

Assignment 1 Feedback

소프트웨어 공학 개론

A반

Minwoo Kim

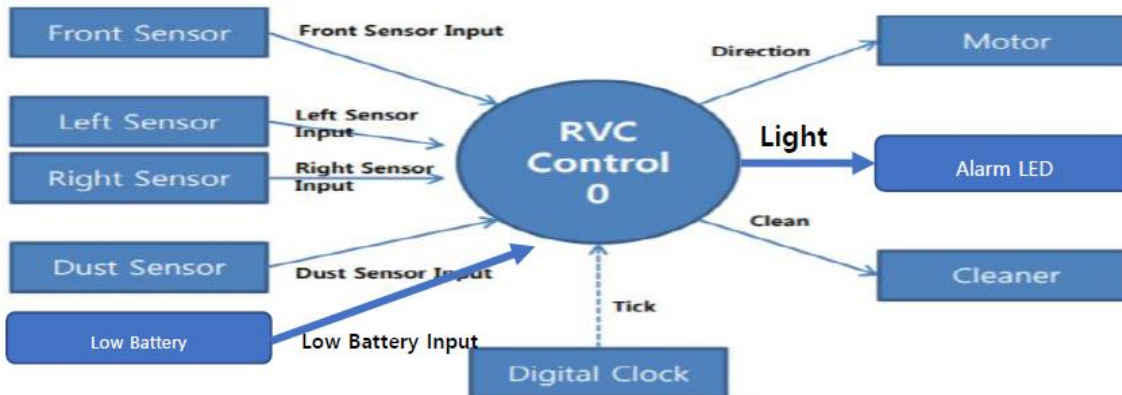
(mwkim.dslab@gmail.com)

Ex 1

- Additional function

- Battery를 추가하여 배터리 잔량이 적을 시 led로 경고

-DFD Level 0



- Sensor terminal의 명칭을 명확히 표기 -

- "low battery"라고 표현된 terminal을 "Battery level sensor" 혹은 "Battery low level sensor"로 수정.

Ex 1

-Data Dictionary

Input/Output Event	Description	Format / Type
Front Sensor Input	Detects obstacles in front of the RVC	True/False , Interrupt
Left Sensor Input	Detects obstacles in the left side of the RVC periodically	True/False , Periodic
Right Sensor Input	Detects obstacles in the right side of the RVC periodically	True/False, Periodic
Dust Sensor Input	Detects dust on the floor periodically	True/False, Periodic
Direction	Direction commands to the motor (go forward/turn left with an angle/turn right with an angle	Forward/Left/Right/Stop
Clean	Turn off/Turn on/Power up	On/Off/Up
Low Battery Input	Detects low battery status of the RVC periodically	True/False, Periodic
Alarm LED	Turn on / Turn off	On / Off

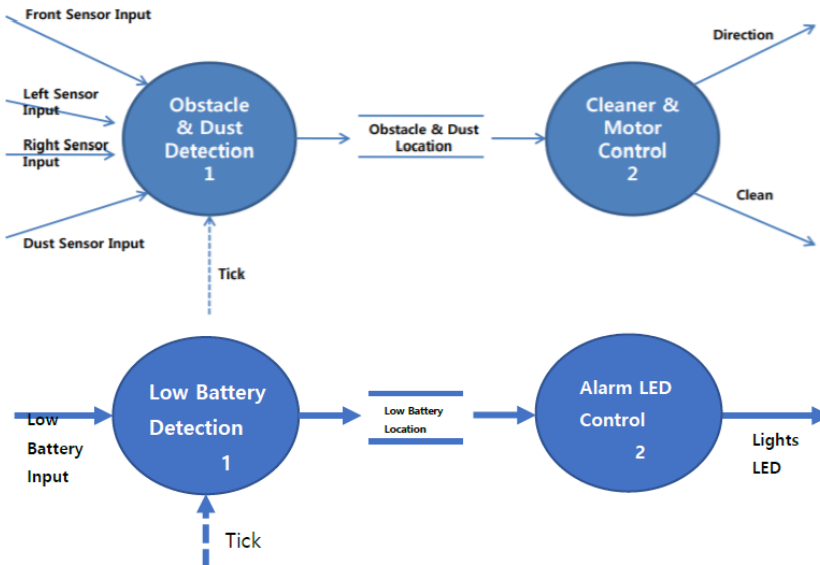
Ex 1

- DFD level numbering 수정

- Numbering은 Input process 부분을 하고 control process 하는 것이 DFD 작성할 때 좀더 수월합니다.

- Low battery detection 1 → Low battery detection 2
- Cleaner & Motor Control 2 → Cleaner & Motor Control 3
- Alarm LED control 2 → Alarm LED control 4

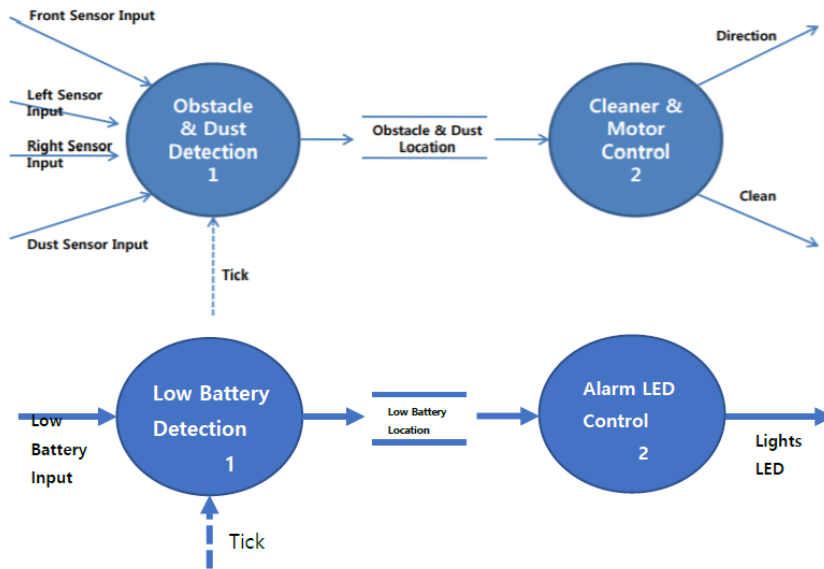
-DFD Level 1



Ex 1

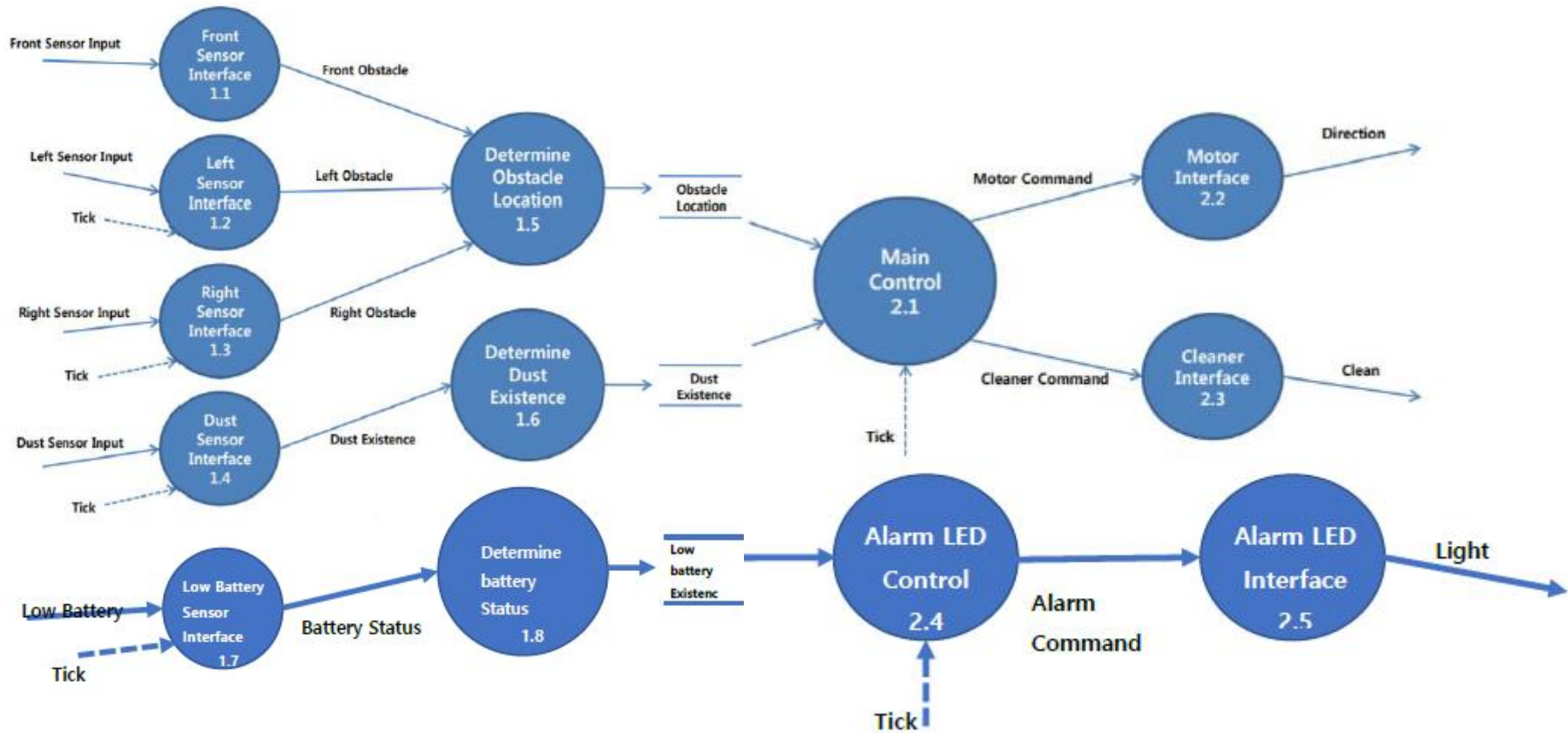
- Data Store가 필요 없다면 제거
 - Data process에서 받은 값이 control process로 data를 전달할 때, 저장하지 않고 바로 사용하고 끝난다면 data store가 없어도 됩니다.

