

## TMS320VC5510 DSK를 이용한 음성인식 로봇

### The Robot Speech Recognition using TMS320VC5510 DSK

최 지 현\*      정 익 주\*\*  
Choi, Ji-Hyun      Chung, Ik-Joo

---

#### Abstract

As demands for interaction of humans and robots are increasing, robots are expected to be equipped with intelligibility which humans have. Especially, for natural communication, hearing capabilities are so essential that speech recognition technology for robot is getting more important. In this paper, we implement a speech recognizer suitable for robot applications. One of the major problem in robot speech recognition is poor speech quality captured when a speaker talks distant from the microphone a robot is mounted with. To cope with this problem, we used wireless transmission of commands recognized by the speech recognizer implemented using TMS320VC5510 DSK. In addition, as for implementation, since TMS320VC5510 DSP is a fixed-point device, we represent efficient realization of HMM algorithm using fixed-point arithmetic.

키워드 : 음성인식, 로봇, HMM

Keywords : *speech recognition, robot, Hidden Markov Model*

---

#### 1. 서론

로봇의 발전 경향을 보면, 사람과 점점 유사해짐을 알 수 있다(대표적인 예가 휴보와 아시모라 할 수 있다.)[1]. 사람을 닮아간다는 것은 바꾸어 말하면 사람의 특징을 지닌다고 할 수 있는데, 그러한 특징 중에 하나가 음성을 이용하여 의사 전달을 한다는 것이다. 음성은 인간이 사용하고 있는 통신 매체 중 가장 자연스러운 형태이며, 따라서 인간의 언어인 음성을 로봇이 이해하게 만드는 것이 관건이라 할 수 있다. 이러한 기능을 하는 것이 음성인식기이다.

음성인식이란, 사람의 음성을 컴퓨터나 기계가 이해할 수 있도록 하는 기술이며, 사용 기술에

따라 몇 가지로 구분된다. 첫째로 그 시스템을 어떤 사용자가 사용할 수 있는지에 따라 화자 독립 방식과 화자 종속 방식으로 나뉘어 진다.

두 번째는 사용자가 어떻게 발음하는 것을 인식 하는지에 따라 고립 단어 인식, 연결 단어 인식, 핵심어 인식, 연속음 인식으로 분류될 수 있다.

마지막으로 인식 가능한 어휘 수에 따라 소용량 어휘 인식 방식과 대용량 어휘 인식 방식으로 나눌 수 있다.

본 논문에서 구현할 음성인식기는 적용화자에 따른 분류로 보면 화자 독립 방식이고, 발음 방식에 따른 분류에 따르면 고립 단어 인식 방식이고, 인식 어휘 수에 따르면 소용량 어휘 인식 방법에 해당한다.

본 논문은(그림 1 참고) TI사의 TMS320VC5510 DSK(이하 5510DSK)를 이용해 음성인식을 하고, 그 결과를 마이콤(송신부 PB-3B)이 무선통신모듈(BIM-433)을 통해 데이터를 무선으로 보낸다. 이렇게 송신부가 이루어지고,

---

\* 강원대학교 전자공학과 석사과정

\*\* 강원대학교 전기전자공학부 교수

수신부에서는 상대편 무선통신모듈(BIM-433)에서 데이터를 받아 수신측 마이콤인 PB-3B가 데이터를 해석해 해당 동작을 로봇에게 지시한다. 또한, 음성인식기의 구현에 있어서 5510 DSP는 고정소수점 DSP이므로, 고정소수점연산을 한다.

본 논문에서 송신 부분은 음성을 인식할 때마다 데이터를 내보내게 된다. 하지만 수신측에서는 로봇을 움직여야하기 때문에, 항상 데이터를 기다릴 수 없으므로, 인터럽트 방식으로 데이터를 수신해 데이터가 없을 경우엔 모터제어에 전념할 수 있게 한다.

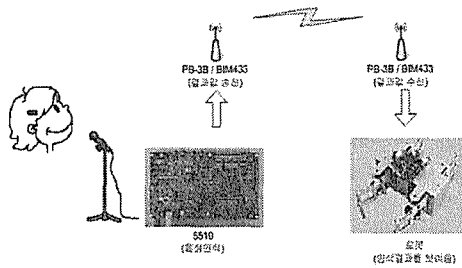


그림 1 시스템의 전체 구성

## 2. 음성인식기

### 2.1 음성인식 시스템의 구조

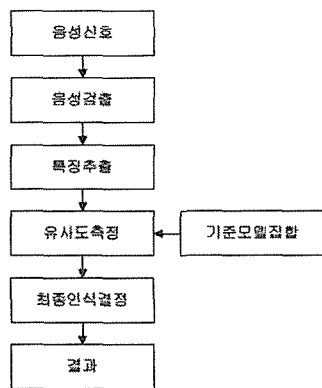


그림 2 음성인식 시스템의 구조

음성인식 시스템은 그림 2의 구성과 같다. 음성 신호가 입력되면, 음성을 검출하게 된다. 음성검출을 하고 나면, 특징추출을 하게 되는데, 이 과정은 음성에 불필요하게 중복되어 있는 정보를 없애고, 다른 음성 신호와의 변별력을 높일 수 있는 정보

