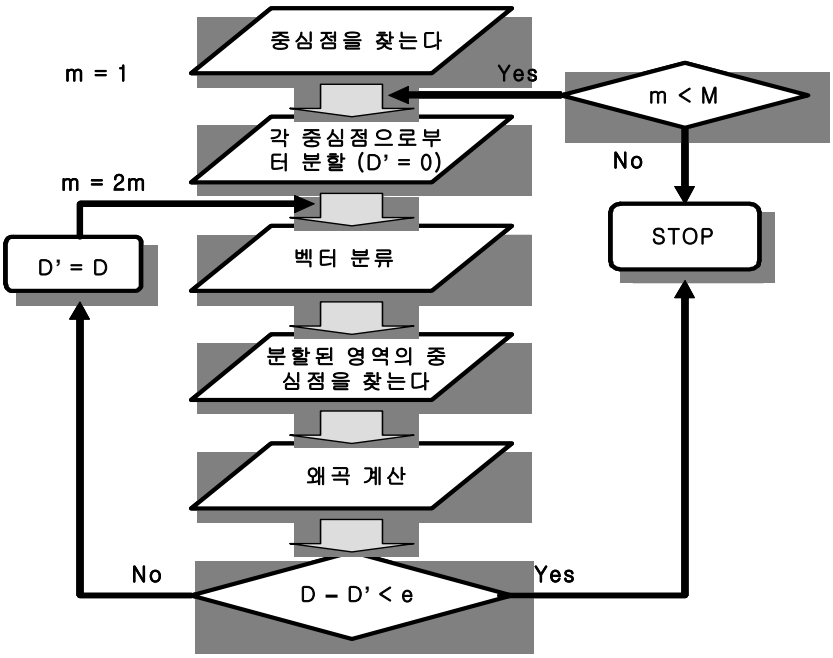


그림 2.5 이진 트리 알고리즘의 순서도

Figure 2.5 Procedure of Binary Tree Algorithm

제 3 장 음성인식 알고리즘

3.1 ART2 알고리즘

ART(Adaptive Reason Theory)는 G. Carpenter와 S. Grossberg에 의해 개발된 자율 신경망으로서, 신경망에서는 학습되지 않은 전혀 새로운 형태의 패턴이 들어오는 경우 이전에 학습한 유사한 패턴으로 분류해 버리는 문제점이 발생할 수 있다. 이러한 문제점을 해결하기 위해서 학습되지 않은 전혀 새로운 패턴이 들어오면 새로운 클러스트를 형성하면서 기존 패턴에 영향을 주지 않는 신경망을 말한다[8],[12],~,[15].

ART는 이진 입력패턴을 처리하는 ART1과 연속적인 패턴을 처리할 수 있는 ART2가 있다. 본 논문에서는 ART2를 사용하고 있다. ART2를 설명하기 전에 ART에서 가장 간단한 ART1에 대해서 간략히 설명하겠다.

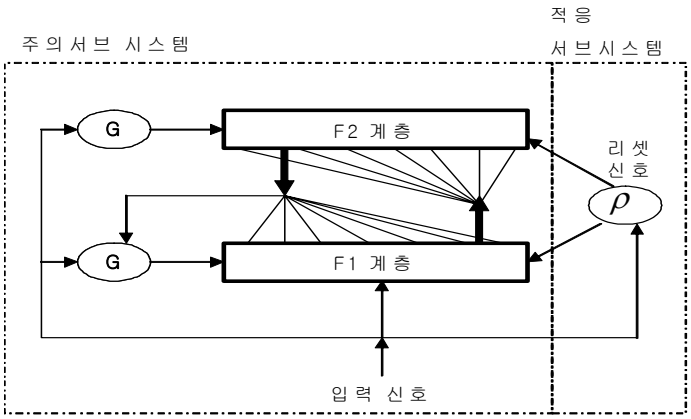


그림 3.1 ART1 구조

Figure 3.1 Architecture of ART1

그림 3.1을 보면 ART1은 다음과 같이 설명된다. ART1은 주의 서브시스템(Attentional Subsystem)과 적응 서브시스템(Orienting Subsystem)으로 구성된다. 주의 서브시스템은 입력신호를 받아 정규화 하는 F1층과 입력 패턴들에 의해 생성된 클러스트들을 저장하는 F2층으로 나누어진다. 각층 사이에는 상향 연결가중치(Bottom-Up) 벡터와 하향 연결가중치(Top- Down) 벡터로 연결되어 있고, 연결 가중치를 조절해서 새로운 패턴에 대한 학습을 수행한다. 적응 서브시스템은 F1층과 F2층의 매칭 실패시 F2층의 활성화를 억제 시킨다.

3.1.1 ART2의 구조

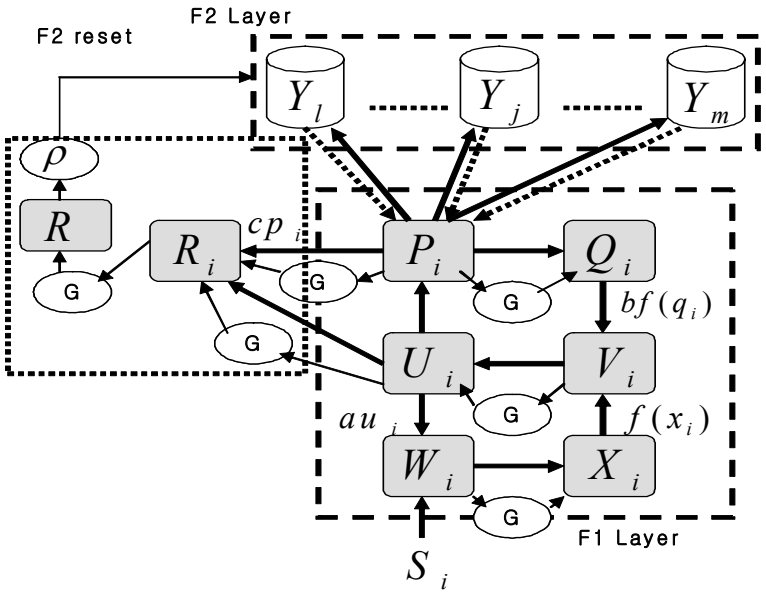


그림 3.2 ART2 구조

Figure 3.2 Architecture of ART2

ART2는 ART1과 비슷한 구조를 가지고 있다. 하지만 ART1에서 처리할 수 있는 이진 입력패턴 뿐만 아니라 아날로그 입력 패턴에 대해서도 학습이 가능하다. 그러나 아날로그 입력 패턴을 처리하기 위해서는 F1층을 여러 개의 서브계층(W, X, U, V, P, Q 유닛)으로 나누어서 피드백(Feedback)과 피드포워드(Feedforward) 처리를 해야 되기 때문에 그 구조는 ART1보다 훨씬 복잡하다. ART2의 기본 구조는 그림 3.2와 같다. ART2는 하부계층(Sub Layer)과 이득제어(Gain Control)부분으로 이루어진 것을 볼 수 있다. F2층에서의 처리는 ART1과 유사하다. 각 층은 다음과 같이 처리된다. 먼저 F1층의 각 구성은 다음 표 3.1과 같이 구성된다.

표 3.1 F1층의 구성

Table 3.1 Configuration of F1 layer

유닛	구성	
W	$w_i = s_i + au_i$	(3.1)
X	$x_i = \frac{w_i}{e + \parallel w \parallel}$	(3.2)
V	$v_i = f(x_i) + bf(q_i)$	(3.3)
U	$u_i = \frac{v_i}{e + \parallel v \parallel}$	(3.4)
P	$p_i = u_i + \sum_j g(y_i) z_{ij}$	(3.5)
Q	$q_i = \frac{p_i}{e + \parallel p \parallel}$	(3.6)
- i,j는 각 하부 계층의 노드를 의미, a, b, e는 상수 - $f(x)$는 F1층의 잡음제거를 위한 임계함수 $f(x) = \begin{cases} 0 & 0 \leqq x \leqq \theta \\ x & x > 0 \end{cases}$		(3.7)

F1층의 각 노드는 새로운 입력 벡터 S_i에 의해 표 1에 의해 초기화 된다. 따라서 새로운 입력 벡터가 네트워크에 투입되면 이전 벡터에 대한 기억이 사라진다. 때문에 F1층을 짧은 시간 메모리(Short Term Memory)라고 한다. F2층에서의 처리는 각각의 노드에 대해 다음 식 (3.8)에 의해 입력 값을 계산한다.

$$y_j = \sum_{i=1}^{n} p_i z_{ji} \tag{3.8}$$

z_{ji} : 상향 연결 가중치, $j = 1$ .... m

F2층은 경쟁 계층이기 때문에 승자 노드만이 값을 출력하고 다른 노드들은 0을 출력한다.

$$g(y_i) = \begin{cases} d & y_j = max\{y_k\}\ \forall k \\ 0 & else \end{cases} \tag{3.9}$$

k는 F2층의 k번째 노드를 의미한다. 따라서 식 (3.5)의 하부 계층의 식은 F2층이 활동하지 않을 시는 $p_i = u_i$로 되고, 그렇지 않으면 $g(y_i) = u_i + dz_{ij}$ (z_{ij} : 하향 연결 가중치)가 되어 d에 따라 값이 갱신되는 것을 확인할 수 있다. 여기에서 F2층은 노드 중 최대 출력 값을 갖는 J를 선택한다. 그리고 리셋 여부를 확인하기 위해서 r을 계산한다. r은 다음 식에 따라서 계산한다.

$$r_i = \frac{u_i + cp_i}{\mid u \mid + c \mid p \mid} \tag{3.10}$$

$$\text{if} \quad |r| < \rho - e \;\; then \;\; y_J = -1 (reset = true)$$

$$\text{if} \quad |r| < \rho - e \;\; then \;\; (reset = false)$$

계산 결과, 리셋이 될 경우는 승자노드의 선택 및 리셋 체크를 반복하게 된다. 그렇지 않으면 노드 J의 가중치 벡터를 갱신하게 된다. 가중치 갱신은 다음 식에 의한다.

$$Z_{ji} = adu_i + \{1 + ad(d-1)\} Z_{ji} \tag{3.11}$$

$$Z_{iJ} = adu_i + \{1 + ad(d-1)\} Z_{iJ} \tag{3.12}$$

갱신된 가중치들은 표 3.1에 의해서 F1층 노드들의 값을 계산해 낸다. 이러한 학습 과정을 통해 노드 J는 입력 벡터들을 기억하게 된다. 이러한 기억은 새로운 입력 벡터가 투입되어도 F2층에 의해 지속적으로 유지된다. 이러한 이유로 F2층을 긴 시간 메모리(Long Term Memory) 라고 한다.

3.1.2 ART2 알고리즘

이 장은 실제 연구에서 ART2가 어떻게 사용되었는지에 대해서 설명하고 있다. ART2의 동작은 총 11단계로 구분된다.

단계 1) 파라미터들을 초기화 한다. -> 다음 범위에 따라 초기화 한다.

$a > 0,\ b > 0,\ 0 \leqq c \leqq 1,\ 0 \leqq d \leqq 1,$

$0 \leqq \theta \leqq 1,\ 0 \leqq \rho \leqq 1,\ e \ll 1,\ \dfrac{cd}{1-d} \leqq 1$

하향 연결 가중치 : $Z_{ij}(0) = 0$

상향 연결 가중치 : $Z_{ji}(0) \leqq \frac{1}{(1-d)\sqrt{M}}$

단계 2) 단계 3) ~ 단계 11)를 수행한다.

단계 3) 각 입력 벡터 s_i에 의해서, 단계 4) ~ 단계 10)을 수행한다.

단계 4) F1층 유닛의 활성화 및 재활성화

F1층 유닛의 활성화	F1층 유닛의 재활성화
$u_i = 0$	$u_i = \frac{s_i}{e + \parallel v \parallel}$
$x_i = \frac{s_i}{e + \mid s \mid}$	$x_i = \frac{w_i}{e + \parallel w \parallel}$
$w_i = s_i$	$w_i = s_i + au_i$
$q_i = 0$	$q = \frac{p}{e + \parallel p \parallel}$
$p_i = 0$	$p_i = u_i$
$v_i = f(x_i)$	$v_i = f(x_i) + bf(q_i)$

단계 5) F2층에서 각각의 노드에 대해 다음 식에 의해서 계산

$$y_j = \sum_{i=1}^{n} p_i z_{ji}$$

단계 6) 리셋이 설정되면 단계 7) ~ 단계 8)을 반복한다.

단계 7) F2층의 노드 중 최대 출력 값을 갖는 J를 선택한다. 그리고 리셋

여부를 확인하기 위해서 r을 계산한다.

단계 8) 리셋의 조건을 검사한다.

조건	$u_i = \frac{v_i}{e + \mid v \mid}$, $p_i = u_i + dz_{ij}$ -> $r_i = \frac{u_i + cp_i}{\mid u \mid + c \mid p \mid}$
	If $\| r \| < \rho - e$, *then* $y_J = -1$ (reset is true ; repeat step 5)
	If $\| r \| < \rho - e$, *then*
	$w_i = s_i + au_i$, $x_i = \frac{w_i}{e + \| w \|}$, $q_i = \frac{p_i}{e + \| p \|}$, $v_i = f(x_i) + bf(q_i)$

단계 9) 리셋이 아니면 단계 10) ~ 단계 11)을 수행한다.

단계 10) F2층의 승자노드의 상, 하향 가중치를 조절한다.

$$Z_{Jk} = adu_i + \{1 + ad(d-1)\} Z_{Jk}$$

$$Z_{iJ} = adu_i + \{1 + ad(d-1)\} Z_{iJ}$$

단계 11) 입력 벡터들을 제거하고, 비 활성화된 F2층의 노드들을 저장한다. 새로운 입력을 받아 들여서 F1층을 활성화 한다.

