

NXC를 이용한 마인드스톰 프로그래밍

LEGO mindstorms
— Platform overview —



차례

- 마인드스톰 - NXT
- NXC
- 예제 소스코드

마인드스톡의 구성



NXT



Servo Motor



Sonic Sensor



Light Sensor



Sound Sensor



Many Bricks



Touch Sensor



마인드스톰의 주요 부품



NXT

마인드스톰의 메인이 되는 부품이다.
Brick들로 만든 로봇을 각 센서들에서 받아들인 정보를 토대로 NXT에서 컨트롤 한다.
NXT 상부의 A, B, C 포트에는 모터를, 하부의 번호 포트에는 센서를 연결한다.
오렌지색 버튼은 전원/확인, 화살표 버튼은 메뉴선택, 어두운 회색 버튼은 취소와 뒤로 가기 기능을 담당한다.



Sonic Sensor

물체와 센서 사이의 거리를 측정 할 수 있다.
255cm까지 가능하며 +/-3cm의 오차가 있을 수 있다. 주기적으로 신호를 받는다.

마인드스톰의 주요 부품



Light Sensor

물체의 색을 구별하는 센서이다.
흑백으로 색을 읽어 들이는 단점이 있지만 물체의 색이 비슷하지 않다면 어느 정도 구별 가능하다. 주기적으로 신호를 받는다.



Touch Sensor

물체의 접촉을 감지하는 센서이다.
Pressed, Released, Bumped(눌렀다가 땀을 반복)을 구별하여 입력 받을 수 있다. 인터럽트로 신호가 들어오며 누른 회수를 측정 할 수도 있다.



Sound Sensor

소리를 감지하는 센서다.
감지한 소리의 크기를 dB 단위로 표현한다. 주기적으로 신호를 받는다.

마인드스톰의 주요 부품



Servo Motor

정방향과 역방향 움직임이 가능하며 360 °회전이 가능한 모터이다. 방향과 회전 각도, 속도 조절(최대속도에 대한 퍼센트)이 가능하여 여러 응용이 가능하며 한번 동작명령을 받으면 다시 동작명령을 받을 때까지 이전의 상태를 유지한다.

마인드스톡의 특징

- NXT자체에서 간단한 프로그래밍이 가능하다.
- 각종 센서를 이용한 상황에 맞는 로봇의 행동설정이 가능하다.
- USB와 Bluetooth를 이용하여 컴퓨터의 프로그램을 다운받아 실행하거나 조종이 가능하다.

NXC

LEGO mindstorms
— Platform overview —



NXC란?

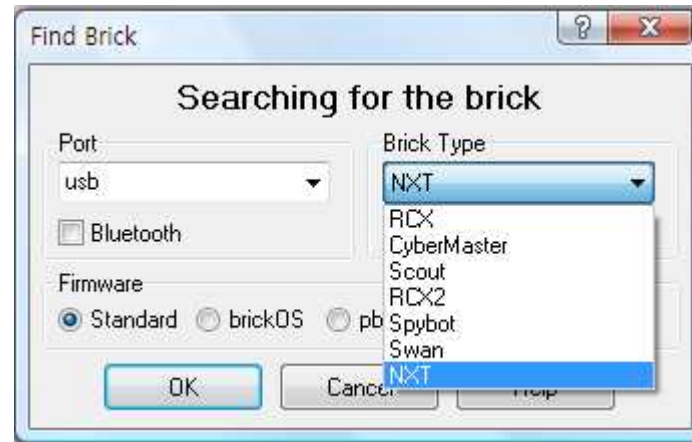
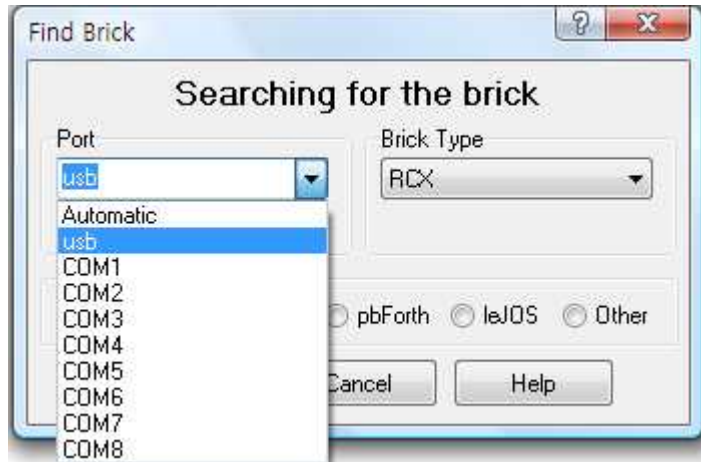
- C언어를 기반으로 NXT에 맞게 수정을 거친 언어
- 개발 툴로 Bricx CC를 사용하며 C언어와 몇몇 명령어만 제외하면 크게 다르지 않기 때문에 C언어를 다룰 줄 아는 사람이라면 쉽게 사용할 수 있다.
- NXC 홈페이지의 NXC Tutorial과 NXC Guide를 참조하기를 권한다.

<http://bricxcc.sourceforge.net/nbc/>

개발 툴 설치

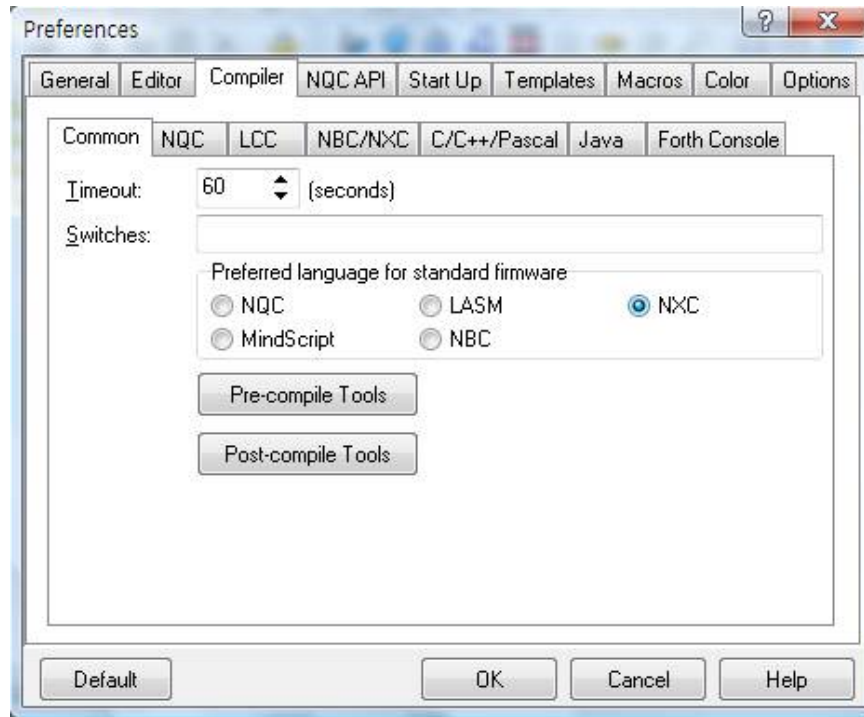
- NXC를 이용한 프로그래밍을 위해 개발 툴인 Bricx CC(Bricx Command Center)를 사용한다.
- <http://sourceforge.net/projects/bricxcc/files/bricxcc/>
- Bricx CC Test Release를 다운 받아 압축을 해제 한다.
- 마인드스톰에 동봉되어있는 CD로 LEGO MINDSTORMS NXT도 설치하여 컴퓨터가 NXT를 인식할 수 있도록 한다.

Bricx CC 설정



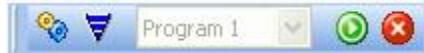
- Bricx CC 실행 시 처음으로 나타나는 화면이며 설정에 맞는 Brick을 찾아 연결하는 과정이다.
- NXT를 컴퓨터에 연결하고 해당 Port로 설정한다.
- Brick Type은 NXT, 그리고 Firmware는 Standard로 한다.
- Cancel을 눌러 취소하고 컴파일전에 메뉴의 [Tools]-[Find Brick]을 실행하여 설정할 수도 있다.

Bricx CC 설정



- 메뉴의 [Edit] – [Preference]를 선택하여 위의 창을 띄운다
- [Compiler] 탭의 [Common]에서 Preferred language for standard firmware항목을 NXC로 설정한다.

컴파일 및 실행



- 컴파일 전에 브릭이 연결되어 있지 않다면 반드시 [Tools]-[Find Brick]를 실행하여 브릭을 연결 후 컴파일을 한다.
- Bricx CC의 메뉴의 오른쪽에 있는 버튼들이다.
- 왼쪽부터 오른쪽으로, 첫 번째 버튼이 컴파일 버튼이다. 소스 코드를 브릭에 맞게 컴파일 한다.
- 두 번째 버튼은 다운로드 버튼이다. 컴파일 된 프로그램을 NXT에 다운로드 시킨다.
- 세 번째 버튼은 프로그램 실행, 네 번째는 실행 중인 프로그램을 중지 시키는 버튼이다.
- 실행과 중지는 NXT의 [My Files]-[Software files]-[소스코드명]을 선택하여 할 수도 있다.

예제 코드

LEGO MINDSTORMS
Platform overview



#1 Motors Test

```
// #1 motors test
task main() {
  int speed = 10;
  int time = 500; // 0.5sec

  while(true) {
    OnFwd(OUT_AC, speed);
    OnRev(OUT_B, speed);
    Wait(1000);

    if(speed < 100)
    { speed += 10; }
    else
    { break; }
  }

  Off(OUT_AC);
  Float(OUT_B); // Coast(OUT_B);
}
```

- 요구사항
 - 모터 A,C : 정방향으로 작동
 - 모터 B : 역방향으로 작동
 - 1초에 10%씩 속도 증가
 - 속도가 100% 이상일 경우 모터 A,B 바로 정지, 모터 B 부드럽게 정지.

#2 Sensors Test

```
// #2 sensors test
#define SOUND 50
#define BRIGHT 40
#define DISTANCE 15

task main() {
  SetSensorTouch(IN_1);
  SetSensorSound(IN_2);
  SetSensorLight(IN_3);
  SetSensorLowspeed(IN_4);

  until(SENSOR_1 == 1);
  OnFwd(OUT_A, 20);

  until(SENSOR_2 > SOUND);
  OnFwd(OUT_B, 50);

  until(SENSOR_3 >= BRIGHT);
  OnFwd(OUT_C, 50);

  until(SensorUS(IN_4) < DISTANCE);
  Off(OUT_ABC);
}
```

• 요구사항

- 다음 입력을 순차적으로 받아들일 것
- Touch Sensor에 입력이 들어오면 모터 A를 20%의 속력으로 회전 시킴
- Sound Sensor에 40% 이상의 소리가 입력 시 모터 B를 50%의 속력으로 회전 시킴
- Light Sensor에 50% 이상의 빛이 들어올 경우 모터 C를 50%의 속력으로 회전 시킴
- Ultrasonic Sensor로 15Cm 미만의 물체를 감지할 경우 모터 A, B, C 작동을 중단시킴.

