

위치 인식 센서 네트워크 기반의 청소 로봇 시스템 구현

김중현^o, 이한선, 정광수
 광운대학교 전자통신공학과
 remos@kw.ac.kr^o, hslee@adams.kw.ac.kr kchung@kw.ac.kr

Development of the robot system based on localization sensor networks

Jonghyun Kim^o, Hansun Lee, Kwangsue Chung
 School of Electronics Engineering, Kwangwoon Univ.

요 약

최근 들어 모든 사물과 대상을 자동화하려는 새로운 통신 패러다임과 함께 센서 네트워크의 필요성이 부각되고 있다. 센서 네트워크는 물리적인 공간의 모니터링이나 위치 추적과 같은 주변의 정보를 얻고자하는 환경에 사용되며, 이러한 센서 네트워크와 특정한 기술과의 결합으로 새로운 가치를 창출하려는 움직임이 일고 있다. 본 논문에서는 위치 추적이 가능한 센서 네트워크 기반의 청소 로봇 시스템을 구현하였다. 구현된 응용은 기본적으로 PC를 통해 모니터링할 수 있으며, 청소 로봇의 이동 위치 정보를 수집하여 빈 청소 구간의 이동, 중복 청소 구간의 회피를 하며, 사용자가 원하는 위치로 이동하는 기능을 구현하였다. 또한 TCP/IP를 통해 기존 인프라와의 네트워킹이 가능하도록 하였다.

1. 서 론

최근 들어 도로, 다리, 터널, 빌딩, 건물벽 등 모든 물리 공간과 객체에 컴퓨팅 기능을 추가하여 모든 사물과 대상을 자동화하고, 전자공간에 연결되어 서로 정보를 주고 받는 공간을 형성하고자 하는 유비쿼터스 컴퓨팅이 각광을 받고 있다. 이러한 새로운 통신 패러다임은 센서 네트워크의 필요성을 부분적으로 부각시키고 있다[1].

센서 네트워크란 여러 개의 저-전력 센서 노드가 무선 통신을 통해 서로 협력하여 특정 작업을 수행하는 새로운 컴퓨팅 분야이다. 센서 네트워크는 주로 물리적인 공간의 모니터링이나 위치 추적과 같은 주변의 정보를 얻고자하는 환경에 사용된다. 특히 주변의 위치 정보는 사용자나 사물에게 주변 환경에 대한 인식을 가능하게 하는 기본적인 요소이기 때문에 유비쿼터스 컴퓨팅에 가장 핵심적인 요소이다. 이러한 센서 네트워크와 특정한 기술과의 결합으로 새로운 가치를 창출하려는 움직임이 일고 있다[2].

로봇 기술은 공장, 산업, 가정 등에 활발히 이용되고 있으며, 특히 가정용 청소 로봇의 등장은 인건비 절감과 편의성 제공으로 인해 인간의 생활에 밀접한 영향을 주고 있다. 현재 판매되고 있는 대부분의 청소 로봇은 자동으로 방의 크기를 측정하고 시간을 설정하여 청소를 할 수 있으며 외출 중에도 청소를 마칠 수 있다. 하지만 이러한 청소 로봇은 개발 초기 단계이니 만큼 인간에 가까운 청소 기능은 기대하기 어려우며, 미리 정의된 청소 알고리즘으로 인해 중복 청소 구간과 청소 미구간이 발생하며, 출입구가 좁은 곳에서는 청소 완료 후 빠져나오지 못하는 경우도 발생한다.

본 논문에서는 이러한 기존 청소 로봇이 갖는 문제점을

보완하기 위해 위치 인식 센서 네트워크 기반의 청소 로봇 시스템을 구현하였다. 구현된 응용은 기본적으로 PC를 통해 모니터링 할 수 있으며, 청소 로봇의 이동 위치 정보를 수집하여 빈 청소 구간의 이동, 중복 청소 구간의 회피를 하며, 사용자가 원하는 위치로 이동하는 기능을 구현하였다. 또한 TCP/IP를 통해 기존 인프라와의 네트워킹이 가능하도록 하였다.

본 논문 2장에서는 전체 시스템 구성과 센서 노드를 이용한 위치 인식 기법, 청소 로봇 컨트롤과 모니터링 등에 대해 기술 하였다. 3장에서는 구현내용을 기술하였고, 4장에서는 결론 및 향후 과제에 대해 기술 하였다.

