

시각 장애자를 위한 음성 지원에 관한 연구

장성호, 함광근, *최승호, **민홍기, 허웅

명지대학교 전자공학과, 동신대학교 정보통신공학과, 인천대학교 정보통신공학과

A Study on Speech Support for the Blind

S.H. Jang, K.K. Ham, *S.H. Choi, H.K. Min, W. Huh

Dept. of Electronic Eng., Myong Ji Univ.
* Dept. of Information Telemetering Eng., DongSin Univ.
** Dept. of Information Telemetering Eng., Inchen Univ.

ABSTRACT

In this paper, we proposed a speech support system of personal computer for the blind. The system is consist of hardware part and software part. The hardware part are consist of personal computer and sound card. The software part are consist of sound driver system, character table and sound output algorithm. This system can recognize inputted characters from keyboard and program produced character strings.

I. 서론

시각 장애자의 시각을 대신하기 위해 촉각과 청각의 형태로 정보를 받아들일 수 있는 시각 보조 장치가 많이 연구된 바 있다.[1]

시각 장애자 중 청각기능이 정상인 경우, 기계적인 도움을 받으면 필요로 하는 정보의 양을 증진시킬 수 있다. 최근에는 시각 장애자를 위한 컴퓨터 지원용 정보전달장치가 개발되고 있다. 현재 실생활에도 많이 보급되어 널리 사용되고 있는 컴퓨터는 기본적으로 시각 장애자를 위해 음성이 지원되는 장치가 적어서 대부분의 정보를 다른 감각기관을 이용해서 수행해야 한다. 그러므로, 시각 장애자들이 다른 보조 기구 없이 직접적으로 컴퓨터를 다룬다는 것은 어려운 일이다.

미국의 TSI(Telesensory System Inc)에서는 점자 디스플레이를 지원하며, 키보드 입력이 가능하고, 음성을 지원하는 장비를 발표하였는데, 이 장비는 RS-232C를 통해 컴퓨터와 통신을 하며 사용한다. 컴퓨터의 확대보급으로 인해 시각 장애자들도 이들을 접할 기회가 많아졌다.

따라서, 본 연구에서는 키입력을 통해 도스 명령 및 입력 문자에 대한 정보를 음성으로 제공하여 정상인과 같이 컴퓨터를 사용할 수 있도록 음성을 지원하는 시스템을 제안한다.

시스템의 구성은 퍼스널 컴퓨터와 사운드 블라스터 음성 지원카드로 된 하드웨어 시스템과 독립된 기능을 지원하는 소프트웨어 시스템으로 구성하였다.

II. 음성 지원 시스템

1. 문자의 음성 출력 알고리즘

음성 출력 알고리즘은 기본적으로 템상주 프로그램을 사용한다. 템상주를 사용하는 이유는 사용자가 키보드를 통해 문자를 입력하는 것을 검사하여 이 문자를 미리 메모리 상에 상주시킨 문자 테이블에 저장되어 있는 사운드 블라스터 카드가 합성할 수 있도록 입력 문자의 한글 발음을 영문으로 만들어 음성을 출력하게 하기 위한 것이다.

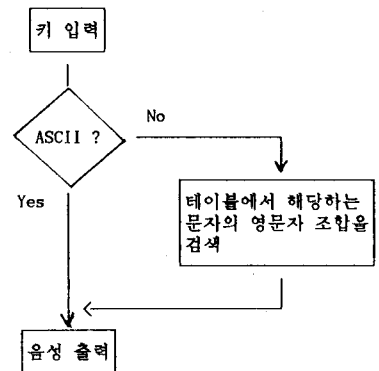


그림 1. 음성 출력 알고리즘
Fig. 1. Algorithm for speech output

예를 들어, 컴퓨터를 사용하는 시각 장애자가 도스 프롬프

트에서 'a' 키를 누르면, 이것을 작성된 음성 출력 루틴으로 넘겨서 음성 카드를 통해 'a' 라는 음성 출력이 이루어지게 된다. 그런데, 입력되는 문자는 일반적인 ASCII 코드(1 byte)나 확장코드(2 byte)이다. 따라서, 입력된 문자가 ASCII 코드인지 확장 코드인지에 따라 출력하는 순서가 달라진다.

ASCII 코드인 알파벳의 경우는 문자의 ASCII 값을 그대로 출력부로 넘겨 주어 음성 출력을 하게되며, 그 밖의 +, - 등 부호의 ASCII 코드와 기능키(F1 - F12), 특수키(INS, DEL, HOME 등)등의 확장코드를 가지는 문자들은 해당하는 한글 발음, 즉 '+' 는 '더하기[do ha ki]', '-' 는 '빼기[pe ki]' 등과 같이 소리나는 대로 영문자로 조합되어 만들어진 문자 테이블에서 해당 문자의 영문자 조합을 찾아내어 음성을 출력하게 된다.