$u_i = \frac{v_i}{e + \| v \|}$
$w_i = s_i + au_i$
$p_i = u_i + dz_{ij}$
$x_i = \frac{w_i}{e + \| w \|}$
$q_i = \frac{p_i}{e + \| p \|}$
$v_i = f(x_i) + bf(q_i)$

그 순서도는 그림 3.3과 같다

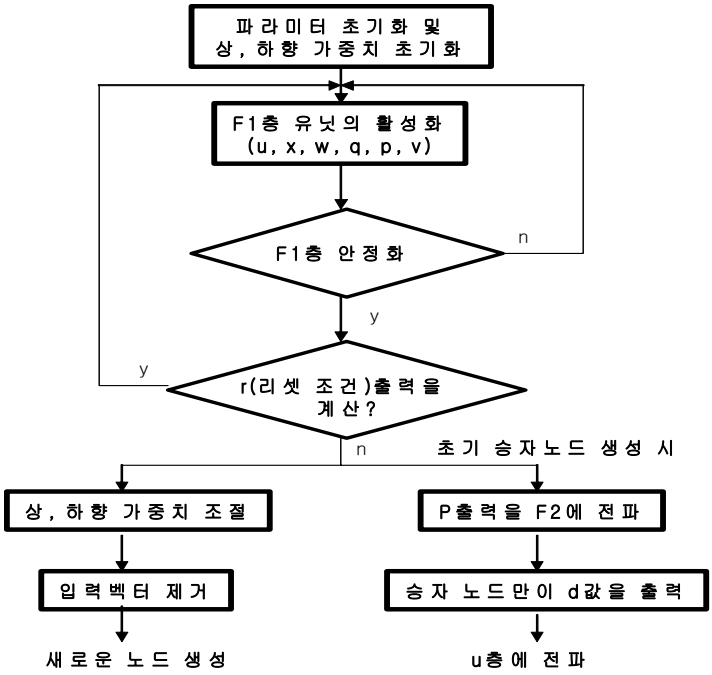


그림 3.3 ART2 순서도

Figure 3.3 ART2 flowchart

3.2 DTW 알고리즘

DTW(Dynamic Time Warping) 는 입력패턴과 참조 패턴 사이의 거리를 측정해서 그 유사도를 측정하는 방법이다. 다시 말하면, 제한된 경로 내에서 단조 증가를 통해서 가장 가까운 거리를 판별, 유사도를 측정한다. 예를 들어, 길이가 M인 입력 음성 패턴을 $T = T(1), T(2), .. T(M)$ 길이가 N인 기준 패턴을 $R = R(1), R(2), .., R(N)$이라고 하면 두 패턴

간의 유사도 d은 다음 식 (3.13)과 같이 누적거리로 표현된다[1],[3],[9],[16],[17].

$$D = \sum_{N}^{n=1} d(R(n), T(W(n))) \tag{3.13}$$

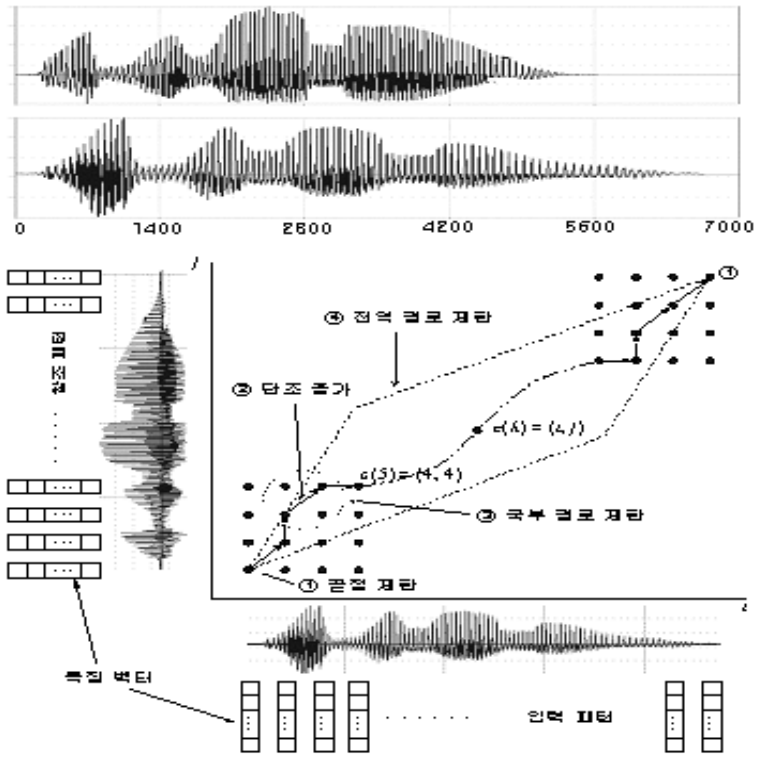


그림 3.4 DTW의 제약된 조건

Figure 3.4 Limited Conditions of DTW

이 때 $d(R(n), T(W(n)))$는 R의 n번째와 T의 $W(n)$번째의 국부적 유사도(Local Distance)이며, DTW는 두 패턴간의 누적 거리 최적화를 하는 (m,n)평면의 최적 경로 $m = W(n)$를 찾는 방법이다. 이 방법은 음성

신호의 특성을 고려해서 최적 경로 탐색에 다음과 같은 제약조건을 가한다. 이것은 그림 3.4에 잘 나타나 있다.

- 끝점 제한 (Endpoint Constraints) : $i(k)=I, \quad j(k)=J$
- 단조 조건 (Monotonic Condition) : $i(k-1) \le i(k),\ j(k-i) \le j(k)$
- 국부 경로 제한 (Local Path Constraints)
- 전역 경로 제한 (Global Path Constraints)
- 기울기 가중치 (Slop Weighting)

다음은 DTW의 알고리즘이다.

단계 1) 초기화 : $D_A(1,1)=d(1,1)m(1)$

단계 2) 반복 : $1 \le i_x \le T_x$, $1 \le i_y \le T_y$인 i_x , i_y에 대해서

$$D_A(i_x,i_y)=\min_{(i'_x,i'_y)}\left[D_A(i'_x,i'_y)+\zeta((i'_x,i'_y),(i_x,i_y))\right]$$

단계 3) 실행 : $d(x,y)=\dfrac{D_A(T_x,T_y)}{M_\phi}$

본 논문에서는 지역거리는 유클리드 거리법을 사용하고 있으며, 지역 제약방법은 ITAKURA방식[1]을 사용하고 있다. DTW의 경우 기준 모델 집합의 작성은 간단하다. 인식하고자 하는 명령어들을 발음하고 분석한 후 연속된 프레임들의 특징 벡터들을 저장하고 있으면 된다. 인식 시에는 입력된 음성을 분석해 특징 벡터를 추출한 후 이들 기준 모델 집합의 구성원과 개별적으로 DTW를 수행하여 가장 적은 누적 거리를 주는 구성원을 찾으면 된다. DTW는 고립단어 인식에 주로 사용되며 인식률이 높다는 장점을 가지고 있는 반면에, 계산량이 많다는 단점으로 고립단어 외에는 적용이 어려운 단점이 있다.

제 4 장 음성인식 시스템 및 이동로봇 제어시스템

4.1 음성인식 시스템의 하드웨어적인 구성

음성은 그 특성상 많은 데이터를 가지고 있다. 그리고 많은 데이터에 대해 압축시키고, 특징 값을 찾아내기 위해서는 빠른 산술 처리능력을 가진 칩이 필요하다. 본 논문에서는 이런 점을 감안해서 그림 4.1과 같은 음성인식 보드를 설계했다[18],[19].

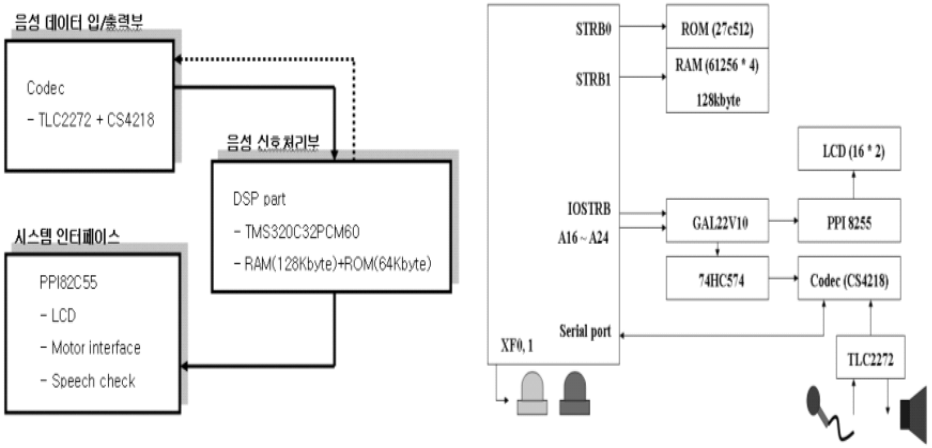


그림 4.1 음성인식보드의 구성

Figure 4.1 Configuration of speech recognition board

보드의 대략적인 구성을 보면 16비트 분해능을 가지고 있는 코덱(CS4218)을 통해서 시리얼로 DSP인 TMS320C32에 전달한다. 전달된 음성 데이터는 DSP내부의 산술연산 처리에 의해서 처리를 수행하고 최종

값은 LCD와 모터 인터페이스, 그리고 음성 처리 값에 대한 감지로 이루어진다. 전체적인 실제 모습은 부록 1에 잘 나타나 있다.

4.1.1 음성 데이터 입/출력부

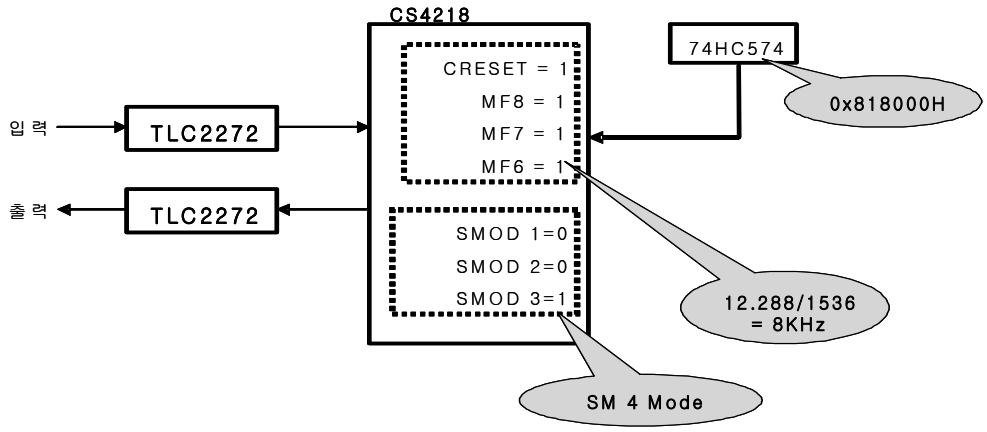
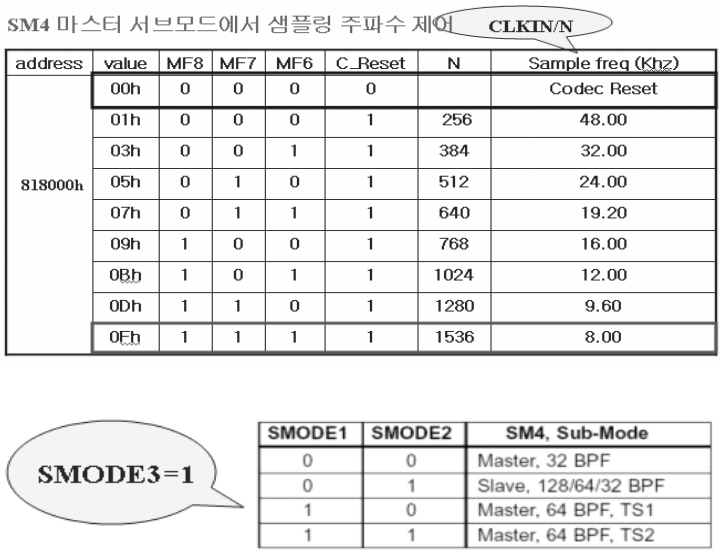


그림 4.2 음성 데이터 입/출력부

Figure 4.2 Speech data input/output part

이 부분은 아날로그 신호, 즉 음성을 디지털 신호로 바꾸어주는 부분으로 부록 11, 12에 잘 나타나 있다. 음성인식은 그 특성상 비교적 정확성을 요한다. 그래서 본 논문에서는 16비트 분해능을 가진 시리얼 오디오 코덱인 CS4218을 사용하고 있다. 그리고 내부에 AD와 DA 변환처리를 할 수 있고, Decimation 필터와 Smoothing 필터를 내장하고 있어서, 별도의 필터를 보유하지 않아도 된다. 처리는 다음과 같이 하고 있다. 그림 4.2를 보면 일단 음성을 증폭시키기 위해서 TLC2272라는 OpAmp를 사용하고 있다. 증폭된 음성은 CS4218의 세팅에 의해서 8KHz로 샘플링된다. 데이

터는 SM4 모드에 SMOD3의 세팅에 의해서 처음 16비트는 좌측오디오 채널로 다음 16비트는 우측오디오 채널로 보내어 진다. 그림 4.3에서 SM4 모드는 다음과 같이 외부 신호에 의해서 설정을 한다. 그리고 8KHz 샘플링은 DSP의 디코더에 의한 74HC574를 통해서 설정한다. 이것은 아래 그림 4.3에 잘 나타나 있다.



address	value	MF8	MF7	MF6	C_Reset	N	Sample freq (Khz)
818000h	00h	0	0	0	0		Codec Reset
	01h	0	0	0	1	256	48.00
	03h	0	0	1	1	384	32.00
	05h	0	1	0	1	512	24.00
	07h	0	1	1	1	640	19.20
	09h	1	0	0	1	768	16.00
	0Bh	1	0	1	1	1024	12.00
	0Dh	1	1	0	1	1280	9.60
	0Fh	1	1	1	1	1536	8.00

SMODE1	SMODE2	SM4, Sub-Mode
0	0	Master, 32 BPF
0	1	Slave, 128/64/32 BPF
1	0	Master, 64 BPF, TS1
1	1	Master, 64 BPF, TS2

그림 4.3 코덱 환경설정

Figure 4.3 Setting of codec

4.1.2 DSP를 이용한 신호처리부

앞에서도 언급했듯이 음성은 그 특성상 많은 데이터를 가지고 있고, 그리고 많은 데이터를 압축하고, 특징 값을 찾아내기 위해서 빠른 산술연산을 필요로 한다. 본 논문에는 이런 점을 감안해서, TI사의 플로팅 포인터

DSP인 TMS320C32(30MIPS)와 SRAM 61C256 * 4(128Kbyte) 그리고 음성 프로그램도 많은 메모리 용량을 감안, 64Kbyte ROM인 27C512를 사용하고 있다. 메인 CPU인 TMS320C32는 그림 4.4와 같이 구성되어 있으며 자세한 회로도는 부록 8, 9에 잘 나타나 있다. A0 ~ A15, D0 ~ D7까지는 많은 포트의 이용으로 인한 팬 아웃 현상을 방지 하기위해서 풀업 저항(10k)을 사용하고 있다. 그리고 프로그램의 다운로드는 MPSD-PP 에뮬레이터를 통한 J-TAG 방식을 사용하고 있다. 시리얼 포트를 통해서 CS4218과 연결, 음성 데이터의 입/출력을 담당하고 있으며, XF0와 XF1의 I/O포트를 통해서 CPU의 상태를 확인하고 있다. INT2와 INT3를 통해서 저장모드와 인식모드를 설정할 수 있게 했다.