2. 구현을 위한 기본 방법

2.1 Cricket을 이용한 거리 산출 기법

실내에서의 위치 기반 시스템은 MIT에서 개발한 Cricket 시스템을 이용 하는 것이 일반적이다[7]. 그림 1은 이러한 시스템을 이용하여 거리를 산출하는 방법을 나타낸 것이다. 거리에 대한 산출은 Beacon과 Listener 사이의 RF 신호와 초음파 신호의 도달 시간 차이를 이용하며, 위치에 대한 추정 은 Listener와 연결된 응용 레벨에서의 추정 알고리즘으로 계산 된다. 즉, 기존 Cricket 시스템은 거리와 위치에 대한 산출 방법이 분리되어 있다.

위치인식을 하기 위해서는 Beacon 과 Listener와의 거리 값이 필요하다. 그림 1과 같이 Beacon에서 Listener로 비동기 적으로 RF신호와 초음파 신호를 보낸다. 타이머는 RF 신호를 수신하면 시작되고 초음파 신호를 수신하면 정지 된다. 타이머를 이용하여 초음파 신호의 도달 시간 계산하고 초당 340m/s인 초음파 속도와 곱하여

거리를 산출 한다[4].

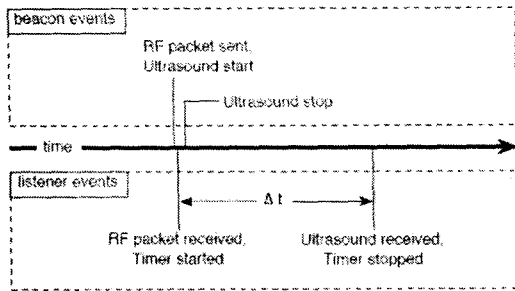


그림 1. Beacon과 Listener의 거리 산출 방법

2.2 청소 로봇 컨트롤 방법

구현에 사용된 청소 로봇은 현재 가정에서 가장 많이 사용되고 있는 i-robot사의 Roomba를 사용하였다. 이 제품에는 Roomba를 제어할 수 있도록 SCI(Serial Control Interface)가 제공되며[5], 제어를 위해 사용자는 정해진 명령 코드를 시리얼 통신으로 입력 가능하다. 이와 같은 시리얼 통신을 위한 별도의 인터페이스 보드가 필요하며, 이러한 보드는 그림 2와 같이 MAX232로 구성된 전압 레벨 컨버팅 회로로 제작할 수 있다.

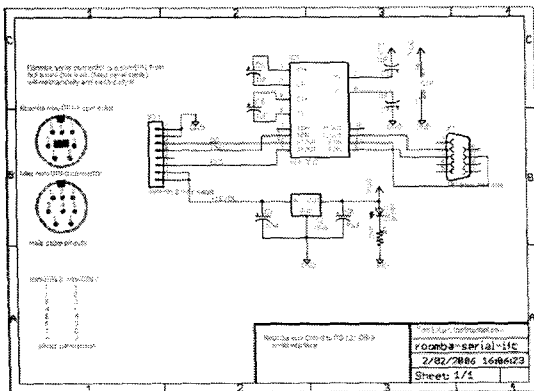


그림 2. Serial Command Interface 회로도

Roomba 제어를 위해 i-robot사에서 제공하는 Roomba SCI는 off, passive, safe, full의 4개의 모드로 동작한다. 사용자가 주는 각 명령은 4가지 모드 중 하나의 모드에 있어야 그 명령에 대한 동작이 수행되며, 수행 후 역시 고유 모드 상태로 놓이게 된다. 예를 들어, Roomba 자체의 청소 알고리즘을 사용하려면 Virtual Button(0x85, 0x86, 0x87, 0x88)을 사용한다. 사용자가 직접 청소기 위치를 제어하고 원하는 알고리즘을 적용시키기 위해서는 Drive(0x89)의 명령을 사용해야 한다.

3. 센서네트워크 위치 기반의 청소 로봇

본 논문에서 구현한 전체 시스템은 그림 3과 같으며 로봇청소기에 탑재된 Listener는 Beacon으로부터 거리 데이터를 받아 자신의 위치 좌표를 산출한다. 산출한 데이터는 PC와 연결된 Listener로 전송되며 PC에서 실행되는 모니터링 툴 프로그램을 통해 청소기의 위치를 보여준다. 또한 모니터링 툴은 청소기로 명령을 전송하여 청소기의 동작을 제어 한다.

