

Practice #2

Software Requirements Specification(SRS) for RVC Controller

컴퓨터 공학과 2학년

201711335 이호용

Table of Contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, acronyms, and abbreviations

1.4 References

1.5 Overview

2. Overall description

2.1 Product perspective

2.1.1 Hardware interface

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

3. Specific requirements

3.1 External interface requirements

3.1.1 Hardware interfaces

3.2 Functional requirements

3.2.1 Mode 1

3.2.1.1 Functional requirement 1.1

3.2.1.2 Functional requirement 1.2

3.2.1.3 Functional requirement 1.3

3.2.1.4 Functional requirement 1.4

3.2.1.5 Functional requirement 1.5

3.2.2 Mode 2

3.2.2.1 Functional requirement 2.1

3.2.2.2 Functional requirement 2.2

3.2.3 Mode

3.2.3.1 Functional requirement 3.1

3.2.3.2 Functional requirement 3.2

3.2.4 Mode 4

3.2.4.1 Functional requirement 4.1

3.2.4.2 Functional requirement 4.2

3.2.4.3 Functional requirement 4.3

3.2.5 Mode 5

3.2.5.1 Functional requirement 5.1

3.2.5.2 Functional requirement 5.2

3.3 Performance requirements

3.4 Design constraints

3.5 Software system attributes

3.6 Other requirements

1 Introduction (소개)

1.1 Purpose (목적)

본 문서는 로봇 진공 청소기(RVC)의 움직임이나 청소 기능을 제어하는 컨트롤러의 제작을 위한 소프트웨어 요구사항 명세서(SRS)이다.

본 문서는 소프트웨어 제작을 요청한 고객을 대상으로 한다.

1.2 Scope (범위)

"RVC Controller"는 센서를 통해 입력 받은 정보를 바탕으로 로봇청소기가 수행할 동작을 명시하는 소프트웨어이다.

RVC Controller는 로봇청소기가 장애물에 가로막히지 않게 하고, 먼지 제거를 위해 청소기능을 스스로 향상시키기도 한다.

1.3 Definitions, acronyms, and abbreviations (용어 및 약어 정의)

RVC : Robot Vacuum Cleaner (로봇 청소기)

Controller : 흐름제어를 담당하는 논리회로

센서 : 로봇 청소기는 센서를 통해 앞, 좌, 우 방향의 장애물과 먼지의 존재 여부를 주기적으로 감지한다.

Tick : 로봇 청소기가 센서를 통해 감지하는 주기

하드웨어 : 센서를 포함해 Controller가 내린 명령을 수행할 단자 (모터, 청소기)

1.4 References (참고자료)

"IEEE Recommended Practice for Software Requirements Specifications", the Institute of Electrical and Electronics Engineers, Inc., 1998.

"Structured Analysis and Structured Design", JUNBEOM YOO, Ver. 2.0.

1.5 Overview (개요)

본 문서는 2장 Overall description에서 RVC Controller의 전반적인 개요를 알아보기 위해서 제품에 영향을 미치는 외부 요인과 소프트웨어가 수행하는 주요 기능의 요약, 사용자 특성, 제약사항으로 나누어 설명한다. 3장 Specific requirements에서는 소프트웨어에 대한 모든 요구사항을 상세하게 포함한다. 이 요구사항들은 이 문서의 작성자 뿐만 아니라 외부적으로 인지할 수 있어야 하며, Finite-State Machine of Controller <그림 1>를 포함하여 가독성을 높였다.

2 Overall description (전체 시스템 개요)

2.1 Product perspective (제품 관점)

Controller 소프트웨어는 외부의 센서들을 통해 데이터를 입력 받고 데이터들을 해석한 결과로 수행할 기능을 하드웨어로 출력시킨다. 다른 시스템이나 소프트웨어와는 독립적이다.

2.1.1 Hardware interface (하드웨어 인터페이스)

센서는 주기적으로 장애물을 감지할 수 있어야 하고 모터는 방향전환이 가능해야 하며 청소기는 끄고 키거나 강약 조절이 가능한 모델이어야 한다.

2.2 Product functions (제품 기능)

Controller는 장애물의 위치와 먼지의 존재 여부를 매 Tick 마다 센서로부터 입력 받은 정보로 갱신해서 상태에 따른 상황 별 기능을 수행하는 명령을 내린다. 센서를 통한 청소의 자동화가 주요 목적이다.

(기능 1) 기본 상태 : 장애물이 감지 되지 않은 상태로 앞으로 움직이며 청소를 수행한다.

(기능 2) 방향 전환 : 장애물이 감지 되어 더 이상 앞으로 움직일 수 없으면 기본 상태에서 수행하던 기능을 멈추고 방향 전환을 수행한 후 다시 기본 상태로 돌아간다.