2. 문자 테이블의 작성

연구에 사용된 사운드 블라스터 카드는 미국에서 만들어진 것으로 영문자를 음성으로 합성하여 출력할 수 있다. 키보드에서 입력된 문자를 음성으로 출력할 때, 알파벳과 같은 순수 영문자는 문자 그대로 문자 테이블을 거치지 않고 음성으로 출력이 가능하고, 그 밖의 키보드상의 문자들이 입력되었을 경우는 영문으로 발음되면 알아듣기 힘들게 된다.

예를 들어, '+' 의 경우 ASCII 코드를 변환하지 않고 음성으로 바로 출력하게 되면, 영어 발음인 '플러스[plus]'로 출력되게 된다.

이 문제를 해결하기 위해 이같은 문자들에 대해서는 특별하게 문자 테이블을 만들었다. 이 문자 테이블은 문자를 읽을 때의 한글 발음이 소리나는 그대로를 실험적으로 다양한 영문자의 조합으로 표현하였을 때 얻어낸 음성 출력 결과를 바탕으로 하여 가장 알아 듣기에 적당한 것만을 모아서 테이블을 만들었다.

3. 램상주의 초기화

입력된 키를 검색하고 음성을 출력하는 모든 루틴은 램상주로 구동된다. 음성 출력 프로그램의 램상주를 위해 가장 먼저 해야 하는 작업이 필요한 초기화를 하는 것이다

먼저, 키에서 입력된 문자를 램상주된 음성 출력 프로그램에서 받아들이고 램상주 프로그램을 활성화시키기 위해 키보드 인터럽트(INT 9H)의 ISR을 음성 출력 프로그램을 활성화시키는 서비스 루틴으로 변경하였다. 키보드 인터럽트는 키가 입력될 때마다 발생하는 것으로 본래의 키보드 인터럽트의 ISR 주소를 다른 곳에 저장하여 보존한 다음 원하는 ISR로 변경하면 이후에 키가 입력될 때마다 음성 출력 프로그램이 활성화된다.

두번째로는 음성 출력 프로그램을 메모리에 상주시키는 작업을 한다. 이는 C 의 표준 함수인 'keep()'을 사용하여 쉽게 할 수 있다. keep() 함수는 실행중인 프로그램을 DOS 의 0x31함수를 사용하여 패러그래프(1 PARAG = 16 byte) 단위의 메모리에 남겨두고 도스로 프로그램제어를 넘겨주는 역할을 한다. 즉, 전체 비상주부분을 차지하고 있던 응용프로그램이 keep()에서 지정한 메모리 공간을 제외한 나머지 메모리 공간을 다른 프로그램이 사용할 수 있도록 반납하게 한다.

상주시 필요한 메모리는 실행 파일의 크기와 전역변수의 데이터형을 모두 더한 것을 16으로 나누어 패러그래프(paragraph)로 나타낸 값이다. 예를 들어, 본 음성 출력 프로그램의 경우는 실행 파일의 크기가 20000 byte를 넘지 않고, 전역변수로 정의된 문자 테이블도 440 byte이고, 여기에 나머지 전역변수들의 크기(2017 byte)를 더해서 상주시 에 필요한 메모리를 구하면,

$$(20000 + 440 + 2017) / 16 \approx 1400 \text{ Paragraph}$$

정도가 되므로 여기에 약간의 여유 메모리를 더해서 2000 Paragraph 정도면 충분하다. 실제로 프로그램을 실행하기 전과 실행 후의 사용 가능한 컨벤셔널(conventional) 메모리의 양을 비교하면, 실행전에 603 Kbyte 정도였으나, 실행된 후에는 540 Kbyte 정도의 자유 메모리가 남아서 다른 응용 프로그램의 실행에 지장이 없다. 또, 음성 출력이 가능하려면 먼저 음성 드라이버를 구동해야 하는데 이는 카드에서 제공하는 표준 음성출력드라이버인 SBTALKER.EXE와 BLASTER.DRV를 사용하였다.

II. 시스템 구성

1. 하드웨어 시스템

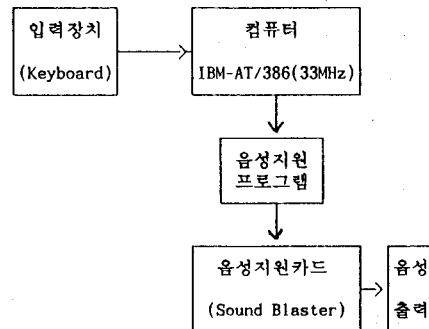


그림 2. 하드 웨어 시스템 구성

Fig. 2. Hardware System Configuration

입력장치는 표준 입력장치인 키보드를 사용하고, 컴퓨터는 IBM-AT/386(33MHz)를 사용하였다. 컴퓨터 내부에서는 프

로그그램을 통해 사운드 블라스터 카드의 음성으로 출력될 내용의 문자열을 만든다. 프로그램에서 넘겨진 문자열은 사운드 블라스터카드로 넘겨지고, 실제적인 음성은 이 카드의 내부 IC에서 합성되어 외부 스피커를 통해 출력된다.