를 추출해낸다. 여기서 추출한 특징을 가지고 기준 모델집합인 일종의 데이터베이스와 비교해 확률을 구하는 과정을 유사도 측정이라고 한다. 마지막으로 최종 인식 결정부에서는 인식된 결과의 타당성을 검증한 후 인식 결과를 출력한다. 실용적인 음성인식 시스템이 되기 위해서는 인식 대상이 아닌 음성이 입력되었을 때 이를 거부할 필요가 있는데, 인식 결정부에서는 이러한 거부기능을 수행한다. 유사도 측정부에서는 기준 모델 집합 중에서 입력된 음성과 가장 가까운 기준 모델을 찾기 때문에 입력된 음성이 인식 대상이 아닌 즉, 기준 모델에 포함되어 있지 않은 단어를 발음하여도 일단 가장 가까운 기준 모델을 찾기 때문에 현재 발음한 음성이 유사도 측정부에서 찾은 기준 모델과 가장 가깝지만, 정말 그 단어를 발음한 것인가를 확인할 필요가 있다. 이를 out-of vocabulary rejection이라고 하는데 음성인식 기술 중에서 해결하기 어려운 난제 중 하나이다. 본 논문에서는 rejection 기능은 구현하지 않았다. 이는 유사도 측정부에서 결정한 결과를 그대로 수용한다는 것을 의미하기 때문에 인식 대상이 아닌 임의의 발음을 하여도 인식 대상 중에 하나로 인식하게 된다.

본 논문에서 구현한 인식기는 항상 대기하고 있다가 음성이 검출되면 음성인식기가 처리하는 방식이므로 오인식률이 높아지는 문제점이 생긴다. 최종인식 결정부에서 rejection 기능을 수행하지 않았기 때문에 잡음이나 비명령어도 거부하지 못하고 가장 유사한 것으로 인식하기 때문이다. 이러한 문제점을 보완하기 위해 버튼을 눌러 음성인식을 시작하는 모드를 추가해 음성인식 개시의 시작을 알려 이러한 문제점을 보완하였다.

### 2.2 음성검출

음성검출은 음성인식 성능에 큰 영향을 미치는 부분 중 하나이다. 잘못된 음성 검출은 어떤 음성인식 알고리즘을 사용하느냐에 따라 차이가 있지만, 이후의 모든 과정에 영향을 끼쳐 인식 시스템의 성능을 저하시킨다. 따라서 정확한 음성검출이 필요하다. 특히 잡음이 심한 환경에서는 잡음의 일부가 검출된 음성구간에 포함되거나 또는 검출 구간에 포함되어야 할 음성 신호 부분이 포함되지 않는 경우가 발생한다.

이상적인 경우, 완벽한 검출기는 다음과 같은 특징을 지닌다. 첫째, 주변 환경에 영향을 받지 않는다. 둘째, 음성 신호에서 추출하기 쉬운 파라미터와 일관적인 알고리즘을 이용하여 음성을 검출한다. 셋째, 실시간으로 음성을 검출한다. 그러나 현실적으로 이상적인 음성 검출기의 구현은 어렵다. 따라서 본 논문에서는 시간영역의 파라미터인 에너지원 영교차출만을 사용하여 실시간으로 음성 구간을 검출하고, 주변 환경의 변화에 따라 음성을

검출하는데 사용하는 파라미터의 문턱값을 적응시킴으로써 주변 환경의 영향을 적게 받도록 설계하였다.

### 2.3 특징추출

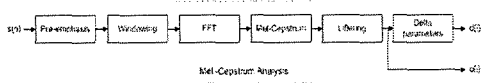


그림 3 음성인식기에서 특징추출을 위해 음성이 처리되는 과정

#### Pre-emphasis

입력된 음성  $s(n)$ 은 pre-emphasis 과정을 거치게 된다. pre-emphasis는 주파수가 높아질수록 성분이 작아지는 음성신호의 고역 부분을 강조하여 음성신호의 주파수 성분을 다소 평탄하게 만들며, 이후 신호처리 과정에서 있을 수 있는 finite precision 효과에 영향을 덜 받도록 하기 위해서이다. pre-emphasis를 위하여 다음과 같이 1차 highpass 필터링을 하게 된다.

$$\bar{s}(n) = s(n) - as(n-1) \quad (1)$$

$a$  는 0.98을 사용하며,  $s(n)$ 과  $s(n-1)$ 은 입력 음성신호를 가리키며,  $\bar{s}(n)$  은 pre-emphasis를 통과한 출력값이다.