#3 Parallel Test

- 요구사항

- 다음 이벤트를 병렬로 처리 할 것
- Touch Sensor에 신호가 '1'일 경우 모터 A를 정방향으로 작동시킨다. 신호가 '0'일 경우 역방향으로 작동시킨다.
- Light Sensor에 40% 이상의 빛이 들어올 경우 모터 B를 정방향으로 작동시킨다. 그 반대일 경우 역방향으로 작동시킨다.
- Ultrasonic Sensor에 15Cm 이하의 물체가 감지되면 모터 C를 정방향으로 작동시킨다. 그 반대일 경우 역방향으로 작동시킨다.

#3 Parallel Test

```
// #3 parallel test
#define BRIGHT 40
#define DISTANCE 15

task motor_A() {
  while(true) {
    if (SENSOR_1 == 1) {
      OnFwd(OUT_A, 75);
      Wait(300);
    }
    else {
      OnRev(OUT_A, 75);
      Wait(300);
    }
  }
}
```

```
task motor_B() {
  while(true) {
    if (SENSOR_3 >
    BRIGHT) {
      OnFwd(OUT_B, 75);
      Wait(300);
    }
    else {
      OnRev(OUT_B, 75);
      Wait(300);
    }
  }
}
```

```
task motor_C() {
  while(true) {
    if (SENSOR_4 < DISTANCE) {
      OnFwd(OUT_C, 75);
      Wait(300);
    }
    else {
      OnRev(OUT_C, 75);
      Wait(300);
    }
  }
}
```

```
task main() {
  Precedes(motor_A, motor_B, motor_C);

  SetSensorTouch(IN_1);
  SetSensorLight(IN_3);
  SetSensorLowspeed(IN_4);
}
```

#4 Problem on Parallel

```
// #4 problem on parallel
task move_fwd() {
    while(true) {
        OnFwd(OUT_A, 20);
        Wait(3000);
    }
}

task move_rev() {
    while(true) {
        if(SENSOR_1 == 1) {
            OnRev(OUT_A, 20);
            Wait(3000);
        }
    }
}

task main() {
    Precedes(move_fwd,
             move_rev);
    SetSensorTouch(IN_1);
}
```

- 두 개의 각기 다른 함수가 동시에 하나의 자원을 사용할 때 예상치 못한 문제점 발생.

#5 Solution - mutex

```
// #4 problem on parallel
mutex motorA;

task move_fwd() {
  while(true) {
    Acquire(motorA);
    OnFwd(OUT_A, 20); Wait(500);
    Release(motorA);
  }
}

task move_rev() {
  while(true) {
    if(SENSOR_1 == 1) {
      Acquire(motorA);
      OnRev(OUT_A, 20); Wait(500);
      Release(motorA);
    }
  }
}

task main() {
  Precedes(move_fwd, move_rev);
  SetSensorTouch(IN_1);
}
```

- Mutex를 이용하여 하나의 자원에 여러 프로세스가 동시에 접근하는 것을 막을 수 있음.

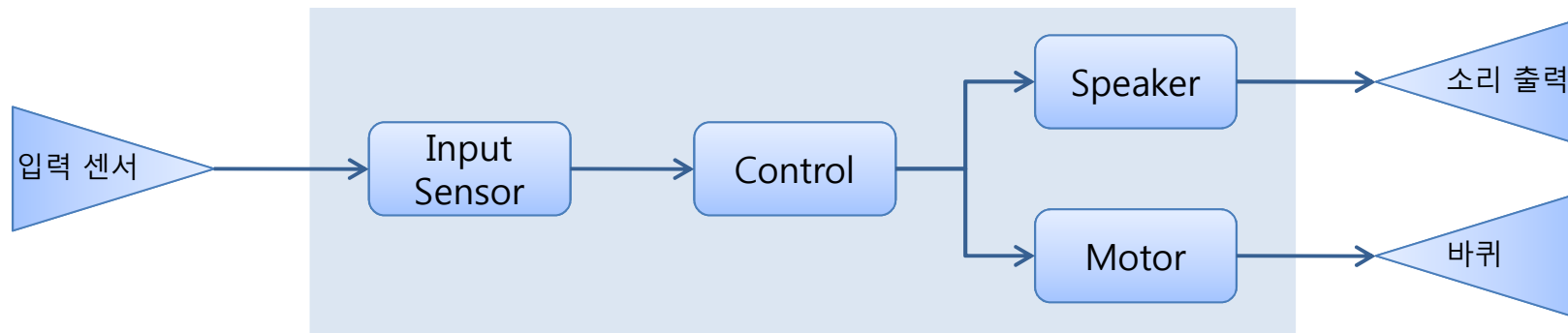
구조적 분석 : 분석서

문제 정의

- 로봇은 정지 상태로 시작한다.
 - 정지 상태에서는 어떤 입력에도 반응하지 않는다.
- 터치센서에 입력이 들어오면 전진하고 누르고 있는 동안 속도가 증가한다.
 - 속도가 최대치(100)에 다다르면 속도를 10으로 돌려놓는다.
- 전진하는 동안 박수를 치게 되면(사운드 센서의 입력이 40 이상이 되면) “! Startup.rso” 소리를 재생한다.
 - “! Startup.rso”소리 파일은 NXT초기 가동 시 출력되는 소리이다.
- 전진하는 동안 밝기가 어두워지면 (라이트센서의 입력이 40 미만이 되면) 동작을 멈춘다.
 - 동작을 멈춘 상태에서는 어떤 입력에도 반응하지 않는다.
 - 다시 밝아진다면 이전 속도로 전진한다.
- 전진 도중에 20Cm 전방의 장애물을 만나면 3초간 후진하고, 다시 이전 속도로 전진한다.
 - 후진 도중에는 어떤 입력에도 반응하지 않는다.

문제 정의

- 시스템 범위 및 구조
 - Input Sensor : 입력 센서를 통해 상황을 인식한다.
 - Speaker : 제어 결과에 따라 소리를 출력한다.
 - Motor : 제어 결과에 따라 로봇을 이동 시킨다.
 - Control : 입력 센서의 입력 정보에 따라 로봇의 작동을 제어한다.



자동 운전 로봇 시스템

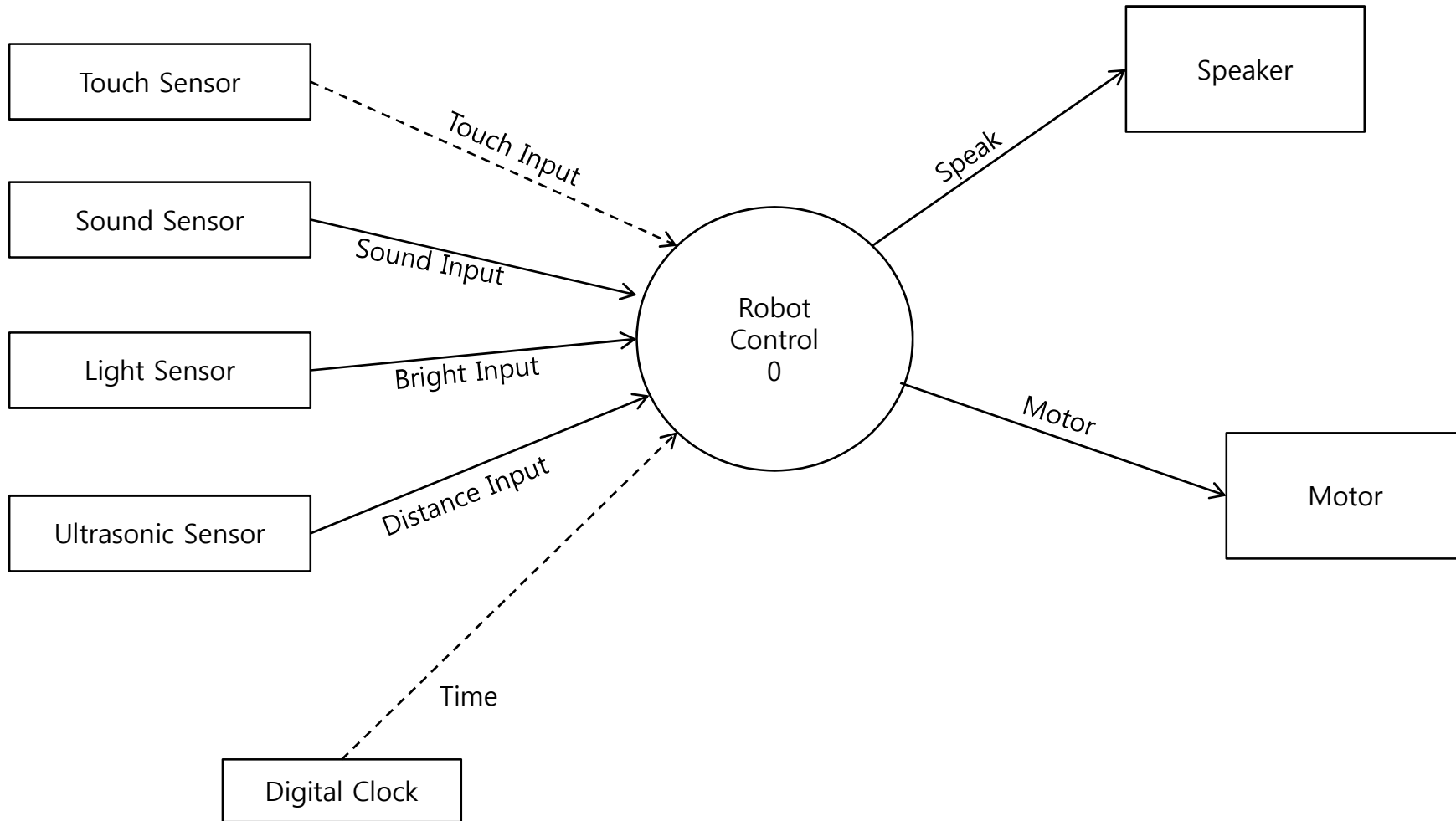
문제 정의

- 입력 센서
 - Ultrasonic Sensor
 - 로봇이 장애물을 인식하게 한다.
 - Sound Sensor
 - 로봇이 사용자의 박수소리에 반응하도록 한다.
 - Light Sensor
 - 로봇이 주변의 밝기를 인식하게 한다.
 - Touch Sensor
 - 사용자가 누르면 반응 하도록 한다.

