

# Software Requirement Analysis for Public Transportation System

Project Team

**Team 3**

Date

**2014-09-21**

---

Team Information

김제헌

박상희

박형민

엄현식

## Table of Contents

1	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.2.1	개발사항	5
1.2.2	제한사항	5
1.2.3	제품의 활용도	5
1.2.4	개발환경	5
1.3	Definition, acronyms, and abbreviations	5
1.4	Reference	6
1.5	Overview	6
2	Overall Description	6
2.1	Product Perspective	6
2.2	Product functions	6
2.3	User characteristics	6
2.4	제약 및 가정 사항	7
3	Structured Analysis	7
3.1	System Context Diagram	7
3.1.1	Basic System Context Diagram	7
3.1.2	Event List	8
3.1.3	The System Context Diagram	8
3.2	Data Flow Diagram	8
3.2.1	DFD level 0	8
3.2.1.1	DFD	8
3.2.1.2	Process Specification	8
3.2.1.2.1	Process 0	9
[텍스트 입력]		2
Team 3		2

3.2.1.3	Data Dictionary	9
3.2.2	DFD Level 1	9
3.2.2.1	DFD	9
3.2.2.2	Process Specification	9
3.2.2.2.1	Process 1	9
3.2.2.2.2	Process 2	10
3.2.2.2.3	Process 3	10
3.2.2.2.4	Process 4	10
3.2.3	DFD level 2	10
3.2.3.1	DFD	10
3.2.3.2	Process Specification	11
3.2.3.2.1	Process 2.1	11
3.2.3.2.2	Process 2.2	11
3.2.3.2.3	Process 2.3	11
3.2.3.2.4	Process 4.1	11
3.2.4	DFD level 3	12
3.2.4.1	DFD	12
3.2.4.2	Process Specification	12
3.2.4.2.1	Process 2.1.1	12
3.2.4.2.2	Process 2.1.2	12
3.2.4.2.3	Process 2.2.1	12
3.2.4.2.4	Process 2.2.2	13
3.2.4.2.5	Process 2.3	13
3.2.4.2.6	Process 3	13
3.2.4.3	Data Dictionary	13
3.2.4.4	State Transition Diagram	14

3.2.4.4.1	STD for Controller 2.1.1 _____	14
3.2.4.4.2	STD for Controller 2.2.3 _____	15
3.2.4.4.3	STD for Controller 2.3 _____	15
3.2.4.4.4	STD for Controller 4 _____	16
3.2.5	Overall DFD _____	16

## 1 Introduction

### 1.1 Purpose

본 문서는 Public Transportation System에 관한 요구사항 명세를 기반으로, Structured Analysis를 수행한 문서이다. 이 문서를 통해 PTS의 필요한 요구사항들의 관계를 명확히 하고, 더 나아가 이것은 실제 구현의 토대가 된다.

이 문서를 읽는 사람에게 PTS의 Analysis가 어떻게 이루어졌는지 명확히 전달한다.

### 1.2 Scope

#### 1.2.1 개발사항

본 프로젝트는 전체 PTS 중 지하철, 버스 및 정산 시스템만을 대상으로 구현하는 것으로 규모를 제한한다. 또한 버스1대와 지하철 2호선 중 5개 역(건대입구, 왕십리, 합정, 신림, 강남)만을 대상으로 한다. 모든 시스템은 SW만으로 구현한다. HW가 필요한 부분은 SW모듈을 만들어 가상의HW를 구현한다.

#### 1.2.2 제한사항

HW(단말기)와 연동을 고려하지 않고, SW로만 구동할 수 있도록 한다.

#### 1.2.3 제품의 활용도

개발이 완료된 후 실제 지하철, 버스 SW(단말기)를 개발하기 위한 프로토타입으로 삼을 수 있다.

