



HAL
open science

Supplementary Document

John Doemer

► **To cite this version:**

| John Doemer. Supplementary Document. 2017. hal-01519184

HAL Id: hal-01519184

<https://hal.science/hal-01519184>

Preprint submitted on 6 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

제32회 한국정보올림피아드
공모대회 (심사 참고논문)

나랏말쌈

영어·불어 외래어의

한글 맞춤법 표기 프로그램 개발

- 사라진 훈민정음 네 글자(뽕, 풍, △, ·) 복원 -

2015. 7. 20

구 분	성 명
소 속	서울 역삼중학교 2학년 3반 10번
출 품 학 생	김 승 민

목 차

1. 주제선정 동기	1
1.1. 한글맞춤법 외래어표기법의 현실화	1
1.2. 외국어 원음과의 발음괴리 해소	2
1.3. 훈민정음 창제 원리 계승	3
1.4. 정확한 외래어 표기의 필요성	4
2. 탐구 목표	5
2.1. 영어 외래어표기법 자동변환 프로그램 개발	5
2.2. 훈민정음의 사라진 4 글자 복원	5
2.3. 원음에 가까운 외래어표기법 구현	6
3. 프로그램 알고리즘	7
3.1. 원어의 외래어표기 변환 방법	7
3.2. 메인 폼(Form) 설계	8
3.3. 원어(영어)를 국제음성기호(IPA)로 변환	9
3.4. 국제음성기호(IPA)를 외래어표기로 변환	13
3.5. 프랑스어 추가사항	20
4. 프로그램 기능	21
4.1. 기본 화면	20
4.2. 발음기호 선택 화면	20
4.3. 원어에 가까운 외래어 표기	22
5. 결론 및 향후 계획	23
5.1. 탐구 결론	23
5.2. 향후 계획	23
6. 참고문헌	23

1. 주제선정 동기

1.1. 한글맞춤법 외래어표기법의 현실화

문자는 자연적으로 발생한 음성언어를 가시적인 방법으로 기록하기 위해 만든 것이다. 이렇게 함으로써 사람들은 문자를 통해 자신들의 의사를 기록하고 다른 사람들에게 전파할 수 있었다. 따라서 문자는 가급적 음성언어를 정확히 표현할 수 있도록 표기되는 것이 바람직할 것이다. 그리고 사람들이 만들어 낼 수 있는 음성의 수는 사실상 무한하다보니 사회 구성원들 간에 사용하는 음성의 수를 한정할 필요가 있으며 또한 이렇게 한정된 음성에 대응하여 문자의 수도 한정되어 사용된다. 그러나 서로 다른 사회의 구성원들 간에서는 사용하는 음성의 종류가 서로 다르다 보니 다른 사회의 음성을 문자로 표기하는데 문제가 발생하는 것은 어쩌면 당연한 일일 것이다.

현대에는 서로 다른 사회들 간의 교류가 과거와는 비교할 수 없을 정도로 빈번하며, 특히 방송, 인터넷의 영향으로 인해 서로 다른 사회들 간의 문화적 교류에는 시간적 공간적 제약이 없을 정도이다. 이로 인해 다른 사회들 간의 언어 교류 역시 급속하게 확대되었으며 이에 따라 다른 사회의 언어에 대한 문자 표기도 확대되었다.

우리나라에서도 그 수가 급격히 증가하는 외래어의 글자 표기를 통일하기 위해 정부(국립국어원)에서 한글맞춤법 외래어표기법이라는 표준을 제정하여 시행하고 있으나 다른 나라의 음성언어를 우리나라 글자로 표기하는데 따른 실제 음성과 글자의 음가에 차이가 발생하는 문제점은 피할 수 없다. 그러다보니 동일한 외래어에 대한 한글 문자 표기가 2개 또는 그 이상 사용되는 경우도 빈번하다.

예) 후레쉬/플래시, 휘날레/피날레, 화이바/파이버 등

실제로 이러한 문제는 언어학자들뿐 아니라 언론이나 방송에서도 종종 제기되고 있는데, 2013년 6월 어느 날 저녁, 모 라디오 프로그램 DJ의 멘트는 이러한 논란이 단지 학계의 문제만이 아니라는 것을 보여주고 있다.

“우리가 흔히 *fighting, fighting*하는데 이걸 *화이팅*이라 해야 할지 *파이팅*이라 해야 할지 늘 헷갈립니다. 외래어의 P 발음과 F 발음이 구분 표기된다면 지금보다 훨씬 사용하기 편할 텐데 말이죠.”

사실 이런 문제점은 외래어표기법 제1장 제1항의 규정으로 인해 현행 맞춤법 체계에서는 불가피한 상황이다.

<외래어표기법 제1장 제1항>

외래어는 국어의 현용 24자모만으로 적는다.

즉, fighting의 영어발음을 정확히 알고 있다 하더라도 현행 외래어표기법에서는 이를 원 발음에 가깝게 표기할 방법이 없는 것이다. 현행 외래어표기법에 따르면 fighting은 파이팅으로, found는 파운드로 적어야 한다. 그러나 이렇게 적으면 found와 pound는 모두 파운드로 표기되어 전혀 구분이 되지 않는다.

1.2. 외국어 원음과의 발음피리 해소

일반적으로 영어 자음 발음의 개수는 24개(반모음 /j/, /w/ 포함)로 보며 국제음성기호(IPA¹⁾)로 표기된 영어 자음 발음은 아래와 같다.

j	w	b	d	f	g	h	k	l	m	n	p
r	s	t	v	z	ð	ŋ	θ	ʃ	ʒ	ʤ	ʦ

- 영어음성학 4.1절 자음의 종류, 구희산, 한국문화사(1998)

이들 중 현대 한글로 구현할 수 없는 음성은 /f/, /v/, /z/, /θ/, /ð/ 등이다. 현행 외래어 표기법에는 /f/는 ‘프’으로 /v/는 ‘브’으로 /z/는 ‘즈’ 등으로 표기하도록 되어 있으나 이럴 경우 /f/는 /p/와, /v/는 /b/와 /z/는 /ʤ/ 또는 /ʒ/와 구분되지 않는 문제점이 발생한다.

모음의 경우 자음보다 복잡하다보니 여러 가지 이론이 있는데 장단음, 이중모음을 고려하지 않는다면 10개 정도로 볼 수 있으며 국제음성기호(IPA)로 표기된 영어 모음의 발음은 아래와 같다.

i	e	æ	ə	ʌ	a	o	u	ɑ	ɔ
---	---	---	---	---	---	---	---	---	---

- 영어음성학 3.2절 모음의 분류, 구희산, 한국문화사(1998)

1) International Phonetic Alphabet: 국제음성학회(International Phonetic Association)에서 제정해 발표한 음성 기호

이들 중 현대 한글로 구현할 수 없는 음성은 /ʌ/, /ɔ/이다. 현행 외래어표기법에는 /ʌ/는 ‘어’로 /ɔ/는 ‘오’로 표기하도록 되어 있으나 이럴 경우 /ʌ/는 /ə/와, /ɔ/는 /o/와 구분되지 않는 문제점이 발생한다.

1.3. 훈민정음 창제 원리 계승

1443년 창제 당시에 훈민정음은 자연에서 나는 대부분의 소리를 표현할 수 있도록 만들어졌으며 지금 기준으로도 자연의 소리를 대부분 표현할 수 있는 세계 유일의 과학적 표음문자로 알려져 있다. 따라서 현대 한글로 구현할 수 없는 음성이라도 훈민정음 글자를 이용할 경우 모두 표현할 수 있으므로 훈민정음을 이용한 외래어 표기법은 좋은 해결 방안이 될 수 있다.

훈민정음이 28자의 기본글자로 이루어 졌다는 사실에 대해서는 대부분의 학자들 사이에서 이견이 없으며 28개의 기본글자는 아래와 같다.

