論文2002-39SC-2-3

여러 개의 모터에 의하여 제어되는 링-코어 자동 선별기 개발

(Development of Ring Core Auto-Classifier by Multi-Motor Control)

朴 寅 圭 *

(In Gyu Park)

요 약

인덕터 전기 부품을 구성하는 핵심 부품인 링-코어를 물질 특성, 인덕턴스 값 및 Q-값에 따라 10등급으로 분류하는 선별기를 설계하는 데 있어, 한번에 10개의 코어를 동시에 자동 분류할 수 있도록 설계한다. 동시에 10개의 코어를 자동 선별하려면 10개의 측정 장비가 필요한데, 이러한 측정 장비는 대단히 고가이므로, 하나의 측정 장비를 사용하여 10개의 링-코어를 검사한 후, 각각의 등급에 따라서 해당 그릇으로 저장하도록 한다. 이러한 시스템을 개발하는 데 있어서, 초당 0.5개의 생산 속도보다 빠른 속도인 초당 1개 정도로 측정 분류할 수 있는 속도가 필요하다. 시스템의 구성물은 먼저 작업 명령을 내려주고, 각종 생산 및 품질 통계를 관리하는 작업 PC와, 100개의 고무호스, 공급기, 측정기, 10개의 등급-상자, 1개의 메인-보드, 10개의 모터-제어-보드, 10개의 모터-드라이버-보드, 10개의 스텝-모터 등으로 구성되어 있다. 따라서 많은 고장, 오동작이 예상된다. 가장 중요한 사항은 일부의 모터 고장, 일부 모터 드라이버 보드 고장, 일부 모터 제어 보드 등의 고장이 발생하여도 작동이 멈추지 않도록 설계하는 것이다. 이를 위하여 센서 회로 추가 및 관련 소프트웨어 알고리즘을 개발한다.

Abstract

Core is the main component of inductor. This core should be classified into around 10 classes according to the value of inductance and Q. The coil should be winded with the outer-boundary of this core by different number of turns. Theses kind of precise inductors would be required in the future environment which PCs and communication devices demand more high speed and lower voltage level. It would be quite unefficient that only one core is classified once a time. There, it will be developed so that 10 cores are classified simultaneously. For the operation of classifying 10 cores once in a time, suppose 10 test instruments could be used. In this case, it would take much cost since a test instrument is expensive. So, by using only one test instrument, it is really more desirable that this system is developed. Each core classified by 10 different classes is to be stored into the corresponding box through the corresponding rubber hose. 10 cores are passed on a serial line and are placed on each testing slot. Here, each core located at each slot is tested, and then the bowl located on the top of a step motor is moved into the corresponding spot by rotating step motor with some angles. Each bowl connected with the corresponding box through rubber hose. Actually 100 hoses are connected, 10 step motors are rotated at 10 different angles, so the size is really so big, the shape of connecting 100 hoses is so complicated. Therefore it is anticipated that

* 正會員, 弘益大學校 電子電氣工學部,
(Department of Electronic and Electrical Engineering)

接受日字:2001年12月3日, 수정완료일:2002年1月14日

Ⅰ. 서 론

인덕터의 구성물을 살펴보면, 페라이트, 철분, RF, 실

the system would be going to be easily out of ordered. In this paper the main purpose is to make several suggestions to be able to work well in these kinds of being affected by the abnormal operation of motors and the flow of cores.

리콘 합금, Sendust, MPP, Hiflux 등의 물질로 링 혹은 캡(모자) 모습의 코어와 코어 외곽을 감는 동선으로 구성된다. 이러한 코어들은 거의 같은 시간에 생산하여도 서로 다른 인덕턴스 및 Q 값을 갖게 된다. 따라서 코어를 하나씩 전수 검사하여 분류하여야 한다. 수동으로 하나의 코어를 측정 장비를 사용하여 측정하면 적어도 30초 정도 걸리게 된다. 그리고 만약 한번에 하나씩 측정하는 자동 검사 장치를 사용하더라도 10초 이내에 검사하는 것은 상당히 어렵다. 한 개의 생산 프레스 장비로 코어는 초당 0.5개 정도로 생산될 수 있다. 따라서 측정 속도가 생산 속도에 비하여 대단히 늦다. 따라서 많은 코어를 동시에 분류할 수 있는 자동 선별기를 개발하여 생산 속도에 맞추어 검사 분류하도록 한다.

더욱이 PC 또는 통신기기 등이 높은 주파수, 낮은 전압에서 작동되는 경향이 점증될 것이다. 이러한 환경에서는 노이즈를 제거하고, 안정된 전원을 유지하여주는 정밀한 코어가 요구된다. 따라서 코어를 Q-값 및 인덕턴스-값에 따라 여러 등급으로 구분할 필요가 있게 된다. 업계에서는 대략 10개 정도로 구분하면 충분하다고 한다. 따라서 본 시스템에서는 10등급으로 분류하도록 한다. 또한 한번에 여러 개 코어가 분류되어야 효과적이기 때문에 본 시스템에서는 10개의 코어가 동시에 분류되도록 설계한다. 문제는 이러한 10개의 코어가 동시에, 그리고 하나의 코어는 각각 10개의 분류그릇으로 저장되어야 하는데, 이러한 배분하는 메커니즘이 단순하면서도 정확하여야 하고, 지속적으로 언제나 정확하게 해당 분류그릇에 저장되는 신뢰성이 보장되어야 한다.

실제로 이러한 10개의 코어가 각각 10등급으로 분류하는 메커니즘의 모습을 살펴보면, 10개의 코어가 동시에 측정을 마친 후에, 각각 해당되는 등급-상자로 저장하여야 한다. 10개의 등급-상자를 맨 아래에 설치하여 놓고, 하나의 등급-상자에는 10곳으로부터 입력 받을 수 있다. 하나의 코어는 10등급 중 하나이므로 10곳 중에서 한 곳으로 갈 수 있고, 동시에 10개의 코어가 내려가므로, 하나의 등급-상자는 10곳으로부터 입력 받을 수 있다. 하나의 코어가 10개의 등급으로 분류하는 방법으로, 하나의 모터에 분류-그릇이 얹어져 있고, 이러한 분류-그릇의 출구가 36도씩 10개의 구멍에 정확히 위치하게 하면, 10개의 구멍은 10개의 등급-상자와 고무호스로 연결되어 있다. 이러한 고무호스는 직경 5,5cm 플라스틱 재료의 호스 100개가 필요하고, 평균길이가 1m 정도 필요하여, 호스 길이가 무려 100m가 되어, 기구적으로 대단히 밀착, 복잡하게 구성되어 있다. 또한 하나의 호스도 막힘이 있거나, 굴러 가는데 막힘이 있게 되면 절대로 안 되록 설계되어야 한다. 또한 스텝모터가 10개나 되고, 각각의 모터는 36도씩 정확한 위치에 위치하여야 한다, 10개의 구멍이 대단히 근접하게 연결되어 있기 때문에 조그만 위치가 이동되어 있어도 코어가 잘 내려 갈 수가 없다. 더욱이 일부 모터의 작동이 잘 되지 않게 되면 코어가 제대로 흘러 갈 수가 없다. 즉 전체 시스템을 정지하고 다시 고친 후 정상적으로 작동시켜야 한다. 따라서 근본적으로 고장확률이 많은 요인을 가지고 있는 시스템이다. 기구적으로 복잡하고, 많은 호스가 연결되어 있고, 모터가 10개가 모두 언제나 10개의 위치 중 하나의 구멍에 정확하게 이동하여야 하는 조건에서만이 정상적으로 작동되는 시스템이므로 고장이 발생하지 않도록, 더욱이 일부 발생하여도 멈춤없이 작동하도록 설계하여야 한다.