#### 1.2.4 개발환경

IDE : Visual Studio C++

Compiler : GCC

### 1.3 Definition, acronyms, and abbreviations

SW : Software

HW : Hardware

PTS : Public Transportation System

CID : Card ID

CR : Card Reader (역 단말기)

태그: 카드와 단말기가 통신할 수 있도록 하는 행위; 승·하차 시 요금 결제를 위한 행위

#### 1.4 Reference

#### 1.5 Overview

### 2 Overall Description

#### 2.1 Product Perspective

SW로 개발된 PTS는 총 3가지로 구성된다. 버스용 단말기와 지하철용 단말기, 정산 시스템이다

대상 제품은 실제 지하철, 버스 단말기에 사용되는 제품이 될 수 있다. HW(단말기)에서 교통카드 감지 센서를 통해 찍힌 교통카드 정보를 읽어온 뒤, HW(화면)에 출력한다. 실제 HW에 의한 동작은 SW 및 console화면으로 처리하여 기능의 동작 유무를 확인하도록 한다.

#### 2.2 Product functions

버스용 단말기는 버스에 부착돼, 탑승 태그와 하차 태그가 가능하다. 기본료를 지불하면 1회 탑승이 가능하다.

지하철용 단말기는 역에 부착돼 탑승 태그와 하차 태그가 가능하다. 1개역 이하를 이동하면 기본료가 부가되며 두 개역을 이동하면 추가 요금이 부가된다.

버스와 지하철 간에는 정해진 시간 내에 환승이 가능하다. 지하철에서 버스로 환승한 경우는 단위 시간 당 버스의 환승 요금이 추가로 부가된다. 버스에서 지하철로 환승한 경우는 한 역당 지하철의 환승 요금 추가로 부가된다. 하차 시 단말기에 태그를 하지 않으면 환승은 적용되지 않는다.

정산은 하루에 한 번 이뤄진다. 버스과 지하철의 기록을 분석하고 버스과 지하철에 각각 수익을 배분한다.

#### 2.3 User characteristics

사용자는 대중교통 승차 시 카드를 태그한다.

사용자는 하차 시 카드를 태그할 수도 있고 아닐 수도 있다

## 2.4 제약 및 가정 사항

버스 승차는 지역에 상관 없이 할 수 있다.

지하철은 2호선 역 중 5개만 고려한다: 건대입구, 동대문역사문화공원, 합정, 신림, 강남.

하루는 3분으로 가정한다.

다음 날 운행이 시작하기 전까지는 정산이 반드시 이루어 져야 한다.

정산후 모든 프로그램의 정보는 초기화 된다. (사용자 카드 정보 제외)

환승이 가능한 시간은 15초 이내다.

버스 환승 시 추가요금의 기준인 단위 시간은 30초이다.

버스와 지하철의 기본료는 1050원이다.

교통카드는 저장된 텍스트 파일로 가정하고, 교통카드 태그 행위를 해당 파일을 입력 하는 것으로 가정한다.

잔액이 모자를 경우 태우지 않는다.

버스 환승 최고 부과금액인 700원이 남아 있지 않으면 버스로 환승시키지 않는다.

지하철 환승 최고 부과금액인 600원이 남아 있지 않으면 지하철로 환승시키지 않는다.

하루(3분) 종료 시 탑승되어 있는 승객은 미정산으로 처리한다.

정산시 소수점 이하는 반올림한다.