(기능 3) 파워 업 : 센서에서 먼지를 감지하면 기본 상태를 유지하면서 청소 기능을 잠시 향상 시키고 일정한 Tick이 지나면 다시 기본 상태로 돌아온다.

2.3 User Characteristics (사용자 특성)

사용자는 어떠한 전문 지식이 필요하지는 않고 가전제품의 전원을 끄고 켜본 정도의 경험만 가지고 있어도 된다. 가정의 청소를 자동화 하고싶은 사용자들이 대다수 일 것으로 예측한다.

2.4 Constraints (제약 사항)

ㄱ. 입력에 대한 Controller의 연산 수행이 Tick/2 안에 완료되어야 한다.

ㄴ. 모터와 청소기의 기능은 병렬적으로 수행되어야 한다.

2.5 Assumptions and dependencies (가정 및 의존성)

로봇청소기의 전원은 켜져 있다고 가정한다.

로봇청소기는 높은 곳에서 떨어지거나 뒤집어지지 않는다.

3 Specific requirements (상세 요구사항)

3.1 External interface requirements (외부 인터페이스 요구사항)

3.1.1 Hardware interface (하드웨어 인터페이스)

ㄱ. 이름 : 센서를 통한 데이터 입력

ㄴ. 목적 / 내용 : 장애물의 위치나 먼지의 존재여부를 아날로그 데이터로 입력을 받아서 디지털 형태로 변환해 출력

ㄷ. 입력 주체 / 출력 대상 : 센서 / 결정 논리회로

ㄹ. 시간 : 매 Tick 마다 입력

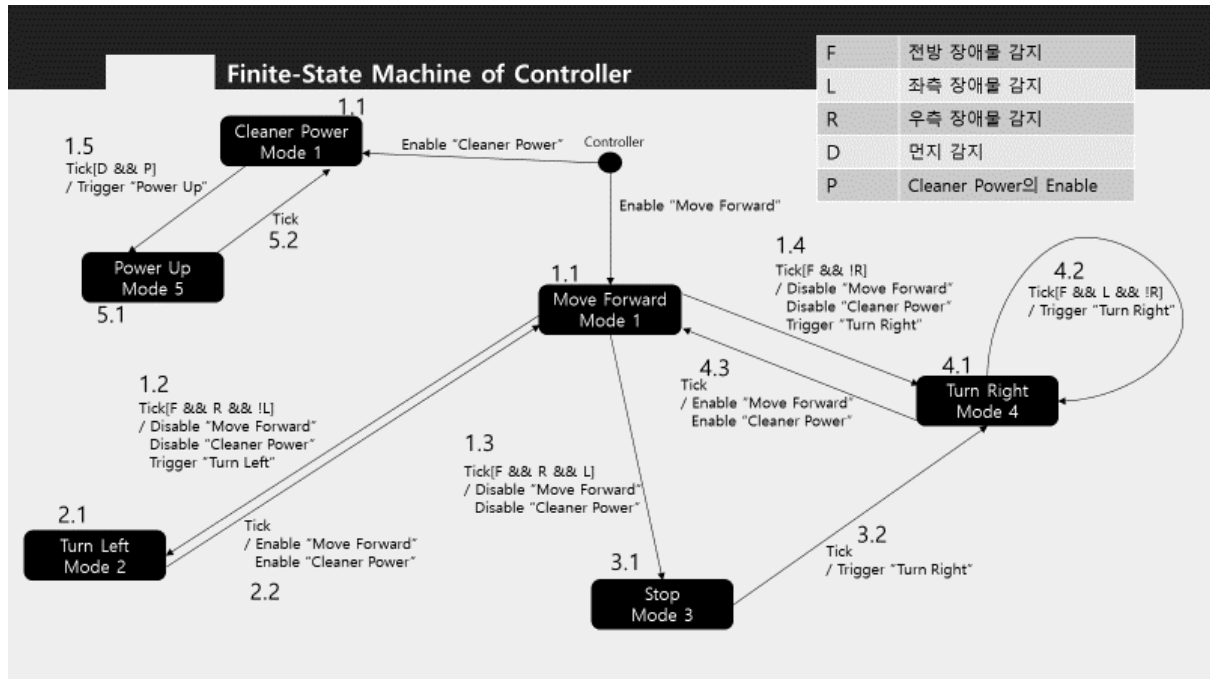
ㄱ. 이름 : Controller의 명령 전달

ㄴ. 목적 / 내용 : Controller가 제어한 흐름대로 모터, 청소기에 수행할 기능 전달

ㄷ. 입력 주체 / 출력 대상 : Controller / Motor , Cleaner

ㄹ. 시간 : 매 Tick 마다 전달

3.2 Functional requirements (기능 요구사항)



<그림 1. Finite-State Machine of Controller>

3.2.1 Mode 1

3.2.1.1 Functional requirement 1.1

Move Forward (Enable), Cleaner Power (Enable)

기본 동작으로써 Tick 마다 입력 받은 데이터에 감지된 장애물이나 먼지가 없다면 앞으로 가면서 청소를 한다.