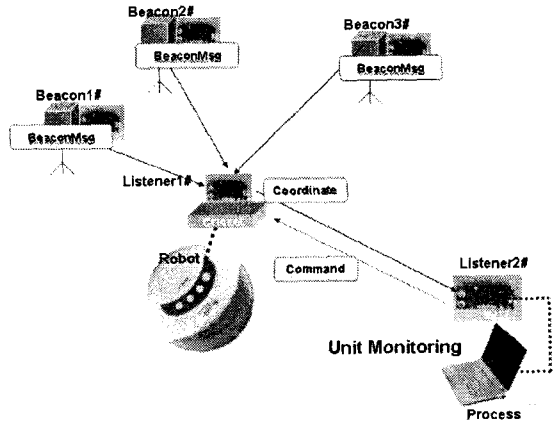


그림 3. 전체 시스템 구성도

3.1 Cricket을 이용한 위치 산출 방법

앞장에서 말한바와 같이 Listener와 Beacon간의 거리 정보를 수집하여 자체 개발한 위치 산출 알고리즘에 대입 하여 Listener의 위치 좌표를 산출 한다. 기존 시스템에서는 Beacon이 PC에 연결된 Listener에게 거리 정보를 전송하면 PC의 응용에서 세 개의 원의 교점을 찾아 좌표를 산출 하는 방식이다. 하지만 정확한 좌표를 위해서는 Listener가 정지 한 후 약3초 정도의 지연이 발생하여 그림4의 구조로 새로운 좌표 산출 알고리즘을 산출 하였다. 또한 기존 Cricket은 거리와 위치에 대한 산출 방법이 분리되어 있었으나 위치 산출 알고리즘을 Cricket에 구현하였다. 그림 4와 같이 Listener는 2개의 Beacon으로부터 거리정보($Distance_1$, $Distance_2$)를 수신 한다. 두 Beacon 사이의 거리 ($Usersetting$)와 높이 (y)는 사용자가 세팅을 한다. Listener와 두 Beacon이 꼭지점인 삼각형이 만들어 진다. Listener로부터 두 노드 사이의 거리와 Beacon 사이의 거리를 모두 알기 때문에 삼각형의 세변의 길이를 알 수 있다.

있어야 명령에 수행된다. 수행 후 고유 모드상태로 놓이게 된다.

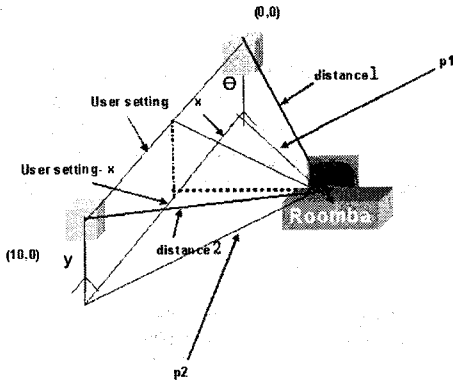


그림 4. 좌표산출을 위한 파라미터

세변의 길이를 알면 제2코사인 법칙에 의해 한각의 Cos 값을 알 수 있다. 그림 4에서 Cos 값은 식 (1)과 같이 산출 된다.

$$\text{Cos} = \frac{\text{UserSetting}^2 + \text{distance1}^2 - \text{distance2}^2}{2 \times \text{UserSetting} \times \text{distance1}} \quad (1)$$

(0,0)으로 부터 x의 상대좌표는 식 (2)와 같이 구할 수 있고 두 Beacon 사이의 거리는 바닥에서의 거리와 같다. 따라서 Listener의 x좌표는 식에서 구한 x 값이다.

$$x = \text{Cos} \times \text{distance1} \quad (2)$$

Listener의 y좌표는 distance1의 거리로부터 계산 된다. Listener, x좌표, Listener에서 Usersetting으로 수선을 내린 변으로 이루어진 직각 삼각형으로부터 피타고라스 정리에 의해 수선의 길이를 구하며, 수선, 높이, 수선의 정사영인 선분으로 이루어진 직각삼각형으로부터 피타고라스 정리에 의하여 y좌표를 산출한다. 이는 식 (3)과 같이 계산 될 수 있다.