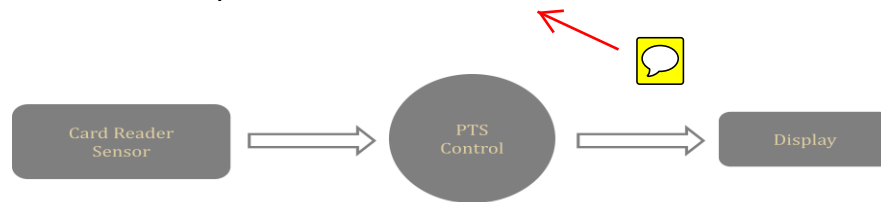
## 3 Structured Analysis

### 3.1 System Context Diagram

#### 3.1.1 Basic System Context Diagram

## System context diagram

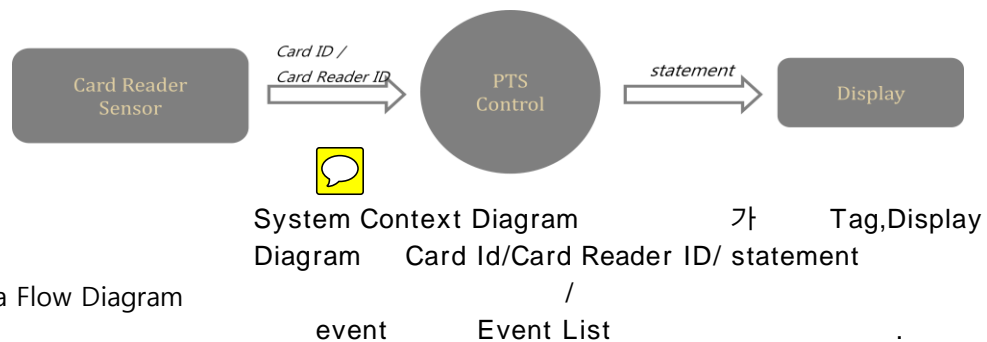
Ver. Identifier



## 3.1.2 Event List

Input / Output Event	Description
<b>Tag</b>	카드 태그가 이루어질 경우 Date Flow가 발생된다.
<b>Display</b>	정상적인 처리가 이루어질 경우, 역 단말기에 해당 카드의 잔액, 사용금액이 출력된다. 그 외의 경우 경고문을 출력한다.