-DFD Level 1



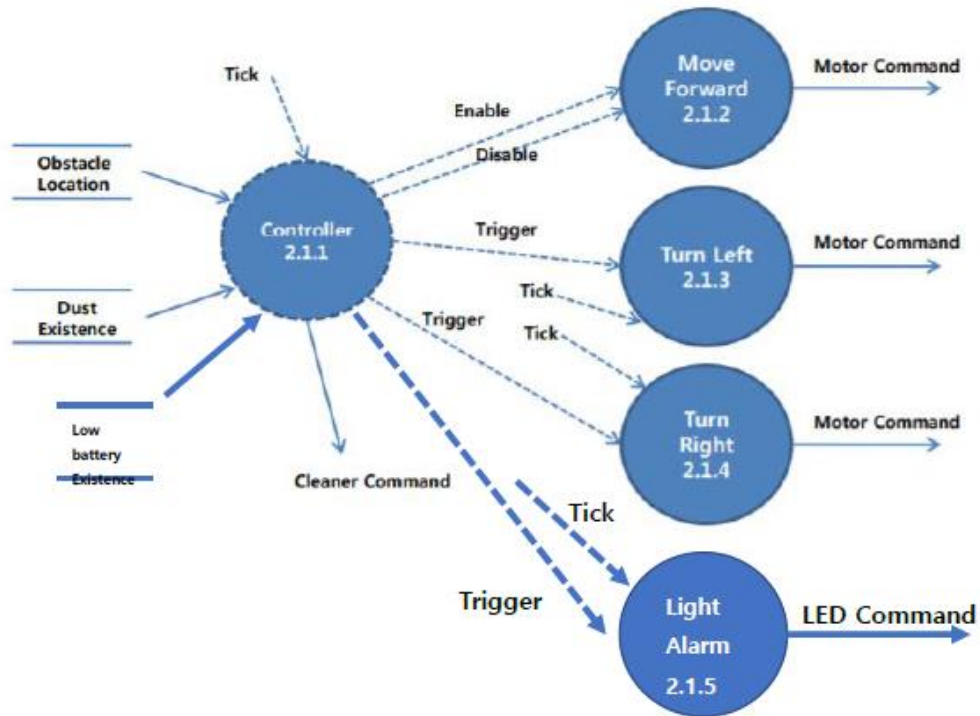
Ex 1

- DFD level 2



Ex 1

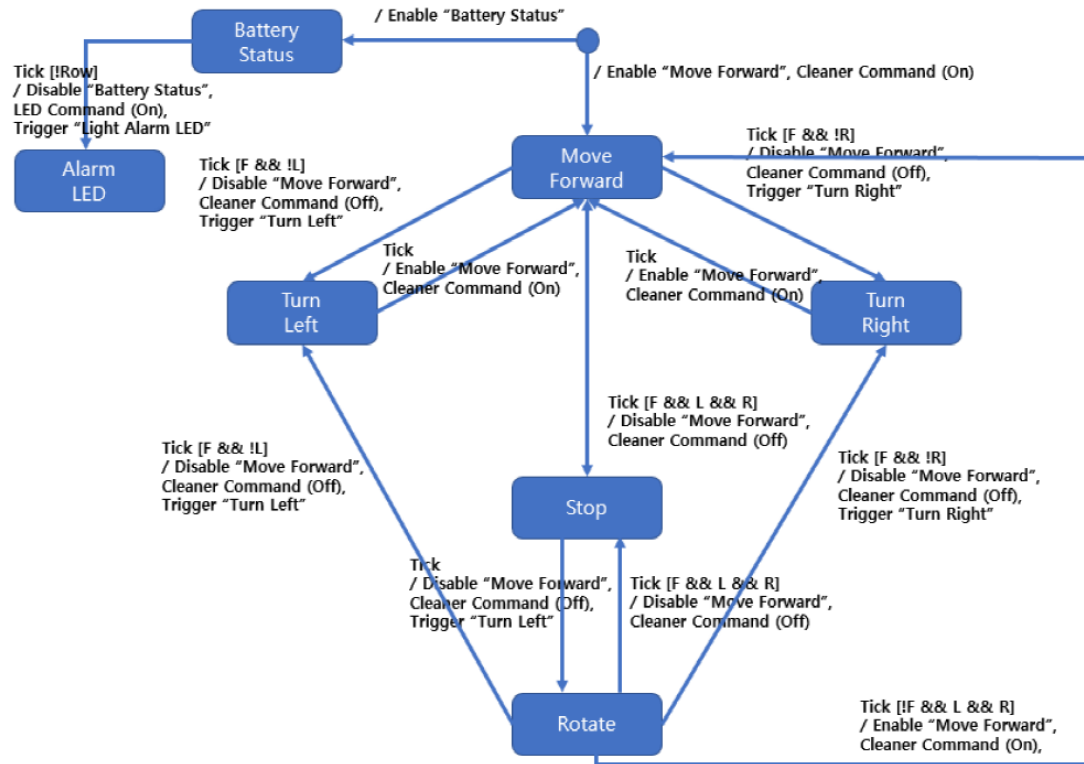
-DFD Level 3



Ex 1

- Initial state 에서 state 전이가 정해지지 않음.
 - “Move Forward” state 혹은 각 state 마다 해당 부분이 필요

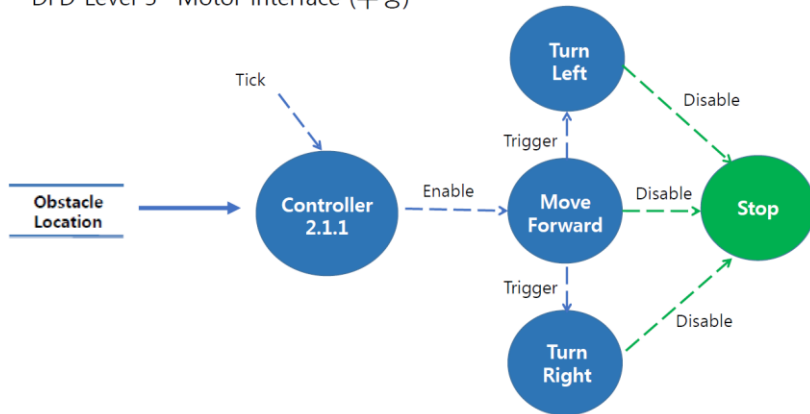
-DFD Level 4



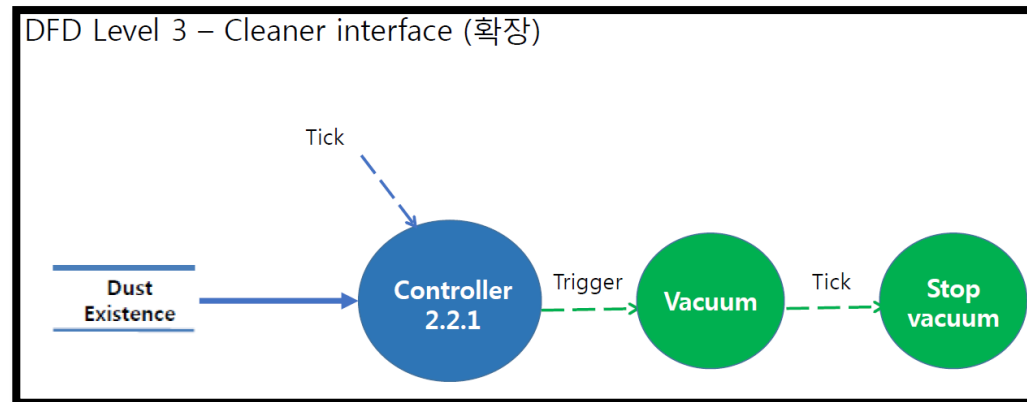
EX 2

- Control Flow 는 Control Process 에서만 발생할 수 있습니다.
- Tick Event 는 vacuum이 아닌 digital clock에서 보내는 control flow 입니다.
- DFD만 있고 설명이 없어서 이해를 제가 못해서 따로 피드백을 못..