위에서 언급한 바와 같이 일부 고장에도 작동 가능하도록 하드웨어 및 소프트웨어를 설계하여야 한다. 이러한 예상 고장에 대하여 나열하여 보면, 먼저 첫째 PC에서 특정 스텝모터에 등급정보를 정상으로 전달 못한 경우, 이러한 스텝모터는 어떠한 등급으로 이동하여야 하는지, 둘째 등급정보를 정상으로 수신하였지만, 스텝모터가 해당 등급-구멍에 정위치 하지 못한 경우, 이 경우에 PC가 스텝모터의 위치를 인지할 수 없다. 그렇다면 PC는 어떻게 대처할 것인가?, 셋째, 스텝모터가 수시로 원점을 잡는다. 스텝모터에 부착된 핀 및 opto-coupler로 원점을 찾는 데, 만약 핀이 떨어져 나가게 되면 원점을 잡을 수가 없게 된다. 그렇다면 원점을 잡을 수가 없는 상태를 어떻게 감지할 것이며, 감지하였어도 어떻게 대처할 것인가?, 또한 스텝모터에 부착되어 있는 핀이 약간 움직이게 되면 원점 잡는 위치가 약간

틀려질 수가 있다. 이러한 경우에 약간 이동되었다는 사실을 어떻게 감지할 것이며 어떻게 대처할 것인가? 더욱이 핀이 약간 움직이게 되면, 때로는 원점을 2번 이상 잡을 수 있다. 물론 위치가 대단히 미소하게 원점 위치가 움직여 있어도 눈으로는 거의 구분할 수 있게 되어 사실상 위치적으로는 문제가 없다하더라도 프로그램으로 원점을 잡으라고 명령을 내리면 경우에 따라 약간 다른 지점에서 원점을 잡게 되는 결과를 가져온다. 이러한 대단히 귀찮은 소프트웨어 문제점을 어떻게 해결할 것인가? 넷째 전력문제로서, 스텝모터가 동시에 10개가 작동되어야 하므로, 일부 스텝모터에 흐르는 전류가 약하게 되는 경우에 제대로 작동되지 못할 것이다. 이러한 경우에 어떻게 전류의 부족을 감지할 것이며, 어떻게 대처할 것인가? 마지막으로 코어의 물질이 일종의 철성분이고, 이러한 코어의 철분가루가 떨어지게 되어 있어 공기 중에 떠 있게 된다. 이러한 철분가루는 여러 가지 전기적, 자기적 노이즈를 발생시킬 것이며, 이러한 철분가루가 실제 하드웨어 보드에 부착하고, 이렇게 부착된 철분가루가 회로 선로에 부착되어 전기적 고장을 초래할 것이다. 따라서 이러한 공중에 날라 다니는 철분가루에 대한 대책을 어떻게 할 것인가?

본 논문에서는 위에서 언급한 여러 에러 요인을 해결하는 방법을 고찰하고 이것에 대한 대책을 세워 하드웨어 및 소프트웨어로 해결을 모색한다. 물론 모든 것을 해결하는 것은 불가능하다. 따라서 최대한으로 해결하도록 한다. 논문의 전개를 소개하면, 먼저 이러한 하드웨어 시스템과 기구적인 메커니즘을 소개한다. 그리고 각각의 구조와 역할에 대하여 논한다. 다음에는 작동 소프트웨어에 대하여 논한다. 마지막으로 전체적인 시스템의 작동 테스트 결과에 대하여 논한다.

Ⅱ. 시스템 구조

시스템 구조에 대하여 논의한다. 먼저 링-코어가 흘러가는 루트에 대하여 설명하여 보면, 분류되지 않은 링-코어가 코어공급기에서 한 줄로 이동하면서 10개의 링-코어가 동시에 코어처리 메커니즘에 안착한다. 여기서 Voltek사의 Ati 장비[2]가 코어를 하나씩 측정하여 RS-232C 통신으로 PC로 측정데이터를 송신한다. 다음에 PC는 측정데이터를 등급 정보로 바꾸어 이러한 등급정보를 메인-보드로 보낸다. 메인-보드는 이러한 10개 코어의 등급정보를 RS-485 통신으로 10개 스레이브-보드에게 보낸다. 그러면 각각의 스레이브-보드는 자기의 등급데이터를 수신하여 자기의 스텝모터를 해당 등급 위치로 이동시킨다. 그러면 자동으로 스텝모터위에 얹어져 있는 분류-그릇의 출구가 각각 해당 등급-구멍에 정확히 위치하게 된다. 스텝모터위에는 분류-그릇이라는 플라스틱 용기가 얹어져 있다. 단지 위와 아

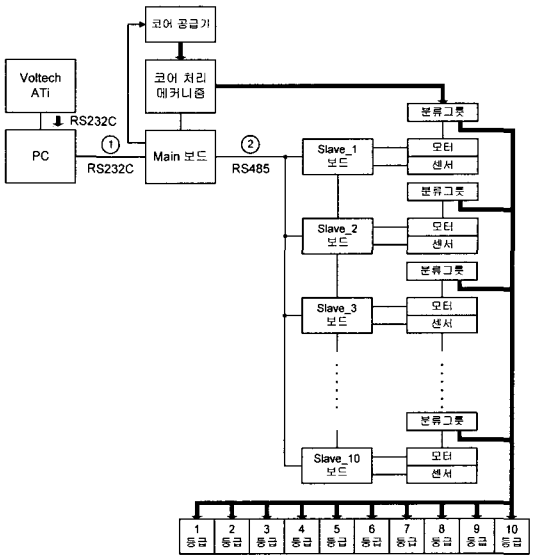


그림 1. 시스템 구조
Fig. 1. System Block Diagram.

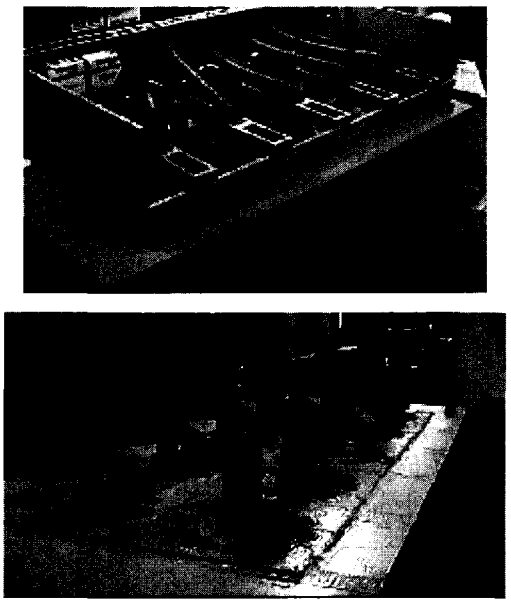


그림 2. 10개의 스텝모터, 분류-그릇
Fig. 2. Step Motor, Classifying Vessel.

래가 열려져 있어, 코어를 해당 등급-구멍으로 출력시켜 주는 역할을 한다. 다음에는 각각의 스텝모터위에 얹어져 있는 분류-그릇의 입구에 측정 완료된 10개의 코어를 동시에 메커니즘을 작동시켜 동시에 쏟아 붓는다. 그러면 자동으로 분류-그릇 출구를 거쳐 고무호스를 거쳐 해당 등급-상자로 굴러 간다.

PC와 측정 장비로부터 RS232C로 등급 정보를 받고, 메인-보드와는 RS-232C로 각종 명령을 보내고, 등급정보를 보낸다. 측정 장비는 영국 Voltek사의 제품으로 동시에 10개의 코어의 인덕턴스-값과 Q-값을 측정할 수 있다. 그러나 내부적으로는 하나씩 순차적으로 측정하여 대단히 빠르게 측정하지 못하는 것이 단점이다. 그러나 비교적 고정밀도로 측정할 수 있음을 확인하였다.