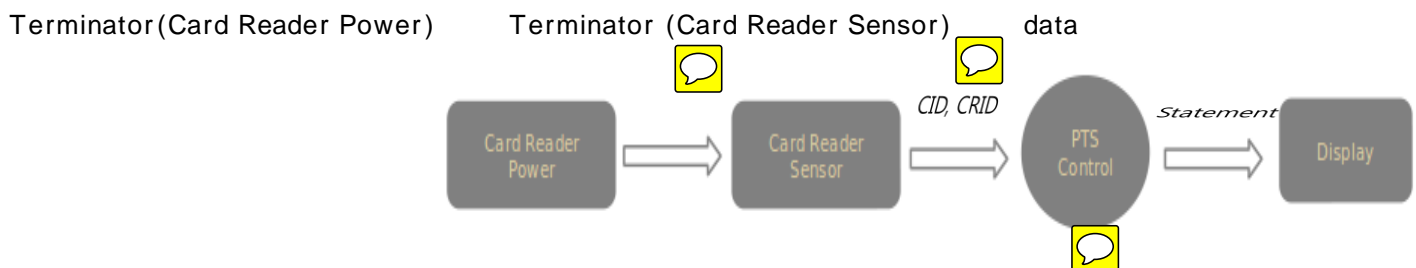
## 3.1.3 The System Context Diagram



## 3.2 Data Flow Diagram

## 3.2.1 DFD level 0

## 3.2.1.1 DFD



## 3.2.1.2 Process Specification



## 3.2.1.2.1 Process 0

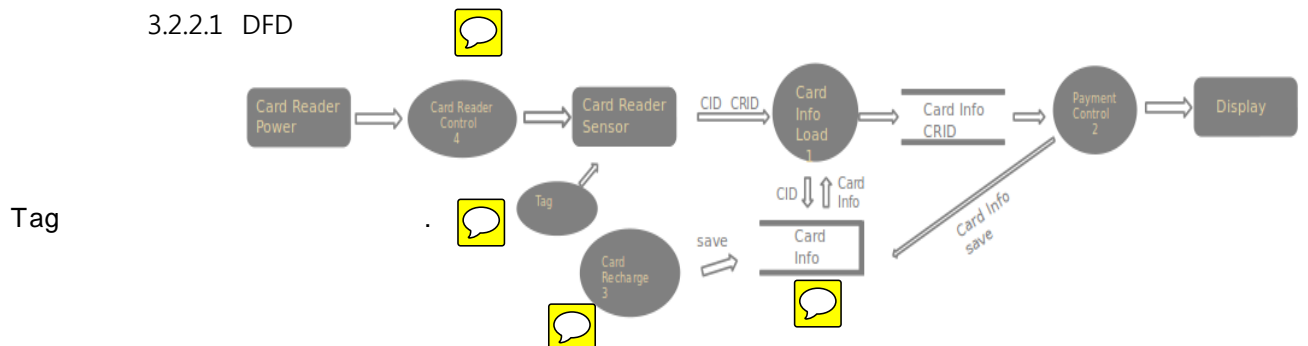
Reference No.	0
Name	PTS Control 0
Input	CID, CRID
Output	Statement
Process Description	CID, CRID를 받아서 요금 계산을 한 뒤, 적절한 statement를 display해준다.

## 3.2.1.3 Data Dictionary

Data Name	Explanation
CID	Tag된 Card의 ID를 INT 형태로 전달한다.
CRID	Tag된 Card Reader의 ID를 INT 형태로 전달한다
Statement	PTS Control에서 계산된 요금 또는 카드의 상태에 따라 적절한 Statement를 Display로 보내준다.

## 3.2.2 DFD Level 1      DFD      Terminator      ..

## 3.2.2.1 DFD




## 3.2.2.2 Process Specification

## 3.2.2.2.1 Process 1

Reference No.	1
Name	Card Info Load DFD      Card info      input
Input	CID, CRID
Output	Card Info(CID,recent(tp, trans_state, cash, CRID,transfer)),CRID
Process Description	CID를 사용해서 카드의 정보를 불러와 Payment Control에 전달한다.



## 3.2.2.2.2 Process 2

Reference No.	2
Name	Payment Control
Input	Card Info, CRID
Output	Card Info, Display 
Process Description	카드 정보를 이용하여 승/하차 시 발생한 요금을 계산하고, 카드 파일에 새로운 값을 저장한다. 그리고 결과(부과 요금, 현재 요금, 현재 시간)를 출력한다.

## 3.2.2.2.3 Process 3

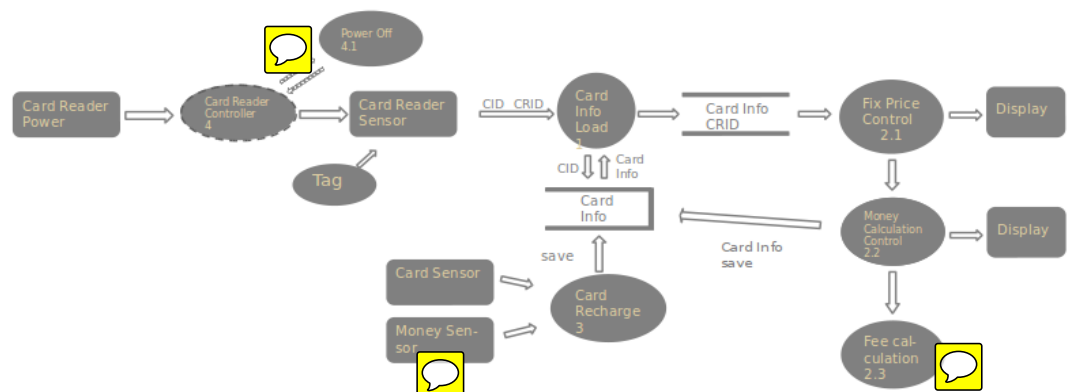
Reference No.	3
Name	Card Recharge Control
Input	CID DFD
Output	Recent_cash => DFD
Process Description	CID를 통해서 카드의 현재 금액에 충전 요금을 더한다.