DFD Level 3 –Motor interface (수정)



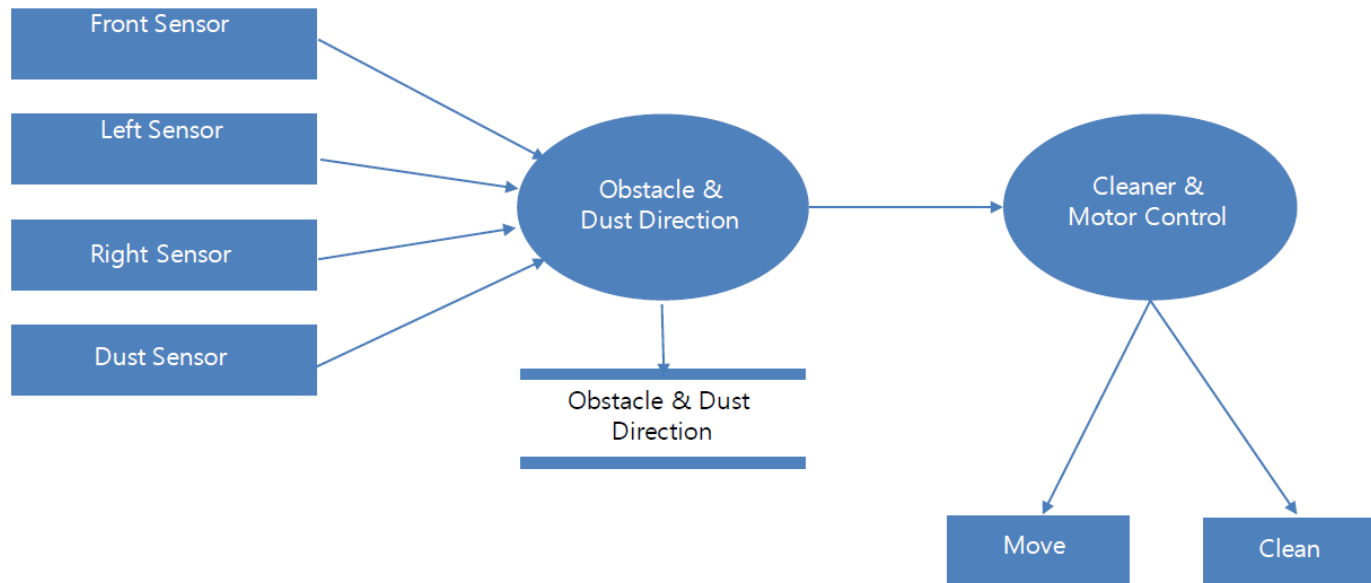
DFD Level 3 – Cleaner interface (확장)



Ex 3

- Data store는 process가 사용하기 때문에 input과 output이 필요

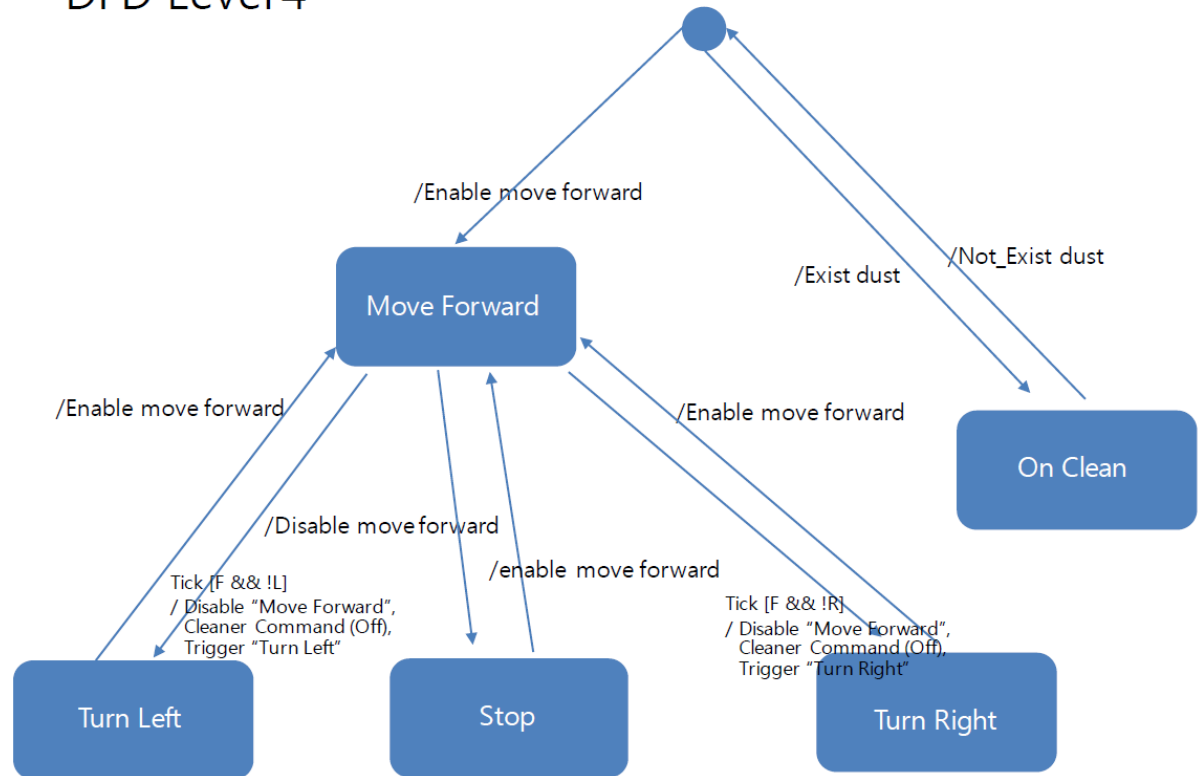
DFD Level 1



Ex 3

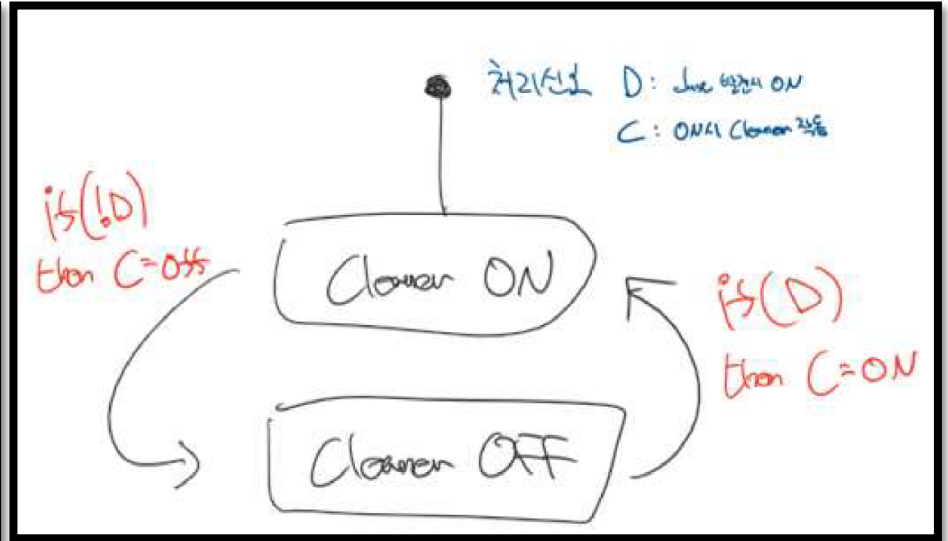
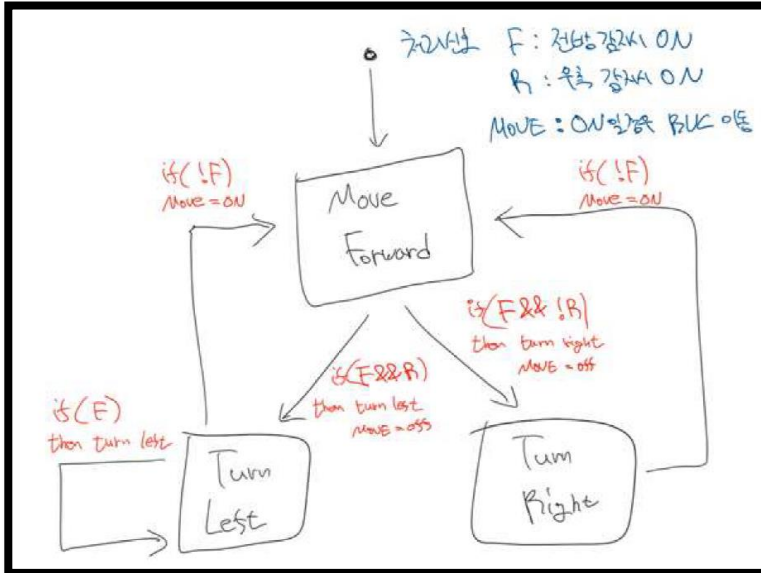
- Move Forward에서 stop으로 바로 가게 됨.
 - 예시로 드린 답안을 참고.