3.2.1.2 Functional requirement 1.2

Tick[F && R && !L]

Mode 1에서 전방과 우측 장애물 감지 데이터가 입력되면 Mode 1의 기능을 멈추고 Mode 2로 전환한다.

3.2.1.3 Functional requirement 1.3

Tick[F && R && L]

Mode 1에서 전방, 좌측, 우측 장애물 감지 데이터가 입력되면 Mode 1의 기능을 멈추고 Mode 3로 전환한다.

3.2.1.4 Functional requirement 1.4

Tick[F && !R]

Mode 1에서 전방 장애물 감지 데이터가 입력되면 Mode 1의 기능을 멈추고 Mode 4로 전환한다.

3.2.1.5 Functional requirement 1.5

Mode 1에서 먼지 감지 데이터가 입력되면 Mode 1의 기능을 유지하고 Mode 5로 전환한다.

3.2.2 Mode 2

3.2.2.1 Functional requirement 2.1

Mode 1 (Disable), Turn Left (trigger)

Func. Requirement 1.2에 의해서 Mode 2로 전환되면 Mode 1의 기능을 멈추고 좌측으로 90° 만큼 회전한다.

3.2.2.2 Functional requirement 2.2

Mode 1 (Enable)

Func. Requirement 2.1에서 회전을 마치면 다시 Mode 1의 기능을 수행한다.

3.2.3 Mode 3

3.2.3.1 Functional requirement 3.1

Mode 1 (Disable), Stop (trigger)

Func. Requirement 1.3에 의해서 Mode 3로 전환되면 Mode 1의 기능을 멈추고 다음 Tick까지 대기한다.

3.2.3.2 Functional requirement 3.2

Tick

Func. Requirement 3.1에서 대기하다가 다음 Tick에 도달하면 Mode 4로 전환한다.

3.2.4 Mode 4

3.2.4.1 Functional requirement 4.1

Mode 1 (Disable), Turn Right (trigger)

Func. Requirement 1.4에 의해서 Mode 4로 전환되면 Mode 1의 기능을 멈추고 우측으로 90° 만큼 회전한다.

3.2.4.2 Functional requirement 4.2

Tick[F && L && !R], Turn Right (trigger)

Func. Requirement 3.2에 의해서 Mode 4로 전환되면 우측으로 90° 만큼 회전해도 다음 Tick에서 전방과 좌측 장애물 감지 데이터가 입력된다. 따라서 Mode 전환을 하지 않고 한번 더 우측으로 90° 회전한다.

3.2.4.3 Functional requirement 4.3

Mode 1 (Enable)

모든 과정(1.4->4.1 , 1.3->4.2)에서의 회전을 마치면 다시 Mode 1의 기능을 수행한다.

3.2.5 Mode 5

3.2.5.1 Functional requirement 5.1

Mode 1 (Enable), Power up (trigger)

Func. Requirement 1.5에 의해서 Mode 5로 전환되면 Mode 1의 기능을 유지하고 다음 Tick까지 Power up을 수행한다.

3.2.5.2 Functional requirement 5.2

Tick

Func. Requirement 5.1에서 다음 Tick에 도달하면 Mode 5의 기능을 중단하고 Mode 1으로 돌아간다.

3.3 Performance requirements (성능 요구사항)

센서로부터 입력 받는 장애물 감지 데이터와 먼지 감지 데이터는 서로 독립적으로 관리되어야 하며 감지 데이터를 기반으로 장애물의 위치나 먼지의 존재 여부를 연산하는 과정이 Tick/2의 시간 내에 수행되어야 한다. 또한 데이터들이 인터페이스를 통해 전달할 때 Tick/4의 시간 내에 수행되어야 매 Tick마다 기능을 수행할 수 있다.

3.4 Design constraints (설계 제약사항)

Controller 설계 시 논리회로에 빈틈이 없어야 하며, 오픈소스 사용 없이 직접 개발한다. 리눅스 운영체제를 사용하고, ANSI C 표준을 따라 프로그래밍 한다.

3.5 Software system attributes (소프트웨어 시스템 속성)

ㄱ. Reliability (신뢰성)

수행할 동작에 대해 정확하게 명령을 내림.

ㄴ. Availability (가용성)

실행 중에 어떠한 상황에서도 Tick시간 내에 계산 가능

ㄷ. Security (보안)

암호화하지는 않지만 Controller가 직접적으로 Network로 데이터 교환을 하지 않기 때문에 비교적 양호한 보안

ㄹ. Maintainability (유지 보수성)

각 하드웨어마다 인터페이스가 정의되어 있어서 유지보수가 쉬움

ㅁ. Portability (이식성)

리눅스를 이용하고 ANSI C 표준을 따르기 때문에 이식성이 좋음

3.6 Other requirements(기타 요구사항)

없음