#### Windowing

특징추출을 위한 분석이 대부분 주파수 영역에서 이루어지므로, rectangular window에서 기인하는 갑작스런 변화를 완화시켜서 의미있는 주파수 분석 결과를 얻기 위하여 Hamming window를 사용한다.

#### Mel-Cepstrum

Mel-Cepstrum은 최근 음성인식에서 가장 선호되는 특징 벡터 중의 하나로서 다음과 같이 구해진다.

$$c(i) = \sqrt{\frac{2}{P}} \sum_{j=1}^P \log(m_j) \cos\left(-\frac{j\pi}{P}(i-0.5)\right), 1 \leq i \leq N \quad (2)$$

여기서  $P$ 는 분석 차수, 즉 mel-filter bank의 분석 대역의 수이며,  $N$ 은 Mel-Cepstrum의 차수이다. 본 논문에서는  $P=24$ ,  $N=12$ 를 사용하였다.

#### Liftering

분석 알고리즘 자체에서 기인하는 변동성 및 개

별 화자에 따른 변동성의 영향을 줄이기 위하여 liftering을 수행한다. 본 논문에서는 다음의 bandpass filter를 사용하였다.

$$w(n) = 1 + \frac{L}{2} \sin\left(-\frac{\pi n}{L}\right), 1 \leq n \leq L \quad (3)$$

여기서  $L$ 은 22를 사용한다.

#### Delta 파라미터

시변 특성을 반영하는 delta 파라미터는 다음과 같이 구하였다.

$$d_t(i) = \frac{\sum_{r=1}^D \gamma(c_{t+r}(i) - c_{t-r}(i))}{2 \sum_{r=1}^D \gamma^2}, 1 \leq i \leq N \quad (4)$$

$D$ 는 2를 사용하며, 따라서 최종적으로 사용한 특징 벡터의 차수는 12차의 mel-cepstrum 계수와 delta mel-cepstrum 계수를 합한 24차이다.

본 논문에서 구현하는 음성인식기의 front-end를 요약하면 다음과 같다.

- ▷ 샘플링 주파수 : 16KHz
- ▷ 한 프레임의 길이 : 400 샘플 ( 25 msec )
- ▷ overlap 간격 : 200 샘플 ( 12.5 msec )
- ▷ pre-emphasis factor : 0.98
- ▷ 사용 window : Hamming window
- ▷ 특징 벡터 : 24차 분석, 12차 Mel-Cepstrum 생성, 12차 delta Mel-Cepstrum
- ▷ Liftering : 22차 bandpass lifter

#### 2.4 유사도측정 (HMM을 이용한 화자독립 인식기)

HMM은 음성의 발생 과정을 확률론적으로 모델링한다. 그림 4는 임의의 음성과 HMM의 관계를 도식화한 것이다.

본 논문에서는 연속 분포 HMM을 사용한다. 연속 분포 HMM에서는 관측 심볼이 발생할 확률을 다음과 같이 가우시안 mixture의 합으로 표현한다 [2].

$$b_j(O_t) = \sum_{m=1}^M c_{jm} \frac{1}{\sqrt{(2\pi)^M |\Sigma|}} e^{-\frac{1}{2}(o-\mu)^T \Sigma^{-1}(o-\mu)} \quad (5)$$

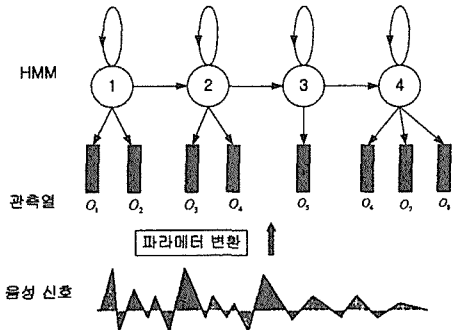


그림 4 HMM과 음성과의 관계

여기서  $\mu$ 는 평균벡터,  $\Sigma$ 는 공분산 (covariance) 행렬이다.  $c_{jm}$ 는  $j$  번째 상태에서  $m$  번째 가우시안 mixture를 위한 가중치에 해당한다. 본 논문에서는 mixture는 연산량을 고려하여 1개만 사용하였으며, 상태의 수는 인식 어휘의 길이에 비례하는 값을 사용하였다. 또한 HMM을 구성함에 있어서 그림 4와 같이 상태 간 skip이 없도록 구성하였다.