DFD Level 4

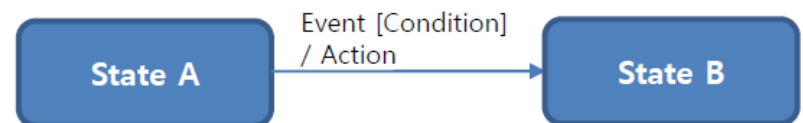


Ex 4

- 제시한 해결방법: **controller**를 두개로 만들자.

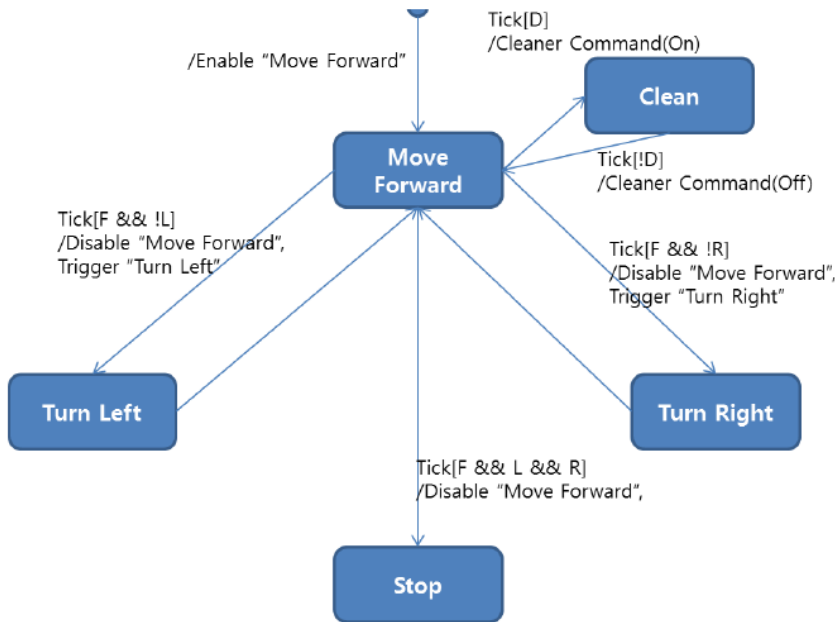


- DFD에 controller 두개에 대한 표현이 없음.
- State Transition Diagram에서 사용되는 notation 이해 필요.



Ex 5

- 제시한 Solution에서 나온 move backward 표현 혹은 state transit이 없다.
 - Ex 1의 "Rotate" state (p. 8) 참고하여 작성가능



제시한 Solution

- 1) Tick조건에 Dust를 추가한다.
 - Cleaner Command를 앞으로 갈 때 작동시키긴 하나 따로 clean 이벤트를 만들어서 Dust가 있을 때 작동하게 한다.
 - Dust를 고려하여 Dust가 있으면 앞에 장애물이 있는 것으로 취급한다. 그러면 공백 없이 청소하게 된다.
- 2) Move Backward를 추가한다.
 - F && L && R이였을 경우 !L이나 !R이 될 때까지 move 한 후 진행한다. 만약 !L && !R일 경우 왼쪽으로 들어서 진행한다고 가정한다.

Ex 6

- 제시한 해결방법

문제점 해결방법

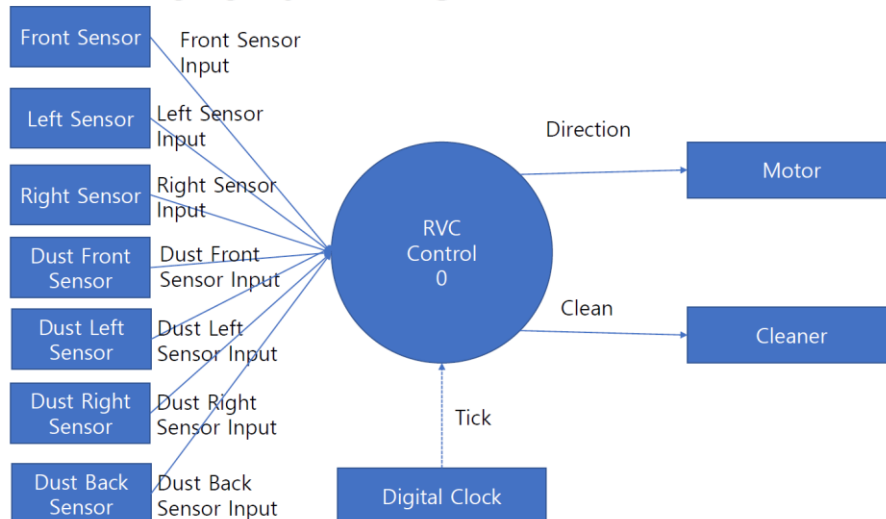
- 전방하고 좌우가 막혀있는 경우 Stop 상태가 되면서 멈춘다.
 - 한방향으로 계속 전방에 장애물이 없을때까지 돌면 된다.
- 먼지를 가장자리만 청소를 하게 된다.
 - 달팽이모양으로 안쪽으로 돌아가면서 청소하게 만든다.
- 청소를 다 한 후에도 계속 작동한다.
 - 자기 주위에 먼지가 없을 경우 작동을 끈다.

Ex 6

Event List - RVC

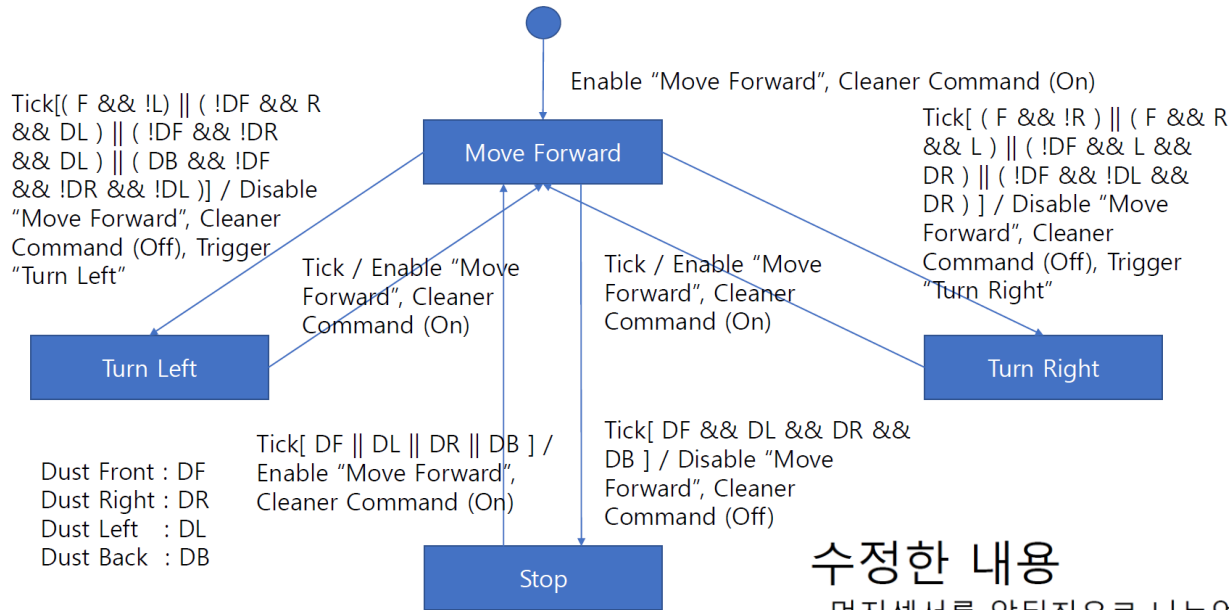
Input/ Output Event	Description
Front Sensor Input	Detects obstacles in front of the RVC
Left Sensor Input	Detects obstacles in the left side of the RVC periodically
Right Sensor Input	Detects obstacles in the right side of the RVC periodically
Dust Front Sensor Input	RVC periodically detects dusts on the front floor
Dust Left Sensor Input	RVC periodically detects dusts on the left floor
Dust Right Sensor Input	RVC periodically detects dusts on the right floor
Dust Back Sensor Input	RVC periodically detects dusts on the back floor
Direction	Direction commands to the motor (go forward / turn left with an angle / turn right with an angle)
Clean	Turn off / Turn on / Power-Up

