

미디 메시지를 이용한 게임이벤트처리

이우석*, 김경식**

호서대학교 컴퓨터공학과 게임공학전공*, 호서대학교 컴퓨터공학과 게임공학전공**
polymode@hanafos.com*, kskim@office.hoseo.ac.kr**

Processing of Game Events Using MIDI Messages

Woo-Seock Lee*, Kyung-Sik Kim**

HO-SEO Univ Computer Engineering*, HO-SEO Univ Computer Engineering**

요약

미디(Musical Instrument Digital Interface)란 마이크로 프로세서를 쓰는 전자적인 음악 기구간의 인터페이스로 연주자의 연주에 대한 정보를 전송하게 되는데 이러한 미디 컨트롤러에서 발생된 미디신호를 이용하여 게임플레이에서 사용되어지는 기존의 입력장치(PC 키보드, 마우스, 조이스틱)와는 다른 사용자 인터페이스로의 확장을 연구한다.

Abstract

MIDI use interface to transmit among the Electronic Musical Instrument. We use signal to happen in MIDI Controller by player and study the expansion user interface different from existing input device(Keyboard, Mouse, Joy stick etc..) to game about the usage the MIDI data.

Key Words: Game, Sound, MIDI, Computer

1. 서론

기존의 마우스와 PC 키보드를 이용한 게임플레이는 입력 장치는 인간의 감각적인 행동을 실시간으로 처리하는데 한계가 있어왔다. 골프게임을 예로 들자면 공의 타점과 힘의 밸런스를 수치화 하여 화면 인터페이스로 나타낸다. 이것은 실제 골프의 느낌을 전혀 주지 못한다.

미디 메시지로 이러한 입력 인터페이스를 교체한다면, 타점은 미디키보드의 Wheel Data로 타력은 건반의 Velocity Data로, 기존의 골프게임에선 채용되진 않았지만 실전 골프에서 중요한 요소인 팔로 스윙은 After touch Data로 처리가 가능하다.

또 탁구게임이나 테니스 게임을 예로 들어 Wheel 과 건반의 타격을 동시에 작동하여 공의 회전력을 조절함으로써 좀더 실전에 가까운 게임 이벤트의 처리를 연구하였다.

미디메시지는 Status Byte (1개)와 Data Byte (0~2개)로 구성되어지므로 MIDI Master Keyboard에서 한 번의 입력으로 복잡한이면서도 정밀한 값을 게임플레이에 적용할 수

있고 MIDI Master Keyboard에 장착되어 있는 Pitch Bender 와 Modulation Wheel을 사용하여 감각을 이용한 게임의 플레이가 가능하다.

MIDI Master Keyboard에서 발생된 미디 메시지를 이용해 기존 입력장치인 마우스와 PC 키보드 한계를 보완하여 새로운 게임 인터페이스의 확장을 시도한다.



[그림 1] 미디 키보드 컨트롤러

2. 미디데이터 구조

미디 데이터의 개관은 다음과 같다.

- ① 미디는 31,25 KBoud 의 전송속도를 가지는 직렬 인터페이스이다.
- ② 이벤트 메시지와 타이밍 참조가 둘 다 전송되며 이진

코드의 디지털 워드로 구성된다.

- ③ 이들은 각 악기 간에 연결된 하나의 선을통해서 전송 된다
- ④ 미디 메시지는 악기에 의해서 사용되는 커맨드와 데이터로 구성되며 이들은 다양한 퍼포먼스와 오퍼레이션 이벤트들을 표시 한다.

미디를 이용하여 악기사이에서나 악기 내에서의 인터페이스에 의해서 전달되어지는 정보는 '이벤트' 라는 것에 의해서표시된다.

이벤트란 '연주자가 악기를 연주하거나 동작시키는 것에 관련된 모든 행위' 를 의미하는데, 연주에 대한 이벤트 (Performance Event)는 음을 연주한다거나 휠이나 페달을 움직이는 등의 행위를 말하는 것이고, 동작에 대한 이벤트 (Operation Event)는 음색 프로그램을 바꾸는 것이나프로그램의 파라미터를 바꾸는 것 같은 작업을 말한다.