## 3.2.2.2.4 Process 4

Reference No.	4
Name	Card Reader Control
Input	Runnable DFD
Output	 output ?....
Process Description	카드 단말기가 돌아갈 수 있게 한다. 


## 3.2.3 DFD level 2

## 3.2.3.1 DFD



## 3.2.3.2 Process Specification

## 3.2.3.2.1 Process 2.1

Reference No.	2.1
Name	Fix Price DFD
Input	Card Info, CRID DFD
Output	Card Info, CRID, price DFD
Process Description	Fix Price를 통해 승/하차 시 필요한 금액을 책정하여 Money Calculation에 전달한다. 


## 3.2.3.2.2 Process 2.2

Reference No.	2.2
Name	Money Calculation DFD
Input	Card Info, CRID, price DFD
Output	Card Info, Display, fee DFD
Process Description	Price를 받아서 현재 잔액에서 뺀 다음 카드 파일에 새로운 값을 저장하고 결과(부과 요금, 현재 요금, 현재 시간)을 출력한다.

## 3.2.3.2.3 Process 2.3

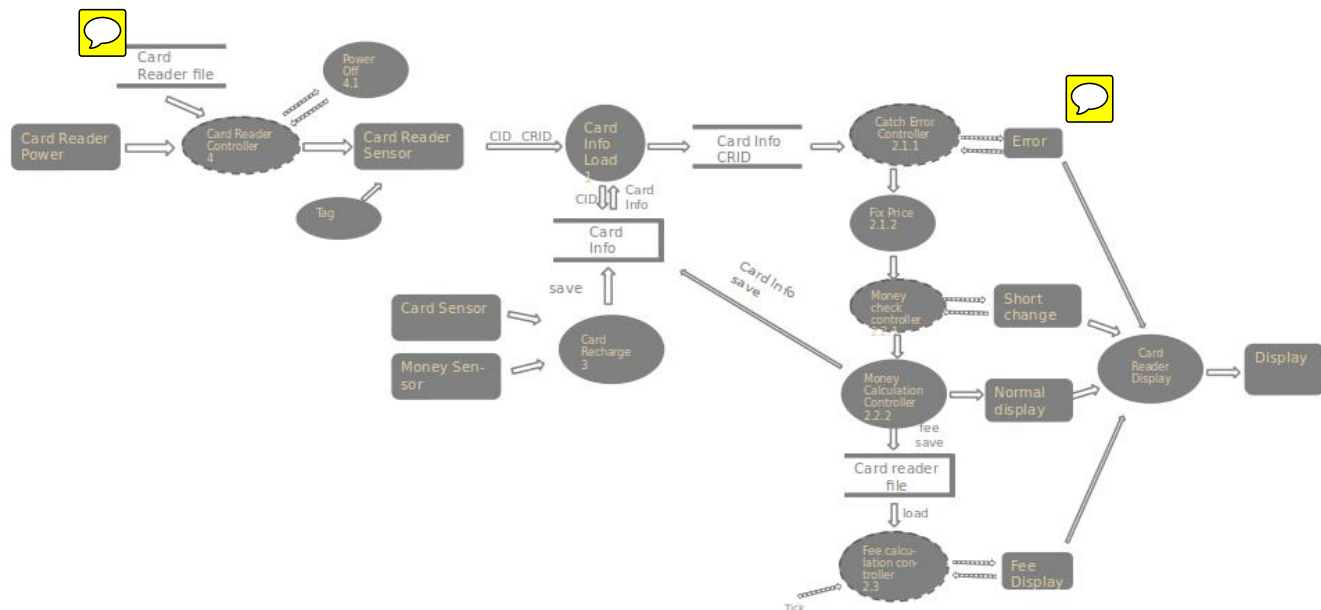
Reference No.	2.3
Name	Fee calculation
Input	fee DFD
Output	Total fee DFD
Process Description	Money Calculation에서 만들어진 부과 요금을 모두 더한다.