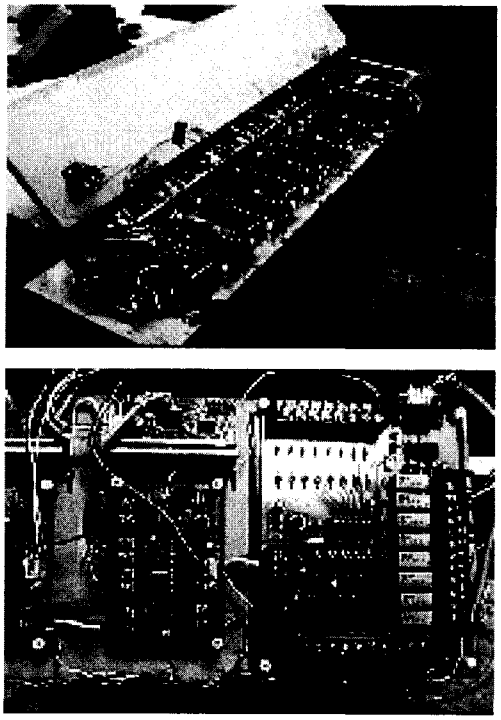


그림 3. 10개의 스레이브-보드와 메인-보드
Fig. 3. Slave-board & main-Board.

스레이브-보드는 각각의 스텝모터를 제어하는 보드로서 하나의 스텝모터마다 하나의 스레이브-보드가 담당하게 된다. 이러한 스레이브-보드들은 RS-485 통신으로 메인-보드와 연결되어 있다. 스레이브-보드에는 먼저 각각의 스텝모터의 알루미늄 외곽 표면에 위치한 핀이 opto-coupler 센서사이를 통과하는 순간을 포착하여 각각의 스텝모터의 원점을 잡을 수 있는 기능을 가지고 있다. 다음에는 스레이브-보드에서 펄스 신호를 하나 주면, 스텝모터를 한번에 0.72도 혹은 0.36도 만큼 돌게 할 수 있다. 따라서 스레이브-보드는 언제나 현재의 위치를 가지고 있다가 다음 위치 정보를 받으면, 계산하여 최대한으로 빠른 방법으로 이동하도록 모터를 제어한다. 그리고 작동이 끝났다는 정보를 다시 메인-보드에게 전달하고, 다시 메인-보드로부터 명령받을 준비에 들어간다.

Ⅲ. 시스템 소프트웨어

본 시스템에서 개발된 소프트웨어를 크게 분류하여 보면, 메인-보드와 스레이브-보드에서 사용한 소프트웨어로서., C 언어를 사용한 기계 운용 프로그램과, 운용 PC의 프로그램으로 Visual Studio 6.0의 Visual Basic을 이용한 작업자 운용 프로그램 등으로 구성되어 있다. 또한 기능에 따라 분류하여 보면, C 언어를 사용한 기계 운영 프로그램은 PIC 칩의 Firmware로써 구성물들 사이에 정보를 전달, 수행, 통신하는 역할을 주로 하게 되며, Visual Basic의 PC 운영 프로그램은 구성물 사이의 정보 교환, 통계 계산, 내부 비교 연산 처리, Data 저장, Data Real Time Display 등의 역할을 하게 된다. 다음은 프로그램의 역할에 따라 각각 자세히 설명한다.

1. PC와 메인-보드사이의 RS232C 통신

PC와 메인-보드 사이의 통신은 RS-232C Interface로 이루어지며, 통신은 미리 규약이 된 통신 프로토콜을 이용하며, 모두 11개의 캐릭터 명령 데이터 포맷을 사용하여 통신한다. PC 메뉴 제어 프로그램은 PC 운영 프로그램 내에 있는 MScomm Control이라는 Visual Basic Component가 담당한다. Event가 발생하면 응답하는 방식으로 인터럽트 방식과 유사하다. 또한 메인보드는 인터럽트로 수신하며, 수신 완료시 명령을 잘 받았다는 "ACK" 신호를 PC에게 보내고, 받은 명령을 처리한 후, PC에게 "End" 신호를 보내어 작업을 완료하고 다음 명령을 기다린다.

PC 운영 프로그램과 메인-보드사이에는 미리 설정한 명령어로써 통신을 하게 된다. 이러한 명령데이터 포맷은 1-바이트 character형으로 11개의 바이트가 어레이 형태로 구성되어 있다. 이렇게 10개의 스레이브-보드로 보낼 명령어를 한 개씩 배분할 수 있도록 10-바이트

명령어를 구성한다. 이렇게 구성하면 하나의 메인-보드에서 10개의 스레이브(slave)-보드에게 동시에 명령을 줄 수 있다. 여기서 첫 번째 1-byte는 PC에서 받은 이러한 명령어가 메인-보드에서 자체 처리하는 명령인지 혹은 스레이브-보드로 리레이 전송해야 하는 명령인지를 구별하기 위한 ID 역할을 하게 된다.

구체적으로 설명을 하면, 메인-보드에서 어떠한 명령을 받으면, 메인-보드에서는 이 명령을 기존의 Command Library에서 비교 검색하여 약속된 명령어와 동일한 명령이 Library에 있으면, PC쪽으로 명령을 잘 받았다는 "ACK"라는 신호를 보내게 된다. 그러면, PC 운영 프로그램에서는 "ACK"신호 수신과 동시에 운영 프로그램의 버튼을 Disable시킨다. 그러면, 메인-보드에서는 수신 받은 명령을 처리하게 되고, 처리가 끝나면, 다시 PC로 "End"신호를 보내게 된다. PC 운영 프로그램은 "End"신호 수신과 동시에 다시 운영 프로그램의 버튼을 Enable 시키게 되어, 다음 절차를 수행할 수 있는 준비가 되는 것이다.

다음은, 메인-보드에서의 첫 번째 1-바이트 정보인 ID에 따른 데이터 선별 과정이다. 그림 2와 같이 메인-보드는 PC로부터 수신 한 명령의 처음 1-바이트 명령의 종류를 분류하여 처리할 수 있다.

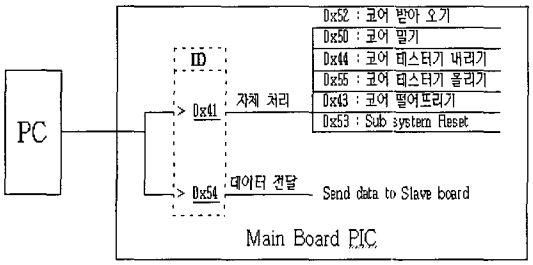


그림 4. ID에 따른 데이터 선별
Fig. 4. Data Format due to ID.

2. 메인-보드와 스레이브-보드와의 RS-485 통신

메인-보드와 스레이브-보드사이에는 RS-485 통신 규약을 따르며, [One-To-Ten] 방식으로 정보를 전달하게 된다. 여기서 [One-To-Ten] 방식이라 함은, 메인-보드와 스레이브-보드 10개가 동시에 통신을 수행하는 것을 의미하며, 또한 동시에 명령을 수행함을 의미한다. 즉 PC에서 보낸 등급 지정 명령을 수신한 메인-보드는 10개의 스레이브-보드에 동시에 명령을 보내게 되는데, 명령은 1-바이트 Character 11개로 구성된 배열 형태를 지니고 있다. 여기서 배열형 명령을 선택한 이유는 따로 특정 스레이브-보드의 어드레싱 작동을 할 필요가 없다는 것이다. 배열 자체가 어드레스를 갖고 있으므로, 각 스레이브-보드는 각 배열 요소 중에서 자신의 어드레스에 맞는 데이터만 가져다 실행하면 된다는 것이다. 이렇게 하므로 정보 전달의 시간적 낭비를 줄일 수 있고, 동시에 10개의 스레이브-보드에 한 번에 명령을 전달 할 수 있다는 것이다. 이것은 아주 효과적인 방법이며, 추후 이용 가능한 가치가 클 것으로 판단된다.