이런 각기 다른 이벤트는 '메시지(Message)' 라는 것에 의해서 표현되는데 이것은 한 이벤트가 수행되면 악기에 의해서 생성되어져서 인터페이스에 의해전달된다. 즉 어떤 한 이벤트는 하나의 메시지로 표현 가능하다.

그리고 메시지는 하나의 커맨드와 그에 따르는 데이터로 구성되어진다.

미디에서의 바이트 단위의 정보, 즉 미디코드는 상태 바이트(Status Byte)와 데이터 바이트(Data Byte)로 나누어진다. 한 바이트중 최상위 비트(MSB = Most Significant Bit)가 '1' 이면 상태 바이트이며, 최상위 비트가 '0' 이면 데이터 바이트이다. 즉 데이터가 127보다 크면 상태 바이트, 작으면 데이터 바이트로 처리된다. 미디에서는 하나의 상태 바이트에 이어서 0 - 2개의 데이터 바이트가 전송되는데, 이것을 묶어서 미디 메세지 라고 하고 이 메세지 중에서 커맨드 부분이 상태바이트이고 데이터 부분은 데이터 바이트가 된다.

$$\text{MIDI Messege} = (\text{Status Byte}) * 1\text{개} + (\text{Data Byte}) * (0 - 2)\text{개}.$$

미디 데이터를 처리 한다는 것은미디 신호를 메세지 단위로 처리함을 의미한다.

이러한 미디 데이터의 형태을 이용하여 게이머가 한번의

키입력을 통해 복합적인 게임플레이 데이터의 전송이 가능함이 본 발표 논문의 요지이다.

미디 메세지는 상태 바이트에 따라서 그 종류가 구분된다. 데이터 바이트는 말 그대로 데이터 이며 이것의 의미도 그 앞의 상태바이트에 따라서 달라진다.미디 메세지의 구분을 위한 상태 바이트도 다시 상위 4비트와 하위 4비트로 나누어서 표시하는데, 일반적으로 미디에서는 '1aaabbbb' 의 형태로 상태 바이트를 나타낸다.

'aaa' 가 '111' 이 아니면 그 미디 메세지를 " 채널 메세지 (Channel Message) " 라 하며 뒤의 'bbbb' 의 값은 0 - 15 까지의 16개 채널을 의미한다. 즉 채널 메세지를 나타내는 상태바이트의 값은 80h(=128d)에서 0EFh(=239d)까지의 값을 가진다.그리고 'aaa' 의 값이 '111' 이면 " 시스템 메세지 (System Message) " 라는 미디정보가 되며 'bbbb' 의 값은 채널값이 아닌 다른 특별한 의미를 지니게 된다.

채널 메세지는 다시 "채널 보이스 메세지" 와 "채널 모드 메세지" 로 나뉜다.

즉 'aaa' 값이 '011' 이고 첫 번째 데이터 바이트가 122 - 127 의 값이면 "채널 모드 메세지" 가 되며 나머지 채널 메세지들은 "보이스 메세지" 가 된다.

3. 채널 보이스 메세지(Channel Voice Message)와 게임플레이로의 적용

미디 인터페이스를 통해서 전송되는 미디 메세지의 거의 대부분은 채널 보이스 메세지 (Channel Voice Message)이다.

이것은 연주의 물리적인 작용으로 생성되는 모든 메세지를 포함한다. (게이머의 게임플레이 이벤트들을 채널 보이스메시지를 이용해 처리할 것이다).

어떤 건반이 눌렀는지, 어떤 스위치가 눌러졌는가 라든지 모듈레이션 휠(Modulation heel),볼륨페달 같은 아날로그적인 컨트롤러가 움직였는지에 대한 것이나 건반이 눌러지는 세기(Velocity)나 누른후에 흔드는것(Pressure)같은 것에 대한 신호가 보이스 메세지이다.