- 자음(17자) : ㄱ, ㅋ, ㅇ(꼭지이음), ㄷ, ㅌ, ㄴ, ㄹ, ㅍ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ(여린히음), ㅎ, ㆁ, ㄷ, ㄷ(반치음)
- 모음(11자) : ㆍ(아래아), ㅡ, ㅣ / ㅜ, ㅛ / ㅗ, ㅑ / ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ

훈민정음 자음 글자의 경우 기본글자 외에도 현대에서 일부 사용되고 있는 병서²⁾와 지금은 사용되지 않는 연서³⁾가 있다. 훈민정음 자음 글자의 구성도는 [표 1]과 같다.

[표 1] 훈민정음 자음 글자 구성도

원형문자	가획자	이체자	병 서		연서
			각자병서	합용병서	
ㄱ ㄴ ㅍ ㅓ ㅇ	ㅋ ㅌ, ㅍ ㅑ, ㅓ ㅕ, ㅗ ㅛ, ㅎ	ㅇ (꼭지이음) ㄷ ㄷ(반치음)	ㄱ, ㅋ, ㆁ ㄷ, ㅌ, ㆁ ㅑ, ㅓ, ㆁ	ㅍ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㆁ ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㆁ ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㆁ	ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㆁ
5자	9자	3자	6자	10자	4자
기본글자 17자			병서 16자		연서 4자

2) 자음 2개 또는 그 이상을 가로로 조합하여 사용하는 것으로 같은 글자를 조합하여 사용하는 각자 병서(ㄱ, ㅋ 등)와 다른 글자를 조합하여 사용하는 합용 병서(ㄱ, ㅋ 등)가 있다.

3) 자음 2개 또는 그 이상을 세로로 조합하여 사용하는 것으로 ㅑ, ㅓ 등이 그 대표적인 예이다.

1.4. 정확한 외래어 표기의 필요성

현행 외래어표기법의 관점에서 볼 때 우리나라의 외래어 표기가 통일된 안정 국면에 들어서기에는 아직 요원한 것으로 보인다. 우리가 사용하고 있는 후라이드 치킨, 바베큐, 주스, 케익, 도너츠 등의 표기법은 현행 외래어표기법에 따르면 모두 틀린 표현이다. 올바른 표기법은 각각 프라이드 치킨, 바비큐, 주스, 케이크, 도넛이다. 이러한 관행적인 오류가 발생하는 이유는 현행 외래어표기법이 복잡하고 배우기 까다롭기 때문이기도 하지만 현행 외래어표기법이 원어의 음가를 제대로 반영하지 못함에 따라 문자의 가장 중요한 요건인 음성언어와의 일치성이 훼손되었기 때문이다.

문자와 음성언어와의 불일치성 문제는 외국인이 한국어를 배우려고 하거나 반대로 영어 등의 외국어를 배우려는 사람들에게 장애요인이 될 수 있다.

2. 탐구 목표

2.1. 영어 외래어표기법 자동변환 프로그램 개발

앞에서 살펴본 바와 같이 외래어표기법은 복잡하고 까다롭기 때문에 이 분야에 전문가가 아닌 이상에는 틀리기 쉽다. 따라서 외국어를 입력하면 그 표기법을 자동으로 제시하는 프로그램이 있다면 올바른 맞춤법 사용에 도움이 될 것이다. 우선 여기서는 외래어 표기법의 상당 부분을 차지하는 영어 외래어에 대한 표기법을 자동으로 제시해 주는 프로그램을 개발하였으며 향후 다른 언어로 확대하고자 한다.

프로그램 개발도구는 Visual Studio C/C++2010의 Form Application을 이용하였다.

2.2. 훈민정음의 사라진 4 글자 복원

1933년 조선어학회는 한글 맞춤법 통일안을 제정하면서 한글 표기방식의 표준을 마련하였으며, 1940년에는 외래어 표기법을 출판하였다. 일제강점기인 이때 만들어진 한글 맞춤법 제1항에는 ‘자모의 수를 40개로 제한’한다고 밝혔고, 자음 19개와 모음 21개로 정해진 자모의 수는 현재까지도 그대로 유지되고 있다.

외래어를 표기하기 위해 자모의 수를 늘이는 것에 대해서는 국어학자들 간에도 논란이 많다. 원음에 가까운 외래어를 구현하기 위해 새 문자를 도입해야한다는 주장과 외래어 표기를 위해 새 문자를 도입하는 것은 주객이 전도된 것이라는 주장으로 나뉘고 있다. 사실 새로운 문자를 도입하는데 따른 비용이 만만치 않다는 점도 새 문자의 도입을 반대하는데 힘을 실어주고 있다. 따라서 이 프로그램에서는 사라진 훈민정음 글자를 복원한 외래어표기와 더불어 현행 한글 맞춤법에 따른 외래어표기를 모두 구현하였다.

이 프로그램에서는 일단 훈민정음 창제 당시 존재했지만 지금은 사라진 글자 중 ㅄ(순경음 비읍), ㅆ(순경음 피읍), ㅅ(반치읍), ㅆ(아래아) 4글자를 복원하여 각각 /v/, /f/, /z/, /ɔ/를 원음에 가깝게 표기하였다. 이 4글자를 우선적으로 복원한 이유는 영어발음을 원음대로 구현하기 위해 필요한 글자이면서도 그 음가가 비교적 정확하게 전해지기 때문이다. 다만 ㅆ(순경음 피읍)의 경우 창제 당시에는 중국어 표기를 위해 사용된 글자로 실제 발음에는 쓰이지 않은 단어로 알려져 있지만 ㅄ(순경음 비읍)이 /v/에 대응되는 원리를 준

용하여 /f/발음으로 택하였다. 복원된 4 글자와 연관된 국제음성기호와 이에 대응되는 한글 글자의 관계는 [표 2]와 같다.

[표 2] 복원된 4글자와 연관된 국제음성기호와 한글 대조표

국제음성기호	p	f	b	v	ɟ	z	o	ɔ
현대 한글 (공식 규정)	ㅍ	ㅍ	ㅂ	ㅂ	ㅈ	ㅈ	ㅅ	ㅅ
훈민정음 (임의 적용)	ㅍ	퐁	ㅂ	빙	ㅈ	Δ	ㅅ	•

- 외래어표기법 제2장 표기일람표, 국제음성기호와 한글 대조표 중 발췌

2.3. 원음에 가까운 외래어표기법 구현

외래어 표기법 제1장 5항에는 ‘이미 굳어진 외래어는 관행을 존중’하도록 되어 있는데, ‘바나나’, ‘라디오’를 그 예로 들 수 있다. 그러나 여기서 말하는 관행의 대부분은 과거의 일본식 발음에서 기인한 경우가 상당수라 음성언어와 글자의 일치성이라는 관점에서 볼 때 올바른 외래어 표기법의 장애요인이 될 수 있다. 따라서 영어 외래어의 현대적 한글 맞춤법 표기를 시도하기 위해 이 프로그램에서는 비록 관행으로 굳어진 외래어라 하더라도 가급적 원음에 가깝게 표기하고자 하였다.