문제 정의

- 주요 시나리오
 - 속도를 조절한다.
 - 속도를 최대로 높였다가 다시 낮아지는 것을 확인한다.
 - 전방에 장애물을 인식한다.
 - 로봇이 3초간 전진 시와 같은 속도로 후진한다.
 - 3초 후 다시 전과 같은 속도로 전진한다.
 - 불을 끈다.
 - 불이 다시 켜질 때 까지 어떤 동작도 하지 않는다.

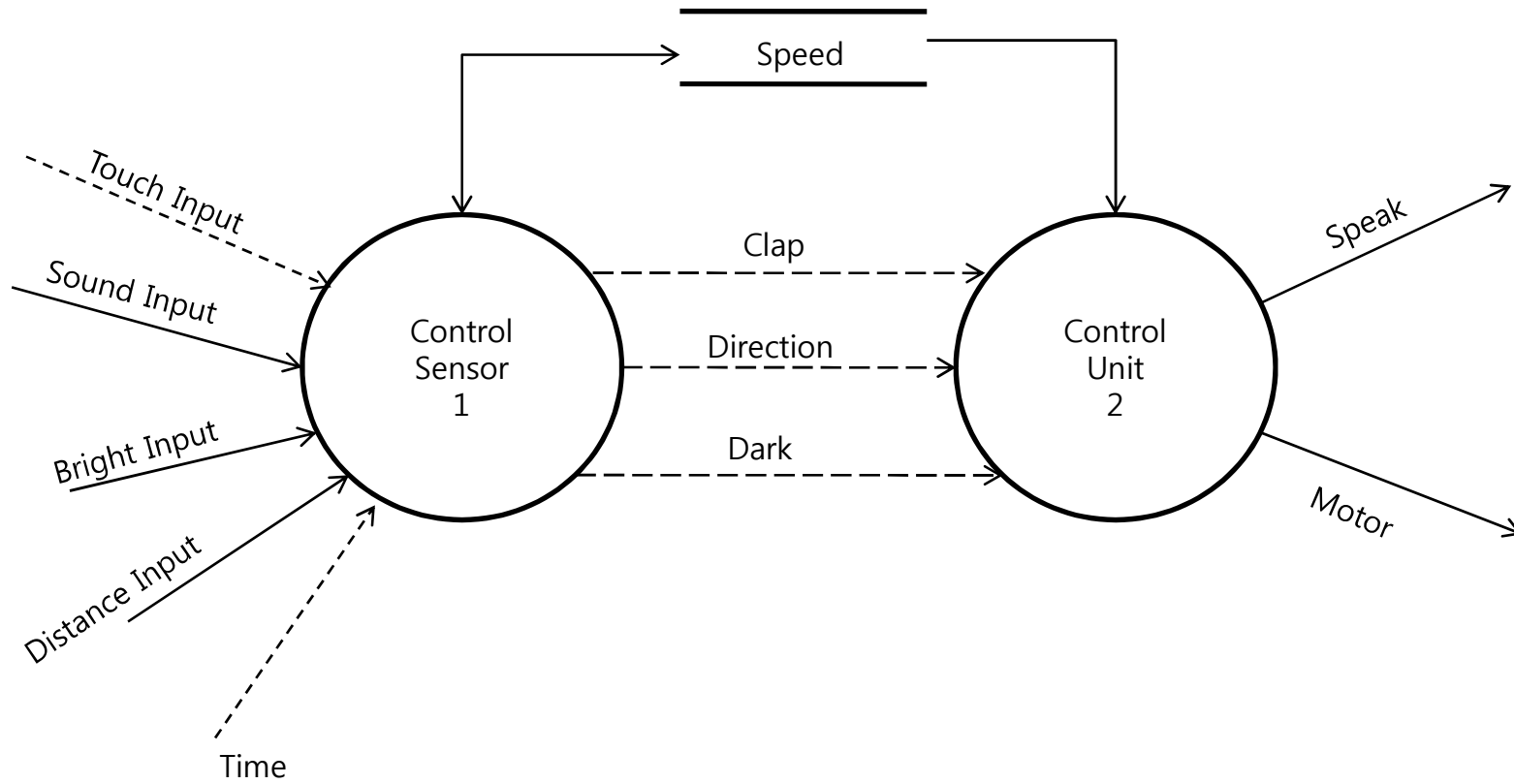
시스템 문맥도 작성



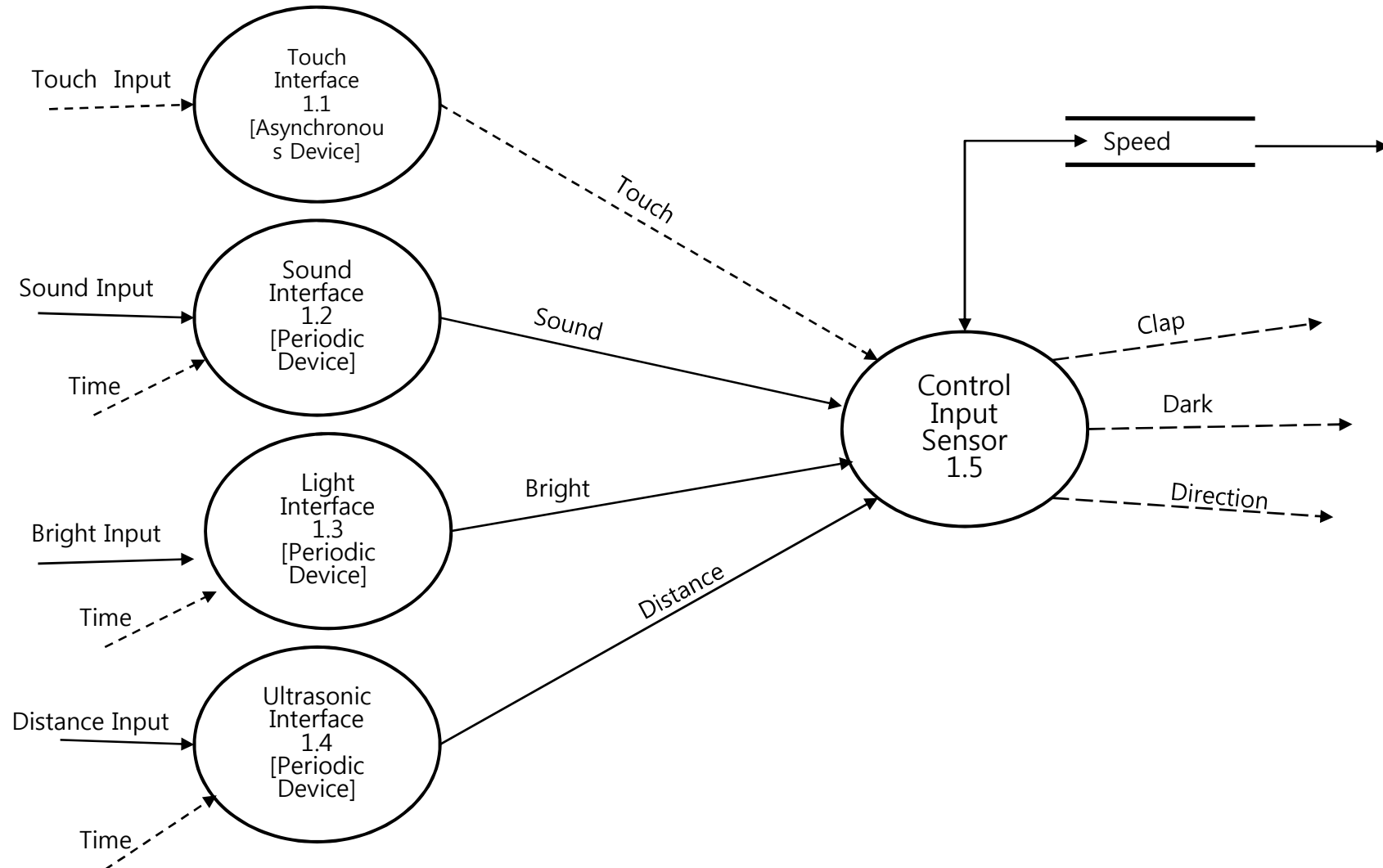
로봇 제어 정보 테이블

정보	설명
Touch Input	터치 센서를 사용자가 눌렀음을 나타낸다.
Sound Input	Sound Sensor를 통해 들어오는 소리의 정보를 나타낸다. 주기적으로 읽혀지며 주기는 10ms이다.
Bright Input	Light Sensor를 통해 들어오는 밝기를 나타내는 정보이다. 주기적으로 읽혀지며 주기는 10ms이다.
Distance Input	Ultrasonic Sensor를 통해 들어오는 사물과의 거리 정보를 나타낸다 주기적으로 읽혀지며 주기는 10ms이다.
Time	타이머 인터럽트로서 10ms 주기로 입력된다.
Speak	Speaker를 통해서 외부로의 출력 정보를 나타낸다.
Motor	로봇 제어에 의한 모터의 제어정보를 나타낸다. 속도와 방향을 가진다.