### 2.5 고정소수점 연산

TMS320VC5510 DSP는 고정소수점 DSP이다. 여기서 고려할 것은 부동소수점 연산과 달리 고정소수점 연산은 연산 방식에 따라 연산의 정밀도가 달라진다는 것이다.

고정소수점 연산을 위한 숫자 표현을 Q-format 이라고 하며, 이런 표현을 이용해서 연산한다고하여 고정소수점 연산을 Q-math라고도 한다. Q-format은 표현하고 싶은 단위가 작고 섬세하다면 fractional bit를 늘리고, 절대값이 크고 소수가 적당히만 존재해도 된다면 fractional bit를 줄이고, 정수 자리값을 늘릴 수 있는 유연성이 있다.

여기서 중요한 것은 round-off 오차를 최소화하여 부동소수점 연산을 하는 인식기와 같은 수준의 인식률을 얻는 것이다. 이것은 연산 결과간의 기준 Q-format 형식의 scaling을 통해 구현한다.

#### ■ HMM과 Viterbi decoding

표 1의 Words 구조체는 한 단어의 HMM을 나타내며, State 구조체는 특정 상태의 가우시안 mixture를 나타낸다. 단일 mixture를 사용하기 때문에 State 구조체 자체가 하나의 가우시안 mixture이며, VECTOR\_SIZE는 delta mel-cepstrum을 포함한 특징벡터의 차수인 24로 정의 되어있다. HMM 데이터들은 고정소수점 형식으로 표현되어 있는데, mean 벡터는 7.9 Q-format 으로 표현되었으며, variance 벡터는 각

각의 엘리먼트에 대하여 서로 다른 Q-format을 적용하였다. variance 벡터의 모든 엘리먼트들을 하나의 Q-format으로 표현하기에는 각각의 엘리먼트의 다이내믹 레인지가 서로 많이 상이하기 때문에 효과적으로 표현할 수 없다. 즉, mel-cepstrum 계수들을 차수에 따른 분포가 매우 차이가 나기 때문에 variance 벡터는 차수별로 서로 다른 Q-format을 적용하였다. 이를 위하여 훈련을 통해 구해진 모든 특징 벡터의 차수에 따른 variance 들의 다이내믹 레인지를 미리 구하고, 이를 이용하여 variance 벡터를 차수별로 서로 다른 Q-format으로 표현하였다. 그리고 차수별로 어떻게 스케일링 하였는지에 대한 정보를 shift[] 배열에 저장하였다.

표 1 HMM을 위한 구조체

```
typedef struct {
    short mean[VECTOR_SIZE];
    // 7.9 format
    short var[VECTOR_SIZE];
    // (10-shift[i]).(shift[i]+6)
Q-format
    short g_const;
    // g_const + offset : 23.9 Q-format
} State;
typedef struct {
    short num_states;
    // number of state
    short *state_vec;
    short **trans_p;
    // transition prob. matrix, 7.9
Q-format
} Words;
```

Viterbi 알고리즘은 t-1에서 계산된 각 상태의 출력 확률에 t 인 순간의 상태로의 전이 확률을 곱한 후, 가장 큰 확률을 가지는 경로를 선택하고, 이렇게 선택된 값에 t인 순간의 출력 확률을 곱하는 것이다. Viterbi 연산하는 과정에서 연속적으로 확률의 곱셈을 수행하게 되는데, 1보다 작은 값을 계속해서 곱할 경우 아주 작은 값이 되기 때문에 이를 해결하기 위해 log 연산을 이용하여 곱셈을 덧셈으로 바꾸어 계산한다.

### 2.6 음성인식 결과

앞에서 언급했던 바와 같이 최종 인식 결정 단계는 구현하지 않는다. 따라서 가장 유사한 값을 출력하게 되고, 동시에 결과값을 포트를 통해 출력

한다(이 값을 이용해 송신측 마이크가 무선 신호를 보낸다). 이런 경우 송신측에서 음성인식의 결과를 확인할 수가 없다. 로봇을 통해서 확인할 수 있지만, 그것은 오인식 때문인지, 데이터의 오류인지 확인할 방법이 없다. 이러한 문제를 해결하기 위해 ADPCM으로 인식 어휘에 해당하는 단어의 음성을 저장해 두었다가, 데이터를 내보낼과 동시에 스피커에서 결과 음성이 나오게 한다. 음성파 로봇의 행동으로 결과를 확인할 수 있게 되는 것이다.