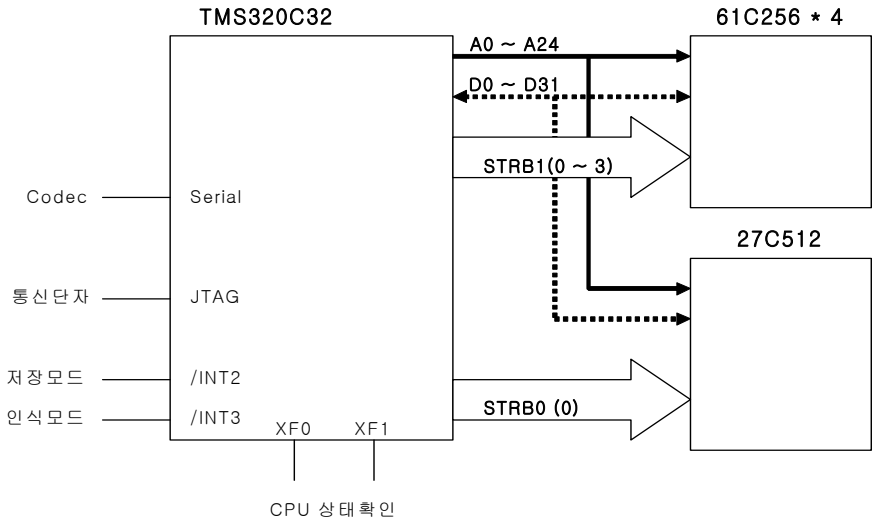


그림 4.4 DSP를 이용한 음성 신호처리부

Figure 4.4 Speech signal process part using the DSP

그리고 STRB0, 1과 IOSTRB는 TMS320C32에서 제공하는 메모리 인

터페이스 포트이다. 이 포트를 이용한 메모리 인터페이스는 그림 4.4와 같이 구성되어 있다. SRAM은 STRB1단자에 4개의 단자를 연결해서 32비트 엑세스를 하고 있고, EPROM은 STRB0단자의 1개의 단자를 연결해서 8비트 엑세스하는 방식을 취하고 있다. 그리고 IOSTRB는 현재 PPI82C55와 코덱 인터페이스에 사용하고 있다. 각 메모리와 주변 장치들의 어드레스 맵은 그림 4.5, 4.6과 같다. 그림 4.5를 보면 STRB0 즉 EPROM의 엑세스 영역은 0x100H이며, STRB1 즉 SRAM의 엑세스 영역은 0x900000H이다. 그리고 IOSTRB를 이용한 8255와 코덱은 디코더를 통해서 영역을 분할한다. 이것은 그림 4.6에 잘 나타나 있다. 그림을 보면 8255는 0x810000H이고 코덱은 0x818000H를 사용하고 있다.

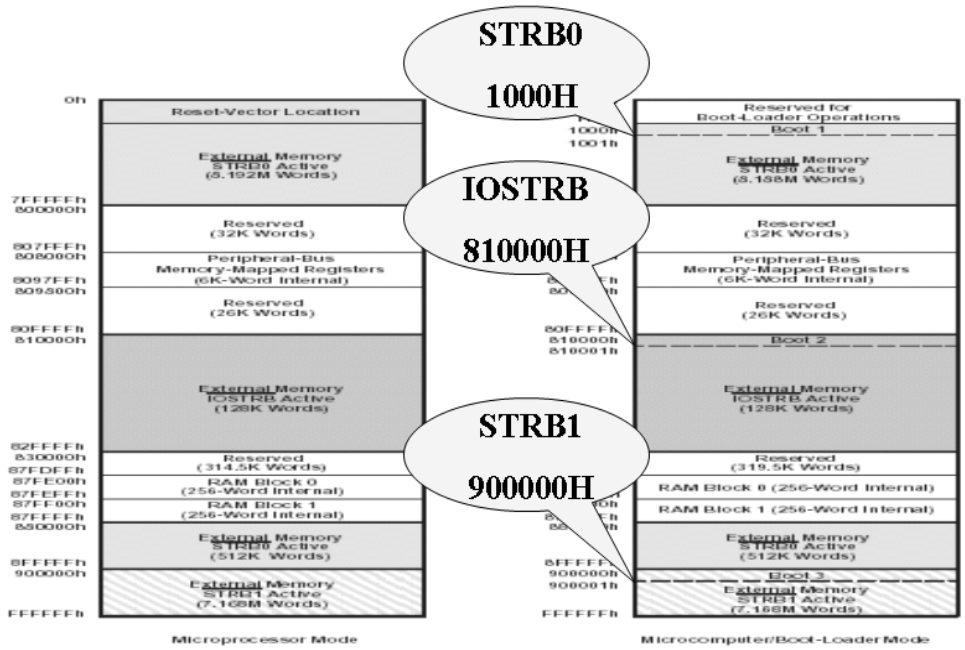
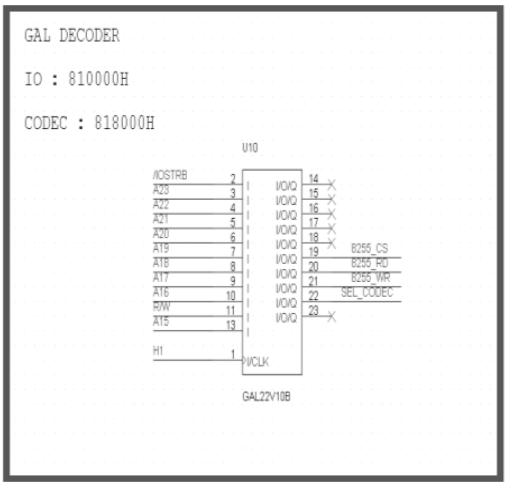


그림 4.5 메모리 맵

Figure 4.5 Memory map



/8255_CS = A23 * /A22 * /A21 *
/A20 * /A19 * /A18 *
/A17 * A16 * /IOSTRB

/8255_RD = /IOSTRB + RW

/8255_WR = /IOSTRB + /RW

/SEL_CODEC
= A23 * /A22 * /A21 *
/A20 * /A19 * /A18 *
/A17 * A16 * A15 */IOSTRB

그림 4.6 I/O 맵

Figure 4.6 I/O Map

4.1.3 시스템 인터페이스부

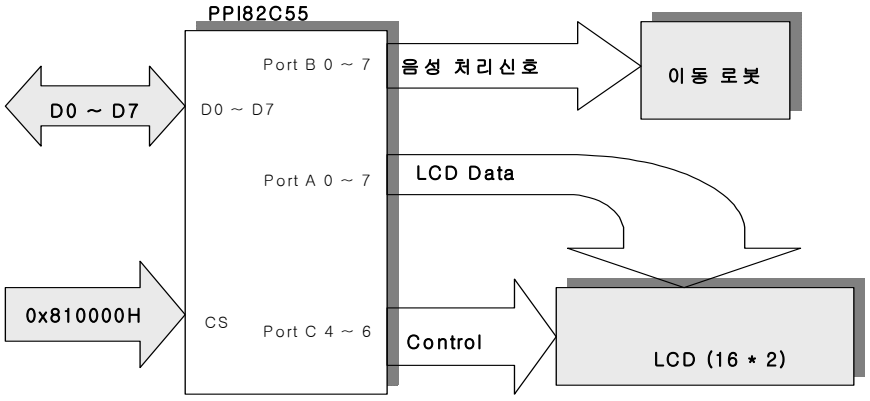


그림 4.7 시스템 인터페이스부

Figure 4.7 System interface part

시스템의 인터페이스는 음성인식 처리된 값을 표현하기위해서 사용한다. 그림 4.7을 보면, 구성은 PPI8255와 LCD를 사용하고 있으며 부록 10에 자세한 회로도가 있다. 포트 A는 LCD의 데이터 버스로 포트 C는 LCD의 제어버스로 사용되며 포트 B는 이동로봇에 최종 처리된 음성 데이터의 신호를 보내줄 때 사용된다.

4.2 음성인식 시스템의 소프트웨어적인 구성

4.2.1 실음성 구간 검출

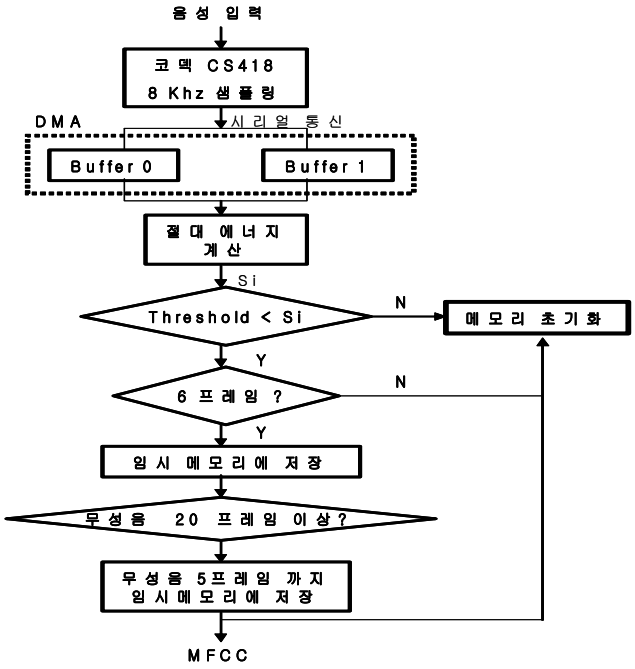


그림 4.8 실음성 구간 검출

Figure 4.8 Region detection of real speech

실음성 구간 검출이라는 것은 실제음성을 추출하는 것을 말한다. 다시 말하면, 실시간적으로 들어오는 음성을 유성음과 무성음으로 구별하고, 유성음이 검출된 부분에 대해서는 프레임으로 분할, 각 프레임을 계산하는 방식을 말한다. 본 논문에서는 절대 에너지 방식을 사용해서 계산하고 있다. 계산된 값이 설정된 값보다 크면 음성으로 간주, 이 부분을 실제음성으로 사용하고 있다.

본 논문에서의 실음성 구간 검출은 그림 4.8과 같이 하고 있다. 코덱(CS4218)을 통해서 음성이 들어온다. 이 때 샘플링은 8KHz로하고 있다. 들어온 음성은 DMA(Direct Memory Access)채널을 통해서 임시 저장된다. DMA채널은 두 개의 버퍼를 할당한다. 두 개의 버퍼 작용은 다음과 같다. 한 개의 버퍼로 음성데이터가 입력되면, 다른 버퍼는 절대 에너지 계산 하도록 설계되어 있다. 이 때 버퍼사이즈(프레임사이즈)는 256으로 설정했고, 프레임 블록킹은 80으로 설정하였다. 그리고 잡음 제거를 위해서 임계값(ITU)을 설정했는데, 이것은 50000으로 설정하고 있다. 임계값을 넘은 데이터 즉 음성으로 간주되어지는 데이터는 6프레임 이상되면 그 부분을 음성으로 간주하고 임시메모리에 저장한다. 이 후 무성음 구간이 20프레임 이상 되면 끝점에 존재하는 무성음 구간을 고려하여 5프레임만 저장시키고, 나머지 15프레임은 버리는 방식을 사용하고 있다.

4.2.2 전처리 단계 및 벡터 양자화

검출된 음성은 전처리 단계인 MFCC를 거쳐서 프레임 당 12계수 값으로 만든다. MFCC 후에 벡터 양자화를 통해서 512개의 코드북을 생성한다. 그림 4.9는 전처리에서 벡터 양자화까지의 과정을 나타낸다.