$$y = \sqrt{\text{distance1}^2 - x^2 - \text{height}^2} \quad (3)$$

Listener에서 산출된 x와 y 좌표의 값을 PC와 연결된 Listener로 RF전송을 한다. 좌표 산출을 위해 Listener는 Beacon으로부터 거리 데이터를 수신한다. 하지만 정확한 거리 데이터가 들어오지 않아 산출 알고리즘 적용 후 오차가 발생 하였다. 이를 해결하기 위하여 Beacon으로부터 거리 정보를 수신 받을 때 일정 값 이상의 데이터는 수신하지 않고 기존 데이터를 사용한다.

3.2 청소 로봇 컨트롤

청소 로봇과 SCI 그리고 Cricket은 그림 5와 같이 연결 된다. Roomba는 off, passive, safe, full의 4개의 동작 모드가 있다. 각 명령 마다 4가지 중에 하나의 모드에

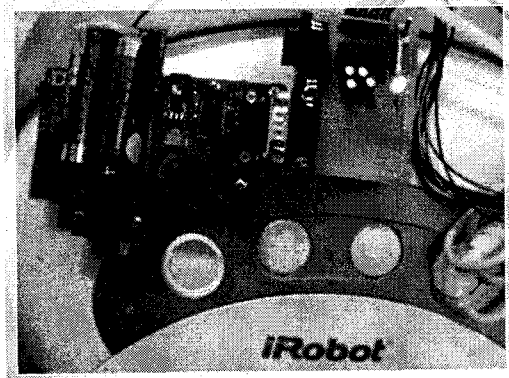


그림 5. 청소로봇과 시리얼 인터페이스와 센서보드 연결

Drive명령은 뒤에 4바이트의 인자를 가지며 앞의 두 바이트는 -500 ~ 500 mm/s의 속도를 나타내고, 뒤의 2바이트는 -2000 ~ 2000의 청소기의 회전 반경을 설정해주는 인자이다. Sensor 명령(0x8E)은 Roomba에 달린 여러 가지 센서, 배터리 잔량, 전류 그리고 온도 값을 입력 받는다. 옵션에 따라 26바이트, 10바이트, 6바이트 등 필요에 따라 센서 값을 받을 수 있다. Roomba의 상태를 나타내주는 Sensor 명령을 사용하여 Roomba의 전방 접촉 스위치 좌우 IR센서의 정보를 시리얼 통신으로 입력 받아 원하는 기능을 Drive, Virtual Button 과 같은 명령으로 Roomba를 Cricket에서 직접 제어한다. 하지만 센서 데이터를 사용하면 일반적으로 마이크로 컨트롤러의 인터럽트 기능을 사용하지 못한다. 센서 패킷의 데이터를 계속 감시하는 풀링 방식으로 센서 값을 읽어와 사용하기 때문에 감지 후 동작하는 속도가 느리다. 이를 보완하기 위하여 26개의 센서 패킷을 모두 감시하지 않고 필요한 부분만 읽는다. 실제로 청소기의 센서 중에 필요한 부분은 앞 범퍼의 상태와 질벽 IR센서인데 바닥에서 사용할 경우 질벽 IR센서는 필요가 없기 때문에 앞 범퍼의 센서 데이터만 사용한다.

주행 알고리즘은 Drive명령을 이용하는데 타이밍으로 제어해야 한다. 제공하는 모터 제어가 Step 단위로 동작하지 않아 타이밍과 센서 데이터로 알고리즘을 제작하였다.

3.3. 모니터링 툴

현재 청소 구역 안에서 청소기의 위치와 청소된 상태를 모니터링하고, 청소기에 직접 임의의 명령을 줄 수 있다. 어플리케이션 환경의 장점을 이용하여 서버-클라이언트 환경을 추가적으로 구현하여 원격지에서도 청소기의 상태를 모니터링 가능하게 한다. 좌표 산출을 위한 Cricket 보드 간의 통신은 RF를 이용하고 있고, Roomba와 Cricket 보드, PC와 Cricket 보드 간의 통신