DFD Level 0 - RVC



Ex 6

DFD Level4 - RVC



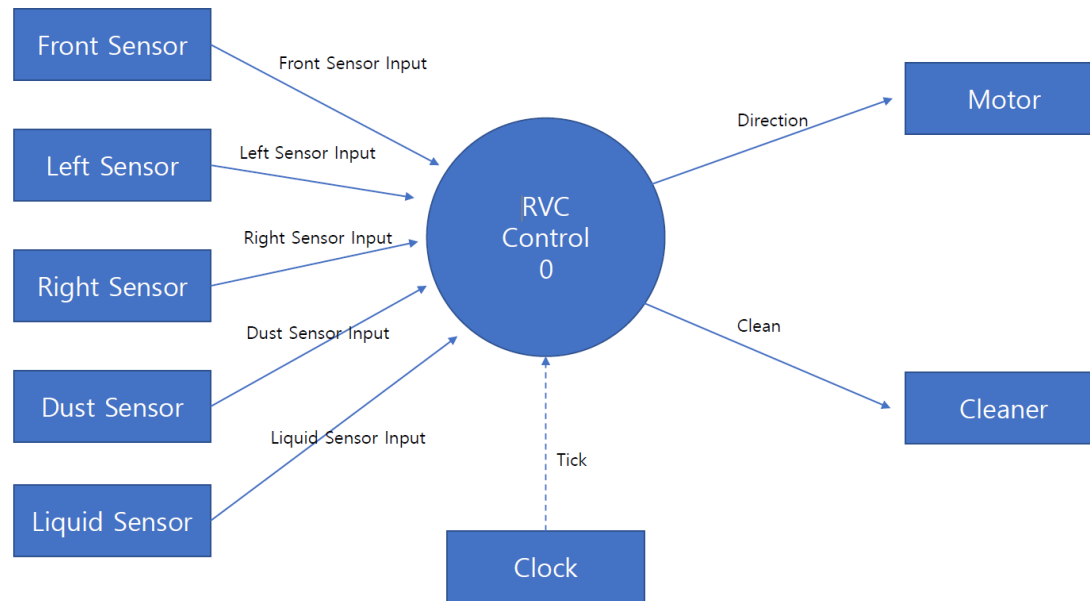
수정한 내용

- 먼지센서를 앞뒤좌우로 나누었다.
- Stop하는 경우는 앞뒤좌우 먼지가 없을시 멈추게 하였다.
- Stop하고 있을때 앞뒤좌우 먼지가 발견하면 움직이게 하였다.
- 왼쪽으로 도는 경우는 앞쪽이 깨끗하고 오른쪽이 막히고 왼쪽에 먼지가 있으면 왼쪽으로 돌고 또는 앞쪽이 깨끗하고 오른쪽도 깨끗하고 왼쪽에 먼지가 있으면 왼쪽으로 돌고 또한 앞좌우가 먼지가 없고 뒤쪽에 있을경우 왼쪽으로 계속 돌게 하였다.
- 오른쪽으로 도는 경우는 앞쪽이 깨끗하고 왼쪽이 막히고 오른쪽에 먼지가 있으면 오른쪽으로 돌고 또는 앞쪽이 깨끗하고 왼쪽도 깨끗하고 오른쪽에 먼지가 있으면 오른쪽으로 돌고 또한 앞좌우가 막혀있으면 오른쪽으로 계속 돌게 하였다.