결과값을 출력하는 포트는 General purpose IO로 8개의 핀이 있지만, 확장포트에 있는 것은 4~7까지의 네 개이다. 따라서  $2^4 = 16$  이므로 16가지의 명령어를 표현할 수 있다. 본 논문은 16개의 단어를 넘지 않는 소용량 어휘 인식기인 관계로, 따라서 0~3 포트를 굳이 칩에 접근해 확장할 필요가 없다[3~5].

다음 그림은 HPI의 GPIO의 포트이다. EXP\_IO\_4~EXP\_IO\_7 로 결과를 표현한다.

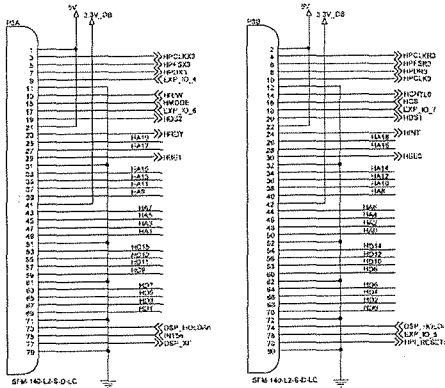


그림 5 결과값을 출력하기위한 GPIO ( EXP\_IO\_4~7 )

### 3. 로봇

#### 3.1 마이크 (PB-3B)

PB-3B에서 중요한 것은 수신측 모듈이므로 송신측에서 보내는 것을 항상 받을 수 있어야 한다는 것이다. 그러기 위해서는 watch dog 방식과 인터럽트를 방식이 있는데, PB-3B의 경우 watch dog 방식으로 데이터를 받으면, 수시로 데이터 확인을 위한 과정 때문에 모터의 움직임이 자연스럽게 않게 된다. 그래서 인터럽트를 사용하게 되는데, 정확한 인식을 위해 level trigger가 아닌 edge trigger 방식으로 인터럽트를 설정한다. (참고로 16

번 핀만 edge trigger를 인식하고, 나머지 다른 포트들은 모두 level trigger로 설정이 된다[6]. 따라서 반드시 16번 핀으로 수신을 해야한다.)

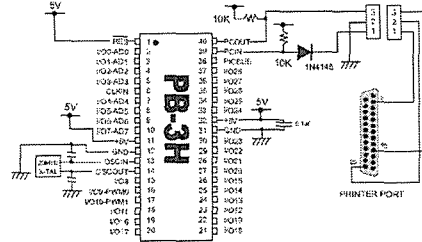


그림 6 PB-3B

#### 3.2 모터 및 모터 드라이버

8개의 서보를 이용한 1족 2자유도 로봇을 사용하기로 한다. 각각의 다리마다 모터가 2개씩, 4개의 다리로 되어 있으며, 각각의 모터는 상/하, 좌/우로 움직이는 기능을 한다. 모터 드라이버로 SMCpro라는 모듈을 사용하게 되는데 이 모듈은 해당 모터 별로 속도와 각도를 지정할 수 있게 되어 있다.

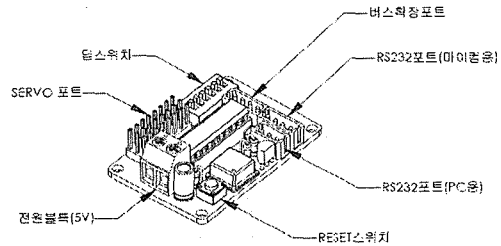


그림 7 SMCpro

#### 3.3 로봇의 움직임

로봇의 움직임은 명령어의 종류에 따라 간다. “앞, 뒤, 좌, 우” 기능과 “인사, 앉아서, 멈춰”의 기능이 있다. 이러한 기능들은 만약 한 동작을 끝내고 비로소 다음 명령어를 받는다면, “멈춰”의 기능은 쓸모가 없어지게 된다. 그러므로 실시간으로 명령어를 전달할 필요성이 생긴다. 그래서 1차로 단계를 작게 나누어 데이터를 확인하는 방식인 watch dog 형태로 구현을 하였으나 모터의 움직임이 자연스럽게 않은 관계로 인터럽트를 사용하기로 한다. 로봇의 움직임은 다음 그림과 같다.

동작의 자연스러움을 위해 동작 단위를 완성된 하나의 가장 작은 단위로 표현하여 움직임의 끊김이 없게 하였다. 그리고 한 cycle의 동작을 끝내면 그 사이 인터럽트에서 동작 MODE가 갱신되었는

지를 검사한다. 만약 동작 MODE가 갱신되었다면 처음으로 돌아가 다시 해당 동작 루틴으로 가고, 아니라면 다시 한 cycle의 동작을 수행하며 이러한 패턴을 반복한다.