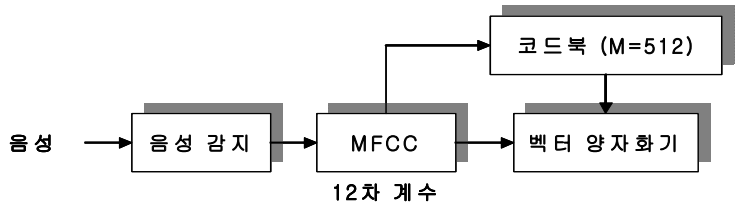


그림 4.9 전처리 단계

Figure 4.9 Preprocessing

4.2.3 음성인식 (DTW + ART2)

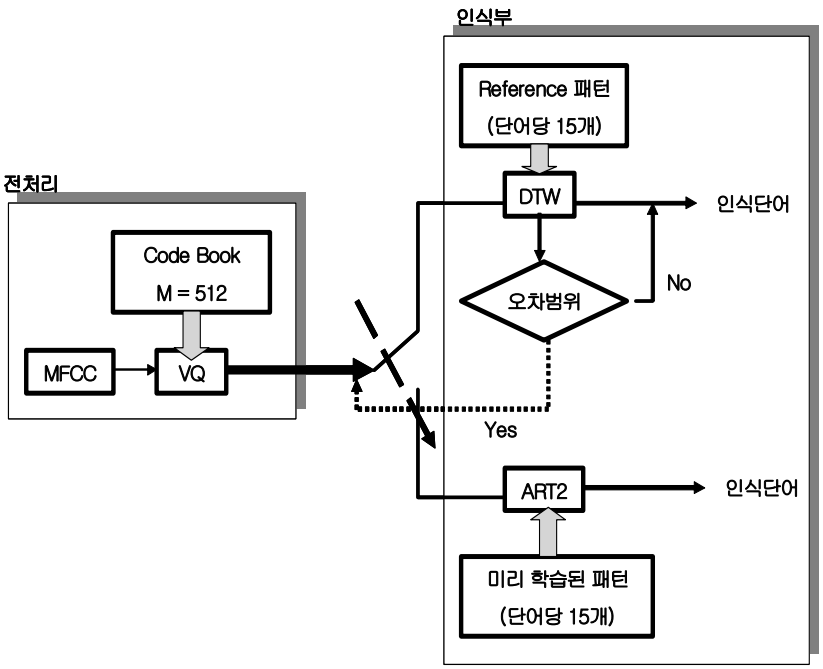


그림 4.10 음성인식 (DTW + ART2)

Figure 4.10 Speech recognition (DTW + ART2)

인식의 절차는 그림 4.10과 같다. 전 처리된 음성은 각각의 거리를 계산한다. 각각의 거리를 계산을 통해서 2순위 거리 값과 1순위 거리 값을 선정한다. 여기서 오차범위를 적용해서 오차 범위와 일치되면 ART2로 전환하는 방식을 취하고 있다. 오차범위는 그림 4.11과 같다.

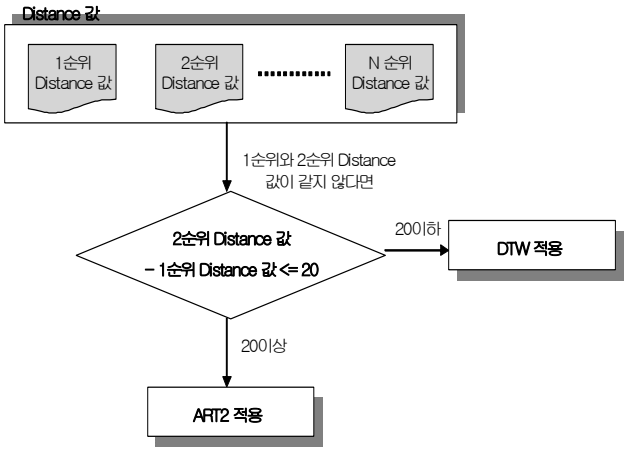


그림 4.11 오차 범위

Figure 4.11 Error range

일단 오차 범위가 존재하려면 1순위와 2순위 Distance 값이 일치하지 않는다는 전제 조건을 가져야 된다. 두 Distance 값을 빼서 나온 것이 20 이상일 경우는 ART2를 적용한다. 적용된 데이터는 어떤 단어의 데이터인지를 결정한다. 만약 레퍼런스에 없는 단어가 생성되면 인식을 중지한다.

4.3 이동로봇 제어시스템

본 논문에서 이동로봇은 전동 휠체어 시스템을 말한다. 전동 휠체어 시

스템의 구성은 그림 4.12와 같으며, 실제 모습은 부록 1에 있다.

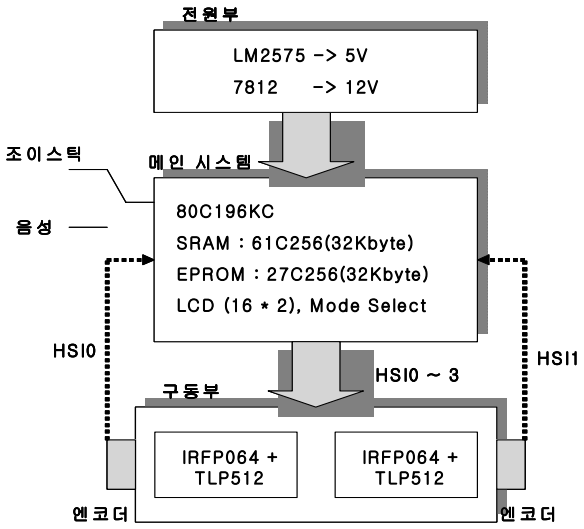


그림 4.12 전동휠체어의 구성

Figure 4.12 Configuration of mobile robot

이동로봇은 음성의 입력과 조이스틱 입력으로 이루어져 있다. 본 논문은 음성입력을 통한 모터 구동에 대해서 설명하겠다. 입력된 음성은 80C196KC의 고속 PWM 출력포트인 HSI0 ~ 3으로 PWM을 출력한다. 출력된 파형은 모터를 구동하고 폐루프 제어시스템인 PID제어 시스템을 위해서 엔코더 값을 HSI0 ~ 1로 받아들이는 방식을 취하고 있다. 이동로봇은 3가지 부분으로 나누어 설명하겠다.

4.3.1 전원부

전원부는 현재 시스템에 필요한 전원을 공급하는 부분이다. 현재 필요

한 전원 24V, 12V, 5V를 필요로 하고 있다. 그림 4.13를 보면, 현재 전원은 24V를 기준으로 각각 레귤레이터를 통해서 가변을 하고 있다. 자세한 회로도는 부록 7을 보면 잘 나타나 있다.

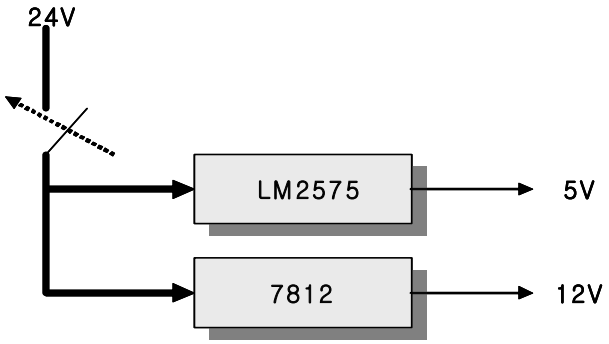


그림 4.13 전원부

Figure 4.13 Power part

4.3.2 메인시스템

메인시스템의 구성은 그림 4.14와 같으며 부록 5에 그 회로도가 잘 나타나 있다. 메인시스템은 80C196KC와 74HC573그리고 61C256, 27C512의 메모리로 구성되어 있는 것을 그림에서 확인할 수 있다. 80C196KC에서는 포트 0을 이용한 조이스틱 입력과 포트 1을 이용한 음성입력으로 나누어진다. 그리고 입력되어진 정보는 80C196KC의 고속펄스 입출력포트인 HSO포트 출력에 의해서 모터 구동부로 보내어진다. HSI0과 1은 엔코더에서 받은 값을 통해서 폐루프제어시스템인 PID제어에 사용되어진다. 모드는 74LS240을 사용해서 포트 0에 7번은 조이스틱모드로 그리고 포트 2에 2번은 음성모드로 전환하는 방식을 사용하고 있다.

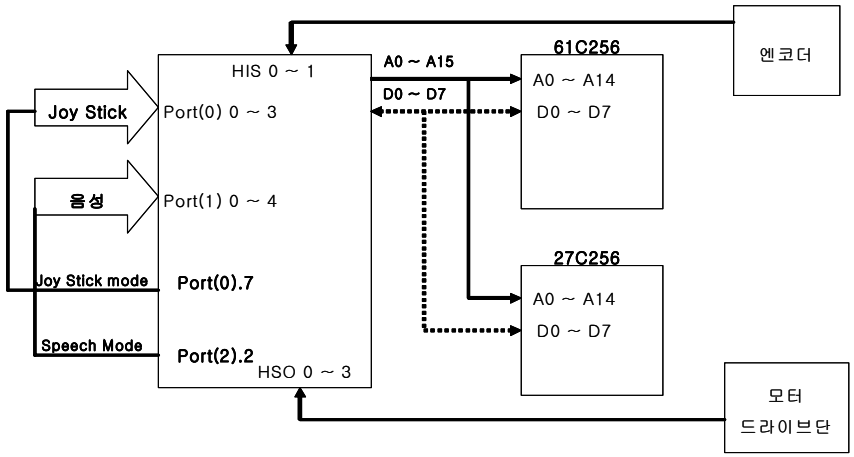


그림 4.14 메인시스템의 구성

Figure 4.14 Configuration of main system

4.3.3 모터 구동

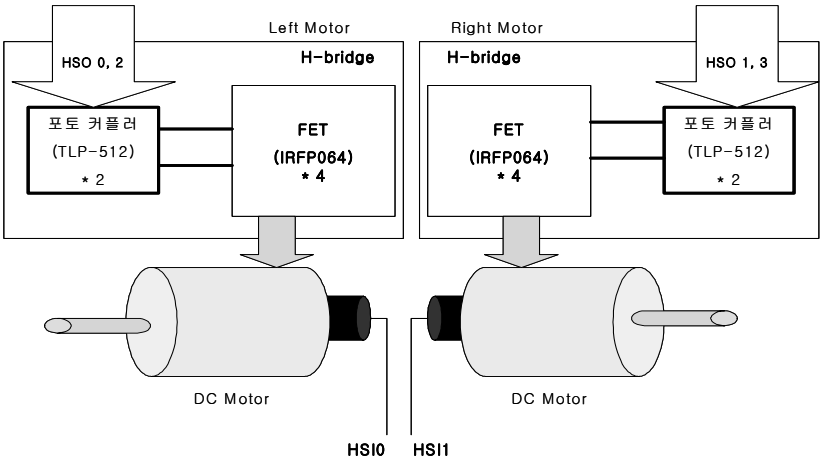


그림 4.15 모터 구동부

Figure 4.15 Motor motion part

현재 사용되어지고 있는 모터는 연속 전류가 최대 8A로 흐르고 무부하시 최대속도가 3000RPM에 감속기의 기어비가 1:22.67인 모터로서 비교적 큰 DC모터를 사용하고 있다. 그래서 TR에 의한 전류제어가 불가피하여 MOSFET를 통한 전압제어를 하고 있다. FET는 IRFP064를 사용하고 있으며 포트 커플러(TLP-512)를 통해서 스위칭하는 H-브릿지회로를 구성한다. 각 모터는 80C196KC에서 나오는 HSO를 통해서 구동을 하고 있는데 좌측 모터에는 0, 2번을 우측모터에는 0, 1을 사용한다. 그림 4.15를 보면 잘 나타나 있고, 부록 6에 그 회로도가 잘 나타나 있다.

4.3.4 센서부

센서부는 이동로봇 즉 전동휠체어가 뒤를 주시하지 못한다는 점을 감안, 이격거리 80Cm정도에 물체가 감지되면 소리를 내어서 주의를 요하게 하는 부분이다. 센서부는 초음파 센서와 PIC16F84를 통해서 80C196KC와는 별개로 설계를 했다. 그 구성은 그림 4.16과 같으며 부록 7에 회로도가 잘 나타나 있다.

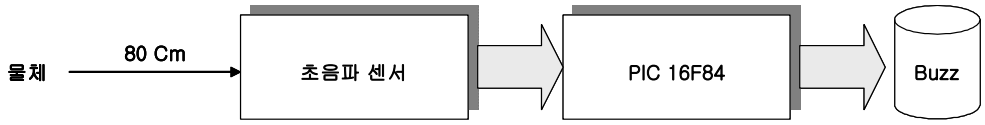


그림 4.16 초음파 센서 회로

Figure 4.16 Ultra sensor circuit

4.4 전체 시스템

그림 4.17을 보면 총 3개의 모듈을 통해서 전체 시스템을 나누어 볼 수 있다. 음성과 조이스틱은 앞에서도 언급했던 것과 같이 스위치의 작용에 따라서 그 모드가 바뀌어서 설정된다. 본 논문에서는 음성에 의한 이동로봇제어 시스템에 대해서만 언급을 하겠다. 음성을 통해서 나온 신호는 이동로봇 시스템에 연결되어서 그 신호 값에 따라 모터를 구동하게 된다. 전체적인 모습은 부록 2에 잘 나타나 있다.

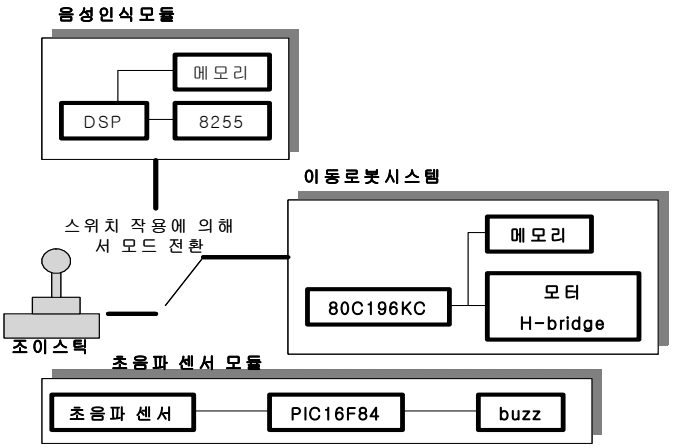


그림 4.17 전체 시스템

Figure 4.17 Total system

제 5 장 실험 및 결과

본 장에서는 4장까지의 내용을 바탕으로 실제로 어떻게 구현되었는지에 대해 설명하도록 하겠다.

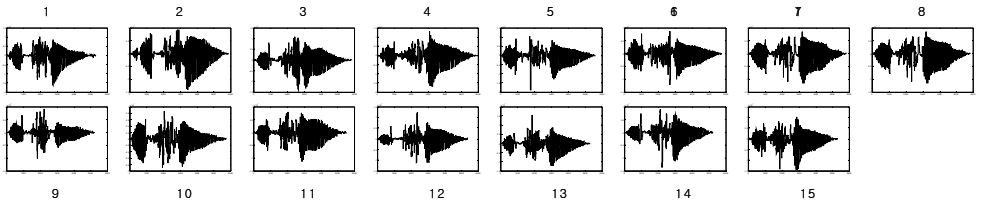
5.1 음성인식 시스템의 실험 및 결과

본 장은 음성인식 시에 변환되는 과정과 인식률에 대해 측정했다. 본 측정은 실시간 상에서는 빠른 데이터 처리와 시뮬레이션이 불가피해서, 오프라인에서 측정한 것을 토대로 했다. 하지만 인식 부분에서는 실시간으로 처리된 것을 중심으로 DTW와 ART2 사용 시를 비교해서 인식률의 차이에 대한 실험에 대해서 설명할 것이다.