한편, RS-485 통신은 Half-Duplex 방식으로 동시에 송수신하는 기능을 지원하지 않는다. 따라서 송신, 수신을 할 경우, 모드를 변경하여야 하는 어려움이 있게 된다. 따라서 타이밍이 중요한 변수로 작용하게 된다. 또한 RS-485 출력 단자 11개가 서로 병렬로 연결되어 있기 때문에 노이즈, 전압 레벨, 임피던스 매칭 등의 문제점을 고려하여 설계하였다.

스레이브-보드는 항상 수신모드로 대기하고 있으며, 메인-보드는 RS-485 통신 포트만 항상 송신 모드로 대기하고 있는 것이다. 이렇게 함으로써 기존에 생길 수 있었던 통신 충돌 에러를 많이 해결하였다. 하지만, 문제점은 스레이브-보드에서는 정보를 메인-보드로 보낼 수 없게 된다는 것이다. 하지만, 쌍방향 통신을 사용한다고 하더라도, 스레이브-보드의 상태는 인지하기가 쉽지 않다. 예를 들어, 어떤 한 모터가 비정상으로 동작을 한다고 가정하여 보자. 가능한 모든 원인을 나열하여 보면, 우선 선로 상의 문제로 신호가 제대로 전송되지 못할 수도 있고, 혹은 보드 자체의 하드웨어가 비정상으로 작동할 수도 있고, 혹은 모터 드라이버-보드가 고장날 수도 있고, 또는 모터 자체에 전류가 충분하게 흐르지 못하거나, 모터여자가 고장난 경우 등으로 가정할 수 있다. 그렇지만, 정확한 원인이 무엇인지를 스레이브-보드 자체가 스스로 감지할 수 없다. 물론 스레이브-보드 하드웨어에 부가 기능을 추가하여 몇 개의 불량을 피해갈 수 있도록 보강하였다. 그러나 완벽하게 모든 원인을 감지하는 것은 불가능하다. 따라서 본 시스템에서는 단방향 통신만을 구현하였고, 또한 이에 대한 에러를 줄이기 위한 노력으로 명령어 데이터 포맷을 Symbol Error를 최소화할 수 있는 Character로 선정되었다는 것이다.

3.3 메인-보드 Data Format

명령어 길이는 전체 시스템의 특성에 따라 달라질 수 있을 것이다. 1:1 통신의 경우 명령의 종류와는 거의 무관하게 명령의 형태를 간단하게 만들 수가 있다. 하지만, 본 시스템에서는 1개의 메인-보드와 10개의 스레이브-보드와의 통신을 수행하기 때문에 명령어의 종류는 그리 많지 않지만, 명령어의 길이는 1-바이트 Character 11개로 이루어진 배열이 필요하다 본 시스템이 고속의 통신 방식을 필요로 하지 않기 때문에 명령어의 길이는 본 시스템의 성능에 영향을 끼치지 않는다.

다음 그림 5는 PC에서 사용하는 명령어의 Data Format이다.

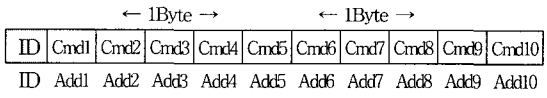
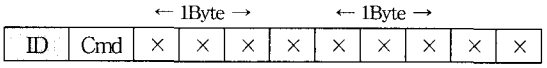


그림 5. PC와 메인-보드사이의 데이터 포맷
Fig. 5. Data Format between PC and Main-Board.

ⓐ ID가 'K' 일 때,

ID가 'K'이면 메인-보드 자체 처리 명령어이기 때문에 단지 1-바이트 명령어만 있으면 되므로 나머지 9-바이트는 필요가 없게 되므로 Don't Care term으로 처리한다.



ⓑ ID가 'T' 일 때,

ID가 'T'이면 스레이브-보드로의 명령이기 때문에 ID 부분만 제외하고 나머지 10-바이트는 유효한 데이터가 된다.

4. 메인-보드 명령어

표1에서 PC와 메인-보드와의 통신 명령에 대한 목록을 보여준다.

표 1. 메인-보드 명령어
Table 1. Main-Board Instruction.

Command Name	PC	Host	Code	Character	Description
Grade_first	➡		0x61	'a'	1등급 분류 명령
Grade_second	➡		0x62	'b'	2등급 분류 명령
Grade_third	➡		0x63	'c'	3등급 분류 명령
Grade_fourth	➡		0x64	'd'	4등급 분류 명령
Grade_fifth	➡		0x65	'e'	5등급 분류 명령
Grade_sixth	➡		0x66	'f'	6등급 분류 명령
Grade_seventh	➡		0x67	'g'	7등급 분류 명령
Grade_eighth	➡		0x68	'h'	8등급 분류 명령
Grade_ninth	➡		0x69	'i'	9등급 분류 명령
Grade_tenth	➡		0x6A	'j'	10등급 분류 명령
Core_receive	➡		0x52	'R'	코어 공급 받으라는 명령
Core_push	➡		0x50	'P'	코어 밀어 넣으라는 명령
Core_down	➡		0x44	'D'	코어 측정대 내리는 명령
Core_up	➡		0x55	'U'	코어 측정대 올리는 명령
Core_dump	➡		0x43	'C'	코어 쏟아 부으라는 명령
Reset	➡		0x53	'S'	pc에서 Main에 보내는 중지 명령
OK_sign	➡		0x4F	'O'	명령을 실행해도 좋다는 신호
Emergency	➡			'Stop'	긴급 상황 비상 스위치
Acknowledge	➡			'ACK'	잘 받았다는 신호
Done	➡			'End'	수신 명령 처리 완료 신호

5. 메인-보드 프로그램

메인-보드의 프로세서인 PIC16F873 내부 프로그램[1]은 시스템이 가동되면 모든 레지스터들이 초기화되고 각각의 I/O Pin들을 사용 목적에 맞게 설정한다. 그 다음으로 작업 지시 명령을 받을 때까지 대기상태를 유지한다. 그리고 인터럽트를 이용하여 명령을 수신하여, 수신 에러를 최소화하였고, 내부 수신 버퍼의 사이즈 크기를 적절히 조절하였다. 여기서 인터럽트에 의한 수신 방법에 대해 살펴보자. PIC16F873의 내부 수신 버퍼 사이즈는 8-비트로 1-바이트의 Character를 저장할 수가 있다. 그런데, 본 시스템에서 요구하는 버퍼 사이즈는 총 11-바이트의 Character를 저장하는 공간이므로, PIC 내부 메모리를 미리 11-바이트 준비해 두고, 인터럽트가 발생할 때마다 메모리에 저장하는 방법을 택하였다. 즉 메인-보드 Firmware에서 사용한 방법은 INT_RDA라는 것으로 CCS-Compile에서 수신 완료시에 발생하는 인터럽트이다. 따라서 PC측에서 11-바이트의 정보를 보내게 되면 메인-보드의 PIC 버퍼는 1-바이트 Character를 받을 때마다 INT_RDA INTERRUPT를 발생하게 된다. 그러면, 그때마다 준비된 메모리에 수신한 Character를 저장하였다가 11-바이트가 되면, 한번의 명령을 실행하는 것이다.