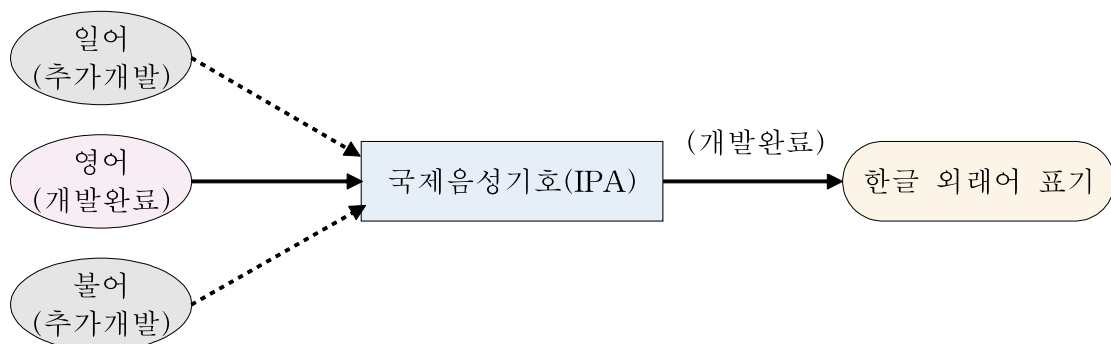
3. 프로그램 알고리즘

3.1. 원어의 외래어표기 변환 방법

원어표기를 외래어표기로 변환하는 방법으로는 우선 원어표기와 외래어표기를 일대일 대응시킨 데이터베이스(DB)를 구축하는 방법이 있다. 그러나 이 방법은 등록해야 할 외래어가 많을 경우 DB를 구축하는데 시간이 많이 걸리며 새로운 외래어가 나타날 때마다 일일이 DB에 추가해야 하는 문제가 있다. 실제로 국립국어원 홈페이지에 있는 외래어표기법 용례가 이런 방식으로 되어 있는데 현재 5만 7천여 건의 데이터가 등록되어 있으며 지금도 계속 추가되고 있다.

이와 달리 개발된 본 프로그램에서는 원어를 음운론에 따라 외래어표기로 변환하는 방법이 사용되었다. 원어가 입력되면 각각의 글자를 미리 프로그램 된 알고리즘에 따라 하나씩 변환하게 된다. 이 방법의 장점은 방대한 DB를 구축할 필요가 없어 프로그램 개발 시간이 짧으며 어떤 원어가 입력되더라도 모두 변환할 수 있다는 것이다. 반면에 언어의 특성상 사전에 프로그램 된 알고리즘에서 벗어나는 예외의 경우 정확하게 변환할 수 없게 된다는 단점이 있다. 따라서 최대한 예외를 줄이는 것이 이 프로그램의 정확도를 높일 수 있는 방법이다.

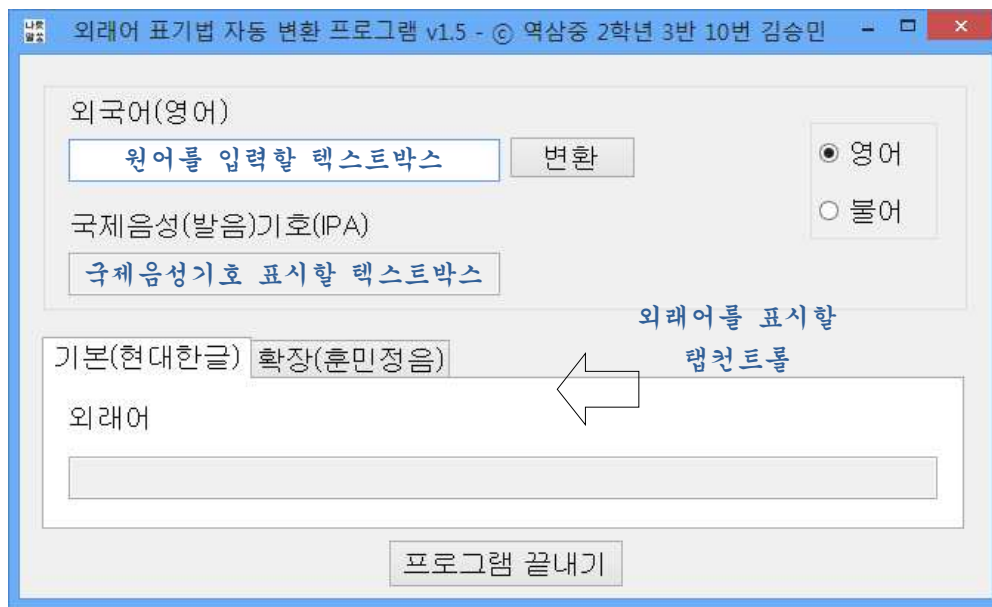
또한 이 프로그램에서는 원어표기를 외래어표기로 바로 변환하는 것이 아니라 국제음성기호(IPA)를 매개로 한 2단계 변환 방식을 적용하였다. 2단계 변환이란 먼저 원어를 국제음성기호(IPA)로 변환한 뒤 다시 국제음성기호를 한글 외래어맞춤법에 따라 변환하는 것이다. 2단계 변환 방식의 장점은 원어를 직접 외래어표기법으로 변환하는 방식에 비해 선행연구 자료가 많아 정확한 변환 알고리즘을 구현하기가 용이하다는 것이다.



[그림 1] 2단계 변환 개요도

또한 2단계 변환은 향후 영어 이외의 다른 외국어를 지원하기 위한 추가 개발시에도 장점이 될 수 있는데 이는 어떤 언어라도 국제음성기호(IPA)로만 변환하면 국제음성기호(IPA)에서 한글 외래어표기법으로의 변환은 기존에 개발된 사항을 그대로 적용하면 되기 때문이다.

3.2. 메인 폼(Form) 설계



[그림 5] 기본 폼 설계 화면

이 프로그램의 메인 폼은 외래어표기로 변환하고자 하는 원어를 입력할 텍스트박스(Text Box), 변환의 시작을 알리기 위한 버튼(Button), 국제음성기호(IPA)를 표시할 텍스트박스, 외래어를 표시하기 위한 탭컨트롤(Tab Control), 프로그램을 끝내기 위한 버튼으로 구성되어 있으며 메인 폼 화면은 [그림 2]와 같다.

외래어를 표기하기 위한 탭컨트롤은 2개의 탭으로 구성되어 있는데 현행 외래어표기법에 따른 외래어를 표시하기 위한 기본(현대한글) 탭(Tab)과 훈민정음 4글자를 복원한 외래어를 표시하기 위한 확장(훈민정음) 탭으로 구성되어 있다.

3.3. 원어(영어)를 국제음성기호(IPA)로 변환

원어(영어)를 국제음성기호로 변환하는 절차를 개략적으로 살펴보면, 먼저 원어가 입력된 텍스트 박스의 내용을 'enWord' 변수에 저장한다. 그리고 저장된 원어의 앞뒤에 붙은 공백문자를 제거하고 원어를 모두 소문자 변환하는데 이렇게 함으로써 프로그램 개발시 소문자만 고려하면 되므로 프로그램 알고리즘의 효율성을 높일 수 있다. 그 다음에는 'enWord' 변수로부터 한 글자씩 국제음성기호(IPA)로 변환하여 'IPAWord'라는 변수에 저장한다.

'enWord' 변수에 저장될 영어 글자는 아스키(ASCII)코드의 범위를 넘어서지 않으므로 'enWord' 변수는 1바이트 크기를 가지는 char 타입 배열로 선언하면 충분하다. 그러나 향후 영어 이외의 언어로의 확장을 고려하면 'enWord' 변수는 유니코드까지 저장할 수 있어야 한다. 유니코드에는 중국어를 포함하여 세계 대부분 국가의 문자와 기호, 구두점이 포함되어 있다. 따라서 'enWord' 변수는 유니코드 문자를 수용할 수 있는 wchar_t 타입의 배열로 선언되었다. 또한 국제음성기호를 저장할 'IPAWord' 변수에는 유니코드(Unicode)까지 저장되므로 wchar_t 타입 배열로 선언되어야 한다.