한 미디 시스템의 각각의 악기는 1-16까지의 채널중 하나가 지정되어 있고, 한번 채널이 악기에 지정되면 그 악기는 오직 할당된 채널에 대해서만 보내거나 받게되며 나머지

다른 채널값으로 들어오는 메시지는 무시한다. 이것이 하나의 케이블을 통해서 나가는 미디 데이터가 한 미디 시스템에서 각기 다른 악기에게 각기 다른 연주를 하게 하는 방법이다.

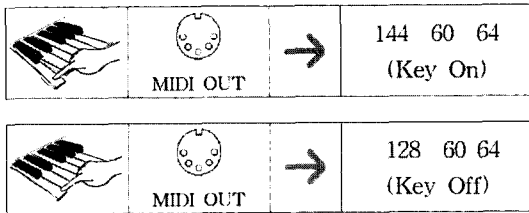
- o Note ON (9nH) / key Number / Attack Velocity
 건반눌림 / 음높이 / 음의 세기
- 타격 / 클립의 종류 / 타력

이 메시지는 한 건반이 눌러졌다는 것을 표시한다. 신디사이저에서는 각 건반에 따라 음 높이가 다르므로 음 높이는 각 건반의 번호를 나타내고 음의 세기는 건반을 누르는 속도로서 표시한다. 음 높이를 나타내는 건반 번호(Key Number)는 건반 128개 중의 하나를 나타내는 것으로 데이터 바이트의 최대 범위인 0 - 127의 값을 가진다.

즉 상태 바이트는 최상위 비트가 '1'이므로 128 - 255의 값을 가지고 데이터 바이트는 최상위 비트가 '0'이므로 0 - 127의 값을 가진다.

'가온 다' (Middle C = C3, 건반 중앙의 '도' 음)를 60으로 두고서 반음에 1씩 증감하여 10 옥타브 이상의 음역을 표시할 수 있다. 음의 세기를 나타내는 건반을 치는 속도(Attack Velocity)는 말 그대로 건반이 얼마나 세게(= 빠르게)눌러졌나를 나타내는 값으로 0 - 127의 값을 가지게 된다.

이것을 골프게임에 적용하게 된다면 골프 클럽의 종류와 타격명령과 타력정도를 하나의 건반입력으로 처리 할 수 있는 데이터를 전송할 수 있다.



[2] 건반의 눌림에 의한 미디데이터의 변화

- o Pitch Bender Change(EnH) / Bender Positon1 / Bender Positon12
 피치벤더 움직임 / 벤더위치1 / 벤더 위치2
- 타점선택 / 타점위치(X or Y) / 타점 위치2

이 메시지는 피치벤더(Pitch Bender)가 움직인 것을 표시하는데 즉 벤더의 위치가 바뀔 때마다 이 메시지가 생성되어 전송된다.

피치 벤더란 신디사이저에서 손잡이 모양으로나 바퀴모양으로 생겨서 건반 옆에 있는 것인데 건반을 누르고서 소리가 나는 상태에서 이것을 움직임으로써 미세한 단위로 음높이의 변화를 일으키는 장치이다.

이것이 움직이는 것은 아날로그의 연속된 값이므로볼륨을 조정하는 페달 등과 같이 '연속적인 조종기구(Continuous Controller)'에 속하는 장치이지만 미디 악기 안에서는 그 변하는 단위를 디지털적으로 정해줄 수 있다. 이는 게임에서 조이스틱의 움직임을 구현할 수 있는 기능이다.벤더 위치(Bender Position)값은 0 - 127의 값을 가지며 현재의 벤더 위치를 나타낸다.64 의 값은 벤더가 중앙으로 왔을 때 전송되는 값이다.