## 3.2.3.2.4 Process 4.1

Reference No.	4.1
Name	Power Off
Input	Runnable Runnable
Output	
Process Description	Runnable이 TRUE이면 power off 실행을 하지 않고, FLASE이면 power off를 실행한다.

### 3.2.4 DFD level 3

#### 3.2.4.1 DFD



#### 3.2.4.2 Process Specification

#### 3.2.4.2.1 Process 2.1.1

Reference No.	2.1.1
Name	Catch Error Controller
Input	Card Info, CRID DFD .
Output	Card Info, CRID, Error, ce
Process Description	에러(승차-승차 && (지-지  버버) && (Time<=15)) 일 경우 메시지를 출력한다.

#### 3.2.4.2.2 Process 2.1.2

Reference No.	2.1.2
Name	Fix Price
Input	Card Info, CRID
Output	Price DFD .
Process Description	카드의 정보를 받아서 부과 요금을 측정하여 Money_Check_Controller에 보낸다.

#### 3.2.4.2.3 Process 2.2.1

Reference No.	2.2.1
Name	Money Check Controller

Input	Card Info, CRID, price DFD
Output	Card Info, Short change DFD
Process Description	Price와 현재 잔액을 비교하여 현재 잔액이 더 적으면 Short change를 실행하고, 더 많으면 Money calculation을 실행한다. short change process

## 3.2.4.2.4 Process 2.2.2

DFD Terminator  
process

Reference No.	2.2.2
Name	Money Calculation
Input	Card Info, CRID, price DFD
Output	Card Info, Display, fee DFD
Process Description	Price를 받아서 현재 잔액에서 뺀 다음 카드 파일에 새로운 값을 저장하고 단말기 파일에 부과 요금을 저장하고, 결과(부과 요금, 현재 요금, 현재 시간)을 출력한다.

## 3.2.4.2.5 Process 2.3

Reference No.	2.3
Name	Fee Calculation Controller
Input	Fee DFD
Output	Total fee DFD
Process Description	카드 단말기 파일에 저장된 부과 요금을 모두 더한다.

## 3.2.4.2.6 Process 3

Reference No.	3
Name	Card Recharge DFD
Input	Card sensor Input, Money sensor Input
Output	Resent_cash DFD
Process Description	Money sensor를 통해 측정 된 금액을 Card sensor를 통해 알게 된 CID의 resent_cash에 측정 된 금액을 더한다.