원어가 저장된 'enWord' 변수에서 한 개 또는 여러 개의 글자를 읽어 국제음성기호(IPA)로 변환하는 과정은 음운론에 따라 알고리즘으로 구현되었다. 우선 기본적으로 영어 1글자 또는 여러 글자를 음성(발음)기호 1개를 대응시켰다. 예를 들면 알파벳 p는 /p/ 발음에, ch는 /k/ 발음에 대응시키는 식이다. 여기서 2개 이상의 글자가 모여 하나의 발음을 구성하는 것을 이중철자(digraph)라 하는데, 예를 들면 단어의 첫 부분에 나오는 이중철자 kn은 항상 /n/으로 발음된다.⁴⁾ 즉 k글자가 묵음(silent letter)이 되는 것이다.

3.3.1. 이중철자(digraph)의 변환

이중철자는 단일철자에 비해 우선적으로 처리될 필요가 있다. 이는 원어를 발음기호로 변환하는데 사용된 알고리즘이 조건분기문(switch... case문, if... else문)을 기본으로 하고 있기 때문이다. 앞에서 예로 든 kn의 경우, 이중철자인 kn을 먼저 처리하지 않고 낱글자인 k를 먼저 처리하게 되면 k와 n이 각각 발음기호로 변환되어 know가 /knou/로 잘못 변환되게 된다.

4) 예를 들면, know : /nou/, knife : /naif/, knock : /nak/ 등이다.

동일한 원리로 3개의 글자로 이루어진 이중철자는 2개의 글자로 이루어진 이중철자보다 먼저 처리되어야 한다. 예를 들면 sc는 /s/로 변환되는데 반해 sch는 /sk/로 변환된다. 만약 sch보다 sc가 먼저 처리된다면 school의 경우 /shul/로 잘못 변환되게 된다.

이중철자와 그 발음기호에 대한 예시는 [표3] 및 [표4]와 같다.

[표 3] 자음 이중철자 및 발음기호

철자	발음	예시	철자	발음	예시
ck	/k/	pick	mb	/m/	dumb
kn	/n/	know	ng	/ŋ/	song
dge	/dʒ/	bridge	ch	/tʃ/	chick
tch	/tʃ/	itch	sc	/s/	scene
th	/θ/	thin	th	/ð/	the
ph	/f/	phone	wh	/h/	whose
rh	/r/	rhythm	wr	/r/	wreck
sch	/sk/	school	gue	/g/	rogue

- p361-362 [도표 13-3] 일부 발췌, 영어음성학, 구산희, 한국문화사(1998)

[표 4] 모음 이중철자 및 발음기호

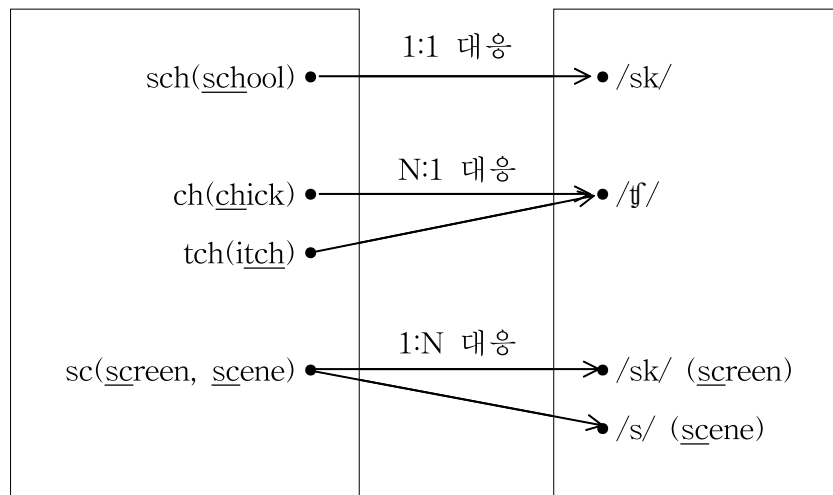
철자	발음	예시	철자	발음	예시
ea	/i/	beat	ay	/ei/	way
ie	/i/	field	ie	/ɛ/	friend
au	/æ/	laugh	ou	/u/	could
ow	/ou/	know	ow	/u/	tow
oo	/u/	too	ou	/ɔ:/	fought
igh	/ai/	high	ough	/au/	bough

- p363 [도표 13-4] 일부 발췌, 영어음성학, 구산희, 한국문화사(1998)

3.3.2. 발음기호 사용자 선택

영어는 한국어와 달리 어떤 영어 글자는 하나의 발음기호로 대응되지 않는다. 즉 영어

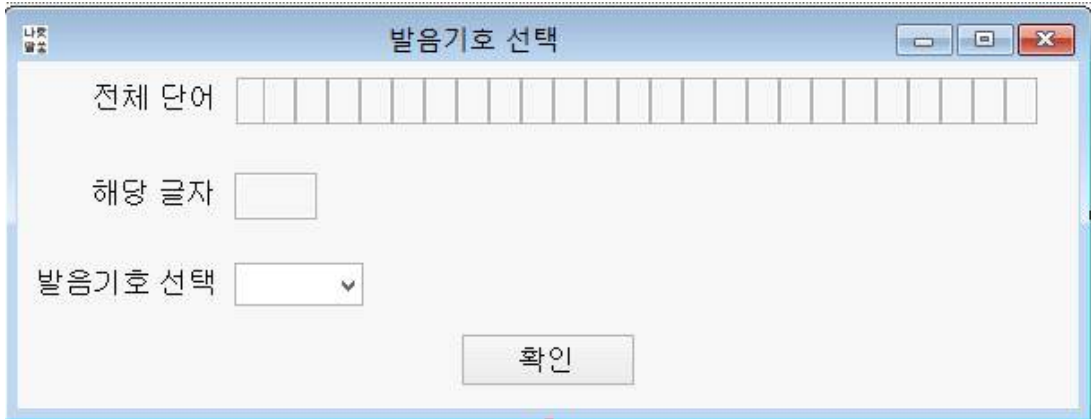
의 철자(이중철자 포함)를 정의역으로 하고 발음기호를 공역으로 하는 함수가 되지 못한다. 이런 현상은 자음에서도 발견되지만 대부분의 경우 모음에서 발견된다. 그런데 어떤 대응관계가 함수가 되지 못할 경우 자동화된 프로그램을 구현하는데 상당한 장애가 된다. 아래 그림은 그 사례를 보여준다.



[그림 6] 영어글자와 발음기호의 대응관계 (예시)

[그림 6]에서 1:1 대응의 경우 영어글자(좌측)와 발음기호(우측)를 대응시키는데 아무런 문제가 없다. 즉 sch란 글자가 나오면 무조건 /sk/로 변환하면 된다. N:1 대응의 경우에도 함수가 되므로 영어글자와 발음기호를 대응시키는데 문제가 없다. ch나 tch란 글자는 모두 /tʃ/로 변환하면 된다. 자동변환에 문제가 되는 것은 1:N(N:N도 포함) 대응의 경우이다. 1:N 대응은 함수가 되지 않기 때문에 정의역(좌측) 원소를 공역(오른쪽)의 어떤 원소에 대응해야 하는지를 정할 수 없다. 예를 들면 sc 글자의 경우 screen이란 단어의 경우 /sk/로 발음되고, scene이란 단어의 경우 /s/로 발음되기 때문에 자동변환이 불가능할 수밖에 없다.

프로그램 상에서 이 문제를 해결하기 위해서는 사용자가 개입하여 발음기호를 선택해야 한다. 이 프로그램에서도 동일한 영어철자에 2개 이상의 발음이 대응되는 경우 발음기호 선택 화면을 띄워 사용자로 하여금 맞는 발음을 선택하도록 한다. 사용자가 발음을 선택하면 선택된 발음을 'IPAWord' 변수에 추가하고 다음 절차로 넘어간다. 발음기호 선택 폼 설계는 [그림 7]과 같으며 사용자는 콤보박스(Combo Box)를 통해 복수의 발음기호로부터 맞는 발음을 선택할 수 있다.