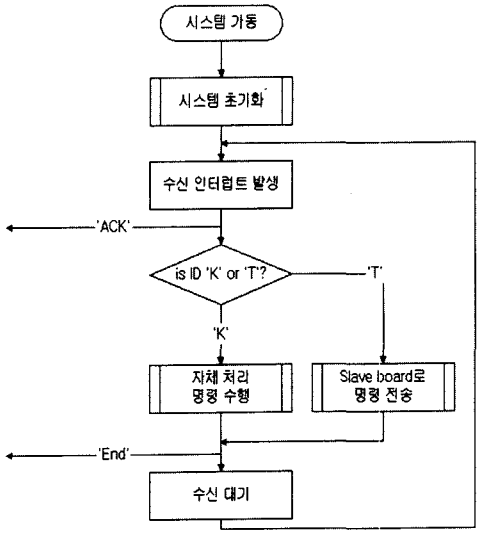


그림 6. 메인-보드 순서도
Fig. 6. Main-Board Flow Chart.

시스템이 가동되어 원점 보정이 완료된 상태에서 스레이브-보드는 스텝모터의 현재의 위치를 각각 기억하게 된다. 그런 다음 작업이 시작되어 명령을 수신하게 되면, 처음에 수신 받은 명령을 그대로 PC로 되돌려 보내게 된다. 다음에는 PC에서는 송신 데이터와 수신 데이터를 비교하여 같은 데이터이면, OK_sign을 다시 보내게 된다. 그러면 메인-보드에서는 다시 'ACK' 신호를 PC로 보내게 된다. 'ACK' 신호를 수신한 PC는 'End' 신호를 수신하기까지 Disable 상태를 유지하게 된다. 그러면 메인-보드에서는 Data Format의 ID를 판별하여 자체 처리 명령인지, 아니면 스레이브-보드로 송신하여야 하는 명령인지를 판단하게 된다.

표-1에서 음영 처리 된 부분의 자체 처리 명령이면 메인-보드는 스레이브-보드와는 상관없이 자체적으로 명령을 처리하고 다시 RS-232C Interface를 통하여 'End' 신호를 PC쪽으로 보내게 된다. 그러면 PC는 'End' 신호의 수신과 동시에 Enable 상태가 된다. 또한 판별한 Data Format의 ID가 스레이브-보드 명령이라면, 메인-보드는 수신된 데이터의 ID를 없애고, 새로운 Data Format으로 RS-485 Interface를 통하여 수신된 명령을 각각의 스레이브-보드로 동시에 전송하게 된다.

6. 스레이브-보드 Firmware

하나의 메인-보드에서 10개의 스레이브-보드에게 RS-485 통신으로 명령을 보낸다. 각각의 스레이브-보드에게 하나씩 명령을 보내는 것이 아니라, 동시에 모든 스레이브-보드의 명령을 RS-485 통신으로 보내고, 그러면 모든 스레이브-보드는 같은 명령어를 스신하게 된다. 다음에는 같은 명령어 중에서 각각의 스레이브-보드에 해당되는 명령어만을 미리 정해놓은 위치에서 각각의 명령을 수신한다. 스레이브-보드의 프로세서 PIC에서는 10-바이트의 Character Array를 명령어로 인식한다. 즉 메인-보드에서 수신한 명령어 중에서 ID 부분인 첫 1-바이트는 더 이상 필요가 없게 되었다. 따라서 스레이브-보드에서 사용하는 Data Format는 그림 7과 같다.

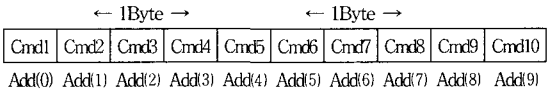


그림 7. 스레이브-보드 Data Format
Fig. 7. Slave-Board Data Format.

각 배열의 Index가 바로 어드레스이기 때문에 미리 1번부터 10번까지 정해진 스레이브-보드는 Arr(N-1)번째 데이터를 자신의 것으로 받아들이면 되는 것이다. 다시 말해서 모든 데이터를 받아서 자신에게 필요한 1-바이트의 데이터를 선택하여 실행한다. 이렇게 하면, 각 스레이브-보드에 대한 어드레싱 없이 명령을 줄 수 있고 또한 동시에 모든 스레이브-보드가 명령을 수신하여 동시에 실행할 수 있게 된다. 따라서 [One-To-Ten] 방식의 통신은 성공적으로 수행될 수 있는 것이다. 여기서 이러한 485-통신을 수행하는 데 있어서 송신하는 순간에 10개의 스레이브-보드가 수신할 수 있

표 2. 스레이브-보드 명령어
Table 2. Slave-Board Instruction.

Command Name	Host	Slave	Code	Character	Description
move_first	--->		0x61	'a'	1등급 분류 명령
move_second	--->		0x62	'b'	2등급 분류 명령
move_third	--->		0x63	'c'	3등급 분류 명령
move_fourth	--->		0x64	'd'	4등급 분류 명령
move_fifth	--->		0x65	'e'	5등급 분류 명령
move_sixth	--->		0x66	'f'	6등급 분류 명령
move_seventh	--->		0x67	'g'	7등급 분류 명령
move_eighth	--->		0x68	'h'	8등급 분류 명령
move_ninth	--->		0x69	'i'	9등급 분류 명령
move_tenth	--->		0x6A	'j'	10등급 분류 명령
Compensation	--->		0x53	'R'	코어 공급 받으라는 명령
Default	--->		0x75	'P'	코어 밀어 넣으라는 명령

도록 타이밍을 맞추어 프로그램하고 많은 수정과 테스트을 하여 개발하였다.

7. 스레이브-보드 프로그램

스레이브-보드의 프로세서인 PIC16F873의 내부 프로그램은 2개의 인터럽트 서비스 루틴(ISR)을 사용하고 있다. 하나는 메인-보드에서 보내는 데이터를 수신하는 인터럽트이고, 다른 하나는 원점을 보정하기 위해 Photo Sensor의 감지를 위한 인터럽트이다. 스레이브-보드의 인터럽트에 대해 좀더 자세히 알아보면, 먼저 데이터 수신 인터럽트인 INT_RDA라는 것은 메인-보드에서의 방식과 동일하다. 따라서 이에 부연 설명은 하지 않는다.

다음에는 Photo Sensor Detection을 위한 인터럽트에 대하여 살펴보자. PIC16F873은 내부적으로 13개의 인터럽트를 지원한다. INT_EXT Interrupt는 PIC16F873의 I/O Pin 중에서 B port의 첫 번째 Pin으로, 즉 Port_B0의 상태가 변화할 경우 일어나는 인터럽트이다.

다음은 프로그램 작동 순서에 대하여 논의한다. 시스템이 가동되면, 스레이브-보드는 원점 보정을 실행한다. 여기서 원점 보정이라 함은 코어 선별기에 있는 분류-그릇의 기준 위치를 말한다. 또한 장시간 작업이 진행되다 보면, 분류-그릇과 등급-구멍간의 간격 오차가 발생하는 것은 당연하다. 따라서 주기적 원점 보정은 필수적임을 알 수 있다. 이 때 쓰이는 인터럽트가 바로 INT_ EXT라는 것이다. 여기서 기준으로 지정된 등급은 아주 중요한 위치가 된다. 왜냐하면, 수신된 명령이 기준위치로 정해진 등급이라면 이는 분류-그릇의 이동 루틴을 거치지 않고, 원점 보정 루틴을 이용하여 원하는 결과를 얻을 수 있다는 것이다. 더 나아가 이렇게 함으로써 주기적 원점 보정의 역할도 충분히 할 수 있는 것이다. 따라서 기준 등급 설정은 매우 중요한 부분임을 알 수 있다. 다시 절차로 돌아가, 인터럽트에 의해 원점 보정(기준 위치로의 이동)이 완료되면 스레리브-보드는 명령 수신 대기모드로 전환을 하게 된다. 그래서 메인-보드에서 명령을 송신하게 되면, 스레이브-보드에서는 Buffer Size 10-바이트로 할당된 메모리가 인터럽트가 발생할 때마다 점점 채워지게 되는 것이다. 그래서 다 채워지면, 미리 정해진 스레이브-보드의 번호를 바탕으로 해당 명령을 판별하여 실행하게 되는 것이다. 그런 다음 다시 자동으로 수신 대기모드로 전환된다.