Ex 7

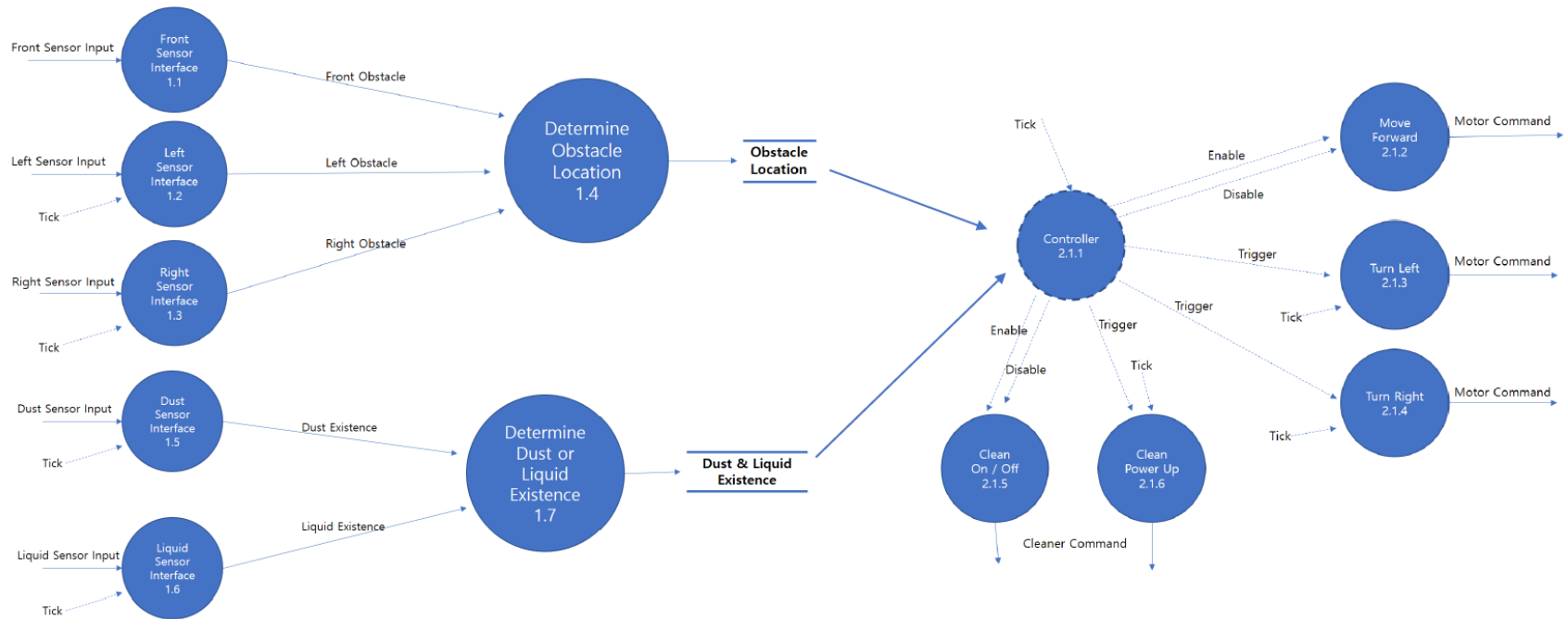
- 액체를 감지해서 어떠한 조치를 취하는 RVC

DFD Level 0 – RVC Example

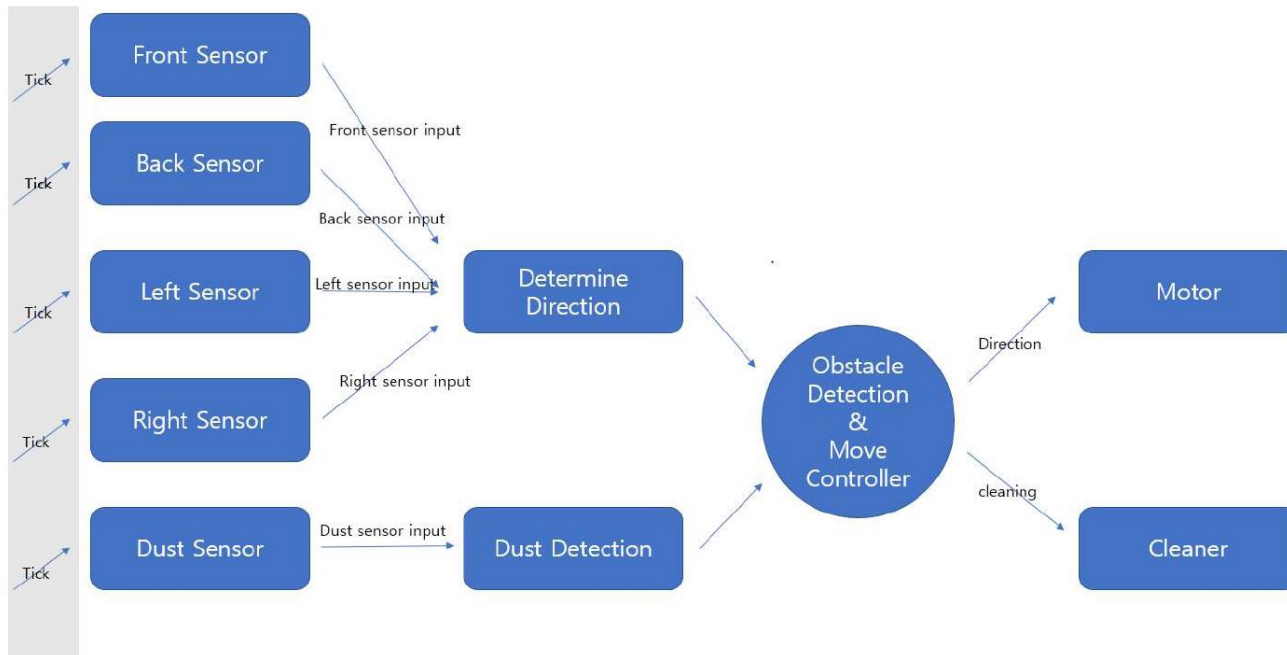


Ex 7

DFD – RVC Example

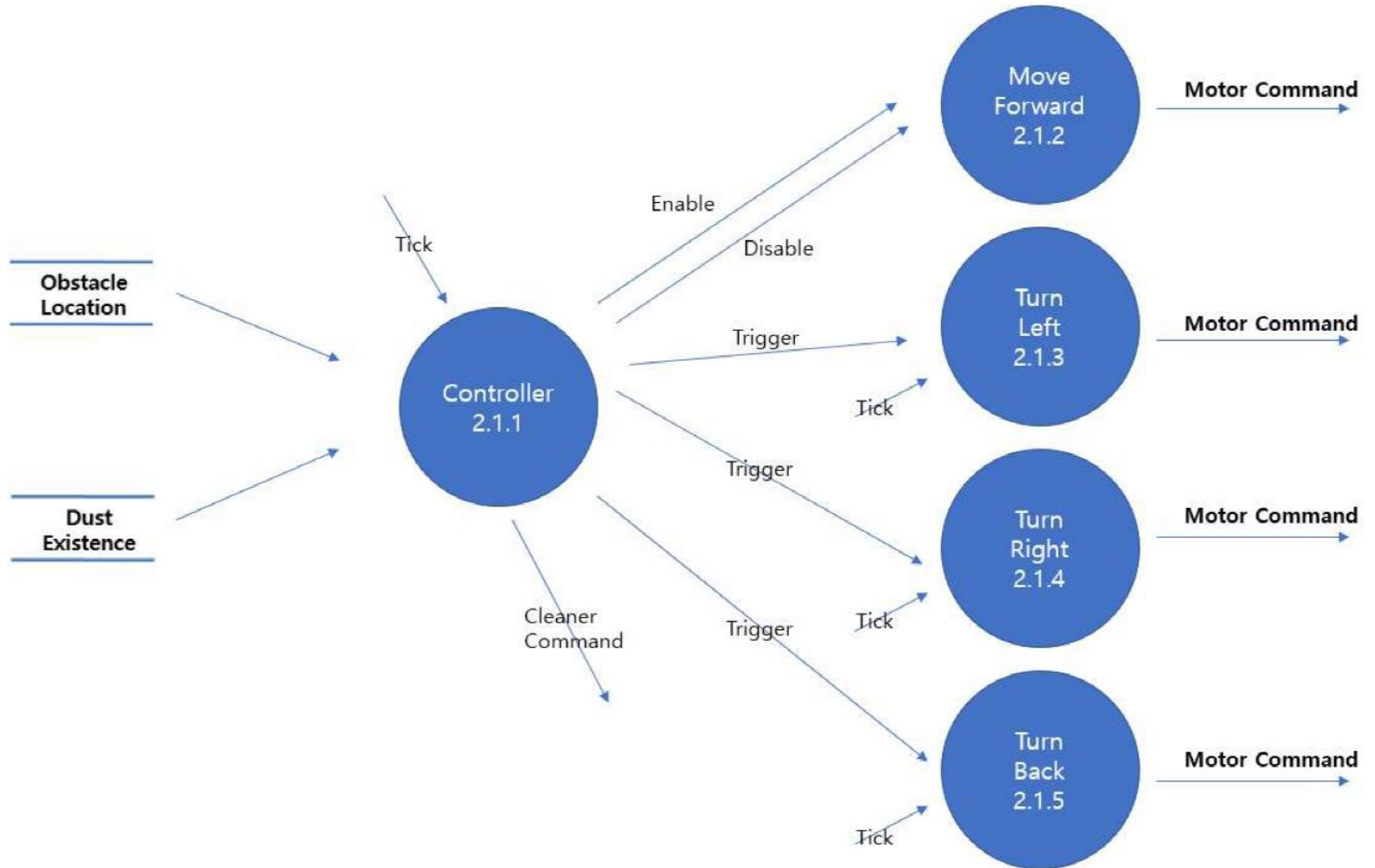


Ex 8

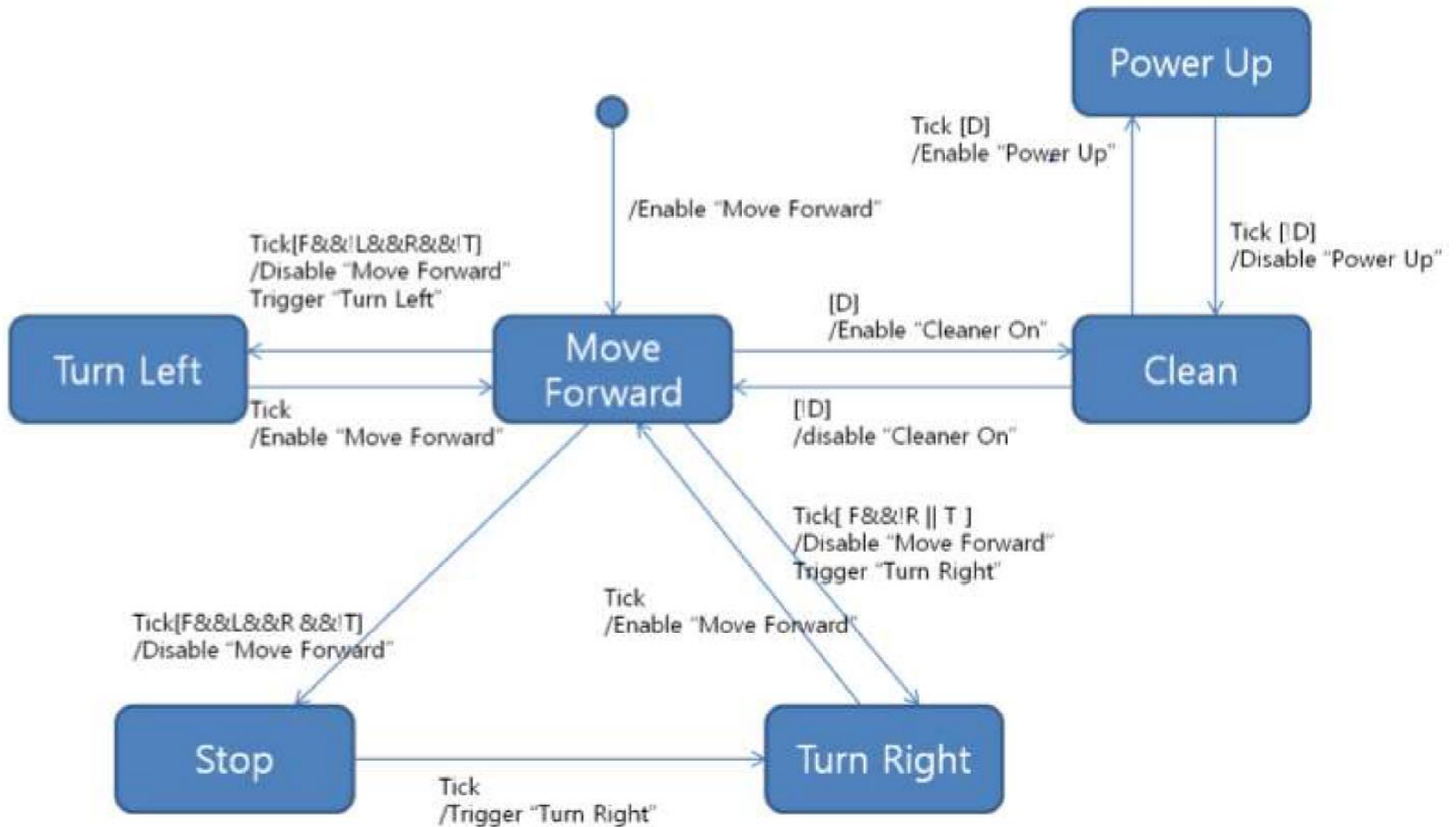


<Dust Sensor를 추가한 DFD LEVEL 0>

Ex 8



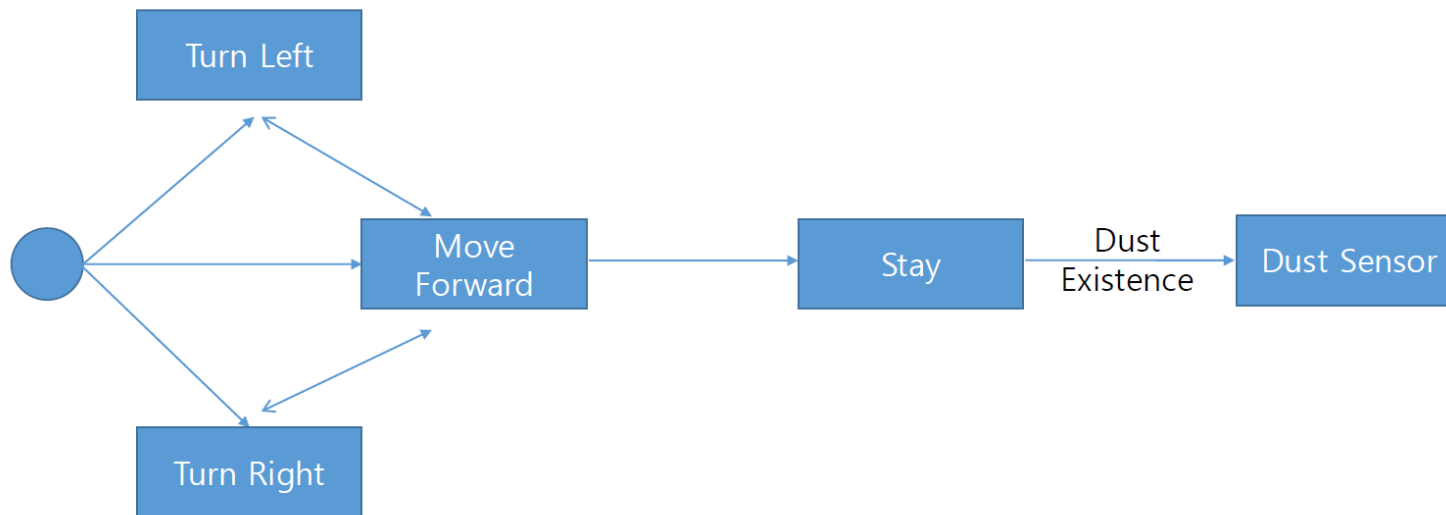
Ex 8



Ex 9

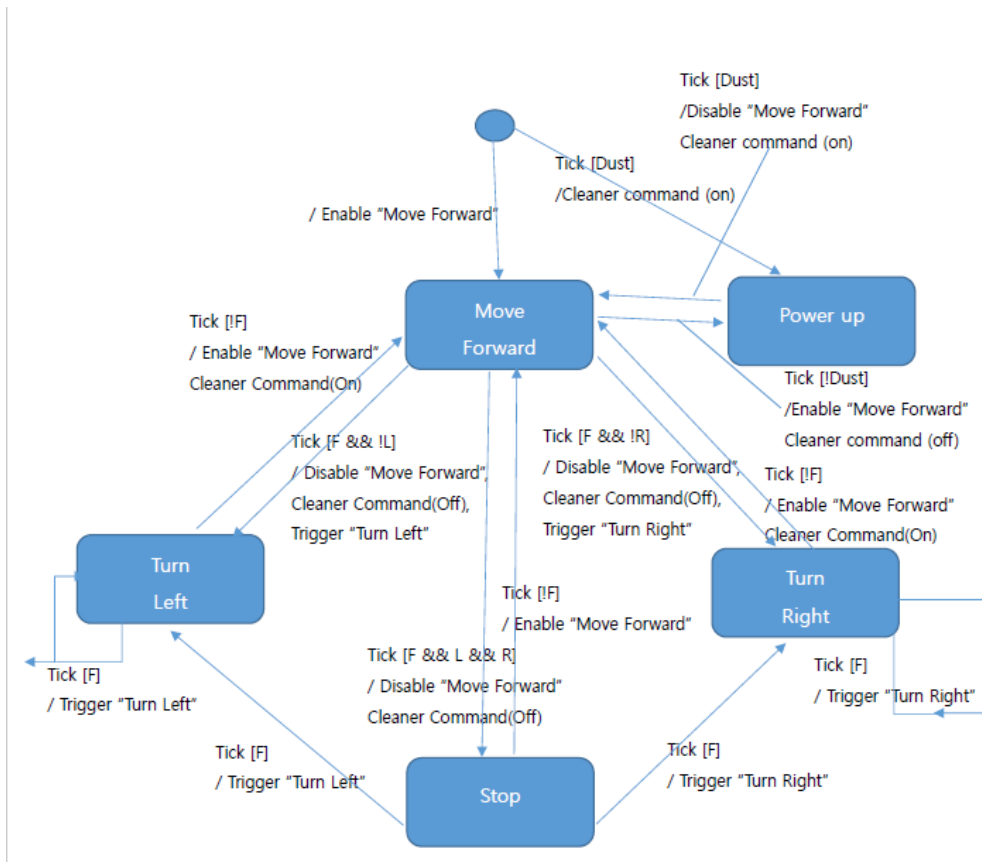