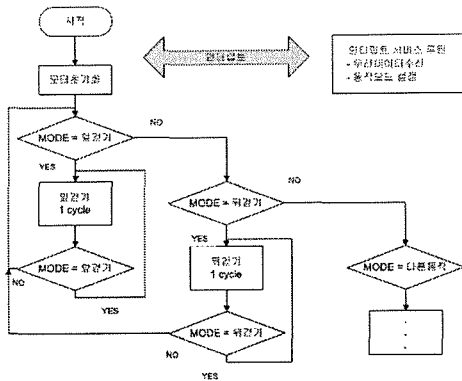


그림 8 로봇 움직임

#### 4. 무선통신

BIM-433 무선 RF 모듈을 사용한다. 통신 모듈의 핀 구성은 그림 9와 같다.

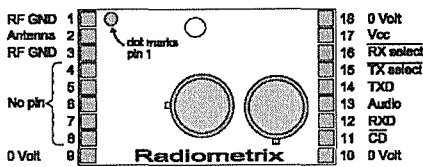


그림 9 BIM-433

무선통신에서 가장 신경써야 할 부분이 바로 정확성이다. 유선통신과 달리 외부의 잡음에 영향을 많이 받기 때문이다. 그래서 보통은 일정한 형식의 패킷 단위로 데이터를 보내게 된다. 패킷의 장점은 다른 것과 구분된다는 점이다. 잡음이거나, 다른 신호원인지를 판단할 수 있는 것이다. 또한 데이터 내의 에러도 알 수 있다. 간단하게는 parity bit, 반전데이터 등으로 데이터의 오류를 찾아낼 수 있다. 본 논문에서는 같은 데이터를 두 번 보내 X-OR 하고, 결과가 0이면 오류가 없다는 것을 확인하는 방법을 사용한다. 패킷의 구성은 표 2와 같으며, 4개의 preamble과 하나의 시작신호(0x00), 그리고 두 개의 데이터로 이루어져 있다.

표 2 패킷의 구성

순서	데이터
1	0x55
2	0x55
3	0x55
4	0x55
5	0x00
6	DATA
7	DATA

송신부에서 이러한 데이터 패킷을 보내면, 수신부는 이 패킷의 구성을 알고 있으므로 검사해 약속된 패킷인지를 검사하고, 약속된 패킷이면 데이터를 추출해 오류가 있는지 확인한 다음 오류가 없다면 최종적으로 해당 MODE를 갱신한다. 이러한 수신부의 동작은 모두 인터럽트 서비스 루틴에서 이루어지므로 MODE 갱신이 끝나면, Main 루틴에서 알아서 동작하게 되어 있다. Main 루틴의 동작은 그림 8과 같고, 인터럽트 수신부는 다음 그림 10과 같다. 그리고 여기서 첫 번째 데이터인 RDATA(0)는 edge trigger라도 시작이라 정확성이 떨어지므로 사용하지 않는다.

하지만 패킷도 완전한 데이터의 정확성을 보장하지는 못한다. 그래서 더욱 정확하게 길고 상세한 패킷을 구성할 수도 있고, 작은 패킷을 여러 번 보낼 수도 있다. 긴 패킷의 경우엔 보내는 보다 확인시 복잡함이 많고, 에러를 정확하게 찾아낼 수 있지만, 다시 보내는 수고 등의 손실이 많다. 따라서 본 논문에서는 작은 단위(표 3)의 패킷을 여러 번 보낸다. 작은 단위의 패킷이라 인터럽트 서비스 루틴에서의 시간 지연도 작고, 같은 동작의 데이터가 여러 번 온다고 해도, MODE의 값이 같으므로 Main 루틴에서의 변화는 없으므로 적당하다 할 수 있다.

#### 5. 완성된 시스템

##### ■ 음성인식부와 송신부

음성인식부분과 송신부의 완성 형태는 아래의 그림과 같다. 개발 단계에서는 CCS(code composer studio)를 이용했지만 완성단계에서는 flash 메모리에 프로그램을 넣어 stand-alone 으로 실행된다.

먼저 음성인식을 하고 그 결과값은 표3의 <5510 GPIO>의 값과 같다.(상위 4 bit를 사용하므로 하위비트는 상관없다.) 송신부의 마이콤은 항상 GPIO의 값을 주시하고 있으며, 값이 갱신 될 때마

다 패킷을 보내며, DATA 값은 GPIO 값에 따라 아래와 같다.