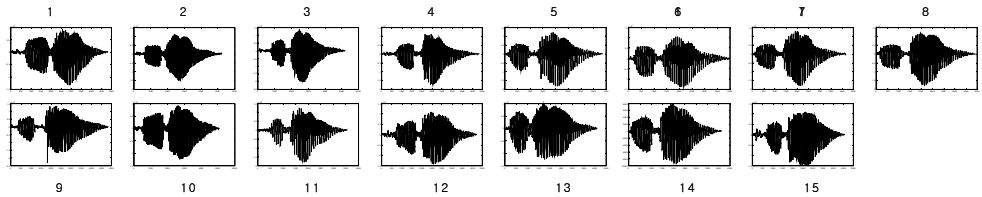
5.1.1 음성 입력

음성은 이동로봇의 진행 방향에 따른 발성을 해야 되기 때문에 다음의 9개단어로 선정을 했고 각 단어 당 15번 씩 발성을 했다. 이것을 오프라인 상에서 *.pcm으로 받아들인다. 아래에서 보여주는 시뮬레이션들은 *.pcm으로 받아들였을 때의 음성의 모양을 나타낸 것이다.

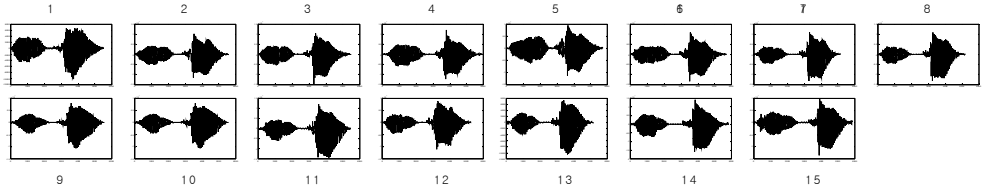
- 앞으로



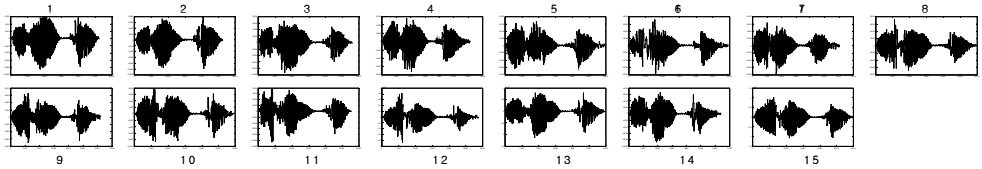
- 뒤로



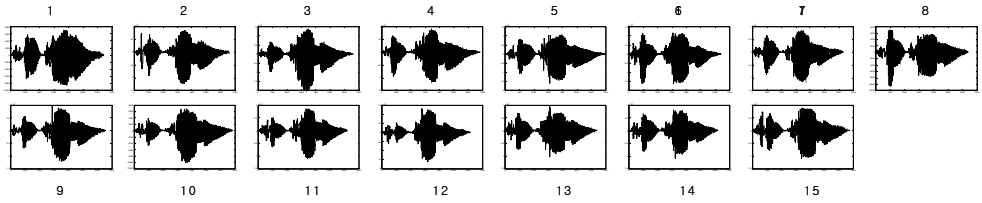
- 왼쪽



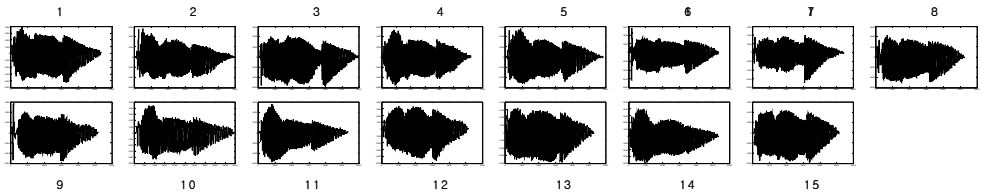
- 오른쪽



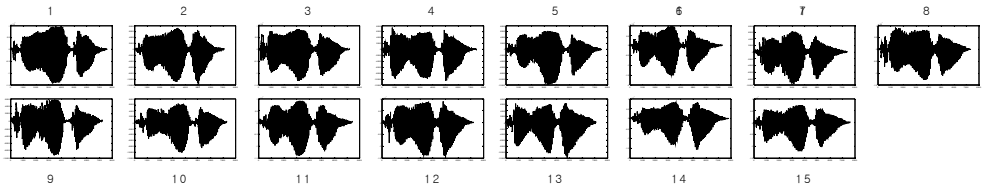
- 천천히



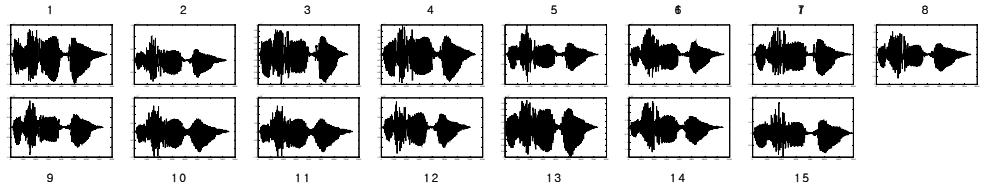
- 빨리



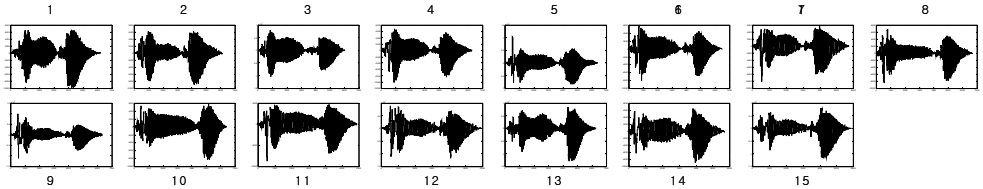
- 좌회전



- 우회전



- 정지

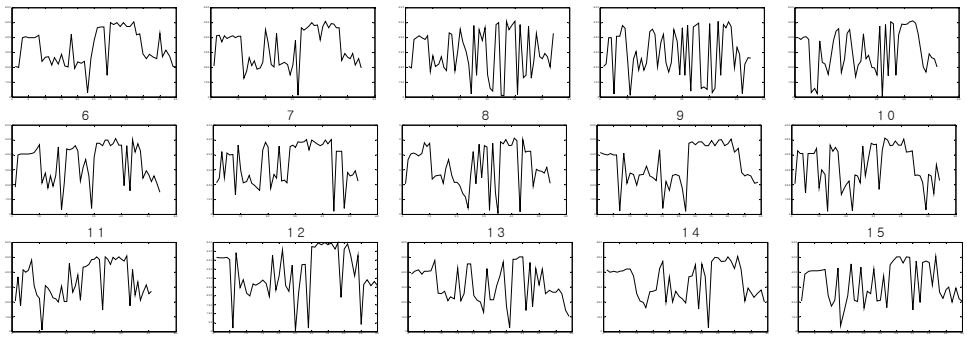


5.1.2 전처리된 음성 데이터

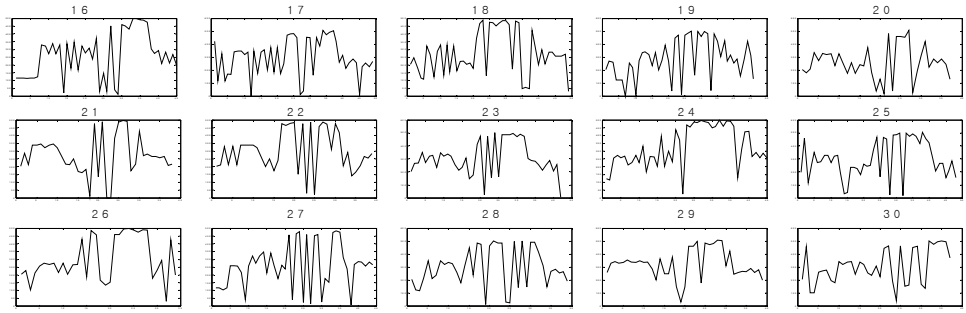
위와 같이 들어온 음성 데이터 135개는 MFCC와 VQ를 통해서 다음과 같은 음성으로 나타난다. 각 데이터는 135개의 배열로 분류를 했다. 이때 사용되어진 1 프레임 사이즈는 256으로 그리고 프레임 블록킹 사이즈를 80으로 하고 있다. 윈도우는 해밍윈도우를 그리고, MFCC를 사용한 전 처리를 하고 있다. MFCC를 통해서 *.pcm데이터를 *.mel파일로 전환을 했다. 그리고 이것을 gathering을 통해서 하나의 파일로 만든다 (*.tab). 만들어진 파일은 코드북을 생성해야 되는데, 이것은 데이터를 압축하는 과정이다. 이 과정에서 코드북사이즈는 총 512개로 분리를 하고 있다

(*.dat). 이렇게 분리되어진 것은 양자화를 통해서 *.mel을 135개의 *.vqd로 바꾼다. 아래 보이는 것은 *.vqd를 표현한 것이다.