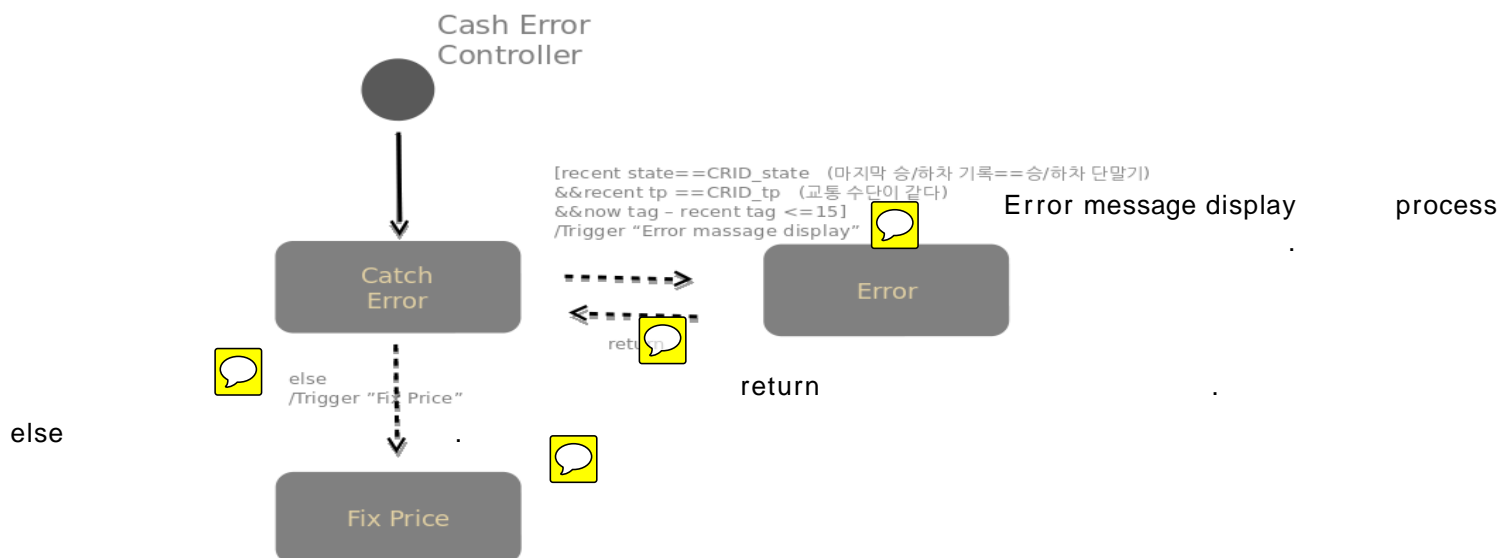
## 3.2.4.3 Data Dictionary

Data	Description	Format/Type
recent_tp	카드 마지막 기록의 교통 수단	TRUE/FALSE
recent_state	카드 마지막 기록의 승/하차 상태	TRUE/FALSE

resent_cash	카드에 남아있는 현재 금액	INT
resent_CRID	카드가 마지막으로 태그 했던 단말기ID	STRING
resent_transfer	카드가 마지막으로 환승으로 승차했는지에 대한 여부	TRUE/FALSE
CRID	단말기의 ID	STRING
resent_tag_time	카드가 마지막으로 태그 했던 시간	INT
now_tag_time	카드가 현재 태그한 시간	INT
price	부과 요금 또는 부과 요금 + 환승 시 필요한 금액	INT
fee	단말기 파일에 저장될 부과 요금	INT
ce	Catch error의 약자로 에러가 났을 경우 FALSE, 에러가 나지 않았을 경우 TRUE를 반환한다.	TRUE/FALSE
Total_fee	단말기 파일에 저장된 부과 요금을 모두 더한 값	INT
Card sensor input	측정 된 카드의 ID(CID)	STRING
Money sensor input	측정 된 금액을 Card Recharge Controller에 전달해준다.	INT
Calculate command	Card reader file에서 Integer 형으로 데이터를 받아서 Calculate command로 데이터를 보내준다.	Integer/Asynchronous
Runnable command	Calculate에서 Calculate command를 수행했는지에 대한 여부를 Boolean 형으로 데이터를 받아 Runnable command로 보내준다.	Boolean/Asynchronous
Total display	Calculate에서 Calculate command를 처리한 데이터를 Tick 마다 Total display에서 출력해준다.	Integer/Asynchronous

### 3.2.4.4 State Transition Diagram

#### 3.2.4.4.1 STD for Controller 2.1.1





Data Name	Description	Format/Type
recent state	카드 마지막 기록에 승차를 했는지 하차를 했는지	TRUE/FALSE
recent tp	카드 마지막 기록에 버스를 탔는지 지하철을 탔는지	TRUE/FALSE
resent tag	카드를 마지막으로 태그한 시간	INT
CRID_state	단말기가 승차인지 하차인지를 구분해서 recent state와 비교한다.	TRUE/FALSE
CRID_tp	단말기가 버스인지 지하철인지 구분해서 recent tp와 비교한다.	TRUE/FALSE
Now tag	방금 태그한 시간과 마지막 태그 시간을 비교한다.	INT