8. 원점 보정 Flowchart

다음은 원점 보정 함수를 호출하는 경우, 실행되는 과정에 대하여 논한다. 원점보정은 수시로 하여야 언제나 정확한 위치를 보장하여 줄 것이다. 원점 보정은 기본적으로 Sensor Detect에 의해 이루어진다. 그런데, 본 시스템에서는 주기적으로 기준 등급에 해당하는 등급 지시 명령이 수신된다는 가정을 전제로 하고 있다. 하지만, 실제 작업에 있어 기준 등급에 해당하는 등급 지시 명령이 주기적으로 안 내려올 수도 있는 것이다. 불행이도 원점에 해당되는 등급구멍을 오래 동안 거치지 않게 되면 문제점이 발생할 것이다 따라서 일정한 기간동안 원점에 해당되는 위치를 거치지 않게 되면 강제적으로 원점 등급 위치로 강제로 이동하도록 한다. 그림 8에서 볼 수 있듯이, 원점 보정함수(Compensate)의 호출이 N만큼 지나도 나오지 않으면, 자동으로 (Compensate) 함수를 호출하게 된다.

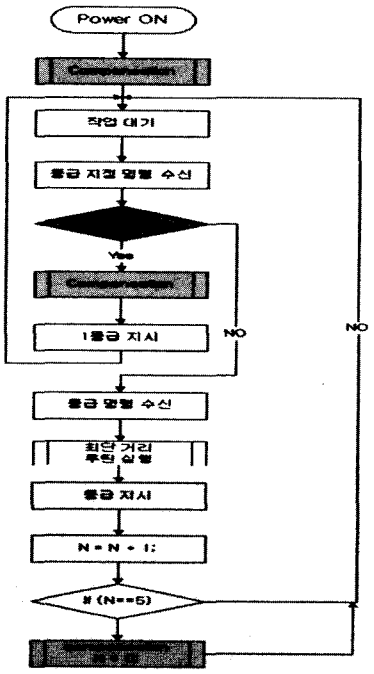


그림 8. 스레이브-보드 순서도
Fig. 8. Slave-Board Flow Chart.

9. PC 운영 프로그램

메뉴 구성은 등급설정, 수동점검, 작업, 통계 부분으로 나뉘어져 있다. 등급설정 메뉴를 다시 세부적으로 나누어, 등급조건, 인덕턴스 범위, 기준치 설정 등이 있

다. 등급조건은 기준치 설정에서 정해준 값에 따라 원하는 분류 조건을 설정할 수 있다. 인덕턴스 범위는 측정하려는 코어의 인덕턴스 값의 범위이다. 이것은 코어를 측정할 때 인가해 주는 Voltage와 Frequency를 자동으로 설정해 주는 역할을 한다. 따라서 코어의 값(L, Q)이 바뀔 때마다 일일이 Voltage와 Frequency를 설정해 주는 번거로움 없이 작업을 진행할 수 있는 것이다.

기준치 설정은 각 등급도 일정 간격을 가지고 있고 각 등급의 기준이 되는 값이다. 이 값에 따라 등급 조건의 오차값이 결정되게 된다. Lot이란 부분은 코어 선별기에서 분류된 후 저장되는 상자에 최대로 저장 할 수 있는 코어의 개수를 의미한다.

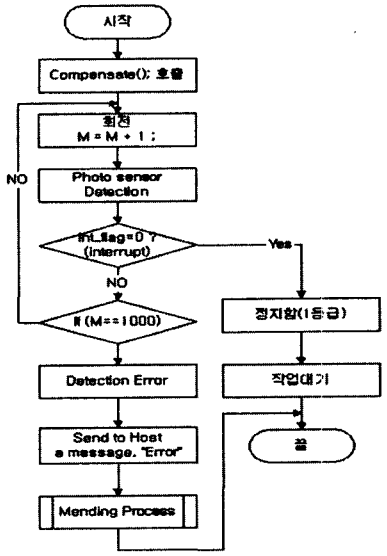


그림 9. 원점 보정 순서도
Fig. 9. Starting Point Compensation Flowchart.

그림 11은 등급 설정에 대한 프로그램 화면이다. 1등급에서 10등급까지 Q-값 및 L-값에 따라 나눌 수 있다. 업체에서 미리 준 값을 저장하여 넣는다. 수동점검 메뉴에는 그림 12에서 볼 수 있듯이 세부적으로 코어 공급기 제어, ATi 테스트, 모터 점검, 분류 현황의 4부분으로 되어 있다. 우선 모터 점검 부분은 좌측에 있는 번호가 바로 모터의 번호이며, 등급과 직결된다. 즉 해당 번호의 모터를 수동으로 위치를 컨트롤할 수 있는데, 모터의 동작 상태를 점검할 때 쓰인다. 다음으로 코어 공급기 제어 부분은 코어 공급기를 수동으로 조작할 수 있도록 여러 명령 버튼으로 구성을 만들었다. 해당 버튼을 클릭하면, 코어 공급기는 수동으로 조작이 가능하게 된다. 측정 장비 ATi 테스트는 ATi를 1번만 작동하여 정상 작동 여부를 점검하게 된다. ATi에 의해 1번 행해진 측정에 의해 분류된 10개의 코어의 분류 결과를 점검하여 실제의 등급과 일치하게 되면 시스템의 정상 작동 여부를 점검할 수 있게 된다. 그림 13에 보여준 작업 메뉴는 작업이 진행 중일 때 이 화면이 표시된다. 작업자가 항상 볼 수 있는 생산현황 메뉴로 현재까지 작업한 코어의 개수와 통계치, 모델명 등이 표시되며, 코어 등급 상자의 저장 상태를 시각적으로 볼 수 있도록 하였다. 10등급은 불량 혹은 측정하지 못한 코어를 지칭한다. 또한 현재까지의 등급별 생산 현황을 실시간으로 보여준다.

통계 부분은 [일통계], [주통계] 2개의 메뉴로 구성되어 있다. 통계에서 [일통계] 메뉴는 그 날 작업한 내용의 각종 통계치를 나타내며, [주통계] 메뉴는 일주일간 작업한 내용의 전체적인 통계치를 볼 수 있는 부분이다. 등급별로 1일간, 혹은 1주간의 생산 현황을 보여주고, 더욱이 등급별로 상대적인 점유율을 나타낼 수 있어 품질 관리할 수 있게 된다.

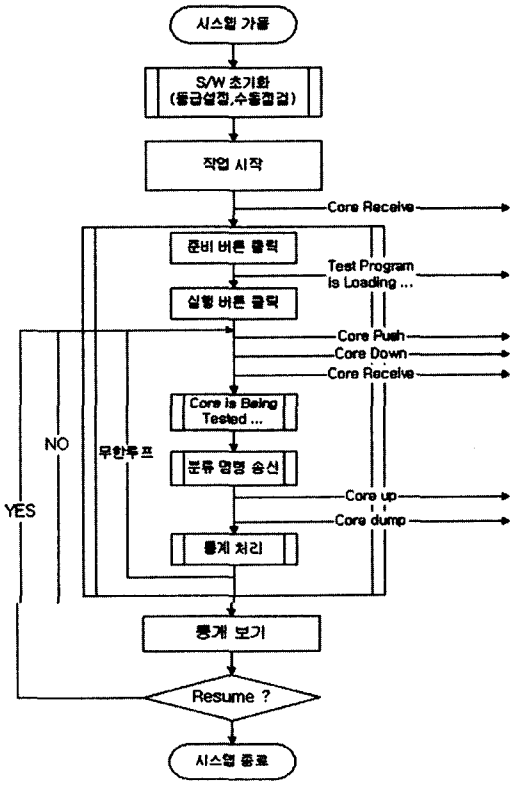


그림 10. PC 운영 프로그램 순서도
Fig. 10. PC Execution Program Flow Chart.