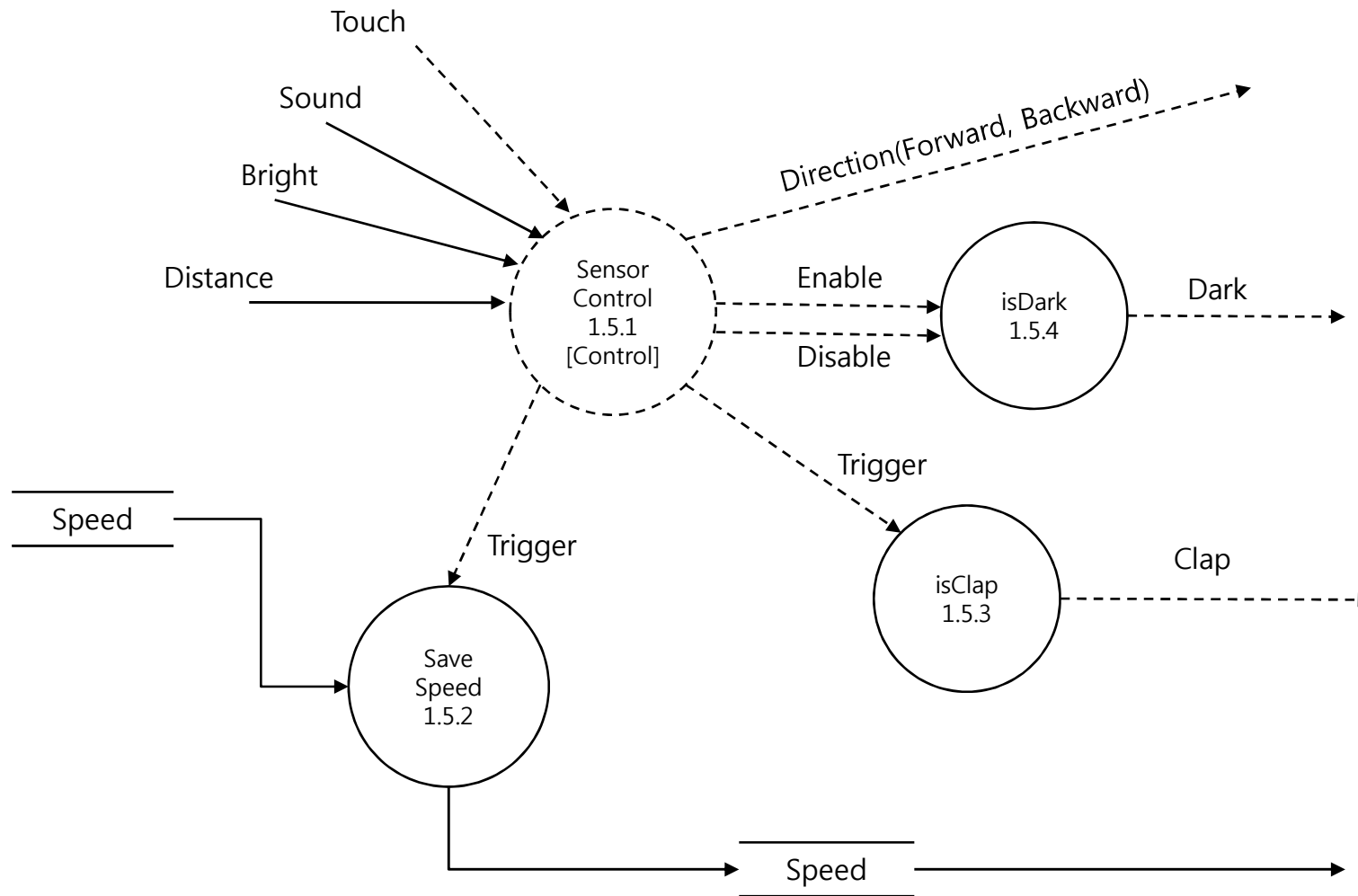
데이터 흐름도 작성



Control Sensor 1 분할

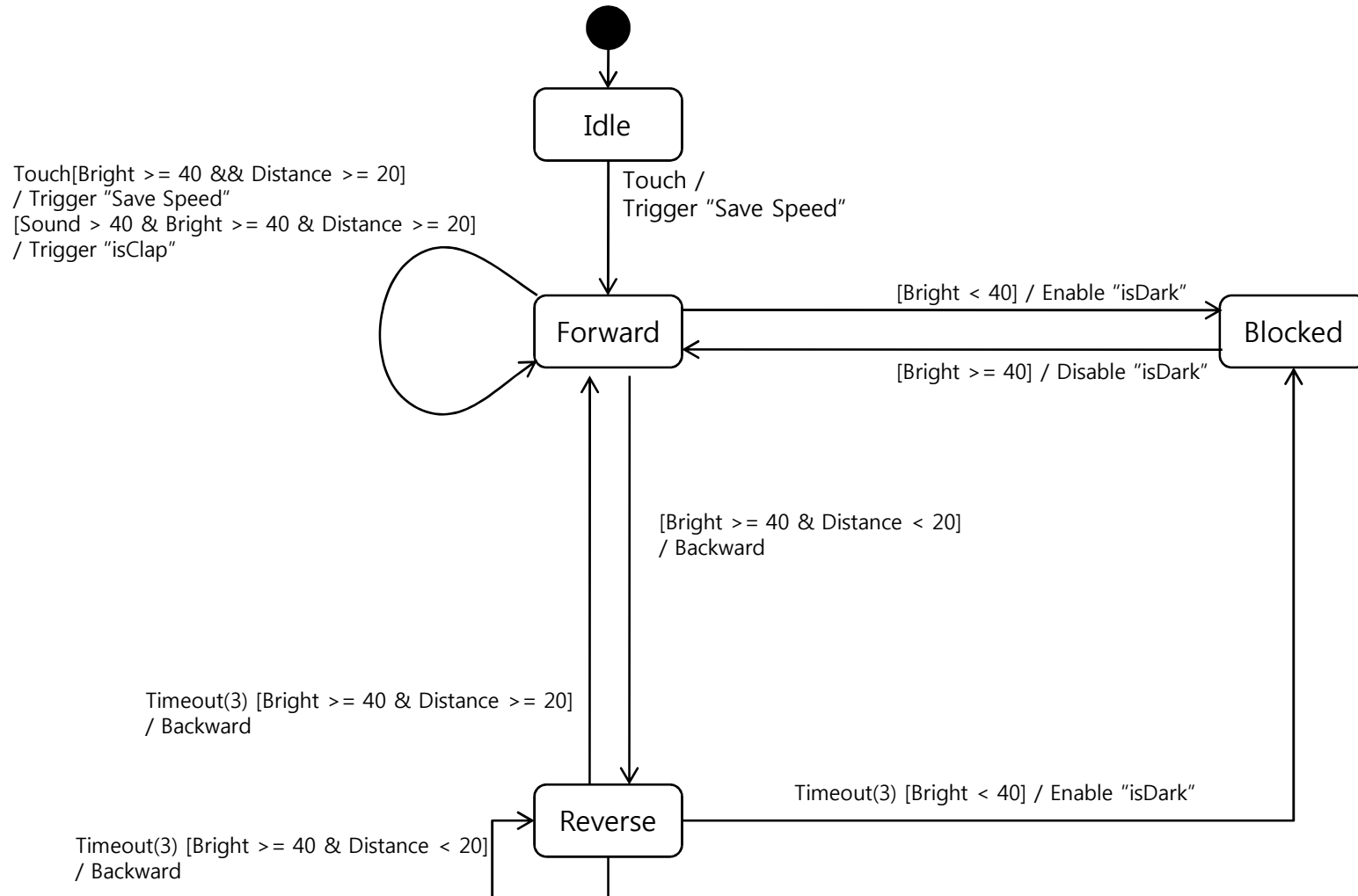


Control Input Sensor 1.5 분할

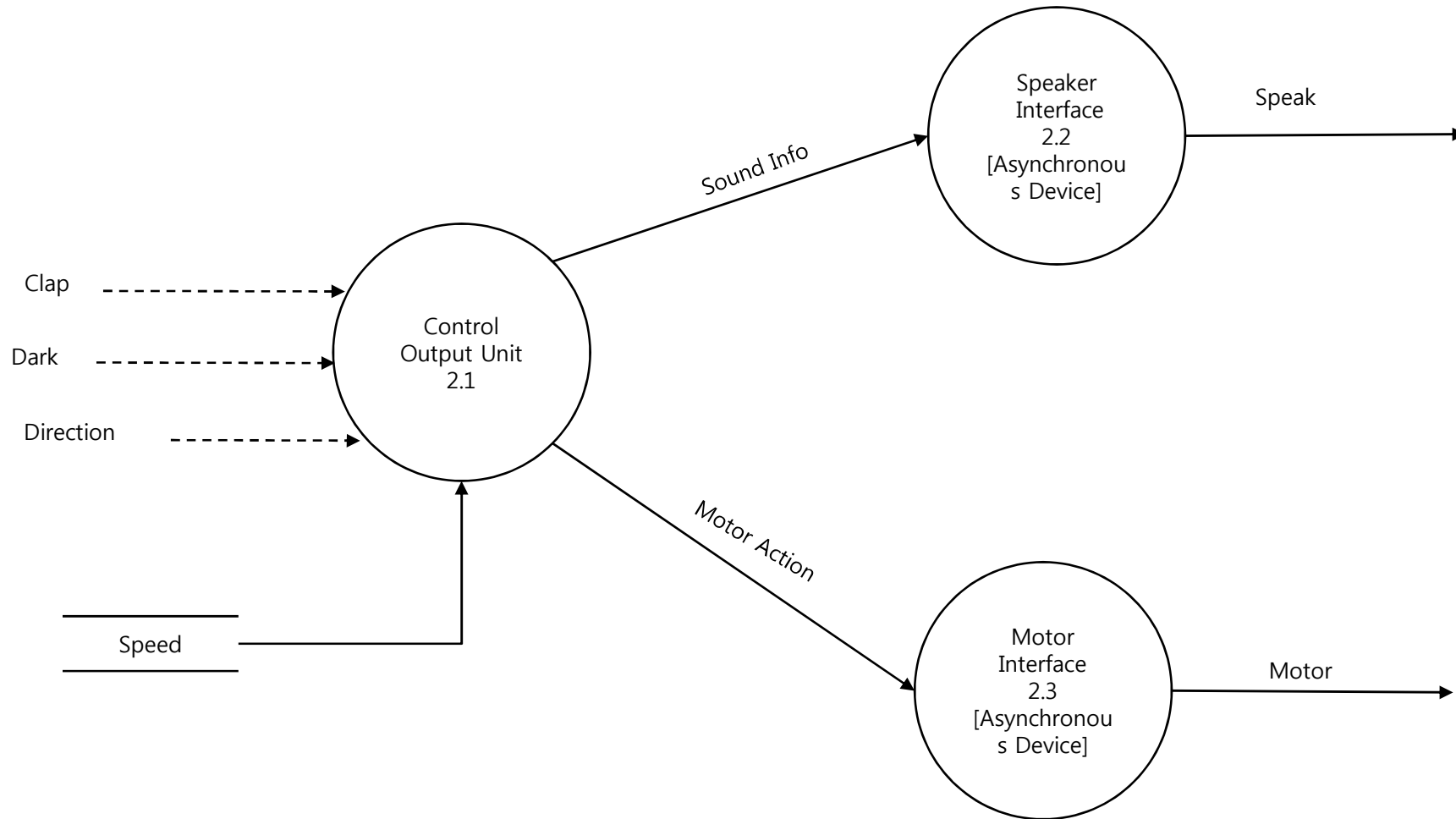


상태 머신 – sensor control

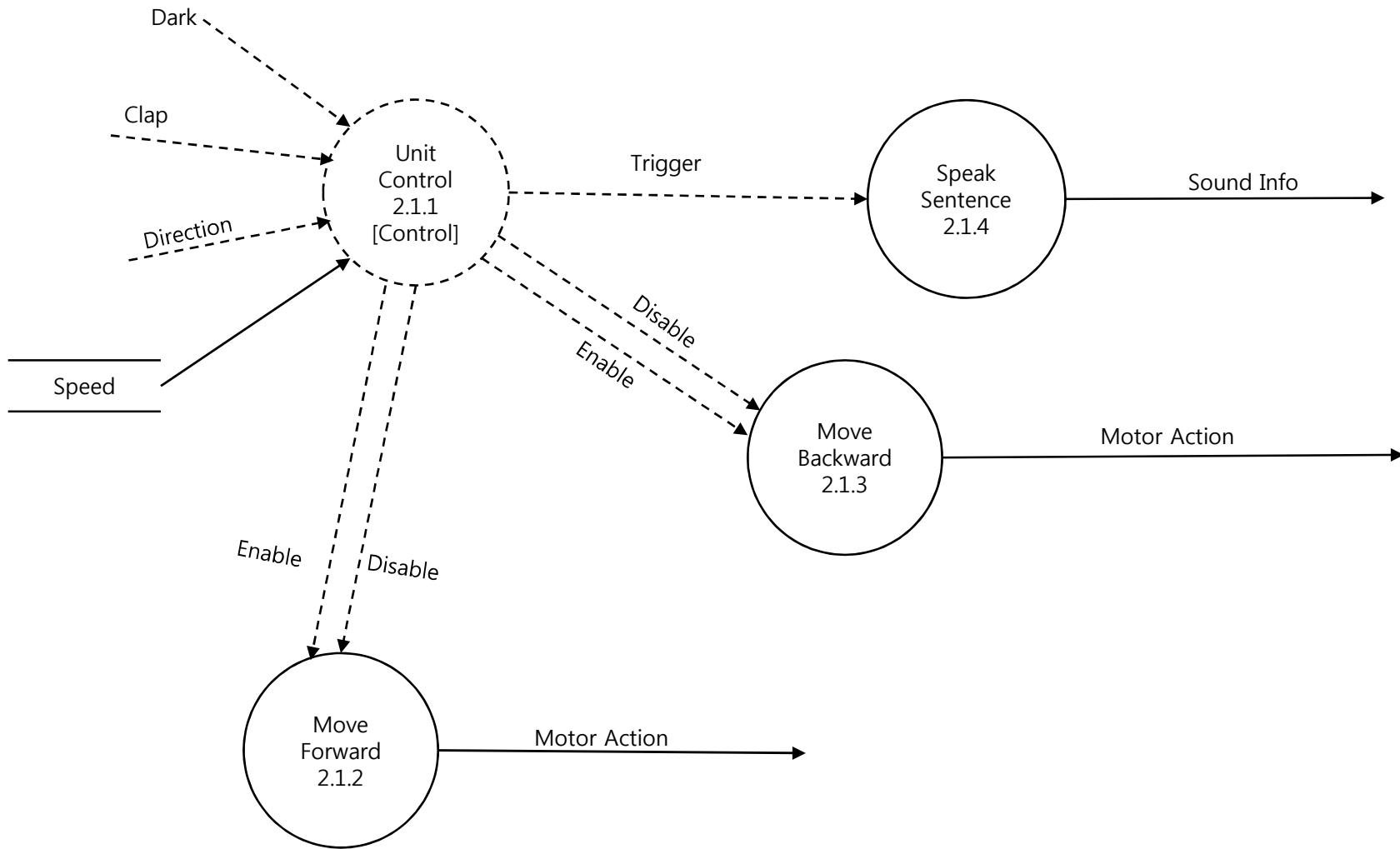
1.5.1



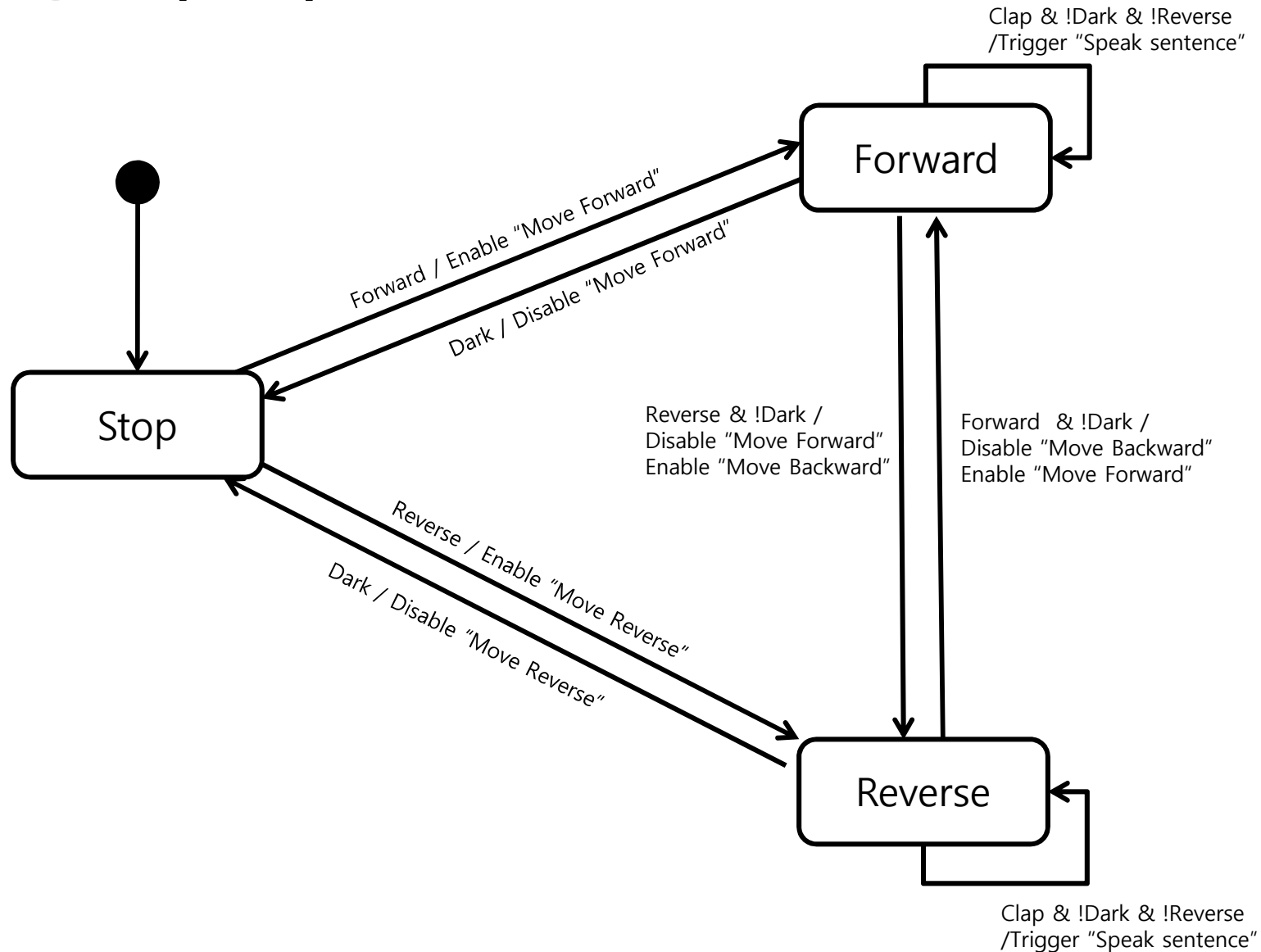
Control Unit 2 분할

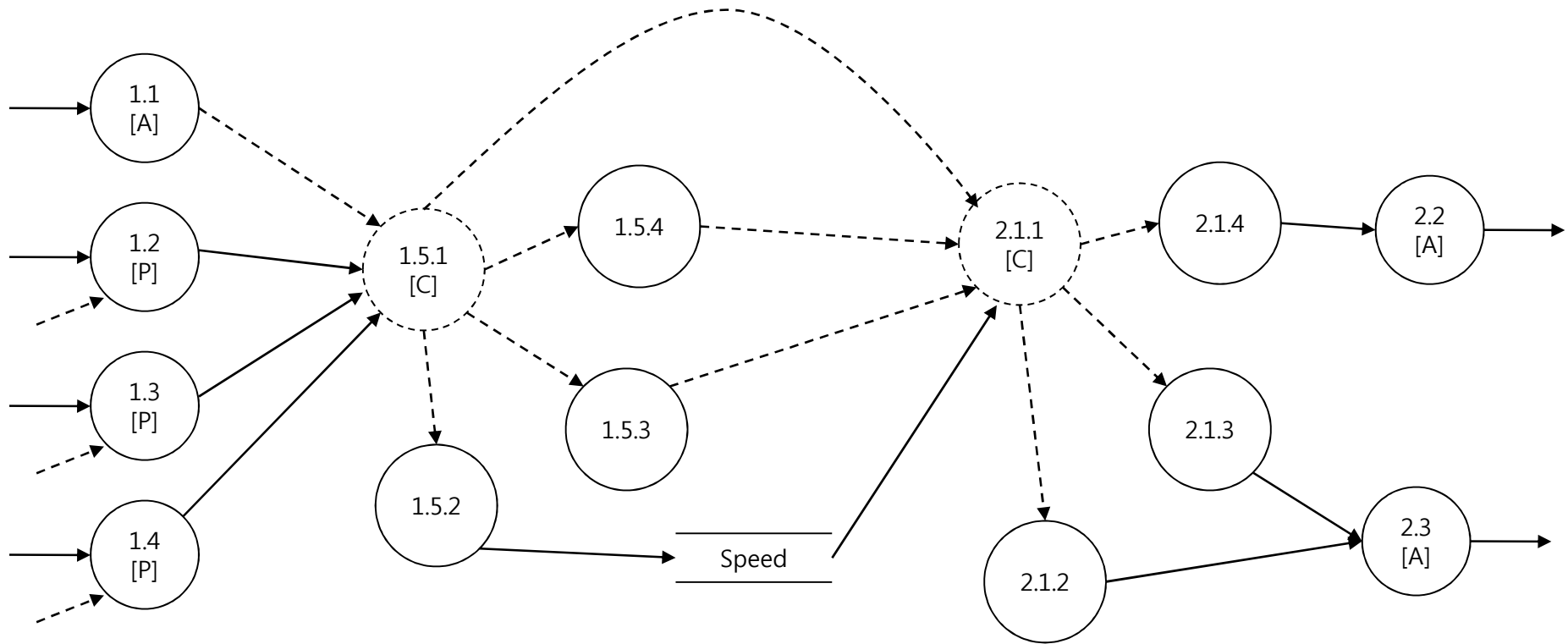


Control Output Unit 2.1 분할



상태 머신 –Unit Control 2.1.1





A : Asynchronous
 P : Periodic
 C : Control

프로세스 명세

PSpec 1.1	Touch Interface
Stereotype	Asynchronous Function
Input	Touch Input
Output	(bool)Touch
Process	
Touch Sensor에서 Touch Input을 받으면 Sensor Control 프로세스에 Touch를 보내어 Touch Input이 들어왔음을 알린다. Touch는 True/False로 구분되며 입력이 들어왔을 때가 True이다.	
PSpec 1.2	Sound Interface
Stereotype	Periodic Function
Input	Sound Input
Output	(int)Sound
Process	
Sound Sensor에서 Sound Input을 받으면 Sensor Control 프로세스에 Sound를 보낸다. Sound는 Sound Sensor에서 감지한 소리의 크기 정보 데이터다.	