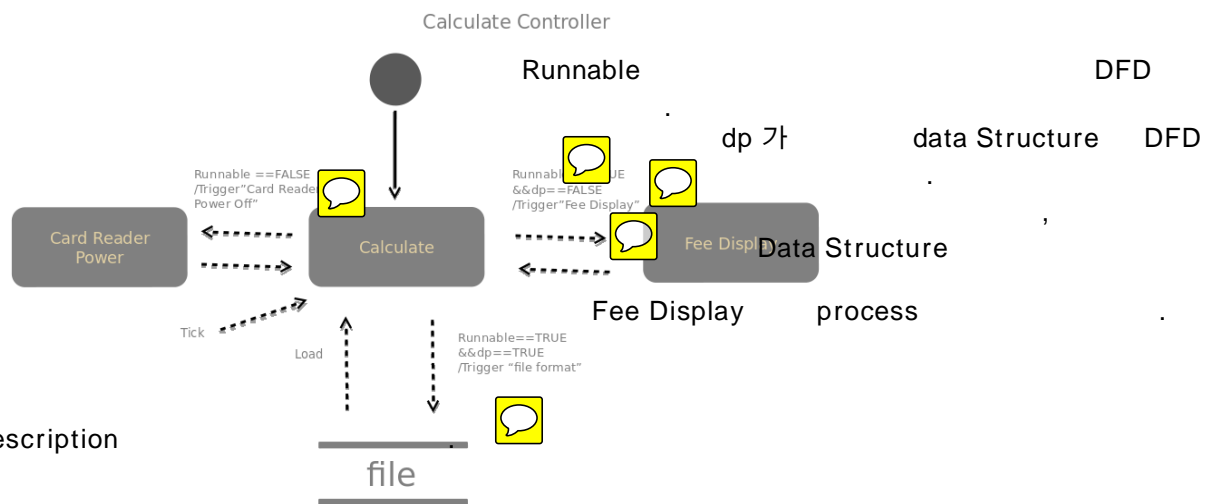
## 3.2.4.4.2 STD for Controller 2.2.



STD

Data Name	Description	Format/Type
Cash	카드에 남아있는 현재 금액	INT
price	(부과 요금) 또는 (부과 요금+최대 요금)으로 환승 시 잔액이 충분한지를 확인한다.	INT

## 3.2.4.4.3 STD for Controller 2.3

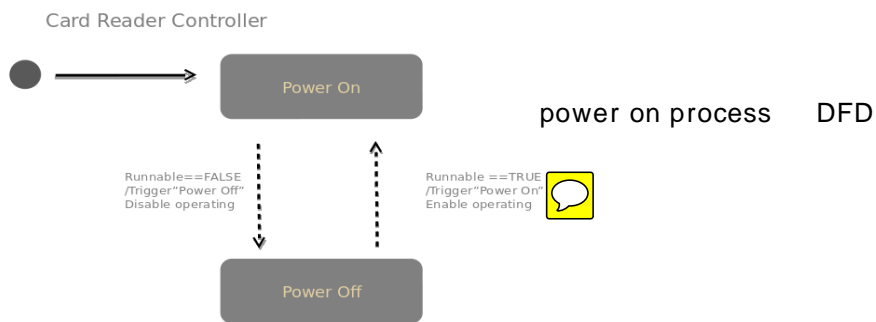


STD description



Data Name	Description	Format/Type
Runnable	단말기 파일의 정산이 잘 됐으면 TRUE, 아니면 FALSE를 넣는다.	TRUE/FALSE
dp	display를 먼저 실행하고 file format하기 위해 만들었다.	TRUE/FALSE
Tick	3분을 주기로 하여 정산을 한다.	

#### 3.2.4.4.4 STD for Controller 4



Data Name	Description	Format/Type
Runnable	TRUE면 Power On하고, FALSE면 Power Off한다.	TRUE/FALSE

### 3.2.5 Overall DFD