표 3 음성인식의 결과값과 명령어에 따른 패킷의 DATA

명령어	5510 GPIO	DATA
앞으로가	0x10	0x41 = 'A'
뒤로가	0x20	0x42 = 'B'
왼쪽으로가	0x30	0x43 = 'C'
오른쪽으로가	0x40	0x44 = 'D'
멈춰	0x50	0x45 = 'E'
앉아	0x60	0x46 = 'F'
일어서	0x70	0x47 = 'G'
안녕	0x80	0x48 = 'H'

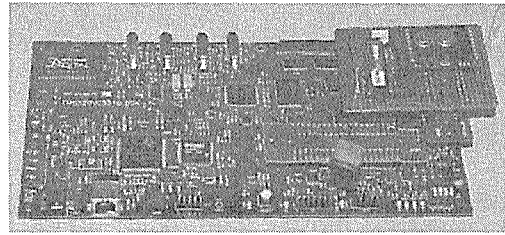


그림 11. 음성인식부와 송신부 완성모습

■ 로봇과 수신부

로봇의 경우 행동 패턴과 무선데이터 수신에 관한 것이며, 이것은 위에서 언급되어 이 부분은 여기서는 생략하기로 한다. 여기서는 수신이 완료되고, 로봇이 움직이므로, 표4와 같이 동작 MODE에 대한 정의만 추가한다.

표 4 로봇의 행동 MODE 값

명령어	MODE
앞으로가	1
뒤로가	2
왼쪽으로가	3
오른쪽으로가	4
멈춰	0
앉아	5
일어서	0
안녕	6

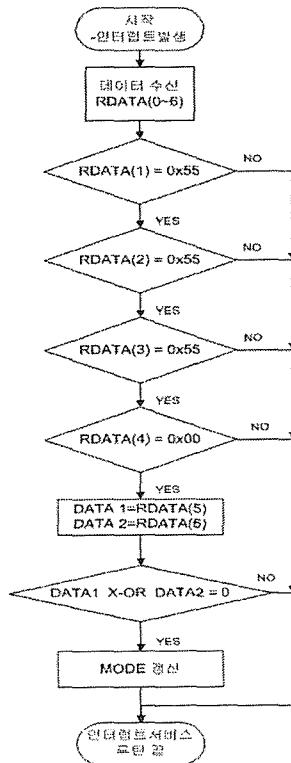


그림 10 수신부의 인터럽트 서비스 루틴

그림 10에서는 단순히 MODE의 값이 바뀌었지만 검사하고 로봇을 움직였다. 하지만 그러한 동작 패턴은 반복된 동작(앞으로가/뒤로가)에서는 문제가 없지만, “왼쪽으로가 / 오른쪽으로가” 등의 어느 일정 시간 동안 또는 어느 일정 각도를 움직이는 것은 계속 반복할 필요가 없으므로, 반복 횟수에 제한을 두어야 한다. 따라서 “앞으로가 / 뒤로가” 명령어의 경우엔 그림 8과 같이 무한 반복을 해도 된다. 그리고 횟수에 제한을 두어야 할 동작은 “앞으로가 / 뒤로가” 동작이다. 보통 1 cycle의 동작으로 30도 정도 회전하게 되는데, 3번 정도를 거쳐야만 방향이 바뀌게 된다. 따라서 1 cycle의 동작을 마치고 MODE를 확인함과 동시에 cycle count도 확인해야 한다. 그리고 3번의 동작이 모두 끝났다면, Main 루틴에서 자체적으로 기본값인 0으로 MODE를 갱신해준다. “멈춰”와 “일어서”의 경우엔 명령어는 다르지만, 같은 행동을

하게 되므로 MODE 값이 같다. 앉아의 경우엔 MODE의 갱신이 필요하지 않고, 계속적인 반복도 꼭 필요하진 않지만, 모터의 피드백을 위해 반복을 하려면, “앞으로가 / 뒤로가”의 행동패턴과 같이 취급한다. 그리고 “안녕” 명령어는 인사는 한번만 하는 것이므로, 항상 그 루틴 마지막에 판단이 아니라 MODE를 0으로 갱신하는 과정과 무조건 시작으로 점프하는 것이 필요하다.

수신부와 송신부 완성한 모양이 그림 12와 같다. 이것은 그림 1의 실제모습이다.