신디사이저를 만드는 회사측에서는 피치 벤더의 음정을 나누는 단위를 더욱 세밀하게 하기위해서 이 메시지와 함께 부가적으로 한 바이트를 더 전송할 것을 선택할 수 있는데, 2 바이트의 벤더 위치 데이터를 전송함으로써 좀 더 정밀한 위치 값을 얻을 수 있다.

- o 컨트롤체인지(BnH) /컨트롤러 번호 /컨트롤러 위치
 /모듈레이션 움직임 /타점위치(X or Y)

이 메시지는 옵션으로 확장될 수 있는 컨트롤러들의 움직임을 나타낸다.

즉 모듈레이션 휠(Modulation Wheel)이나 각종 레버나 스위치로써 음향효과를 내개위한 것에 관계된 메시지가 '컨트롤 체인지'이다.

이어지는 데이터 바이트중에서 첫번째가 컨트롤러의 종류를 나타내며 두번째 바이트가 그 데이터가 된다.

$$\text{컨트롤러 번호 바이트} = 0ccccccc = 0edcccc$$

앞의 0ccccccc는 뒤의 테이블에서 쓰는 기호로서 단지 컨트롤러 종류를 나타내는 바이트이다.

(게임플레이상에서 모듈레이션 휠을 이용하기 위해서는 그것을 의미하는 바이트값인 1이 전송되어야한다. 이것은 휠을 움직이는 순간 자동으로 전송되며, 움직인 정도가 그

의 데이터 값으로 처리된다.)

뒤의 0edcccc는 그 바이트를 더욱 세분한 것으로 'e'는 'Effector'를 나타내는 것으로 컨트롤러(= Effector)가 아날로그 값을 갖는 것이나 스위치의 ON, OFF로서 나타내어지는 것이나를 구분하는 비트로서 0 - 63까지의 값과 64 - 127의 범위로 나누어 준다.

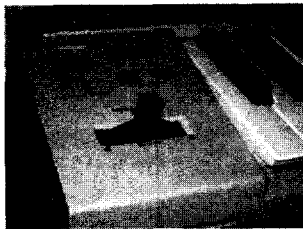
'cccc'는 컨트롤러 번호를 나타내는 것으로 32가지의 컨트롤러를 표시할 수 있는데 이것이 앞의 'e' 비트와 조합되어 0 - 31은 아날로그 컨트롤러를 나타내는데 쓰이며 64 - 95의 값은 스위치조작에 의한 컨트롤러를 표시하게 된다.

'd'는 'Detail bit'로서 'cccc' 비트에 종속된다.

0 - 31의 아날로그 컨트롤러의 경우 뒤따라오는 '컨트롤러 위치' 바이트의 0 - 127 값을 가지고서는 정확한 표현이 어렵기 때문에 32 - 63의 값은 0 - 31이 나타내는 컨트롤러의 데이터를 더 전송한다는 신호로 쓰인다. 이렇게 함으로써 좀 더 세밀한 위치 데이터의 전달이 가능하다.

또한 'e' 비트 값이 '1' 인 경우 'd' 비트 값도 '1' 이면 96 - 127의 값을 가지게 되는데 논리상 64 - 95의 스위치 컨트롤러의 확장 명령으로 써야 하지만 스위치는 ON, OFF의 값만 전달하면 되므로 다른 용도로 쓰이는데, 122 - 127 값을 제외한 나머지 값들은 지금 쓰이지 않고 나중의 확장을 위해 남겨둔 것이고 122 - 127의 값은 범위가 바뀌어 "채널 모드 메시지"를 나타내는데 쓰인다. 다시 말하면 '채널 모드 메시지'는 '채널 보이스 메시지' 안의 '컨트롤러 체인지' 메시지에 숨어 있는 것이다.

컨트롤 체인지의 둘째번 데이터 바이트는 컨트롤러의 위



[3] 미디 컨트롤러 휠의 여러 가지 형태

치를 나타내는데 아날로그 컨트롤러의 경우 0 - 127의 값으로 그 위치를 나타내며 스위치 컨트롤러의 경우 '0'은 'OFF'를 나타내고 '127'은 'ON'을 표시한다.