[그림 7] 발음기호 선택 폼 설계 화면

3.3.3. 발음기호 표기법

영어 글자는 알파벳 26자 이내에서 표기되므로 표기법에 있어서 아무런 문제가 없지만 발음기호의 경우 학자나 단체에 따라 그 표기법이 상이하여 어느 것을 기준으로 할 것인가에 따라 프로그램의 내용이 달라질 수밖에 없다. 발음기호는 대부분 국제음성기호(IPA)를 참조하여 만들어져 있으나 국제음성기호(IPA)의 기본문자만도 107개(2005년 기준)에 달하는 등 지나치게 많은 측면이 있어 국제음성기호(IPA) 모두를 이용하는 것은 효율성이 많이 떨어질 수밖에 없다. 따라서 대부분의 발음기호 표기법은 국제음성기호(IPA)의 일부를 선택하여 구성되다보니 각각의 표기법마다 선택한 기호가 약간씩 다른 것이 현실이다. [표5]에서는 대표적인 영어 발음표기법인 Jones, Kenyon & Knott, Trager & Smith 표기법을 비교하였다.

[표 5] 영어 발음표기법 비교

표기법	Jones	Kenyon & Knott	Trager & Smith
shy	ʃ/ʃai/	ʃ/ʃa/	š /šay/
yes	j /jes/	j /jes/	y /yes/
sit	i / sit/	I /sIt/	i / sit/
boy	oi /boi/	oi /boI/	oy /boy/

이 프로그램에서는 우리나라에서 가장 많이 사용되어 우리나라 사람에게 가장 익숙한 Jones 표기법에 따라 발음기호를 표기하였다. 참고로 Jones 표기법은 다음, 네이버 영어 사전에서도 사용하는 표기법이다. 프로그램에서 국제음성기호(IPA)를 사용하기 위해서는

각 기호에 해당되는 유니코드(Unicode)를 알아야 한다. 국제음성기호(IPA)에서 사용하는 기호는 라틴어에서 차용한 것을 제외하면 대부분 유니코드 차트 0x0250-0x02AF (IPA Extensions)에 포함되어 있다. 이 프로그램에서 사용하는 발음기호 및 그에 해당되는 유니코드는 [표6]와 같다.

[표 6] 이 프로그램에서 사용된 국제음성기호(IPA) 표기의 유니코드 테이블(36자)

Index	0	1	2	3	4	5	6	7	8
IPA	a	e	ɛ	i	o	u	æ	ɑ	ɔ
유니코드	0061	0065	03B5	0069	006F	0075	00E6	0251	0254
Index	9	10	11	12	13	14	15	16	17
IPA	ə	ʌ	j	w	b	d	f	g	h
유니코드	0259	0245	006A	0077	0062	0064	0066	0067	0068
Index	18	19	20	21	22	23	24	25	26
IPA	k	l	m	n	p	r	s	t	v
유니코드	006B	006C	006D	006E	0070	0072	0073	0074	0076
Index	27	28	29	30	31	32	33	34	35
IPA	z	ð	ŋ	θ	ʃ	ʒ	dz	dʒ	tʃ
유니코드	007A	00F0	014B	019F	0283	0292	02A3	02A4	02A7

3.4. 국제음성기호(IPA)를 외래어표기로 변환

이 프로그램에서 국제음성기호(IPA)를 외래어표기로 변환하기 위한 기본적인 알고리즘은 ‘외래어표기법 제 2장 국제음성기호’ 및 ‘제 3장 표기세칙의 제1절 영어’를 따랐다. 영어 원어에서 국제음성기호(IPA)로의 변환과는 달리 국제음성기호(IPA)에서 한글 외래어표기로의 변환은 함수관계가 성립한다. 즉 어떤 국제음성기호(IPA)에 대응되는 한글 외래어표기는 하나만 존재한다. 따라서 사용자의 개입 없이도 국제음성기호(IPA)를 외래어표기로 자동 변환할 수 있다. 국제음성기호(IPA)와 한글 글자와의 대응관계는 [표7] 및 [표8]와 같다.

[표 7] 국제음성기호(IPA)와 한글 글자 대응표 (자음)

IPA	한글		IPA	한글	
	모음 앞	자음 앞 또는 어말		모음 앞	자음 앞 또는 어말
b	ㅃ	ㅍ	s	ㅅ	ㅆ
d	ㄷ	ㅌ	t	ㅌ	ㄷ, ㅌ
f	ㅍ (퐁) ^{<주1>}	ㅍ	v	ㅃ (빙) ^{<주1>}	ㅍ
g	ㄱ	ㄲ	z	ㅈ (△) ^{<주1>}	ㅉ
h	ㅎ	ㅎ	θ	ㄷ	ㅌ
k	ㅋ	ㄱ, ㅋ	ŋ	ㅇ	ㅇ
l	ㄴ	ㄴ	θ	ㅅ	ㅆ
m	ㅁ	ㅂ	ʃ	ㅅ	ㅆ, ㅅ
n	ㄴ	ㄴ	ʒ	ㅈ	ㅉ
p	ㅍ	ㅃ, ㅍ	ʒ	ㅈ	ㅉ
r	ㄹ	ㄹ	ʧ	ㅈ	ㅉ

<주1> ()안은 훈민정음 글자

[표 8] 국제음성기호(IPA)와 한글 글자 대응표 (모음 및 반모음)

IPA	한글	IPA	한글	IPA	한글
a	아	u	우	ə	어
e, ε	에	æ	애	ʌ	어
i	이	ɑ	아	j	이
o	오	ɔ	오 (·) ^{<주2>}	w	오, 우

<주2> ()안은 훈민정음 글자

3.4.1. 한글 초성, 중성, 종성의 분리 변환

한글은 영어와 달리 초성, 중성, 종성으로 이루어져 있어 같은 발음기호라 할지라도 초성에 오는 경우와 종성에 오는 경우가 그 쓰임이 다르다. 따라서 이 프로그램에서는 발음기호를 한글로 변환할 때 우선 초성, 중성, 종성을 분리하여 변환한 뒤 나중에 이를 합하여 하나의 글자를 만드는 방법을 택하였다. 초성, 중성, 종성의 인덱스를 저장하기 위해 int 타입 배열 변수 3개를 각각 'iniLetter', 'medLetter', 'finLetter'로 선언하였다.

발음기호가 모음인 경우 한글의 중성으로 처리하면 되나 자음인 경우 앞글자의 종성으로 들어가야 할지 뒷글자의 초성으로 들어가야 할지에 대한 알고리즘 처리가 쉽지 않다. 이에 대한 알고리즘은 ‘외래어표기법 제 3장 표기세칙’을 참조하면 구현할 수 있다. 예를 들면 ‘표기세칙 제 1항 무성파열음(/p/, /t/, /k/)’에 따르면 짧은 모음 다음의 어말 또는 짧은 모음과 유음·비음(/l/, /r/, /m/, /n/) 이외의 자음 사이에 오는 무성파열음은 받침으로 적고 이외의 경우에는 ‘으’를 붙여 적도록 규정되어 있다.

- book/buk/의 /k/는 짧은 모음 /u/ 다음의 어말에 왔으므로 받침으로 적으므로 ‘북’
- apt/æpt/의 /p/는 짧은 모음 /æ/와 유음·비음 이외의 자음 /t/ 사이에 왔으므로 받침으로 적으므로 ‘앳트’
- stamp/stæmp/의 /p/는 유음·비음 /m/ 다음에 왔으므로 ‘으’를 붙여 적어 ‘스탬프’
- make/meik/의 /k/는 짧은 모음이 아닌 /ei/ 다음에 왔으므로 ‘으’를 붙여 적어 ‘메이크’

현대 한글은 유니코드 0xAC00-0xD7AF (Hangul Syllables)에 완성형 형태로 모두 수록되어 있다. 그러나 발음기호로부터 한글로 변환할 때 초성, 중성, 종성을 분리하여 그 인덱스만 저장하였기 때문에 한글을 표기하기 위해서는 초성, 중성, 종성 인덱스로부터 그에 해당되는 한글의 유니코드를 생성해야 한다. 따라서 애초에 초성, 중성, 종성 인덱스를 정할 때 한글 유니코드로의 변환을 고려하여야 한다. 이 프로그램에서 정한 초성, 중성, 종성 인덱스는 각각 [표9], [표10], [표11] 같다.