2. 소프트웨어 시스템

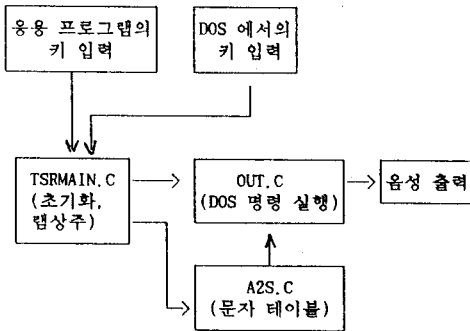


그림 3. 소프트웨어 시스템 구성
Fig. 3. Software System Configuration

1) TSRMAIN.C

램상주를 위한 여러가지 처리를 한다. 커서위치의 저장, 현재 실행중인 프로그램의 인터럽트 벡터 테이블을 저장한 다음에 새로운 벡터로 변경하고, 램에 상주하고, 키버퍼를 점검하여 키가 입력되었는지를 검사한다. 그리고, 키가 눌리면 그 문자값을 OUT.C로 넘긴다.

2) OUT.C

음성 출력 지원 드라이버가 설치되었는지를 점검하고 설치된 것을 확인하면, TSRMAIN.C에서 넘겨 받는 문자를 ENTER가 눌릴 때까지 버퍼에 저장하여 나중에 명령을 실행하는 과정에 사용된다. 그리고, 다시 문자를 A2S.C로 넘겨준다.

버퍼에 저장된 개개의 문자들은 ENTER가 입력되면 한번에 음성 출력이 이루어진다. 그리고, 그 문자들이 도스의 명령, 예를 들어, 'dir'이라는 명령의 경우는 그 명령을 실행한다.

3) A2S.C

문자에 대해 실제의 음성 출력을 이루는 부분으로 입력된 문자가 ASCII인지 확장키인지를 판별하여, 미리 만들어진 문자 테이블에서 해당하는 문자 발음의 영문자 조합을 찾아서 사운드 블라스터의 개발자 들에서 제공하는 sbts_say() 함수를 이용하여 음성으로 출력한다. 이 테이블은 영문자의 경우는 그대로 출력하고, 그 밖의 문자나 기능

키는 한글 발음을 기준으로 만들어져 있다.

IV. 실험 및 고찰

프로그래밍 언어로는 Turbo-C와 Borland C++ ver 3.1 컴파일러를 사용하였다.

본 프로그램을 사용하여 도스 프롬프트에서 문자를 입력했을 때, 개개의 문자가 음성으로 출력되는 경우에 영어 알파벳의 발음은 듣기에 지장이 없으나 한글 발음으로 바꾼 부호나 그 밖의 키의 경우는 약간 모호하므로 학습이 필요했다. 또, 버퍼에 저장하였다가 도스 명령을 실행할 경우에 일반적인 영어 단어가 아니면 엉뚱하게 발음하는 경우도 있었다.

문자 출력 속도는 보통의 문자 출력때보다 음성 출력이 이루어지는 시간이 좀더 걸렸고, 메모리는 음성출력 드라이버 프로그램과 본 프로그램이 메모리를 차지하고나서의 사용가능한 메모리는 540 Kbyte정도 남았다. 이는 다른 응용 프로그램을 실행시킬 수 있는 크기이다.

V. 결론

본 알고리즘은 다른 기존의 시각보조장치를 사용한 컴퓨터 이용 방법보다 비교적 저렴한 장비인 음성합성카드만을 사용하여 이루어졌기 때문에 좀더 쉽고, 보편적으로 사용될 수 있다.

음성카드를 사용하여 키보드에서 입력된 문자를 음성으로 출력하게 함으로써 키 입력이 바쁜지를 알수있고, 잘못된 키를 눌렀을 경우도 쉽게 수정할 수 있다. 입력이 끝나면 지금까지 입력된 명령을 다시 확인하여 들을 수 있다.

본 시스템을 사용하면 도스 프롬프트에서 시각 장애자들도 여러가지 도스 명령을 실행하는 것이 가능하다. 화면을 볼 필요가 없이 단지 키를 입력함으로써 출력되는 음성을 들으며 작업을 할 수 있다.

그러나, 아직 램상주부분이 불안전하고, 다른 프로그램을 실행했을 때의 결과를 기대할 수는 없다. 앞으로 이부분 대한 연구를 계속할 계획이다.

[참고 문헌]

[1] 이양희 "음성합성 기술 개발 현황 및 전망" 음성통신 및 신호처리 워크샵 논문집, pp. 88 - 93(1992).
 [2] 伊藤部達, "感覺代行", BME, vol. 2, No. 11, pp. 715-721, 1988
 [3] 임안수, "시각장애자의 정보접근", 視覺障礙研究, 제7집, pp. 3-24, 1991
 [4] Steven Holzner, "Advanced Assembly Language", Brady, pp. 187-211, 1991.