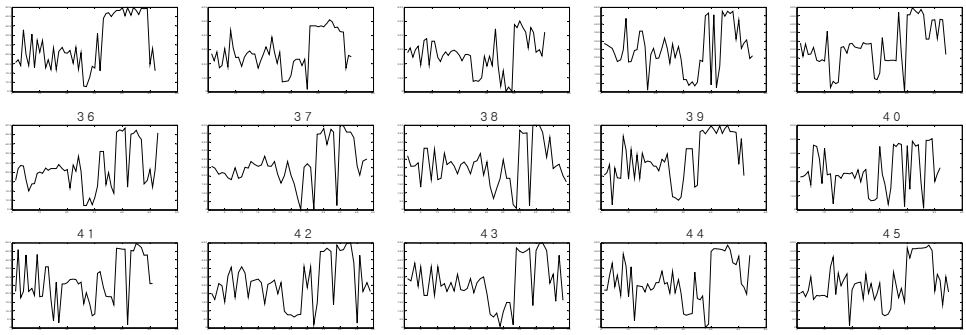
-앞으로



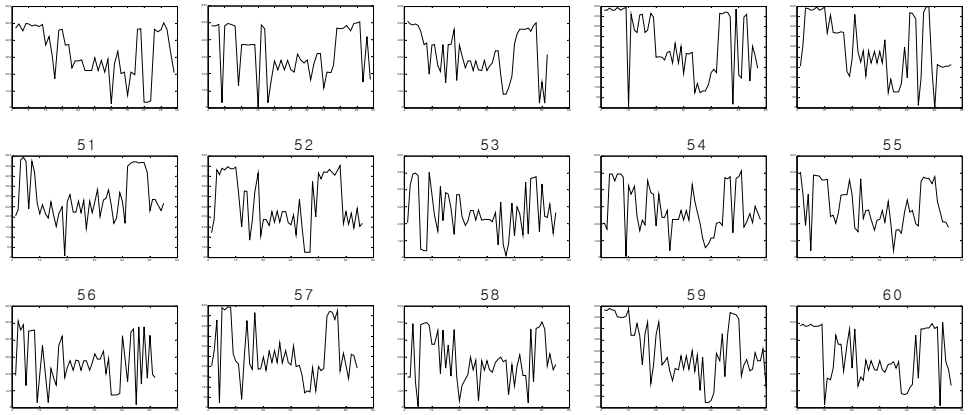
- 뒤로



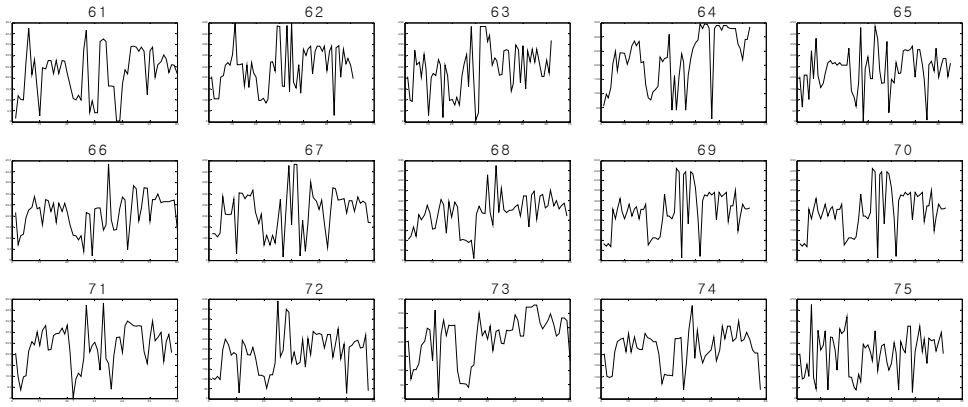
- 왼쪽



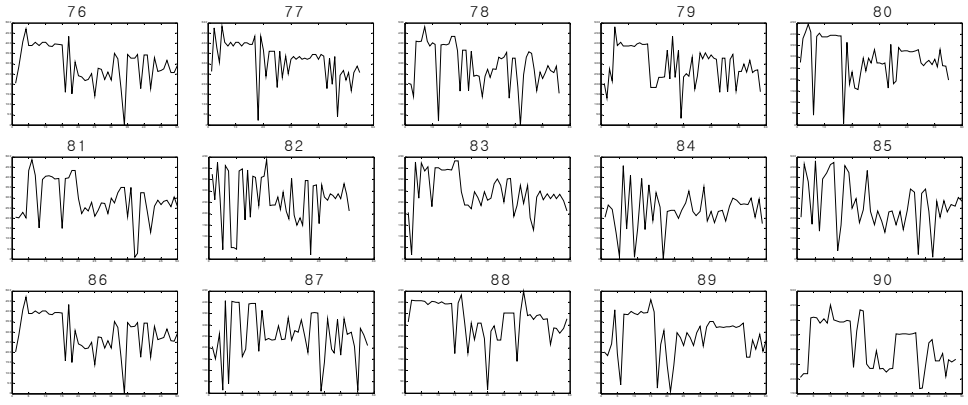
- 오른쪽


51
52
53
54
55
56
57
58
59
60

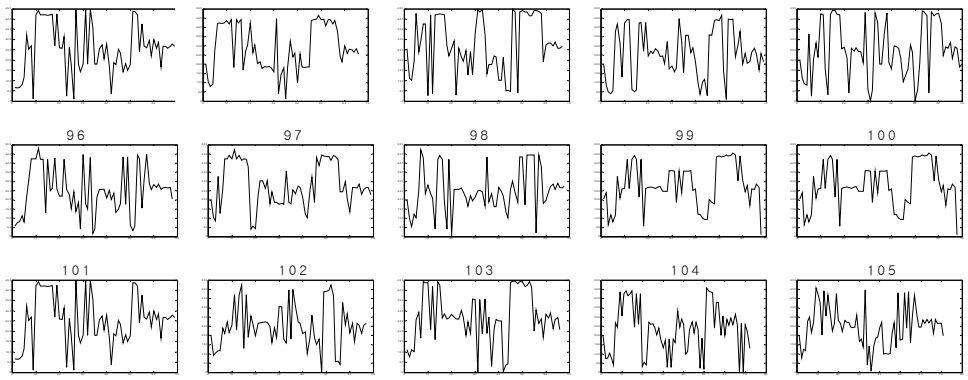
- 천천히


61
62
63
64
65
66
67
68
69
70
71
72
73
74
75

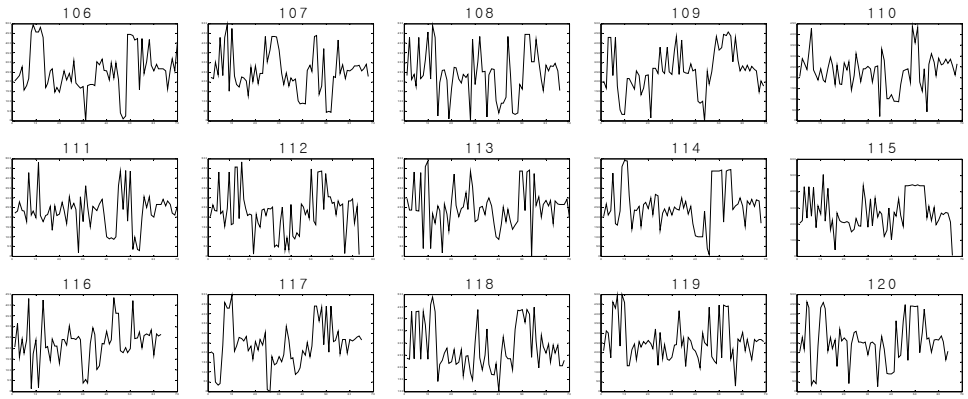
- 빨리


76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

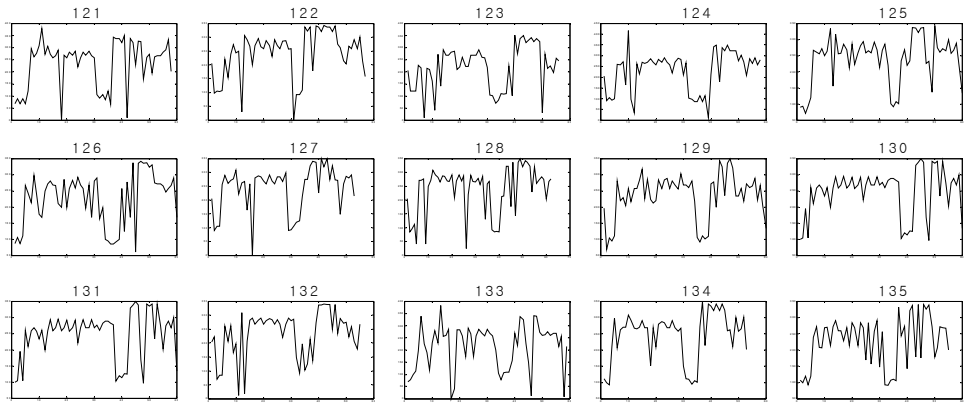
- 좌회전


96
97
98
99
100
101
102
103
104
105

- 우회전


106
107
108
109
110
111
112
113
114
115
116
117
118
119
120

- 정지


121
122
123
124
125
126
127
128
129
130
131
132
133
134
135

5.1.3 ART2 분류

벡터 양자화를 통해서 만들어진 것은 ART2를 통해서 비슷한 패턴으로 분류된다. 분류되어진 패턴은 다음 표 2와 같았다. 표 2와 같이 분류되기 전에 ART2를 위한 파라미터는 다음과 같이 설정했다.

- a = 10, b = 10, c = 0.1, d = 0.995, ($\frac{cd}{1-d} \leq 1$)
- 입력 패턴 : 135(총 음성 수) x 80 (최대 프레임 수)
- 승자노드(M) = 60 (최대로 인가될 수 있는 개수)
- 에러율 : 0.01
- [ ]는 승자노드

표 5.1 ART2에 의한 패턴 분류

Table 5.1 Pattern distribution of ART2

명령어	승자노드	특징 벡터
앞으로	[1]	1, 2, 3, 6, 7
	[2]	4, 8, 13
	[3]	5, 9, 10, 15
	[4]	11, 12, 14
뒤 로	[5]	16, 17, 27
	[6]	18, 19, 20, 21, 26, 29
	[7]	22, 23, 24, 25, 28
	[8]	30
왼 쪽	[9]	31, 32, 33, 35, 37, 39, 44, 45
	[10]	34, 36, 38, 41

	[11]	40, 42, 43
오른쪽	[12]	46, 47, 48, 49, 50, 56, 57
	[13]	51, 52, 54
	[14]	53, 55
	[15]	58, 59, 60
천천히	[16]	61, 62, 68, 69, 71, 75
	[17]	63, 64, 66, 67, 72
	[18]	65, 70, 73, 74
빨 리	[19]	76, 77, 78, 79, 85, 86, 88, 90
	[20]	80, 81, 82, 87, 89
	[21]	83, 84
좌회전	[22]	91, 92, 93
	[23]	94, 95, 96
	[24]	97, 98, 99
	[25]	100, 101, 104
	[26]	102
	[27]	103, 105
우회전	[28]	106, 108
	[29]	107, 119, 120
	[30]	109, 110
	[31]	111, 114, 118
	[32]	112, 117
	[33]	113
	[34]	115, 116
정 지	[35]	121, 122, 123, 124, 125, 126
	[36]	127, 131, 132, 134
	[37]	128, 129, 130, 133, 135

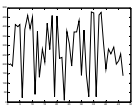
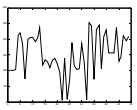
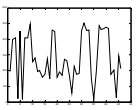
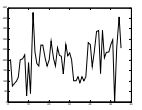
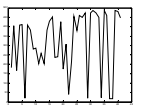
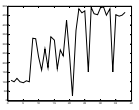
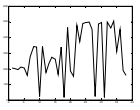
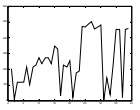
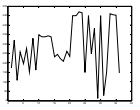
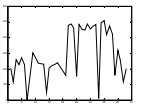
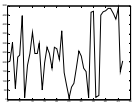
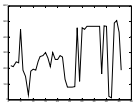
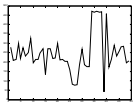
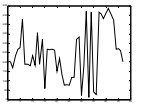
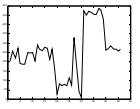
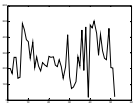
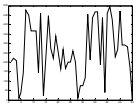
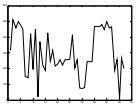
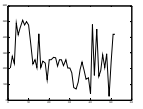
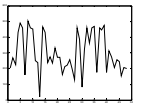
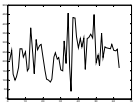
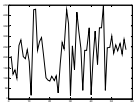
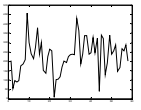
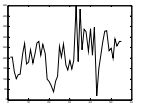
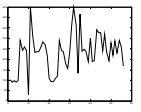
5.1.4 음성인식

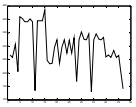
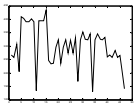
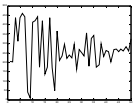
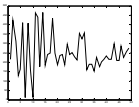
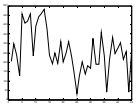
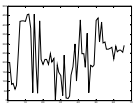
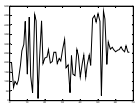
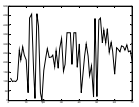
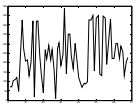
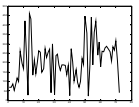
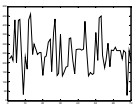
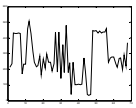
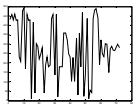
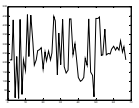
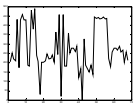
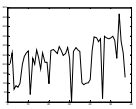
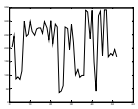
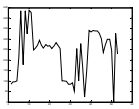
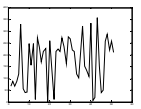
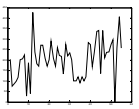
1) ART2에 의한 인식

ART2에 의해서 분류되어진 음성을 다섯 번의 입력 음성을 통해서 인식을 해보았다. 이 실험에서 [ ]는 같은 승자노드이며, X는 인식되지 않은 단어를 얘기한다. 그 결과는 표 5.2와 같다.

표 5.2 ART2에 의한 인식

Table 5.2 Recognition of ART2

앞으로					
	[4]	[1]	[3]	[3]	X
뒤 로					
	[6]	[5]	[7]	[5]	[7]
왼 쪽					
	[11]	[9]	[9]	[10]	[9]
오른쪽					
	X	X	[13]	[14]	[13]
천천히					
	[16]	[17]	[16]	[18]	[18]

빨 리					
	[19]	[19]	[20]	[21]	[19]
좌회전					
	[25]	[22]	[26]	[24]	X
우회전					
	[33]	[34]	X	[31]	X
정 지					
	[35]	[36]	[37]	[37]	[35]

2) 음성인식

인식은 두 알고리즘, 즉 DTW만 사용했을 때의 인식률과 ART2를 적용했을 때의 인식률을 비교해서 설명하도록 하겠다. 본 인식은 전동 휠체어가 실지 장애인에 접목 시에 잡음환경이라는 것을 고려해서 실험을 했다. 그 실험 결과는 표 5.3과 같다.

표 5.3 음성인식 테스트 결과

Table 5.3 Result of speech recognition test

명령어	테스트 횟수	DTW		ART2	
		인식 수	인식률	인식 수	인식률
앞으로	50	46	92%	49	98%

뒤로	50	45	90%	49	98%
왼쪽	50	46	92%	48	96%
오른쪽	50	47	94%	49	98%
천천히	50	45	90%	48	96%
빨리	50	45	90%	48	96%
좌회전	50	44	88%	46	92%
우회전	50	44	88%	47	94%
정 지	50	47	94%	49	98%

위의 결과로 DTW단독으로 사용할 때보다 ART2사용 시 전체 평균인식률이 약 5%정도 상승된 것을 확인할 수 있다. 위의 실험들로 볼 때 후처리알고리즘을 사용하게 되면 그 만큼에 인식률 향상을 가져온다는 것을 발견했다.

5.2 이동로봇에 적용되어진 음성인식

이동로봇 즉 전동 휠체어에는 음성인식을 통해서 나온 값을 PPI8255의 포트 B를 통해서 출력을 했다. 그리고 이동로봇의 포트 1에서 0 ~ 4까지의 포트를 통해서 받게 된다. 이것은 현재 PPI82C55 B 포트에 LED를 부착해서 그 상태를 확인할 수 있다.

모터의 속도는 명령어에 상관없이 첫 발성 시 3 Km/h의 속력을 가진다. 후에 '빨리'라는 명령어를 받게 되면 1 Km/h라는 속력이 추가된다. 이렇게 해서 최대 속도는 7 Km/h의 속력을 만든다. '뒤로'라는 명령어를 발성하면 1 Km/h로 속도가 줄게 된다. '왼쪽'과 '오른쪽'은 각 1번씩 발성할 때마다 30도씩 방향을 전환하게 되고, 3번 발성 시에는 '좌회전'과 '우회

전’과 같은 턴(Turn)의 개념으로 사용된다. 위의 설명을 표 5.4와 같다.

표 5.4 음성 인식에 따른 모터의 이동

Table 5.4 Move of motor for speech recognition

명령어	신호	이동 로봇의 작용
앞으로	0xF1	3km/h로 전진한다.
뒤 로	0xF2	3km/h로 후진한다.
왼 쪽	0xF3	한번 발성 시 30도씩 왼쪽으로 이동, 3번 발성하면 좌회전과 같은 턴(Turn)의 기능
오른쪽	0xF4	한번 발성 시 30도씩 오른쪽으로 이동, 3번 발성하면 우회전과 같은 턴(Turn)의 기능
천천히	0xF5	발성 시 1km/h의 속도 감소
빨 리	0xF6	발성 시 1km/h의 속도 증가 (최대 7km/h까지)
좌회전	0xF7	좌측으로 턴해서 돌아간다.
우회전	0xF8	우측으로 턴해서 돌아간다.
정 지	0xF9	정지 상태에 들어간다.