PSpec 1.3	Light Interface
Stereotype	Periodic Funtion
Input	Bright Input
Output	(int)Bright
Process	
Light Sensor에서 Bright Input을 받으면 Sensor Control 프로세스에 Bright를 보낸다. Bright는 Light Sensor에서 감지한 밝기 정보 데이터다.	

PSpec 1.4	Ultrasonic Interface
Stereotype	Periodic Function
Input	Distance Input
Output	(int)Distance
Process	
Ultrasonic Sensor에서 Distance Input을 받으면 Sensor Control 프로세스에 Distance를 보낸다. Distance는 Ultrasonic Sensor에서 감지한 물체와의 거리 정보 데이터다.	

PSpec 1.5.1	Sensor Control
Stereotype	Control
Input	(bool)Touch, (int)Sound, (int)Distance In, (bool)Dark
Output	Trigger, (bool)Direction
Process	
현재의 상태에서 입력에 따라 행동을 결정하고, 행동을 수행하기 위한 이벤트를 발생시킨다.	

PSpec 1.5.2	Save Speed
Stereotype	Synchronous Function
Input	Trigger, (int)Speed
Output	(int)Speed
Process	
Trigger 이벤트를 통해서 수행되며 현재의 Speed 데이터를 읽어서 크기가 100 이상이면 10으로, 100미만이면 현재 Speed에서 10 추가한 값을 저장한다.	

PSpec 1.5.3	isClap
Stereotype	Synchronous Function
Input	Trigger
Output	(bool)Clap
Process	
<p>Trigger 이벤트를 통하여 수행되며 Unit Control 프로세스에게 Clap 이벤트를 보내어 Speak Sentence 프로세스에게 Trigger 이벤트를 보내도록 하는 프로세스이다.</p>	

PSpec 1.5.4	isDark
Stereotype	Synchronous Function
Input	Enable, Disable
Output	(bool)Dark
Process	
<p>Enable 이벤트가 발생시 수행되며 Unit Control 프로세스에게 Dark 이벤트를 보내어 Move Reverse 프로세스에게 Enable 이벤트를 보내도록 하는 프로세스이다.</p>	

PSpec 2.1.1	Unit Control
Stereotype	Control
Input	(bool)Dark, (bool)Clap, (bool)Direction, (int)Speed
Output	Trigger, Enable, Disable
Process	
현재의 상태에서 입력에 따라 행동을 결정하고, 행동을 수행하기 위한 이벤트를 발생시킨다.	

PSpec 2.1.2	Move Forward
Stereotype	Synchronous Function
Input	Enable, Disable
Output	(int, bool)Motor Action
Process	
Enable 이벤트가 발생시 수행되며 Motor Interface에게 Motor Action 데이터를 보내어 Motor가 앞으로 동작하도록 하는 프로세스이다. Motor Action에는 Motor의 진행 방향과 속도가 포함된다.	

PSpec 2.1.3	Move Backward
Stereotype	Synchronous Function
Input	Enable, Disable
Output	(bool, int)Motor Action
Process	
<p>Enable 이벤트가 발생시 수행되며 Motor Interface에게 Motor Action 데이터를 보내어 Motor가 뒤로 동작하도록 하는 프로세스이다. Motor Action에는 Motor의 진행 방향과 속도가 포함된다.</p>	

PSpec 2.1.4	Speak Sentence
Stereotype	Synchronous Function
Input	Trigger
Output	Sound Info
Process	
<p>Trigger 이벤트가 발생시 수행되며 Speaker Interface에게 Sound Info 데이터를 보내어 Speaker가 소리를 내도록 하는 프로세스이다. Sound Info는 Speaker가 낼 실제 소리 정보다.</p>	

PSpec 2.2	Speaker Interface
Stereotype	Asynchronous Function
Input	Sound Info
Output	Speak
Process	
Sound Info 데이터를 받아 Speaker에게 Speak 데이터를 보내 소리를 내게 하는 프로세스이다.	

PSpec 2.3	Motor Interface
Stereotype	Asynchronous Function
Input	Motor Action
Output	Motor
Process	
Motor Action 데이터를 받아 Motor에게 Motor 데이터를 보내어 움직이게 하는 프로세스이다.	

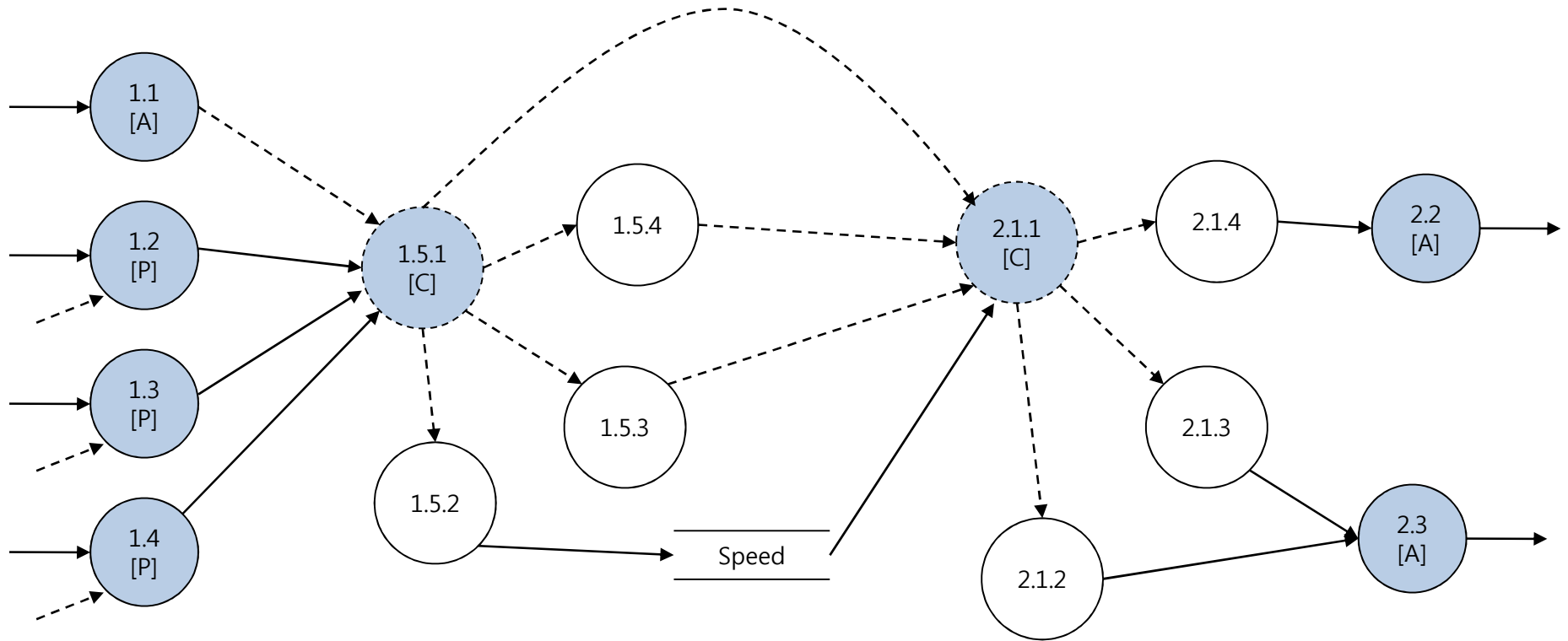
데이터 사전 작성

데이터 명	설명
Clap	Sound Sensor를 통해 들어오는 소리의 크기가 40을 넘었음을 알려, Speaker를 통해 Speak를 내기 위해 보내는 정보다.
Dark	Bright가 40미만이면 로봇의 모든 동작을 멈추기 위해 보내는 정보이다. True와 False로 구분되며 True일 때 Bright가 40미만임을 나타낸다.
Direction	Distance의 정보에 따라 모터의 방향을 결정하기 위해 보내는 정보다. Direction의 값은 Distance가 15미만일 때 Backward, 15이상일 때 Forward가 된다.
Speed	0이상 100이하의 값을 가지며 값은 모터의 최대속도에 대한 비율을 나타낸다. Touch가 들어올 때마다 10씩 증가하며, Speed의 값이 100일 경우 Touch가 들어오면 Speed의 값은 10이 된다. 초기값은 10이다.