[표 9] 현대 한글 초성 인덱스 테이블 (19자)

Index	0	1	2	3	4	5	6	7	8	9
초성	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅁ	ㄴ	ㅂ	ㅅ
Index	10	11	12	13	14	15	16	17	18	
초성	ㅆ	ㅇ	ㅈ	ㅊ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	

[표 10] 현대 한글 중성 인덱스 테이블 (21자)

Index	0	1	2	3	4	5	6	7	8	9	10
중성	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅚ	ㅜ
Index	11	12	13	14	15	16	17	18	19	20	
중성	ㅝ	ㅟ	ㅛ	ㅜ	ㅠ	ㅡ	ㅞ	ㅟ	ㅛ	ㅜ	

[표 11] 현대 한글 중성 인덱스 테이블 (28자)

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
중성		ㄱ	ㄲ	ㄳ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ
Index	14	15	16	17	18	19	20	21	22	23	24	25	26	27
중성	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ

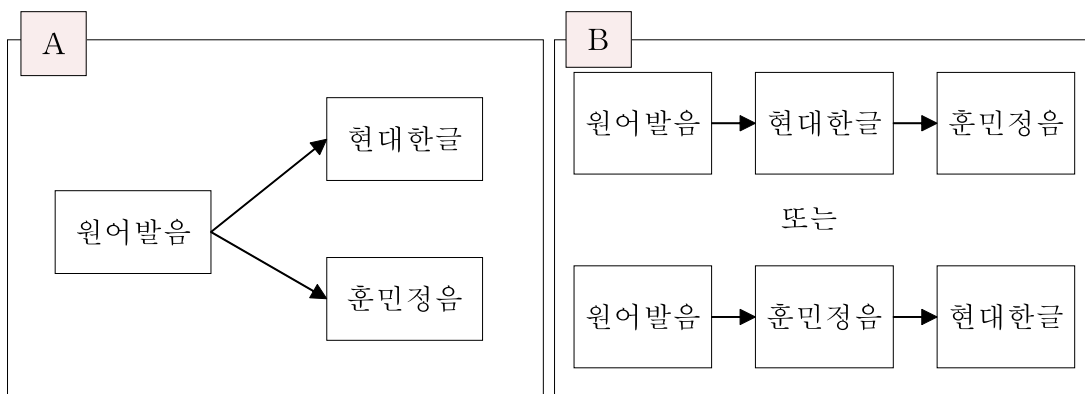
한글 유니코드의 첫 번째 글자인 ‘가’의 유니코드는 0xAC00이다. 따라서 초성, 중성, 중성 인덱스에 해당하는 한글 유니코드는 다음과 같이 구할 수 있다.

$$\text{한글 유니코드} = \text{초성_인덱스} * (21*28) + \text{중성_인덱스} * 28 + \text{중성_인덱스} + 0xAC00$$

이렇게 변환된 한글 유니코드는 최종적으로 wchar_t* 타입의 배열변수 ‘koWord’에 저장되며 기본(현대한글) 탭의 외래어를 표시하기 위한 텍스트박스에 표시된다.

3.4.2. 훈민정음 고어 구현

이 프로그램은 원어발음을 현행 외래어표기법과 훈민정음의 사라진 4글자를 복원한 표기법 2가지로 변환되어야 한다. 변환 방법으로는 일단 크게 2가지를 생각할 수 있는데 첫 번째가 원어발음을 현행 외래어표기법과 훈민정음 표기법으로 각각 변환하는 방법이며 두 번째는 원어발음을 현대한글 또는 훈민정음 표기법으로 먼저 변환한 뒤 변환된 현대한글 또는 훈민정음 표기법을 나머지 표기법으로 변환하는 방법이다.

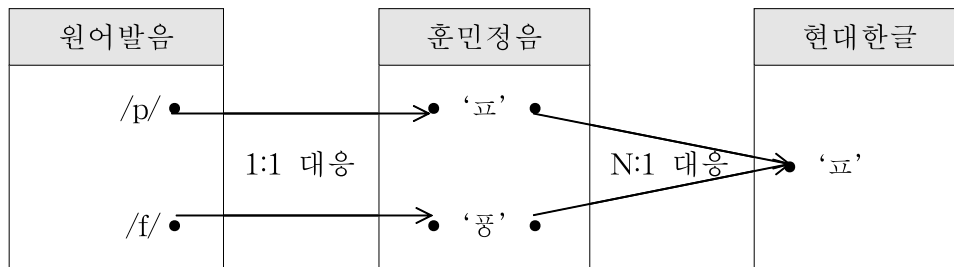


[그림 8] 원어발음의 한글 변환 방법론

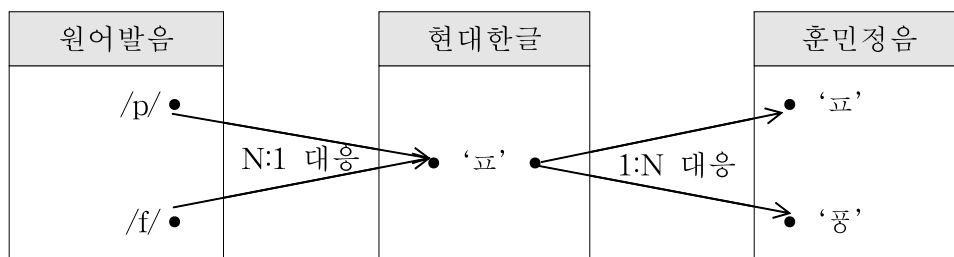
현대한글과 훈민정음 간에 차이가 많이 나서 현대한글↔훈민정음 간 변환이 복잡하다면 A 방법이 유리할 수 있으나 이 프로그램에서는 현대한글을 기반으로 훈민정음의 사라진 4글자를 추가하는 등 비교적 간단한 작업으로 현대한글↔훈민정음 간 변환이 가능하므로 굳이 원어발음을 현대한글과 훈민정음으로 각각 변환하는 것은 비효율적이다. 따라서 이 프로그램에서도 B 방법을 선택하였다.

다음으로 원어발음을 현대한글로 먼저 변환하여야 하는지 훈민정음으로 먼저 변환하여야 하는지를 결정하여야 한다. 여기서 가장 중요한 판단기준이 되는 것은 두 번의 변환에 따른 대응관계가 모두 함수가 되도록 결정하는 것이다. 이렇게 할 경우 사용자 개입이 없이 자동으로 변환되도록 알고리즘을 구현할 수 있다.

[표 7]에서 알 수 있듯이 현대 한글의 경우 /f/ 발음과 /p/ 발음이 모두 ‘ㅍ’으로 변환된다. 반면에 훈민정음의 경우 /f/ 발음은 ‘푹’으로 /p/ 발음은 ‘ㅍ’으로 변환된다. 이 상황을 도식화해서 나타내면 [그림 9] 및 [그림 10]과 같다.



[그림 9] 원어발음→훈민정음→현대한글 변환 방법론



[그림 10] 원어발음→현대한글→훈민정음 변환 방법론

원어발음→훈민정음→현대한글 변환([그림 9])의 경우 1:1 대응과 N:1 대응으로 모두 함수가 된다. 반면에 원어발음→현대한글→훈민정음 변환([그림 10])의 경우 N:1 대응은 함수가 되나 1:N 대응은 함수가 되지 않음을 알 수 있다. 따라서 이 프로그램에서는 원어 발음→훈민정음→현대한글 변환([그림 9]) 방법을 적용하였다.