- Control flow는 control process 에서 발생
- Notation 꼭 지켜주세요.

DFD Level 4 – RVC



Ex 10

- 초기 상황에 먼지가 있으면 바로 Power-up 으로 가서 먼지를 닦고 시작.



Ex 11

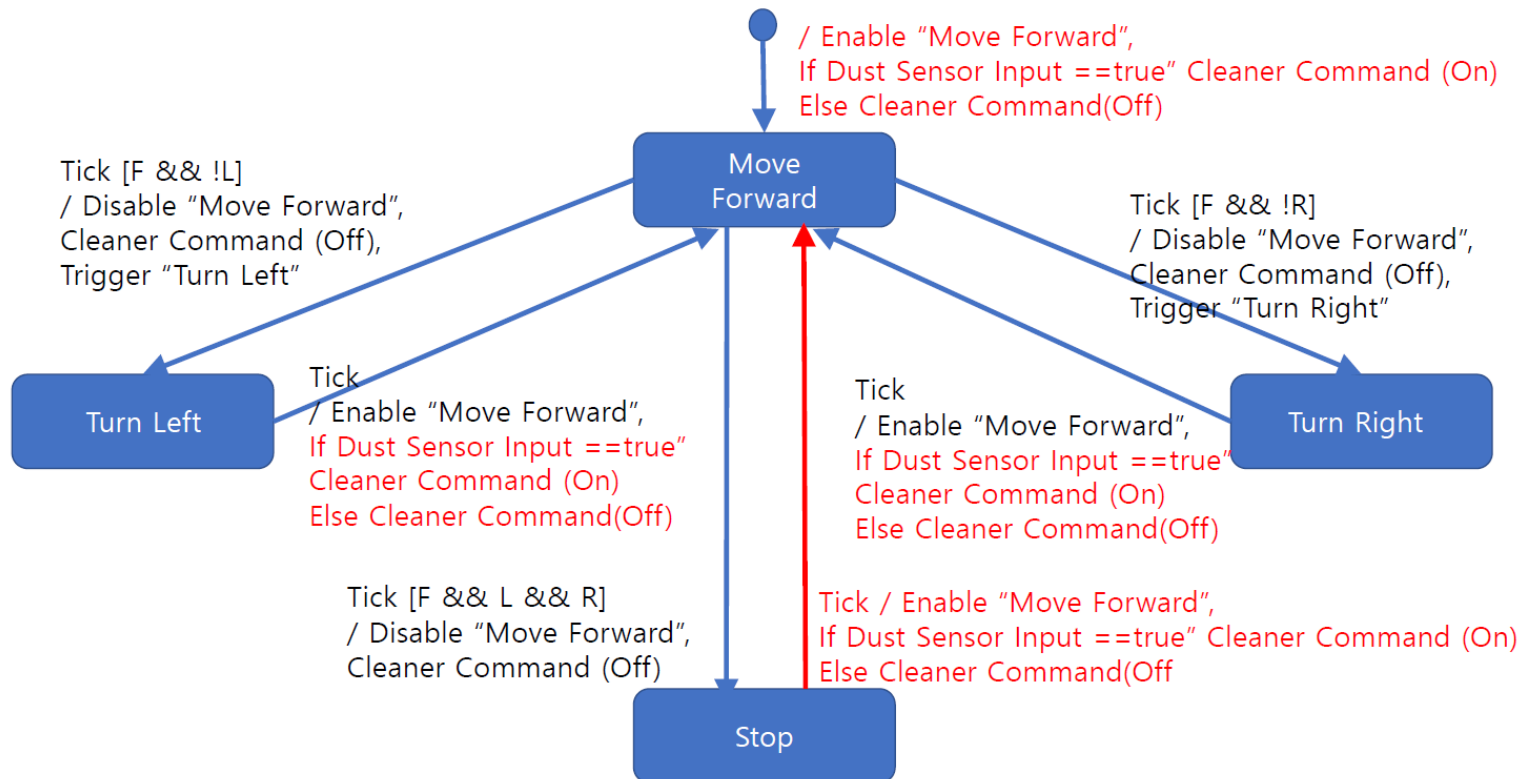
- 잘못 설명하여 틀린 내용이 전달 되었네요.
- 절차지향적인 언어를 대상으로 한다는 말은 SASD 수행 대상과는 다른 내용입니다.
- SASD는 시스템 혹은 소프트웨어를 대상으로 수행하는 개발 방법론입니다.
- 실습 과정에서 cyclic 한 시스템을 절차지향적 언어인 C로 구현하는 과정을 가지는데 설명을 부족하게 하여 틀린 내용을 알려드린 것 같습니다.