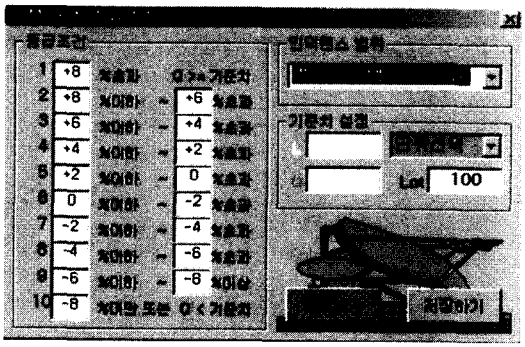


그림 11. 등급설정
Fig. 11. Grading Setting.

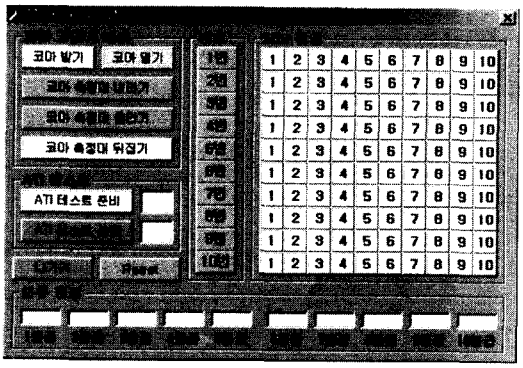


그림 12. 수동 점검
Fig. 12. Manual Test.

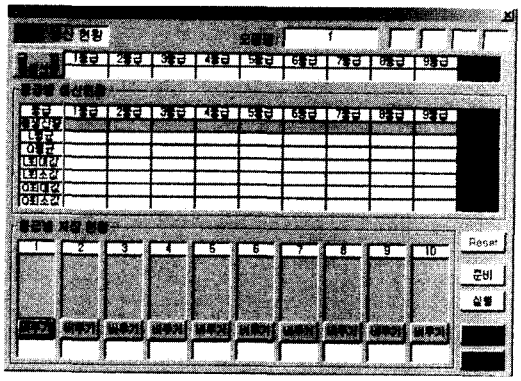


그림 13. 작업 메뉴
Fig. 13. Woring Menu.

Ⅳ. 실 험

먼저 1994년도에 제작된 선별기[4]를 참조하여 실험의 내용을 정하였으며, 에러 가능한 여러 상황을 반복 실험하면서, 이러한 에러를 없애거나 피해가는 방법 및 과정에 대하여 논의한다.

PC 응용 프로그램은 Visual Basic으로 작성되었으며, 코어 선별 작업의 진행을 시각적으로 알아 볼 수 있게 했으며, 실시간으로 데이터의 통계치가 작업 화면상에 표시되도록 하였다. 그리고 코어를 측정하는 Voltech ATi와는 RS-232C Interface(COM2)로 연결되었으며, 내부적으로는 ATi Server Program과 연결되어 ATi에서 측정되는 코어의 Inductance값과 Quality Factor값을 Visual Basic상에서 연산 처리할 수 있도록 하였다. 또한 동시에 RS-232C Interface(COM1)으로 메인-보드와 연결하여 실시간으로 연산 처리된 결과를 참조하여 메인-보드를 거쳐 각각의 스레이브-보드에게 명령을 송신할 수 있다. 다음은 이러한 상황에서 나타날 수 있는 에러의 발생원인 및 이에 대한 대책에 대하여 논의한다.

- Handling 에러

S/W에서는 ATi로부터 받은 데이터를 비교 연산하여 주기적으로 명령을 메인-보드로 송신하게 되어 있다. 여기에는 코어 공급기를 수동으로 조작하는 명령도 포함되어 있다. 즉 데이터를 비교 연산하고, 명령을 송신하는 과정에서, 선행 처리되어야 하는 명령이 있기 마련인데, 만약 선행 처리되지 않은 상태에서 다음 명령을 송신하게 되면, 전혀 다른 등급으로 판정하는 치명적인 오류가 나타날 수도 있다.

① 증상

ⓐ 모터가 이동하여 분류-그릇이 먼저 등급-구멍에 정확히 위치하기 전, 즉 전혀 움직이기 전에 코어가 분류-그릇 입구에 떨어 질 수 있다.

ⓑ 분류-그릇이 이동하는 동안에 코어가 떨어진다.

② 조치 : 이러한 오류가 발생하지 못하도록 많은 실험을 거쳐 프로그램을 개발하는 방법이 없음.

③ 대안 : 각 공정에서 Photo Sensor에 의한 감지로 Timing을 조절하여 코어가 정상으로 흘러가는 과정을 감지하여 Event 발생시에만 해당 명령을 실행하게 하는 것이다.

- 송신 에러

S/W에서 명령을 메인-보드로 보내면, 메인-보드에서는 잘 받았다는 'ACK' 신호를 PC 역 전송하게 된다. 하지만, 메인-보드에서 어떤 오류가 생겼다면, 'ACK' 신호를 PC로 보낼 수 없을 것이다.

① 증상 : S/W 상에서 버튼 동작을 할 수 없게 된다. 버튼을 눌러도 작동하지 않는다.

② 조치 : 해당 S/W창을 닫고 다시 실행하고, 코어 선

별기 우측 상단의 붉은 버튼을 누른다. 이러한 붉은 버튼이 눌려지면, 모든 시스템이 리셋 작동을 시작하여 메인-보드가 정상으로 작동한다.

- 스레이브-보드 에러

스레이브-보드는 메인-보드와 RS-485 Interface로 연결되어 명령을 수신하는 기능과 모터 드라이버를 제어하는 기능이 있다. 메인-보드로부터 명령을 수신하는 방식은 Interrupt를 이용한 것으로 첫 번째 명령을 받고 다 수행하기도 전에 두 번째 명령이 들어 왔을 때도 에러 없이 잘 동작하도록 Firmware를 설계하였다. 또한 스텝모터의 반복적인 회전으로 인한 등급-그룻의 출구와 등급-구멍과의 불일치에 대한 주기적 보상으로 Photo sensor를 부착하여 특정 등급에 대한 명령을 수신하였을 때에는 센서 감지에 의한 Interrupt를 이용하여 정확한 원점보상이 주기적으로 수행되도록 설계하였다. 이러한 경우, 스레이브-보드에서 발생할 수 있는 에러에 대하여 분석해 보자.

- 분류-그룻과 등급-구멍 불일치

스텝-모터에 매달려 있는 핀과 광센서가 만나면 원점 보상이 자동으로 수행된다. 만약 이러한 원점 보상이 되지 않는 위치로만 장기간 이동하게 되면 원점보상이 이루어지지 않게 되어 분류그룻과 등급-구멍사이에는 불일치 현상이 나타날 수 있을 것이다. 그리고 이러한 상황이 발생하지 못하도록 하여야 한다.

① 증상 : 분류그룻과 해당 등급-구멍이 어긋나 있다.

② 조치 : 일정한 기간 동안 원점 보상이 안 되면 강제적으로 원점 보장이 되도록 한다.