유니코드에는 훈민정음 고어가 포함된 글자는 지원하지 않기 때문에 현대한글과 같은 방법으로 훈민정음을 구현할 수 없다. 다만 훈민정음을 구성하는 자모(자음과 모음)는 유니코드 0x3130-0x318F (Hangul Compatibility Jamo)에 포함되어 있다. 따라서 이 프로그램에서는 훈민정음을 자모단위로 표시하였다. 이 프로그램에서 사용되는 훈민정음 자모 및 유니코드는 [표12], [표13] 및 [표14]와 같다.

[표 12] 훈민정음 한글 초성 및 유니코드

Index	0	1	2	3	4	5	6	7	8	9	10
초성	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ	ㅆ
유니코드	3131	3132	3134	3137	3138	3139	3141	3142	3143	3145	3146
Index	11	12	13	14	15	16	17	18	19	20	21
초성	ㅇ	ㅈ	ㅊ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	ㅍ	ㅍ	ㅍ
유니코드	3147	3148	3149	314A	314B	314C	314D	314E	3178	317F	3184

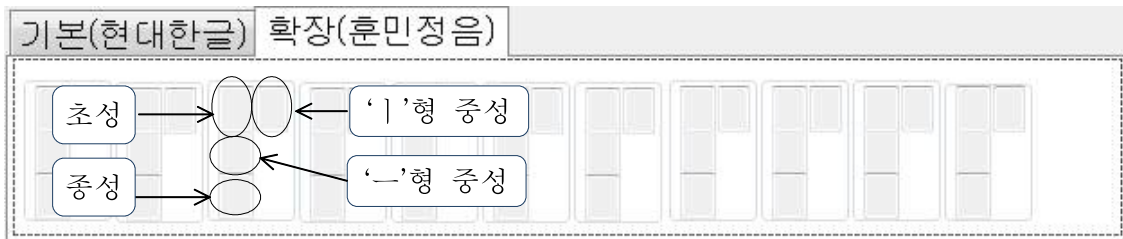
[표 13] 훈민정음 한글 중성 및 유니코드

Index	0	1	2	3	4	5	6	7	8	9	10
중성	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅓ	ㅕ
유니코드	314F	3150	3151	3152	3153	3154	3155	3156	3157	3158	3159
Index	11	12	13	14	15	16	17	18	19	20	21
중성	ㅖ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ
유니코드	315A	315B	315C	315D	315E	315F	3160	3161	3162	3163	318D

[표 14] 훈민정음 한글 종성 및 유니코드

Index	0	1	2	3	4	5	6	7	8	9
종성		ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ
유니코드	3130	3131	3132	3133	3134	3135	3136	3137	3139	313A
Index	10	11	12	13	14	15	16	17	18	19
종성	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ	ㅆ	ㅈ
유니코드	313B	313C	313D	313E	313F	3140	3141	3142	3144	3145
Index	20	21	22	23	24	25	26	27		
종성	ㅊ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ		
유니코드	3146	3147	3148	314A	314B	314C	314D	314E		

훈민정음 고어를 글자 단위로 표현할 수 없는 한계로 인해 이 프로그램에서는 훈민정음 고어를 자모단위로 표시하였다. 먼저 초성을 표시할 텍스트박스를 좌측 상단에 배치하고 종성을 표시할 텍스트박스를 하단에 배치하였다. 중성의 경우 3가지 형태로 나타날 수 있는데 ‘ㅣ’ 형태와 ‘ㅡ’형태, 그리고 ‘ㄴ’형태가 그것이다. 따라서 중성을 표시할 텍스트박스는 2개로 좌측 가운데와 우측 상단에 각각 배치하였다. 훈민정음 고어를 표시할 텍스트박스 배치 화면은 [그림 11] 과 같다.



[그림 11] 훈민정음 고어 표시 텍스트박스 배치 화면

‘ㄴ’형 중성의 경우 ‘ㅣ’형 중성 자리와 ‘ㅡ’형 중성 자리에 모두 배치한다. 아래는 영어 fighting에 대한 현대한글 및 훈민정음의 외래어표기의 예이다.



[그림 12] 현대한글 및 훈민정음 외래어표기 예

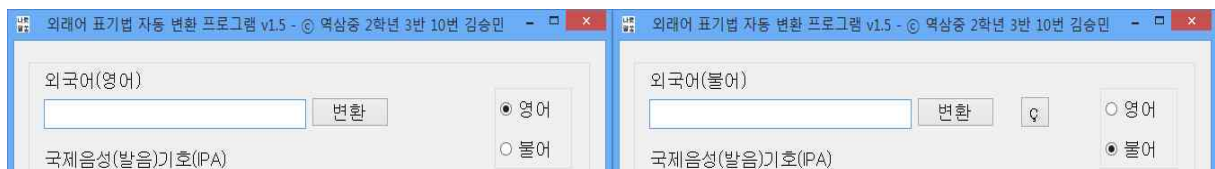
3.5. 프랑스어 추가사항

대중적인 영어 외래어 표기 프로그램에 우선 도전했고, 2007년부터 2011년까지 캐나다에 거주했던 경험(필수 제2외국어 프랑스어 수업 수강)을 토대로 프랑스어를 추가 개발했다. 프랑스어 외래어 표기 알고리즘은 영어와 크게 다르지 않다. 다만 프랑스어에는 영어와 다른 특징이 몇 가지 있어 프로그램 개발시 이를 고려하여야 한다.

먼저 영어에서는 반모음이 /j/와 /w/만 있으나 프랑스어에서는 이에 더하여 반모음 /ɥ/도 존재한다. 두 번째는 영어에는 없는 비자음 /ɲ(뉴)/가 프랑스어에 사용된다. 세 번째는 영어의 /n/, /m/에 해당하는 비음이 음절의 마지막에 올 경우 프랑스어에서는 /ŋ(웅)/으로 발음된다. 이런 문제는 원어(프랑스어)를 음운론에 따라 국제음성기호(IPA)로 변환한 뒤 국립국어원의 외래어표기법 제 2장 국제음성기호, 제 3장 표기세칙 제 3절 프랑스어의 표기에 따라 외래어로 변환함으로써 해결할 수 있다. 특히 프랑스어의 경우 영어에 비해 예외가 적어 국제음성기호로 변환하는데 사용자의 개입이 적다는 장점이 있다.

프랑스어를 외래어표기로 변환하는데 있어 또 하나의 문제점은 영어와 달리 프랑스어 알파벳(철자)은 우리나라 키보드에 없는 것이 있다는 것이다. 특히 Cédille(세디유, ç, 유니코드 0xE7)의 경우 유사한 철자인 ‘c’와는 발음 규칙이 달라 다른 철자로 처리해야 한다. 이 문제를 해결하기 위해 이 프로그램에서는 버튼 컨트롤을 이용하여 가상의 키보드를 만들어 필요할 때마다 해당되는 버튼을 누르면 외국어 입력 텍스트 박스에 입력되도록 구현하였다.

그리고 가상 키보드는 해당되는 언어를 선택할 경우에만 표시되도록 하여 사용자가 혼돈을 겪지 않도록 배려하였다. 예를 들면 ‘ç’ 글자의 경우 프랑스어를 선택하면 가상 키보드가 표시되지만 영어를 선택할 경우 가상 키보드가 사라진다.