은 시리얼 통신을 이용한다. Listener로부터의 정보를 전달 받는 PC에는 모니터링 서비스 모듈이 실행되고, TCP/IP 환경을 이용하여 클라이언트 모듈과 데이터를 주고받게 된다. Server 쪽 Monitoring Tool은 시리얼 포트에 연결된 Beacon을 통해 전달 받은 Listener의 위치 정보를 출력하고, 직접 청소기를 컨트롤 할 수 있다. 전달 받은 정보는 클라이언트에 전달해주고, 클라이언트로부터 받은 명령은 Listener로 전달한다. 클라이언트 모니터링 툴은 서버 모듈과 동일한 이이지만, 시리얼 포트가 아닌 TCP/IP를 이용하여 Server 모듈로부터 들어온 정보를 바탕으로 모니터링을 수행한다. 마찬가지로 직접 청소기를 컨트롤하는 명령을 줄 수 있다. 테스트 콘솔 클라이언트는 모니터링 툴의 각 종 통신 데이터를 테스트하기 위해 TCP/IP를 통해 직접 패킷을 만들어 보낼 수 있는 툴이다.

시스템 구현은 사용자 입장에서 실제 생활에서 사용 가능한 기능을 구현 하였다. 먼저 모드를 2가지로 나누어 자동모드, 사용자 모드로 구성 된다. 자동모드는 청소 시작을 모니터링 툴에서 지시 해주면 Roomba의 내장된 알고리즘으로 청소를 시작 한다. 청소 시작 후 일정 청소가 진행 되었다고 판단되면 청소 맵의 상태를 확인 한다. 확인 하여 청소가 미진한 부분이 있다면 그곳으로 자동적으로 이동하여 청소를 한다. 또한 청소가 된 위치는 되도록 사용자 모드는 원하는 위치를 지정하면 청소기가 그곳으로 움직이며 여러 액추에이터 역시 사용자가 제어 할 수 있다. 사용자 모드로 원하는 곳에 이동시키고 원하는 일을 한 후 자동모드로 전환하여 동작시키면 효율적인 청소가 될 수 있다.

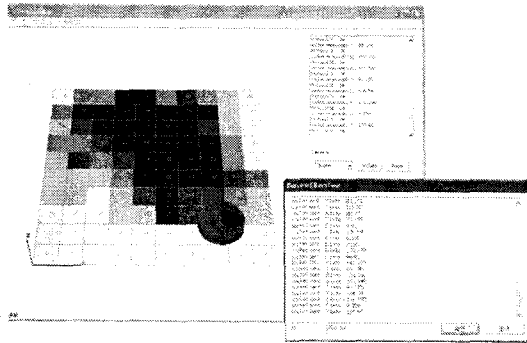


그림 6. 청소기 모니터링

모니터링 툴에서 네트워크 기능을 추가 하여 인터넷이 가능한 곳에서 제어가 가능하도록 구현 하였다. 모니터링 구현은 크게 5가지 모듈로 이루어져 있다. 유저 인터페이스는 툴의 전체적인 모양과 각종 다이얼로그 및 컨트롤들을 MFC로 구현하였다. 렌더링 모듈은 OpenGL을 이용하여 모니터링 간단한 3D 환경으로 볼 수 있도록 하였다. 수학 모듈은 3D 환경에 적합한 계산을 유용하게 하기 위해 Vector 등의 수학 객체를 간단하게 구현하였다. 맵 구조 관리 모듈은 청소 구역을 관리하기 위해 맵 구조체를 만들고, 이동 할 때마다 궤적을 판단하여 청

소 여부와 횟수를 체크하여 표시한다.

통신 모듈은 윈도우 소켓을 이용하여 서버 모듈과 클라이언트 모듈을 Select 방식으로 구현하고, Cricket 보드와의 통신을 위해 시리얼 통신 관련 부분도 제작하였다. 청소 로봇을 동작 시키면 그림 6과 같이 모니터링 툴에 청소 궤적이 나타난다. 구역으로 지정해 놓은 셀 당 지나간 횟수를 카운트 하여 셀의 색을 바꾸어 주었다. 현재 5단계로 세팅이 되어 있고 청소기가 많이 지나 갈 때마다 진한 색으로 바뀌는 것을 볼 수 있다. 청소기의 자동 이동에 관한 것은 위와 같은 맵핑 정보에 따라 청소가 미비한 구간을 판단하고 청소 로봇에게 이동 명령을 내린다.