- 광센서 감지 에러

원점 위치 보정용 광센서에 의한 Interrupt를 이용하여 원점을 보상하게 된다. 만약 일정 기간이상 이러한 원점 보상이 안 나타나면 에러로 취급한다.

① 증상 : 광센서의 고장이나, Pin의 고장으로 분류-그룻이 멈추지 않고, 계속 회전을 한다.

② 조치 : 일정한 기간동안 발생이 안 되면 PC 운용 화면에 표시하여 사용자로 하여금 인지하도록 한다.

- 모터 고장 에러

스텝 모터는 펄스의 개수로서 모터의 회전을 조정할 수 있는 장치이며, 여기서 쓰인 모터는 5상 스텝 모터로써 정밀도가 매우 크다. 또한 각 등급에 맞게 모터의 펄스 수가 이미 조정이 되어 있는 상태이므로, 분류-그룻과 해당 등급-구멍사이가 정확하게 일치하지 않으면, 모터의 이상으로 볼 수 있을 것이다.

① 증상

ⓐ 위치 보정 후에도 계속 분류-그룻과 등급-구멍이 정확하게 일치되어 있지 않고 약간 어긋나 있다.

ⓑ 모터에서 소리가 나며 탈조 현상을 일으킨다.

② 조치 : 모터를 교체하여야 한다.

- 명령어 중복 수신 에러

통신을 하다 보면 명령어를 중복으로 받아들이는 경우가 있을 것이다. 이런 경우를 대비하여 스레이브-보드에서는 Interrupt를 이용한 수신 방법을 채택하고 있어 어느 정도의 중복을 허용하고 있다. 하지만, PIC STACK size에도 한계가 있기 때문에 100% 에러에 대한 해결을 보장할 수 없다.

① 증상 : 잘 작동하던 분류-그룻이 어느 순간 명령 수행을 하지 않고, 멈춰 있다.

② 조치 : Reset 버튼을 누른다.

- 보드 고장

스레이브-보드 기판이, 작업 여건상 철분 가루의 누적으로 서로 연결될 수도 있을 것이다. 이런 에러는 향후 반드시 예상될 것이다.

① 증상 : 임의의 모터가 오동작을 하거나, 여러 다른 조치를 취하여도 동작하지 않는다.

② 조치

ⓐ 작업을 멈추고, 유사 오류에 대한 조치를 취해 본다.

ⓑ 그래도 안 되면, 해당 스레이브-보드 기판에 쌓인 먼지를 청소해 준다.

- 작업자 실수

작업 중에 작업자의 실수로 스텝모터위에 얹어져 있는 분류-그룻을 건드려 이미 원점 보상이 되어 있는 상태에서 벗어나게 된다.

① 증상 : 이러한 경우에 코어가 제대로 흘러 갈 수 없게 된다.

② 조치 : 작업 멈춤 상태에서 Reset 버튼을 클릭하면, 다시 원점 보정이 이루어진다. 그런 후 '실행' 버튼을 누른다. 그러면 작업이 계속 될 것이다.

V. 결 론

코어가 생산되는 속도보다 빠르게 선별할 수 있는 시스템을 개발하는 방향은 한 개의 코어를 10등급으로 분류하고, 한번에 10개의 코어를 선별하도록 설계한다. 이러한 경우에 코어를 공급기에서 10개씩 한 줄로 공

급하는 메커니즘을 거쳐 측정-slot에서 테스트 장비가 등급을 측정한 후, 10개의 측정된 코어를 해당 상자에 저장시키기 위하여 10개의 스텝모터를 해당 등급위치에 이동시킨 다음, 동시에 측정된 10개의 코어를 분류-그릇에 쏟아 부으면 종료된다. 이러한 시스템을 개발하는 데 있어서 하드웨어 구조 및 소프트웨어를 개발함에 있어, 초점은 10개의 스텝모터가 작동되고, 각각의 스텝모터는 정확히 10등급으로 나누어진 등급-구멍에 위치하여야 하고, 더욱이 코어가 공급기에서 측정 메커니즘을 거쳐 스텝모터위에 놓여진 분류-그릇을 거쳐 100개의 호스가 총길이가 무려 100m되는 메커니즘으로 구성되어 있다. 이러한 구성물은 고장이 나기 쉬운 것은 당연하다. 이러한 근본적으로 취약한 환경에서 고장이 발생하지 않도록 설계하는 것은 불가능할 것이다. 문제는 고장이 발생하는 경우에 얼마나 하드웨어적으로 소프트웨어를 어떻게 감지하고, 대처하는 측면에서 개발하였다.

소프트웨어 부분을 2가지로 분류하여 보면, 먼저 PIC16F873을 사용한 메인-보드, 스레이브-보드의 Firmware와 PC 운영 프로그램을 들 수 있으며, 프로그램에 사용된 언어를 살펴보면, PIC16F873의 경우 C-Language를 사용하였고, PC 운영 프로그램은 BASIC Language를 각각 사용하였다. PIC16F873 Firmware의 구성은 메인-보드의 경우, 자체 명령 처리부와 정보 전달부로 나뉘고, 스레이브-보드의 경우, 인터럽트에 의한 원점 보정 루틴과, 등급 지시 명령 수행부로 나눌 수 있다.

메인-보드에 있어서, 자체 명령 처리부는 PC로부터 수신한 데이터의 ID가 0x41(Character 'K')일 경우에 수행되는 부분으로, 코어 공급기의 수동 제어 단자와 연결되어 있는 보드의 Relay를 제어하게 되어 있다. 그리고 PC로부터 수신한 데이터의 ID가 0x54(Character 'T')일 경우는 ID 1-바이트를 제외한 나머지 10-바이트를 그대로 스레이브-보드로 중계 송신하게 되어 있다.

또한, 스레이브-보드에 있어서, 인터럽트에 의한 원점 보정 루틴은 PIC16F873 내부적으로 지원되는 INT_EXT를 응용하여 Photo Sensor Detect을 이용한 것이다. 그리고 등급 지정 명령 수행부는 5상 스텝 모터의 펄스 입력 응답을 이용한 것이다. 즉, 1 Pulse 에 0.72° 를 이동하므로, 원판에 균일한 간격으로 10개의 구멍이 나 있는 코어 선별기에서는 0.72° × 50 = 36° 인 것을 이용, 50 Pulse를 기본으로 출력하는 함수를 만들어 사용하였다.

PC 운영 프로그램은, 전체 시스템의 운영을 위한 프로그램이며, PIC16F873과의 통신, 측정 데이터의 비교 연산, 통계 계산, Database 등의 역할을 수행하게 된다. 또한 작업자가 쉽게 이해하고 접근할 수 있는 메뉴 구성과 사용 방법을 채택한 윈도우용 Application 응용프로그램이다.

참 고 문 헌

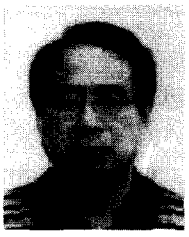
[1] Data Sheet, PIC16F8xxx, Microchip Inc., may, 2001.

[2] Manual, ATi Inductance Measurement Tool Tester, Voltek, march 2000.

[3] User Manual, C&C Compiler.

[4] 산업자원부, 최종보고서, 부품자동선별기 개발에 관한 연구, 1994

저 자 소 개



朴 寅 圭(正會員)

1972년 : 서울대학교 전자공학과 졸업. 1982년 : The Ohio State University EE 석사 졸업. 1986년 : Purdue University EE PhD. 1986~1987년 : University of Houston 조교수. 1987~1988년 : 삼성전자 종합 연구소 제 5연구실장. 1988년~현재 : 홍익대학교 전자전기공학부. <주관심분야 : 컴퓨터 구조, 멀티미디어 시스템 등임>