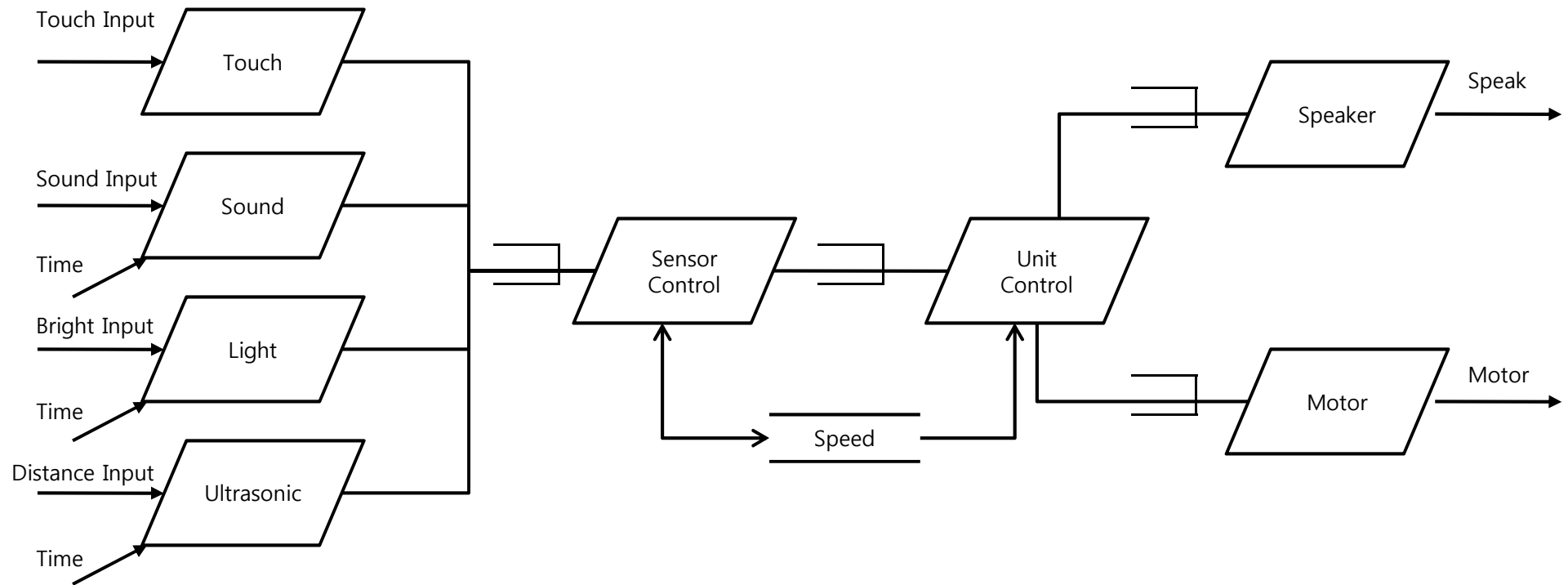
데이터 명	설 명
Sound Info	Speaker가 실제로 낼 소리 데이터를 나타낸다. NXT에 내장되어 있는 소리 데이터 베이스를 이용하며 프로그램 코드상에서 지정 할 수 있다.
Motor Action	모터의 속도와 방향 정보를 포함하는 데이터 이다. 데이터 저장소인 Speed와 Direction을 사용한다.
Touch	Touch Sensor로부터 Touch Input이 들어왔음을 나타내는 이벤트이다. True/False로 구성되며 True일 때가 Touch Input이 들어왔을 때이다.
Sound	Sound Sensor로부터 들어오는 Sound Input의 크기 데이터이다.
Bright	Light Sensor로부터 들어오는 Bright Input의 크기 데이터이다.
Distance	Ultrasonic Sensor를 통해 들어오는 Distance Input의 크기 데이터이다.

구조적 설계 : 설계서

후보 태스크 선정



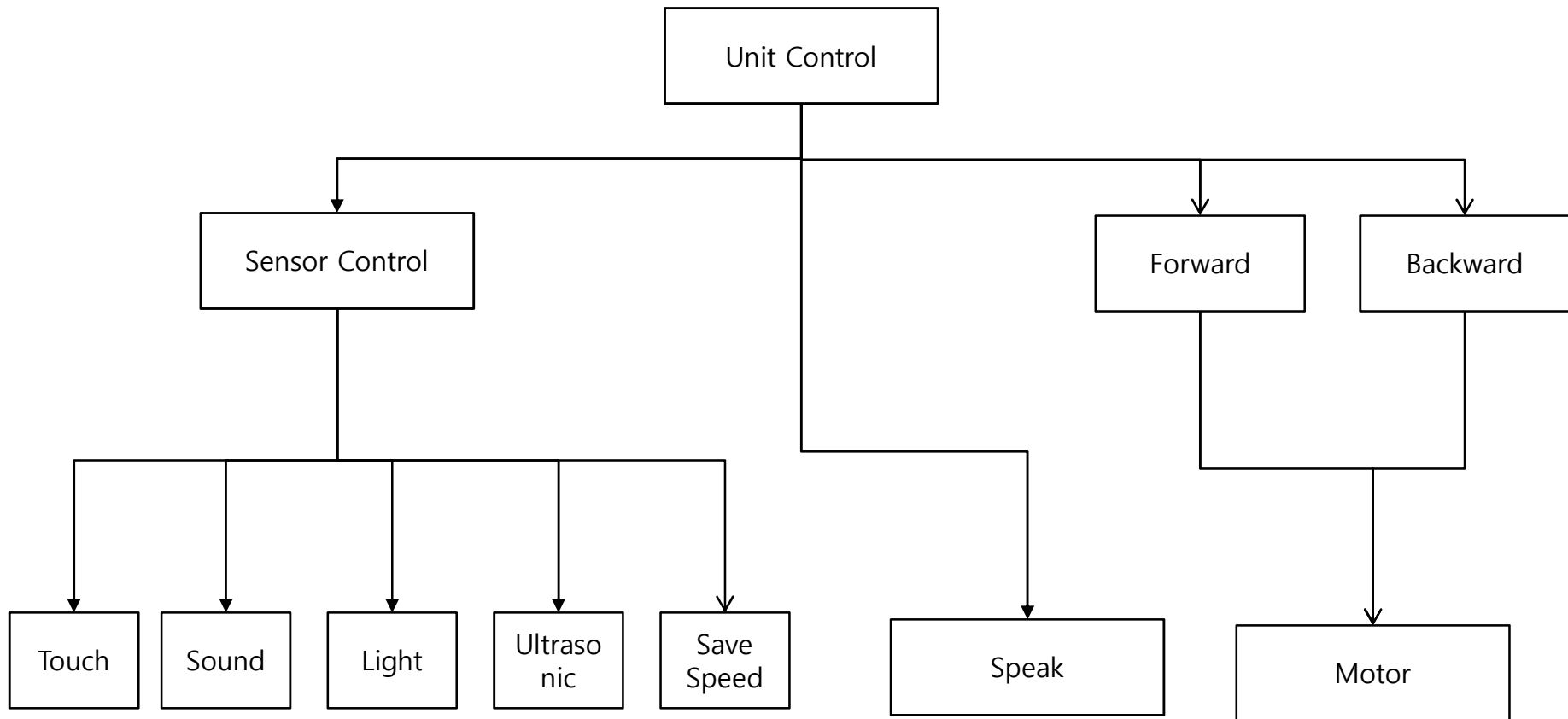
TASK MODEL



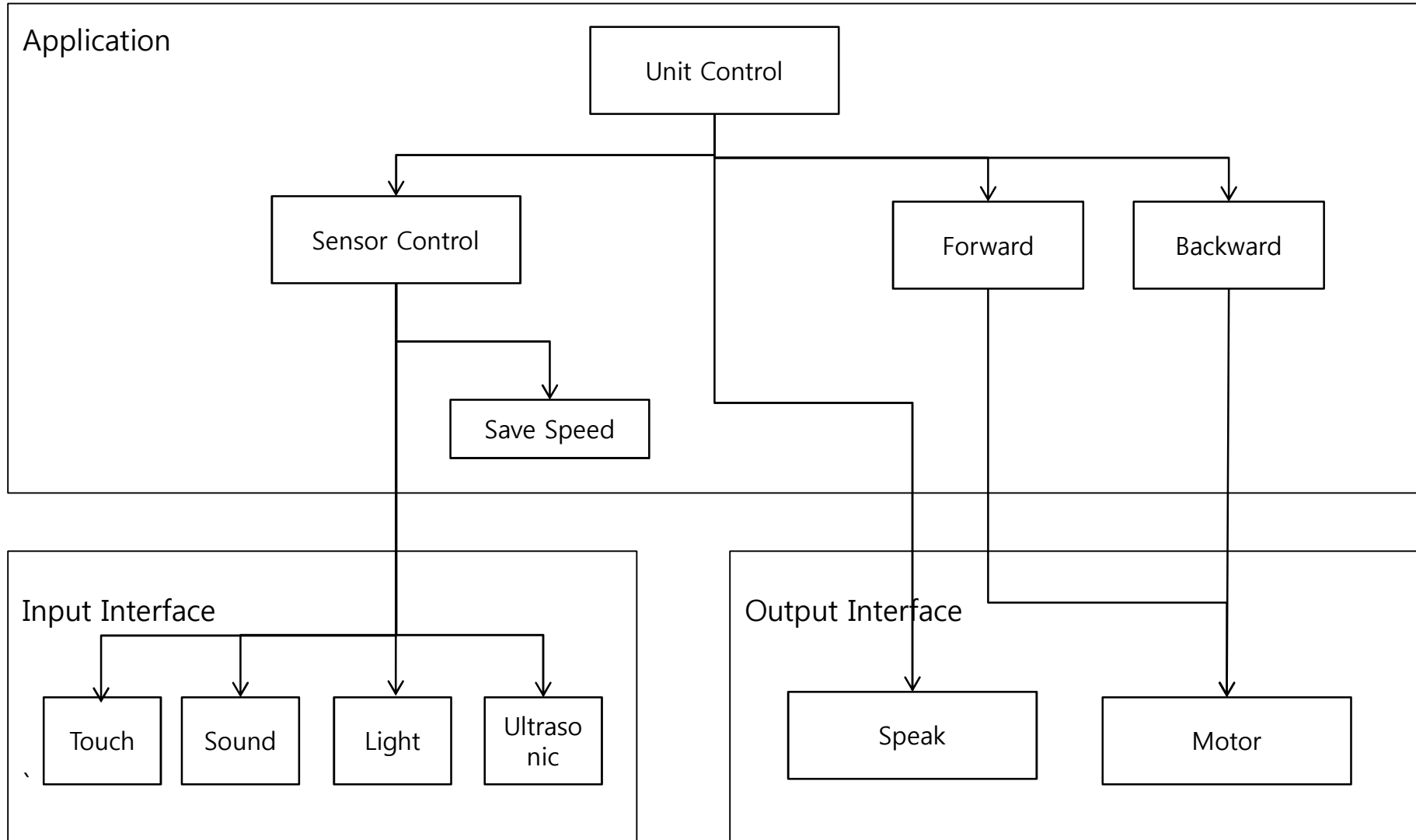
식별된 태스크에 대한 모듈 정의

	프로세스	선별기준	모듈
Task 1	Touch Interface 1.1	Asynchronous	Touch
Task 2	Sound Interface 1.2	Periodic	Sound
Task 3	Light Interface 1.3	Periodic	Light
Task 4	Ultrasonic Interface 1.4	Periodic	Ultrasonic
Task 5	Sensor control 1.5.1 Save Speed 1.5.2 isClap 1.5.3 isDark 1.5.4	Control 제어 응집도 제어 응집도 제어 응집도	Sensor control
Task 6	Unit control 2.1.1 Speak Sentence 2.1.2 Move Forward 2.1.3 Move Backward 2.1.4	Control 제어 응집도 제어 응집도 제어 응집도	Unit control
Task 7	Speaker Interface 2.2	Asynchronous	Speaker
Task 8	Motor Interface 2.3	Asynchronous	Motor