제 6 장 결 론

본 논문에서는 음성인식 중에서 화자종속형에 많이 사용하면서 비교적 인식률이 높은 DTW를 사용해서 전동휠체어에 적용시켜보았다. 여기서는 그 대상이 전동휠체어이기 때문에, 임베디드 시스템에서 구현을 했으며, 실시간이라는 개념을 도입, 시간에 따라 연속적으로 흘러 들어오는 음성을 감지해서 그것을 빠르게 처리하고 빠르게 인식시켜서 전동휠체어에 적용시키는 것이 그 목적이다. 그래서 빠른 인식을 위해서 비교적 적은 메모리에 저장이 가능해서 화자독립의 압축 알고리즘으로 많이 사용된 벡터 양자화를 도입해서, 빠른 인식을 통해 다소 안정적인 시스템을 구현할 수 있었다. 하지만 벡터 양자화의 사용으로 인해서 인식률이 다소 저하되는 것이 확인할 수 있었다. 여기에 ART2알고리즘을 하이브리드형으로 적용해서 인식률 향상을 가져올 수 있었다. DTW와 ART2는 오차범위라는 것을 적용해서 사용하고 있다. 오차범위는 인식거리 2순위에서 1순위를 뺀 것이 20이하가 되면 DTW알고리즘을 20이상이 되면 ART2알고리즘을 적용시켰다. 이렇게 함으로써 약 5%라는 인식률 향상을 가져올 수 있었고, 그 결과 보다 빠른 처리와 정확한 인식효과를 얻을 수 있었다.

음성인식 시스템에서는 신호처리 전용칩인 DSP(TMS320C32)를 통해서 음성 신호처리만을 하였고, 전동휠체어 시스템에서는 16비트 마이크로콘트롤러(80C196KC)를 사용하여, 보다 더 안정된 시스템을 구현할 수 있었고, 오인식에 대비해서 초음파 센서를 이용, 80cm 거리의 물체들에 대해 주의를 요할 수 있게 설계를 했다.

앞으로의 연구방향은 연속 음성 인식 및 화자독립형 음성인식을 임베디드 시스템에 적용시키는 것과 위의 음성인식 시스템을 다른 부분에 접목시켜서 구현 해 보고자 하는 것이 과제이다.

참고 문헌

[1] Lawrence Rabiner and Biing Hwang Juang, " *Fundamentals of Speech Recognition* ", Prentice Hall, 1993

[2] 정익주, 정훈, " TMS320C32 DSP를 이용한 실시간 화자 종속 음성인식 하드웨어 모듈(VR32)의 구현 ", *한국음향학회, Vol.17., No.4. 14-22, 1998.*

[3] 김창근, 한학용, " TMS320C32를 이용한 실시간 음성인식 무선자동차의 구현 ", *신호처리 시스템학회, 2001*

[4] 오경환, " *음성언어정보처리* ", 홍릉과학출판사, 1997

[5] 유강주, " DHMM을 이용한 숫자음 인식의 Data Fusion에 관한 연구 ", *한국해양대학교 공학석사학위 논문, 1998*

[6] Lawrence Rabiner, " A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition ", *Proc. IEEE, Vol 77, No. 2, FEBRUARY 1989*

[7] 이상배, " *퍼지-뉴로 제어 시스템* ", (주)교학사, 1999

[8] 김태영, " Signal Processing using Fuzzy Logic and Neural Network for Welding Gap Detection ", *한국해양대학교 석사학위 논문, 1999.*

[9] 김정훈, " 음성인식처리용 임베디드 시스템의 설계 및 구현에 관한 연구 ", *한국해양대학교 공학석사학위 논문, 2000*

[10] Steven L.Gay, " Jacob Benesty, *Acoustic Signal Processing for Telecommunication* ", Kluwer Acacemic Publishers, 2000.

[11] John Makoul, Salim Roucos and Herrert Gish, " Vector Quantization in Speech Coding ", *Proc. IEEE, Vol. 73, No.11,*

1985.

[12] L.R Rabiner, S.E. Levinson and M.M. Sondhi, " On the Application of Vector Quantization and Hidden Markov Models to Speaker Independent Isolated Word Recognition ", *Bell System Technical Journal, Vol. 62, No.4 APRIL 1983*

[13] Chin-Der Wann, Stelios C. A. Thomopoulos, " A comparative study of self-organizing clustering algorithms dignet and ART2, *Neural Networks ", Vol. 10, No. 4, pp. 737-753, 1997*

[14] Carpenter, G.A., Grossberg S., " The ART of adptive pattern recognition by a self-organizing neural network ", *IEEE Comput. Vol. 21, No. 3, pp. 77-88, 1988*

[15] Carpenter, G.A., Grossberg S., " ART2: Self-organization of stable category recognition codes for analog input patterns, *Applied Optics ", Vol. 26, No. 23, pp. 4919-4930, 1987*

[16] 김정훈, 류홍석, 강성인, 강재명, 김관형, 이상배, " 화자 독립형 음성인식 모듈 설계에 관한 연구 ", *제어자동화 시스템 공학회 추계 학술대회, P109 ~ 112, 2002. 12.*

[17] 류홍석, 김정훈, 강재명, 강재명, 김관형, 이상배 " 다기능 전동휠체어의 음성인식 모듈에 관한 연구 ", *대한전자공학회 제 25권 1호, P83 ~ 86, 2002. 6.*

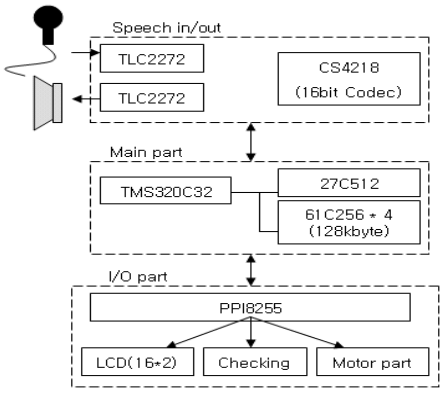
[18] " *TMS320C32 General Purpose User's guide* ", Texas Instrument

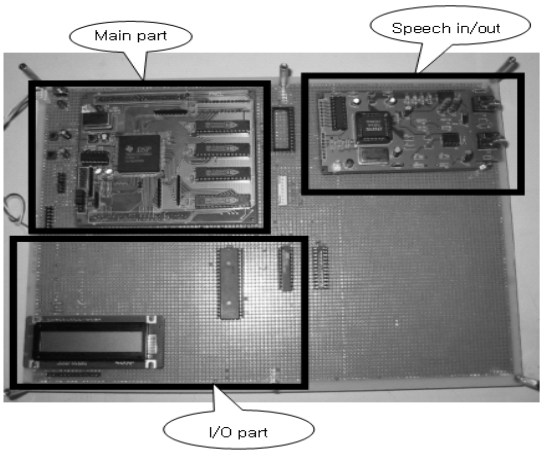
[19] 윤덕용, " *TMS320C32 마스터* ", Ohm사, 1999.

[20] Claudio Becchetti, " *Speech Recognition Theory and C++ Implementation* ", John Wiley & Sons Ltd, 1999.

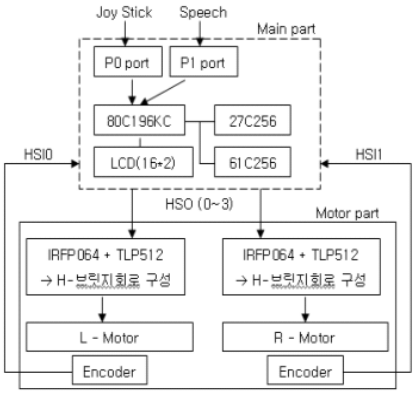
부 록

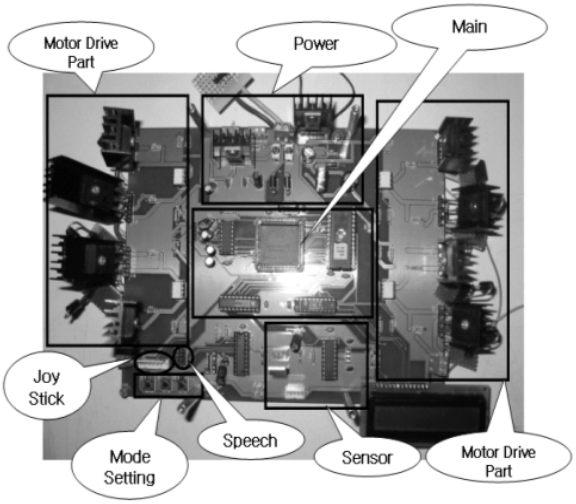
1. 음성인식 시스템



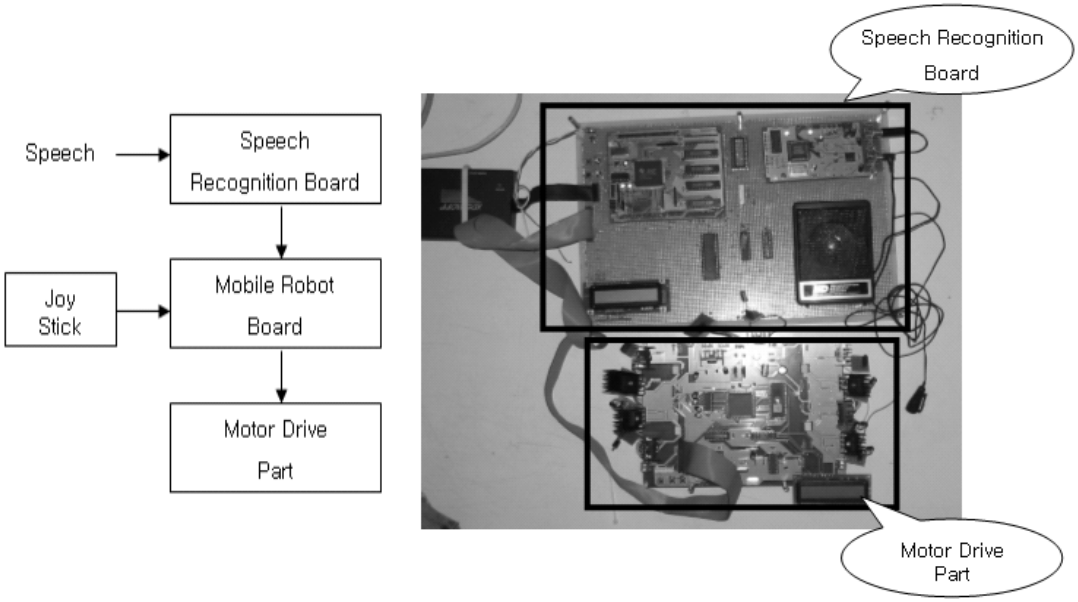


2. 이동로봇(전동휠체어) 시스템



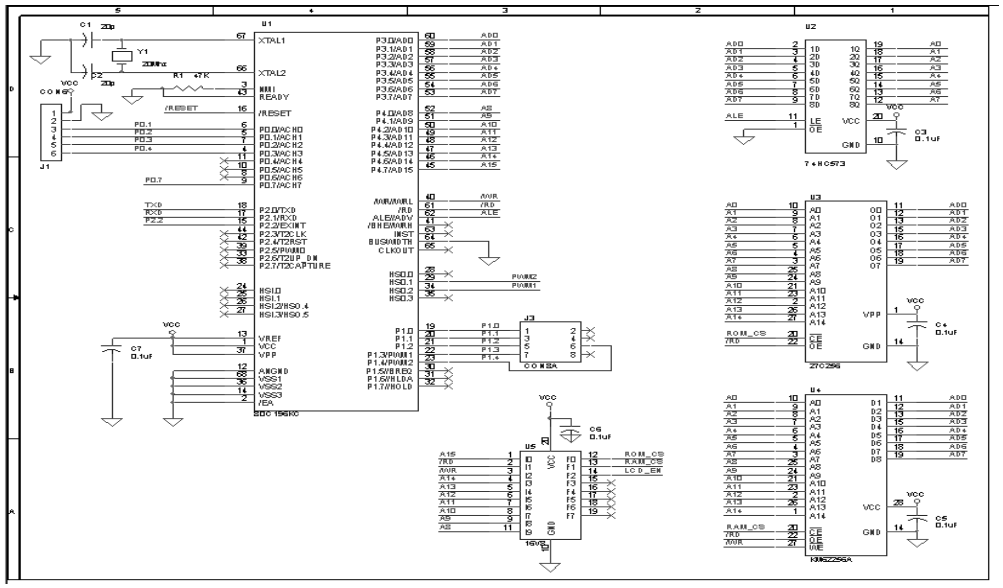


3. 전체 시스템

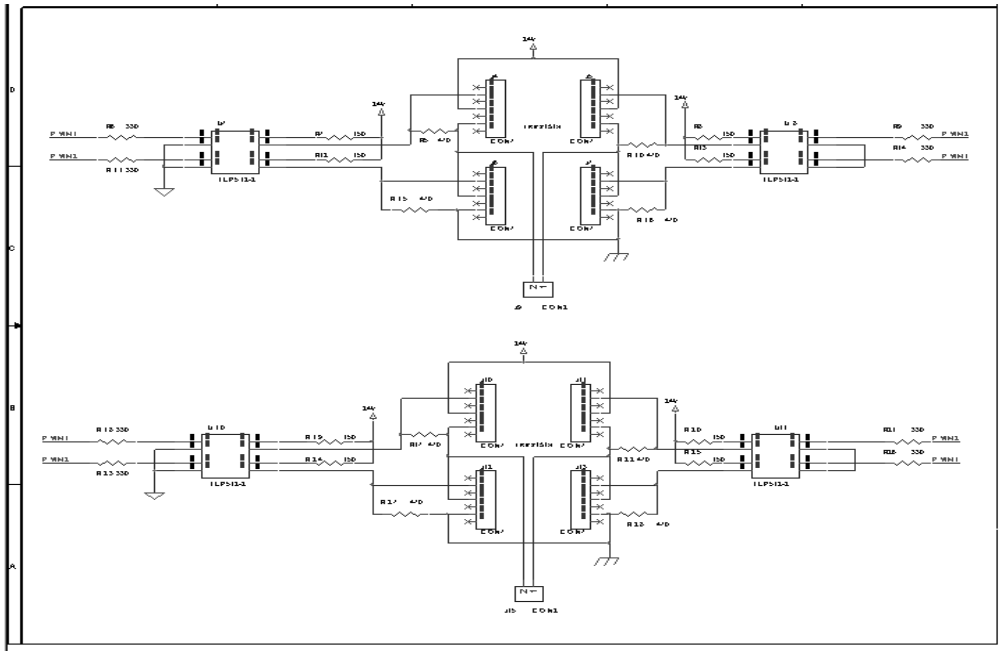


4. 실제 이동로봇의 모습(전동휠체어)