앞서 언급한 피치벤더와 모듈레이션 휠을 통해 감각적으로 입력되어진 데이터는 실시간으로 게임상의 좌표설정이나 위치선택에 이용 되어질 수 있다.

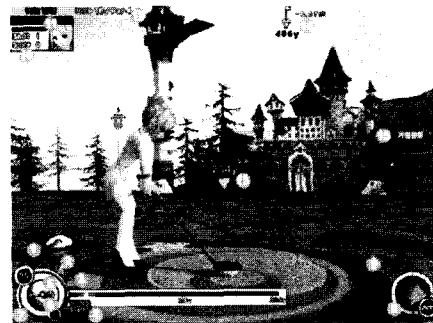
- 키 프레스어(Key Pressure) (AnH) / 건반 번호 / 애프터 터치(After touch)

키 프레스어는 말 그대로 악기의 건반이 눌러진 후의 압력에 얼마나 반응하느냐에 대한 메시지이다. 이런 효과를 '애프터 터치(After touch)'라고 하는데 비브라토(Vibrato)나 트레몰로(Tremolo) 등의 효과를 내는데 쓰인다.

이는 실제 골프스윙동작에서 중요한 요소인 임팩트후의 Fallow 스윙에 해당하는 데이터로 처리될 수 있다.

4. 결론

대부분의 골프게임의 경우 게이머의 플레이테이터를 화면 인터페이스를 통해 공의 타점을 지정하고, 표시된 게이머의 움직임을 멈추고 공을 임팩트하게 된다. 이는 실제 골프와는 많은 차이가 있다



[4] 온라인 골프게임 '팡야'의 인터페이스

앞서 설명한 미디 데이터로 게임플레이 이벤트를 처리할 수 있는 장점은 실시간으로 복합적인 데이터 값을 한번에 처리할 수 있고, 미디 컨트롤러 자체가 아날로그이기 때문에 섬세한 힘의 조절과 판단을 요구함으로써 실전골프에 가까운 긴장감을 줄 수 있다고 본다.

본 발표논문은 미디신호를 이용한 게임플레이 인터페이

스의 아이디어와 가능성을 서술하였고 차후 연구과제로는 본 논문의 프로그램 구현과채널 보이스 메시지 이외에 채널 모드 메시지(Channel Mode Message)와 시스템 메시지(System Message)를 이용한 게임시스템 제어연구와 그것의 프로그램 구현이다.

또한 MPU-401 기본 미디스펙에서 제공하는 16채널을 이용, 다수의 미디 컨트롤러를한 대의 PC 플랫폼에 연결하여 게임플레이를 진행하는 게임 인터페이스의 연구도 병행될 것이다.

참고문헌

- [1] 전형진미디핸드북교회사
- [2] 전형진MIDI 1.0 Spec 機電硏究社
- [3] www.borg.com/~jglatt/tech/midispec.htm
- [4] <http://krypt.dvndns.org:81/joy2midi/>
- [5] Popular음악이론 후반기 출판사
- [6]<http://www.pangya.com> 온라인골프게임광야



이우석

1995년 호서대학교 컴퓨터공학과 (학사)
 1999년 입투메이트-삼국통일
 2001년 디머스 소프트- Fidex 게임사운드 디렉터
 2003년 호서대학교 컴퓨터공학과 (석사)
 2003년~현재 호서대학교 게임공학과 박사과정 재학 중
 관심분야: 게임사운드, 미디어 프로그래밍



김경식

1982년 서울대학교 전산기공학과 (학사)
 1990년 서울대학교 컴퓨터공학과 (석,박사)
 1984년~1991년 한국전자통신연구원
 1991년~현재 호서대학교 게임공학전공
 관심분야: 게임프로그래밍, 게임제작전반