이해한 내용 정리

- 소프트웨어의 유형: 주문형 / 패키지형 / 임베디드(하드웨어 장치에서 수행되는 소프트웨어로, 이 수업과 관련)
- SASD(Structured Analysis and Structured Design) : 구조적 분석 + 구조적 디자인 **절차지향적인 언어를 대상으로 함**
- SASD Process의 과정으로 모델링이 이루어짐

Ex 11

- 수업시간에 보여드린 실습 답안 참고해주세요.
- State Transition Diagram의 notation 확인!

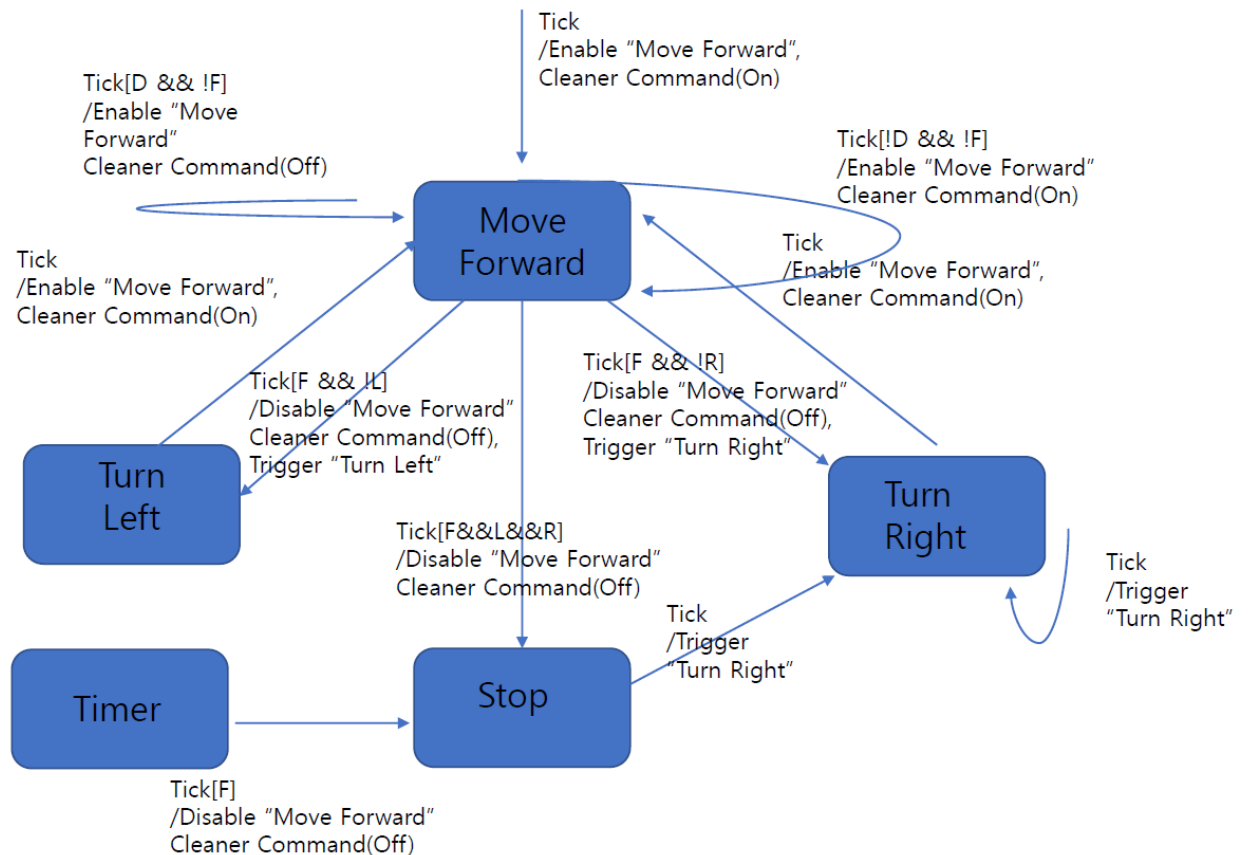


Ex 12

- 제시한 방법
 - 1. Stop
 - Turn Right
 - 2. Dust
 - 먼지 보이면 cleaner off
 - 추가 기능. Timer, battery
 - Timer 기능 표현.

Ex 12

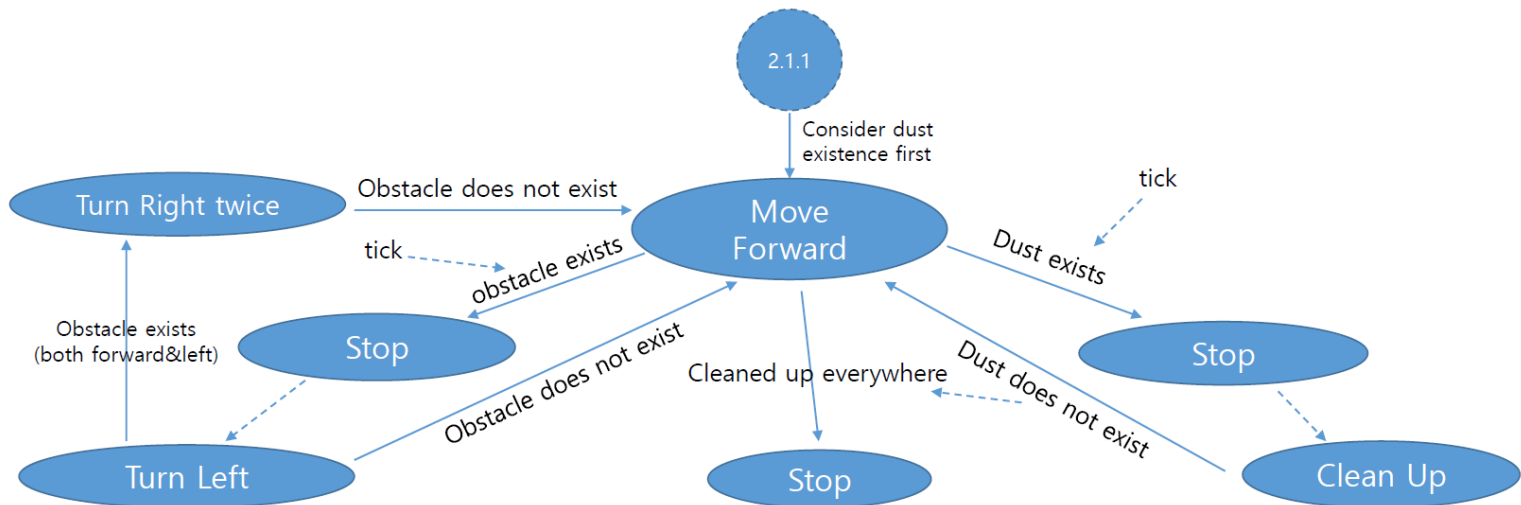
- DFD가 표현되었으면 더 좋았을 것.
- “Timer” State 로 진입하지 않음.



Ex 13

- Uh.. Notation!

RVC example DFD 4 level



우선 컨트롤러에서 움직임을 결정하기 이전에 dust가 존재하는지를 탐지한다. 만약 dust가 존재한다면, move forward를 disable(stop)하고 우선 청소를 한 후 더 이상 dust가 detect되지 않으면 move forward를 enable한다. 이 때 obstacle이 존재할 경우 우선 멈춘 후, 왼쪽으로 먼저 turn한 후에, obstacle의 존재 유무에 따라 존재한다면 turn right을 두 번 실행해 기존 진행 방향의 우측으로 turn하여 앞으로 이동하며 다시 앞의 process(dust detection & clean up)를 반복하고, 존재하지 않을 경우에는 왼쪽으로 turn한 상태에서 다시 앞의 process를 반복한다.

Ex 13

- Overall DFD에 맞지않는 level 표현
 - State로 보이는 표현도 보임.
 - Control과 data flow 구분 필요.

RVC example DFD