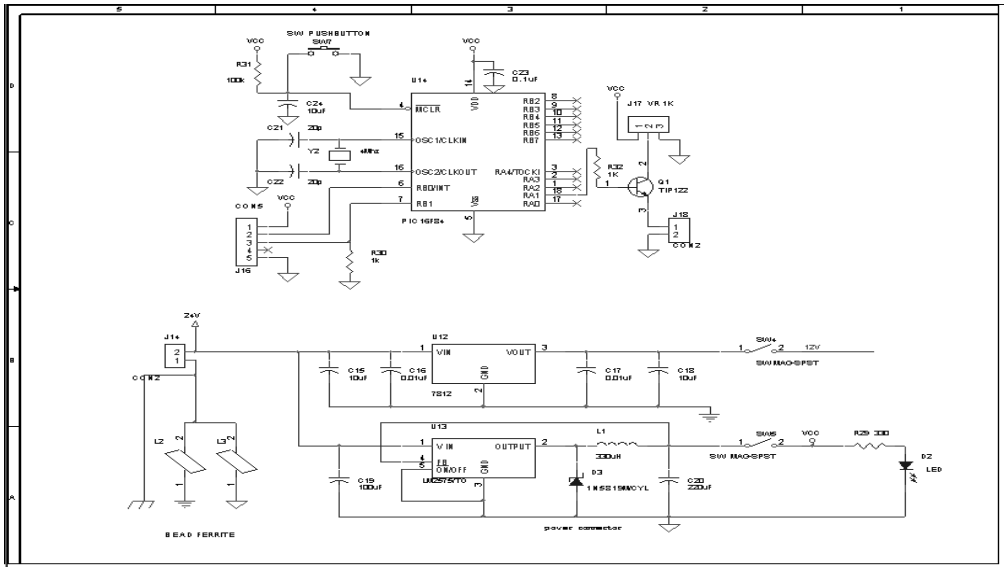
5. 이동로봇 메인시스템



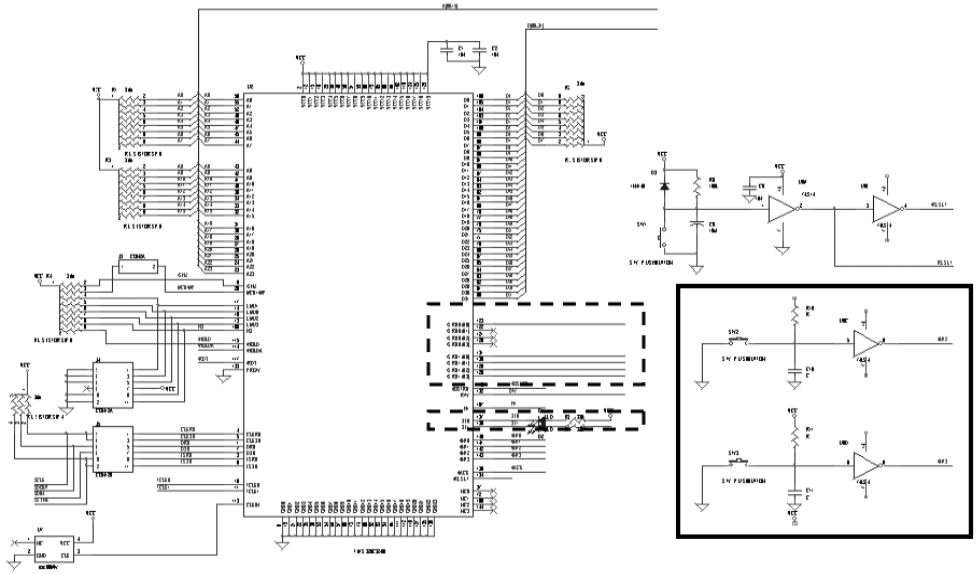
6. 모터 드라이브단



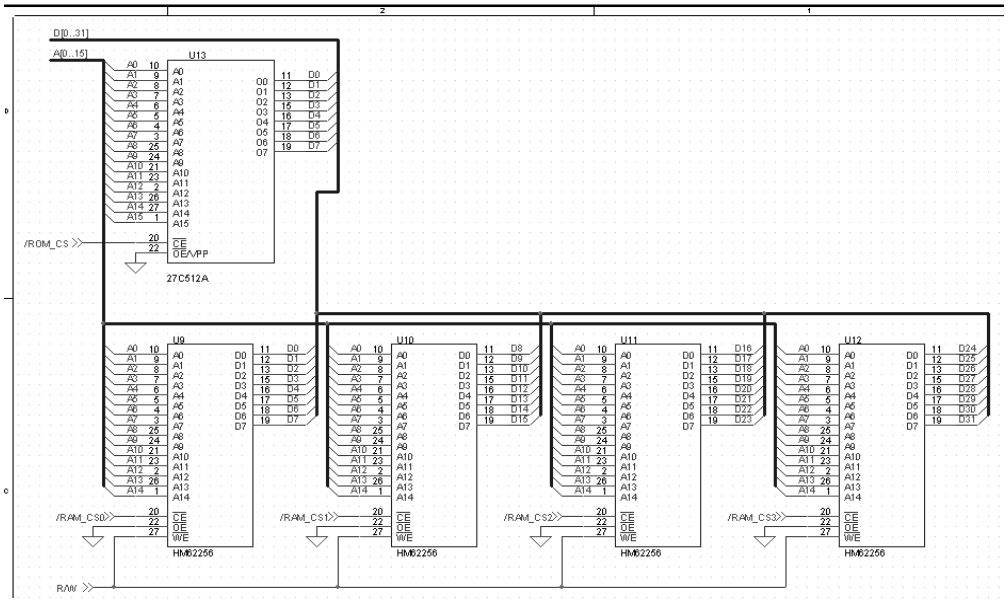
7. 전원부 및 센서부



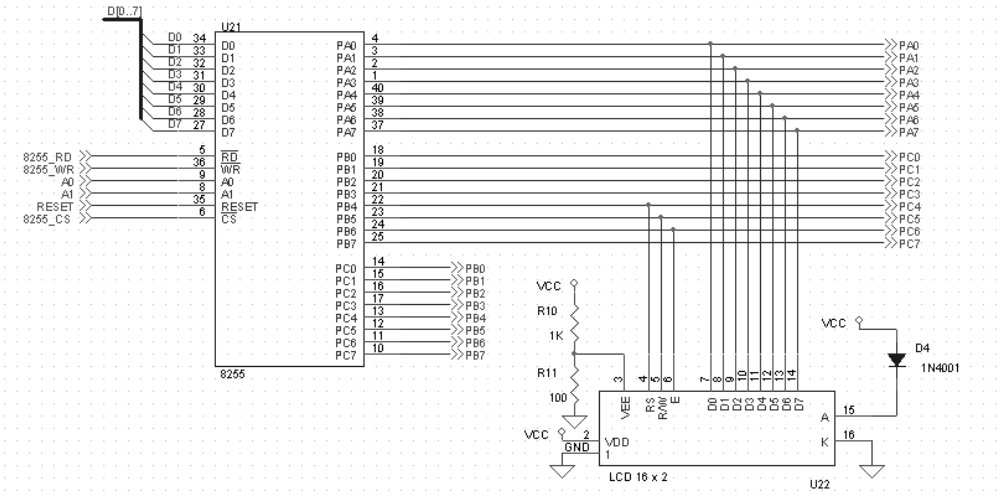
8. 음성 신호처리부



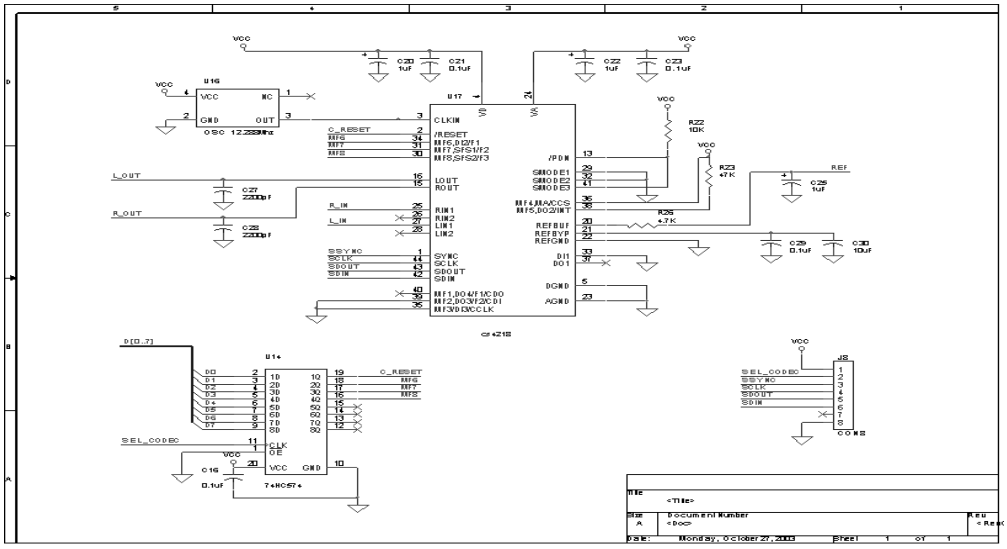
9. 메모리 인터페이스



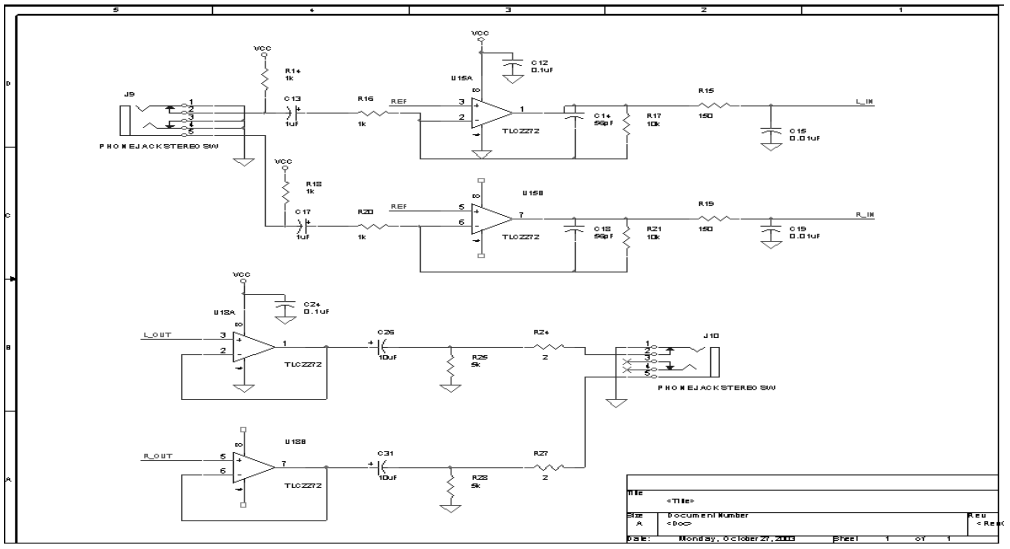
10. I/O 인터페이스



11. 코덱 회로



12. 음성 입/출력부



감사의 글

조도의 바다 바람을 맞으면서 공부한지도 어느새 2년, 짧으면 짧고, 길면 길다는 시간 동안 부족한 저에게 깊은 배려와 관심을 가져주신 지도 교수님 이상배 교수님께 머리 숙여 진심으로 감사의 말씀을 드리고 싶습니다. 그리고 바쁘신 가운데도 부족한 저의 논문을 심사해 주신 박동국 교수님과 임재홍 교수님께도 감사드립니다.

오늘에 제가 있기 까지 학부과정에서 여러모로 힘써주신 최대우 교수님, 그리고 권순량 교수님, 이응주 교수님, 최영복 교수님, 손영선 교수님, 김민성 교수님 김은주 교수님께 진심으로 감사드립니다.

연구실에서 같이 생활한 선배들, 프로젝트와 생활에 많은 도움을 준 김정훈, 김영탁, 이창규(희정씨), 문희근, 강재명 선배님에게 진심으로 감사드리며, 그리고 항상 저에게 힘이 되어주고, 같이 동고동락 했던 우리 동기 강종윤, 김수정, 부족한 저를 잘 따라 준 친구이자 후배인 장원일, 박주원, 정성훈에게도 고맙다는 말을 전합니다.

항상 나와 같이 했던 고등학교 친구 태준과 전문대 친구인 희호, 명섭, 인철, 희정, 성환, 동길형, 승진이형 영규형, 학부 친구이자 형들인 일식, 도진, 순영, 진영, 승훈, 용수형 그리고 경호, 종민, 승현, 정락, 수현, 일원, 경태, 필모형과 형수님들 그리고 정훈형 학부때 같은 연구실에 있었던 동생 영민이와 성진, 태영, 선용형에게 감사드리며, 하시는 일 모두 잘 되시길 빕니다. 항상 학교생활에 활력이 되어 준 전통연구회와 회장 세윤씨, 또 인용, 천수, 동식, 원희, 상현씨에게도 고맙다는 말을 하고 싶습니다. 제가 직장생활 시 많은 힘든 상황에서도 언제나 용기를 주었던 철현, 민주, 동수형에게도 감사합니다.

끝으로 오늘 제가 있기 까지 항상 지켜 봐 주신 어머니, 아버지 앞에 크게 고개 숙여 감사하다는 말 드리고 싶고, 그리고 언제나 따뜻한 말과 힘이 되어준 누나와 자형, 형과 형수님, 그리고 외삼촌과 외숙모, 우리 이모님들과 사촌동생 수정, 해영, 성대에게도 감사하다는 말 전해 드리고 싶고, 저를 위해서 항상 아끼고 성원 해주신 모든 분들에게 이 논문을 바칩니다.

* 미처 적지 못한 _____ 에게도 고맙다는 말 전합니다.