모듈 구조 - 초기



모듈 구조 - 상세



모듈 정의

모듈 ID	RC_AM01	모듈 명	Unit Control
모듈 개요	Sensor Control로 부터의 입력값을 이용해 Speaker와 Motor을 제어함		
인터페이스	없음 - 프로그램 구동 시에 실행		

모듈 ID	RC_AM02	모듈 명	Sensor Control
모듈 개요	4개의 센서로부터 입력을 받아 처리함		
인터페이스	get_sensors() - 입력된 값들을 처리하여 구조체(m_sensors)를 반환하는 함수		

모듈 ID	RC_AM03	모듈 명	Save Speed
모듈 개요	Sensor Control에 의해 불러져 모터의 속도를 저장해 줌		
인터페이스	save_speed(int m_current_speed) - 현재 속도를 인자로 받아 속도를 확인한 후 계산하여 다시 반환 해주는 함수		

모듈 정의

모듈 ID	RC_AM04	모듈 명	Forward
모듈 개요	현재 저장된 속도에 맞게 모터를 정방향으로 회전시킴		
인터페이스	set_forward(int m_current_speed) – 현재 저장된 속도를 인자로 받아 모터를 작동시키는 함수		

모듈 ID	RC_AM05	모듈 명	Backward
모듈 개요	현재 저장된 속도에 맞게 모터를 역방향으로 회전시킴		
인터페이스	set_backward(int m_current_speed) – 현재 저장된 속도를 인자로 받아 모터를 작동시키는 함수		

모듈 ID	RC_IM01	모듈 명	Touch
모듈 개요	외부에 있는 Touch Sensor로 부터의 입력을 소프트웨어에서 사용할 수 있도록 처리함.		
인터페이스	get_touch_input() – Touch Sensor의 입력을 받아 (bool)m_touch로 반환하는 함수		

모듈 정의

모듈 ID	RC_IM02	모듈 명	Sound
모듈 개요	외부에 있는 Sound Sensor로 부터의 입력을 소프트웨어에서 사용할 수 있도록 처리함.		
인터페이스	get_sound_input() – Sound Sensor의 입력을 받아 (int)m_sound로 반환하는 함수		

모듈 ID	RC_IM03	모듈 명	Light
모듈 개요	외부에 있는 Light Sensor로 부터의 입력을 소프트웨어에서 사용할 수 있도록 처리함.		
인터페이스	get_light_input() – Light Sensor의 입력을 받아 (int)m_bright로 반환하는 함수		

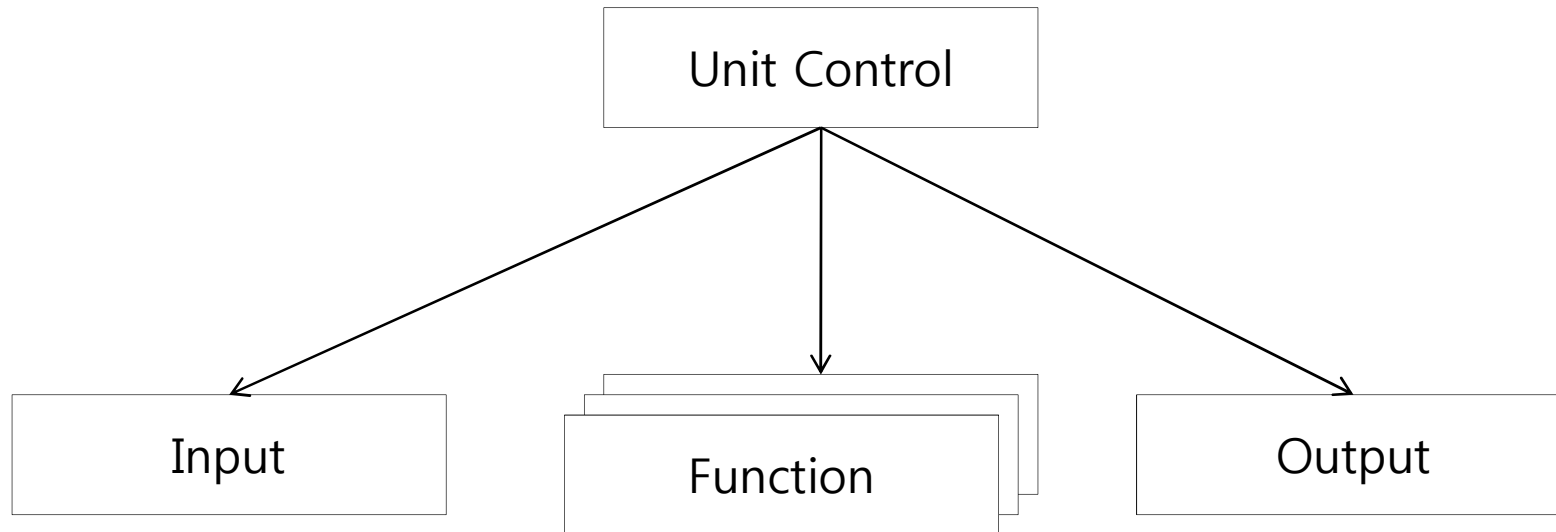
모듈 ID	RC_IM04	모듈 명	Ultrasonic
모듈 개요	외부에 있는 Ultrasonic Sensor로 부터의 입력을 소프트웨어에서 사용할 수 있도록 처리함.		
인터페이스	get_ultrasonic_input() – Ultrasonic Sensor의 입력을 받아 (int)m_distance로 반환하는 함수		

모듈 정의

모듈 ID	RC_OM01	모듈 명	Speaker
모듈 개요	소프트웨어에서 처리된 출력값을 이용해 외부 스피커를 동작할 수 있도록 처리함		
인터페이스	set_speaker() - 외부 장치인 Speaker를 동작하게 하는 함수		

모듈 ID	RC_OM02	모듈 명	Motor
모듈 개요	소프트웨어에서 처리된 출력값을 이용해 외부 스피커를 동작할 수 있도록 처리함		
인터페이스	set_motor() - 현재 속도 값인 (int)m_current_speed를 인자로 받는 함수 set_forward()또는 set_backward()에 의해 불러지는 함수. 모터의 동작을 제어하는 함수이다.		

Balanced System



외부 입력

- Dark
- Clap
- Direction

데이터 저장소 입력

- Speed

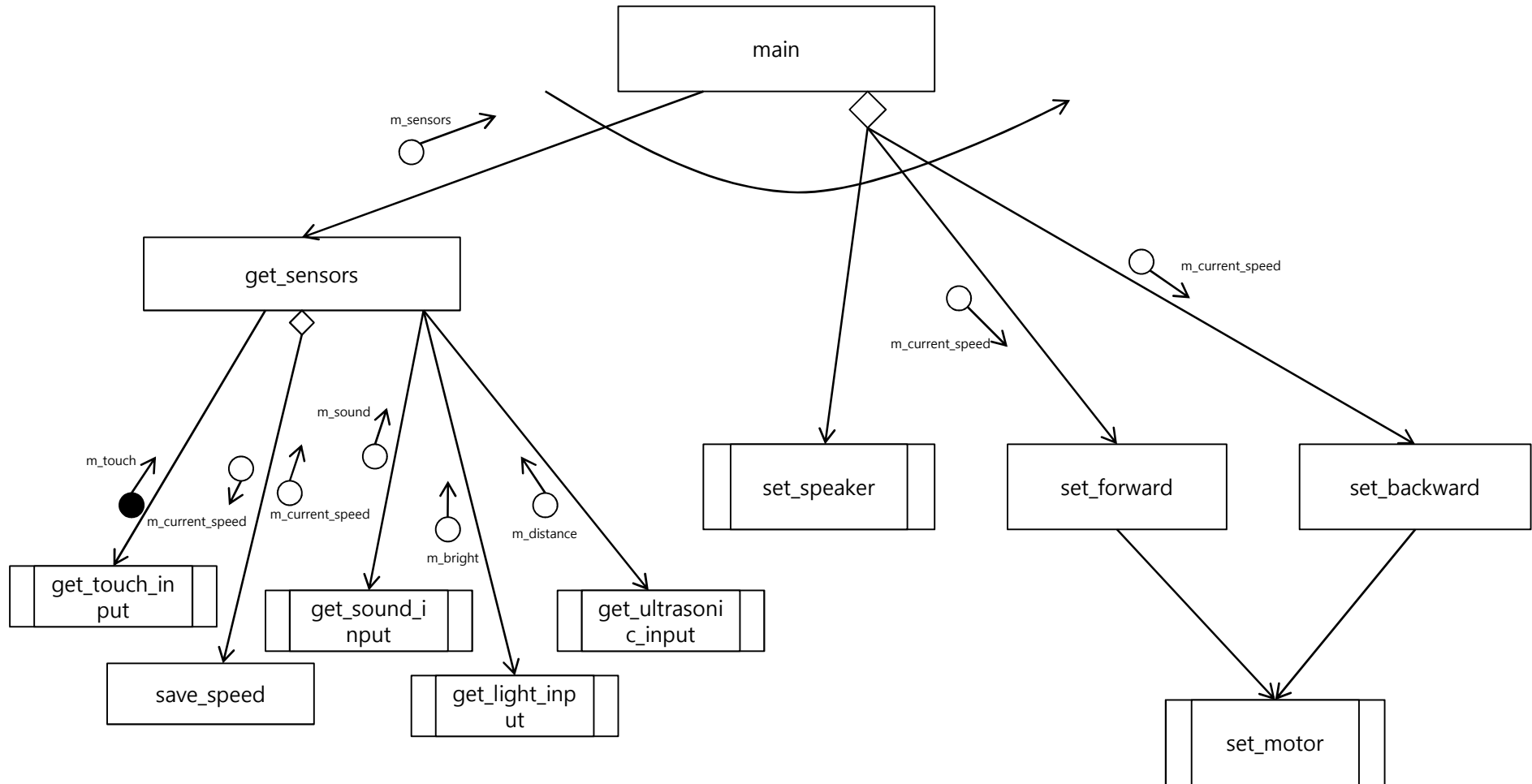
내부 프로세스

- Speak Sentence
- Move Reverse
- Move Forward

외부 출력

- Sound Info
- Motor Action

구조 차트



데이터 정의

데이터 명	설명	타입
m_touch	터치 센서의 입력 정보를 가진 변수.	bool
m_current_speed	로봇이 움직이는 속도 정보를 가진 전역 변수	int (0~100)
m_sound	사운드 센서의 입력정보를 가진 변수	int(0~100)
m_bright	라이트 센서의 입력정보를 가진 변수	int(0~100)
m_distance	울트라소닉 센서의 입력정보를 가진 변수	int(0~255)
m_sensors	센서들의 입력정보가 유효한지 여부를 가진 구조체	struct (m_sound, m_bright, m_distance)