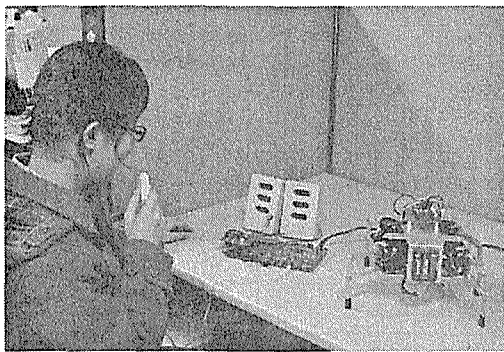


그림 12 실행 모습

## 6. 결론

본 논문에서는 최근 지능형 로봇에서 가장 핵심적인 기능인 음성인식 기능을 고정소수점 DSP로 구현하고 인식된 결과를 무선으로 전송하여 로봇을 음성으로 제어하는 음성인식 로봇을 구현하였다. 본 연구에서는 무선을 이용한 음성인식 제어 방식의 로봇의 실제 응용 가능성을 보이는 것이 목표이므로 전체적인 시스템의 구현에 중점을 두었으나 구현 과정에서 다음과 같은 세부적인 구현 방안을 제시하였다.

DSP를 이용한 음성인식 구현부에서는 고정소수점 연산방식으로 구현이 까다로운 HMM 알고리즘을 연산오류를 최소화하는 방식으로 구현하였다. 이를 위하여 부동소수점으로 구현된 가우시안의 mixture의 분산벡터의 엘리먼트들을 고정소수점 연산에서 사용할 때 차수별로 개별적으로 스케일링하여 연산 오류를 최소화하였다. 또한 전처리 과정에서는 주변 잡음의 영향을 최소화하기 위하여 음성 검출의 문턱값을 주변잡음에 적응하도록 하였다.

두 번째로 인식된 결과를 전송하고 로봇을 제어하는 모듈을 구현하고 이 과정에서 발생한 실제적인 문제들을 해결하는 방법을 제안하였다. 무선 전송 모듈부에서는 DSP와 마이컴 사이의 데이터 전

송을 위한 인터페이스 회로를 구현하였으며 마이컴이 전송 받은 데이터를 무선 송신기를 이용하여 로봇의 수신기 쪽으로 전송하였다. 무선 전송과정에서 발생하는 전송오류를 보완하기 위하여 작은 단위의 패킷을 여러번 전송하는 방법을 제안하였다. 로봇의 수신 모듈 부분에서는 수신된 데이터를 로봇의 마이컴이 해석하여 로봇을 구동하도록 하였다.

결과물의 경우, 개선의 여지가 남아있다. 그중 하나는 데이터 수신 시에 모터의 정상적인 동작에 어려움이 있었기 때문에 polling 방식 대신 인터럽트 방식을 사용하였다는 점이다. 행동 패턴의 경우, 패킷의 구성이 다르므로 바뀌지 않지만, 데이터가 입력될 때마다 인터럽트 서비스 루틴으로 분기하게 되면, 시간적 지연으로 정상적인 동작을 기대하기 어렵다. 한편, BIM의 경우엔 상대편이 없으면, 잡음이 입력되게 되는데, 이런 경우엔 인터럽트 루틴에서 메인 프로그램으로 복귀하지 못해 정상적인 동작이 불가능하다. 만약 또 다른 BIM의 방해전파가 존재한다면, 정상적인 동작이 불가능해지게 된다. 무선통신이라는 관점에서 볼 때 인터럽트 방식보다는 polling 방식이 더 나을 수도 있을 것으로 예상되며 향후 개선해야 할 부분이라고 여겨진다.

음성인식 부분에서는 거절 기능이 구현되지 않았기 때문에 이를 보완하기 위하여 버튼 트리거 방식을 제안하였지만 궁극적으로는 거절 기능을 구현하여 음성 명령 시에 버튼의 조작 필요성이 없도록 해야 할 것이다.

음성인식은 상용화가 된 기술이지만, 아직 단어의 수준에 머물고 있으며, 사람의 대화를 이해하는 수준에는 이르지 못하고 있다. 지금까지의 성과도 있지만, 아직은 가야할 길이 멀다고 할 수 있다. 사람과 컴퓨터가 원활히 대화할 수 있을 정도로 기술이 발전한다면, 음성인식 기술의 적용 분야는 지금보다 더욱 더 넓어질 것이다.

## 참 고 문 헌

- [1] 뉴미디어통신공동연구소, <http://147.46.66.129/>
- [2] 김윤경, 주상열, “확률의 이해” 敎友社, 2002
- [3] TMS320C55x DSP CPU Reference Guide
- [4] TMS320C55x DSP Peripherals Reference Guide
- [5] TMS320VC5510 / 5510\_dsk\_scheme
- [6] BASIC으로 움직이는 초소형 컴퓨터 User's Manual Vol 10, Comfile