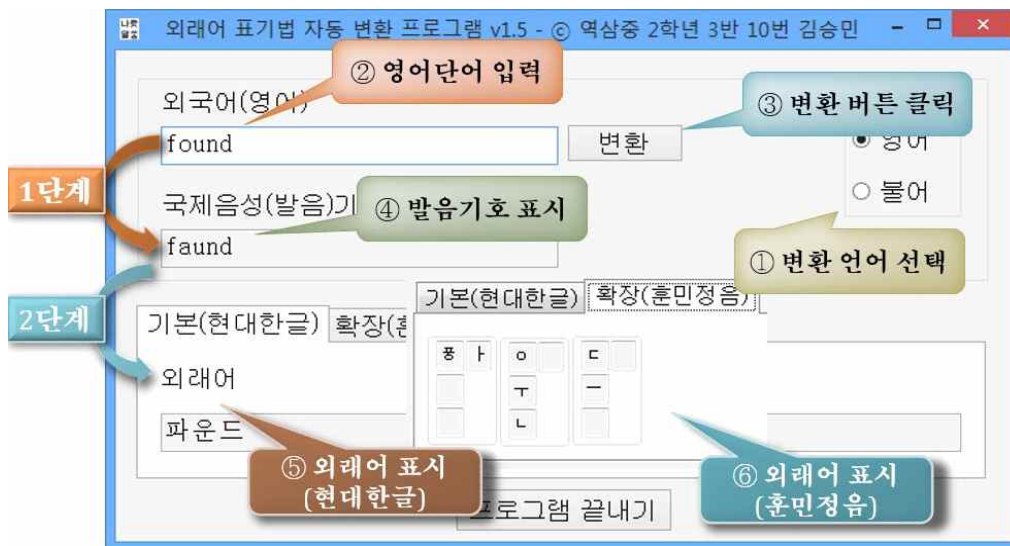
[그림 13] 가상 키보드 표시 화면

[그림 10]에서 외국어를 불어로 선택하면 ‘ç’ 글자 버튼이 표시되며(좌측), 영어 선택시에는 표시되지 않는다.

4. 프로그램 기능

4.1. 기본 화면

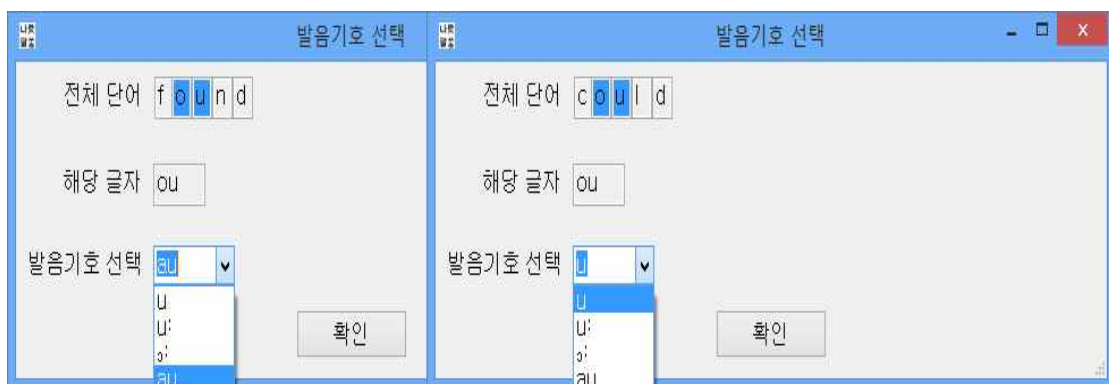
프로그램을 실행하면 [그림 14]와 같은 기본화면이 나타난다. 프로그램 사용법은 먼저 ① 원하는 영어단어를 입력하고, ② 변환 버튼을 클릭하면, ③ 국제발음기호(IPA)가 표시되며 동시에 ④ 현대한글 외래어와 ⑤ 훈민정음 외래어가 같이 표시된다.



[그림 14] 프로그램 기본 화면

4.2. 발음기호 선택 화면

한글과 달리 영어는 하나의 글자가 여러 가지 소리로 날 수 있는데 이럴 경우 발음기호 선택 화면이 나타나며 사용자는 맞는 발음기호를 선택할 수 있다.

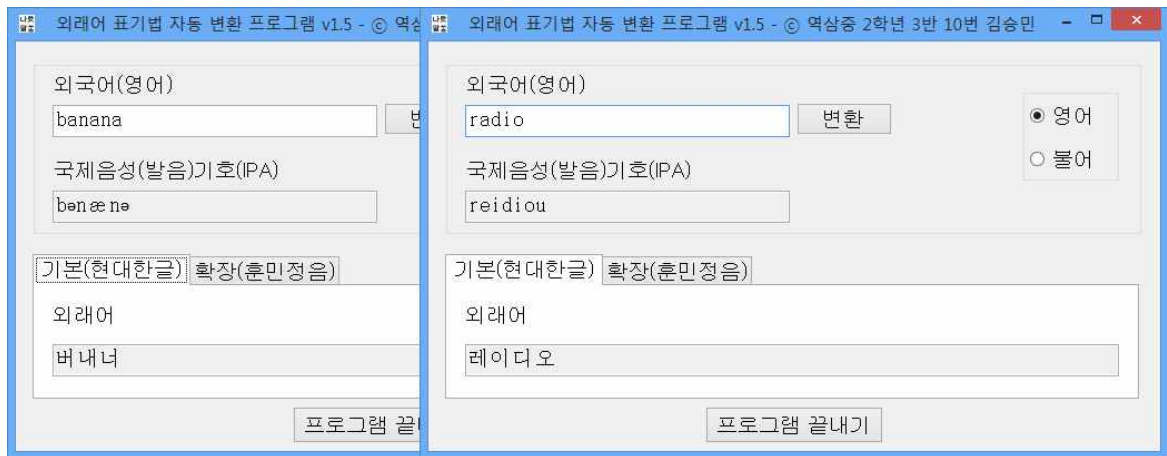


[그림 15] 발음기호 선택 화면

[그림 15]에서와 같이 동일한 ‘ou’ 글자라도 found의 경우 /au/로 could의 경우 /u/로 발음된다.

4.3. 원어에 가까운 외래어 표기

관행으로 굳어진 외래어(일본식 외래어 표기)라도 가급적 원어에 가까운 발음으로 표시된다. 예를 들면 ‘banana’의 경우 외래어 관행은 ‘바나나’이나 원어에 가까운 발음인 ‘버내너’로 표시한다. 또한 ‘radio’의 경우 외래어 관행은 ‘라디오’이나 원어에 가까운 발음인 ‘레이디오’로 표시한다.



[그림 13] 원어에 가까운 외래어 표기 예

5. 결론 및 향후 계획

5.1. 탐구 결론

- ◆ 영어, 프랑스어 원어를 외래어 표기법으로 자동으로 변환할 수 있는 컴퓨터 프로그램의 가능성 확인
- ◆ 외래어 표기법에 현재 사용되는 40개의 자모 외에도 훈민정음의 사라진 글자를 활용할 경우 더 정확한 외래어 표기가 가능하다는 점 확인
- ◆ 이 외래어 자동 변환 프로그램이 어린이를 위한 영어, 프랑스어 발음 교육 및 외국인을 위한 한글 교육 프로그램에 활용 가능성이 있음을 확인

5.2. 향후 계획

- ◆ 영어, 프랑스어 뿐만 아니라 다른 외국어의 외래어 변환 기능 추가
- ◆ 훈민정음의 사라진 글자 추가 복원
- ◆ 원어 발음에 대한 음성지원(Text to Speech) 기능 추가

6. 참고문헌

- ◆ 영어음성학(1998), 구희산, 한국문화사
- ◆ 영어음성학개론(2005), 전상범, 을유문화사
- ◆ 28자로 이룬 문자혁명 - 훈민정음(2007), 김슬옹, 아이세움
- ◆ An outline of English phonetics(1960), Jones, Cambridge: W. Heffer & Sons Ltd.
- ◆ La Phonetique, Presses Universitaires de France(1994), Bertil Malmberg
- ◆ 국립국어원-외래어표기법(<http://www.korean.go.kr>)
- ◆ 유니코드 차트 (<http://www.unicode.org/charts/>)