전체 적인 시스템 구현 확인을 위하여 그림7과 같은 데모 환경을 구축 하였으며 위에서 기술된 구현 내용을 확인 하였다.

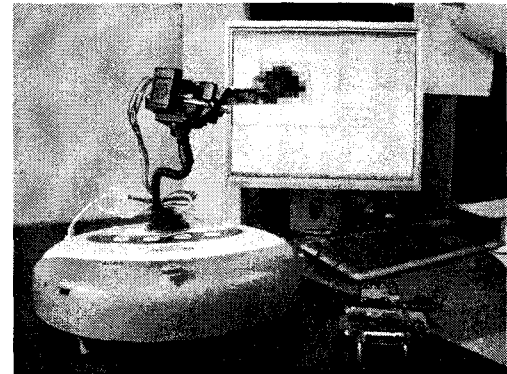
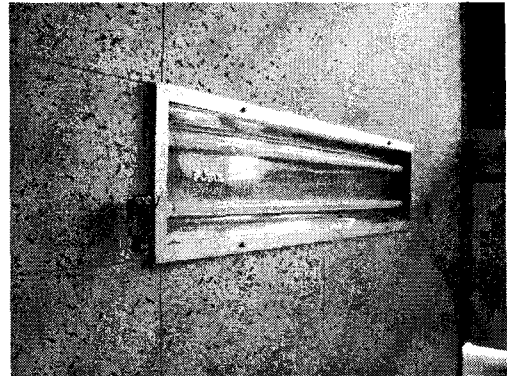


그림 7. 데모 환경 설치 모습

4. 결론 및 향후 과제

센서 네트워크를 이용하여 센서 노드의 위치를 측정하고 센서노드를 I-robot사의 청소로봇 Roomba에 연결하여 위치기반의 청소 로봇 시스템을 구현하였다. 기존 청소 로봇과 달리 PC에서 모니터링이 가능하고 청소 로봇 스스로 미흡한 청소 구간에 이동하여 청소 동작을 수행 한다. 사용자가 원하는 위치로의 이동과 청소 로봇의

개별 제어를 할 수 있는 시스템을 구현 하였다.

청소기의 위치를 인식함으로써 청소기의 위치 파악 뿐만 아니라 지나간 자리를 로깅 하여 청소가 된 부분과 안 된 부분을 판단 할 수 있게 되었다. 또한 사용자 중심의 모니터링 툴에서 무선으로 청소기를 제어 하여 사용자의 명령에 따라 청소가 가능한 청소로봇을 구현 하였다. 또한 모니터링 툴에 TCP/IP 기능을 구현하여 인터넷이 되는 어느 곳에서든 청소기의 상태를 파악 할 수 있고, 청소기의 상태를 제어 할 수 있다. 기존 청소 로봇이 가지는 닫힌 공간에서의 무한 루프 발생 문제를 해결 하였으며, 지능적으로 청소 상태를 파악하여 청소 동작을 결정하여 청소 품질을 향상 되었다.

센서 네트워크 관점에서 추적좌표의 속도가 증가함에 따라 위치 측정값이 한계를 갖는다. Cricket에서 들어오는 초음파와 RF 신호 시간차의 결과인 두 노드 간의 거리 값이 Listener에 비동기 적으로 들어온 데이터에 대한 추정기법을 향상시킬 것이다. 기능면에서 청소 로봇 동작 알고리즘을 향상 시켜 동작의 신속성과 청소 기능의 효율성을 높이고, 모니터링 툴에서는 PDA와 같은 모바일에서의 구현을 하여 사용자 이동성에 대한 연구를 진행 할 것이다.

참 고 문 헌

- [1] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research," Dartmouth Computer Science Tech Report TR2000-381, 2000.
- [2] M. Satyanarayanan, "Pervasive computing: vision and challenges," IEEE Personal Communications, pp.10-17, Aug. 2001.
- [3] <http://cricket.csail.mit.edu/>
- [4] Nissanka B. Priyantha, Anit Chakraborty, Hari Balakrishnan, "The Cricket Location-Support system," Proc. 6th ACM MOBICOM, Boston, MA, Aug. 2000.
- [5] http://www.irobot.com/images/consumer/hacker/Roomba_SCI_Spec_Manual.pdf