

PIC를 이용한 엔코더 계측방법 개선

°김종대*, 송덕현**, 이보희**, 김진걸*

* 인하대학교 자동화공학과 ** 세명대학교 전기공학과

The improvement of measuring method for the Optical Encoder using PIC.

°J.T. Kim,* D.H. Song** B.H. Lee** J.G. Kim*

* Dept. of Industrial Automation, Inha Univ.

** Dept. of Electrical Engineering Semyung Univ.

abstract : This paper proposed a new method for the measurement of optical encoder. So far several methods were used for the encoder measurements, there were noise problems and needed many space to realize it. Specially, it is more serious under the multi-motor system. In this paper, we adapted the PIC microcontroller and replaced the TTL logics with the PIC software. Therefore, the effects of noise can be reduced, and we can realize the measuring method for the optical encoder under multi-motor system within one millisecond time base.

해능을 높여 주는 체배 회로와 검출된 각 에지마다 작은 펄스로 환산하기 위한 테이블이 기록된 PROM. 그리고 체배된 펄스들 모터 회전 방향에 따라 UP/DOWN 카운트 할 수 있는 카운터를 연결하여 만든 방법[2]이 있다. 또한 TTL 로직과 카운터(74HC193)를 이용하여 계수하는 완전한 하드웨어를 이용한 방법도 있다. 그런데 여기서 하드웨어만으로 이루어 지는 종래의 방법에 있어 몇가지 문제점들을 지적할 수 있는데, 우선 다축 구동시에 아날로그 노이즈의 영향에 매우 민감하다는 것과 각 모터마다 각각의 회로를 필요로 하므로 부품수가 많아지게 되고 이는 소형화에 적합하지 않다는 점등을 들 수 있다.

1. 서론

많은 시스템에 있어서 모터를 제어하기 위해 회전속도와 회전위치에 대한 정보를 알아내는 일은 매우 중요한 일이며, 이를 검출하기 위한 센서로써 회전식 엔코더가 가장 많이 사용되고 있다. 회전식 엔코더는 절대위치형과 자연중가형으로 나누어 지며, 이러한 광학식 엔코더는 회전하면서 그 때마다 펄스를 발생하게 되고, 이 펄스를 통하여 모터에 대한 정보를 알아낼 수 있다. 회전속도를 검출하는 방식으로 주파수/전압변환 방식과 주기측정 방식등이 사용되고 있으며 특히 펄스카운트 방식은 회전속도 뿐만 아니라 회전위치 검출 방법으로도 많이 사용되고 있다[1].

이러한 펄스카운트를 위해서 몇가지 방법이 사용되고 있는데, 그 중 한가지는 엔코더 신호로부터 보다 정밀한 정보를 얻을 수 있도록 분

본 논문에서는 PIC를 이용하여 펄스카운트에 있어서 필요한 회로들을 프로그래밍화 함으로써 성능 향상과 동시에 소형화를 가져오는 새로운 방법을 제안하고자 한다.

2. 시스템 설계

시스템을 설계하고 제작할 때 범용 IC로 만들기는 너무 큰 경우에 마이컴을 사용하여 이를 간단히 실현할 수 있다. 현재 많이 사용되고 있는 고성능 CPU를 사용하면 좋은 시스템을 만들 수 있을 것이다. 그러나 필요한 I/O핀수를 생각할 때 이런 원 보드 마이컴을 사용한 경우 공간 활용면이나 경제적인 면에서 비 효율적일 수 있다. 이런 경우에 가격이 저렴하고 공간을 적게 차지하며 프로그램이 용이한 Microchips사의 Micro Controller인 PIC16CXX 시리즈를 사용하면 바람직하다. PIC시리즈 중에는 PIC16C54 및 PIC16C84가 있으나, PIC16C54 및

84의 경우는 입출력 단자가 12핀에 불과하여 같은 구성을 가진 PIC16C55를 사용 하였다. PIC16C55의 대략적 사양을 보면 다음과 같다.[3]

- 모든 명령어는 한사이클(200ns)로 구성.
- 동작 속도는 20Mhz
- 12비트 명령어, 8비트 데이터
- 8비트 타이머(카운터) 내장(8비트 프리스케일 리내장)
- 파워온 리셋(Power On Reset)
- 위치독 타이머 내장
- 내부 RAM 25Byte, EPROM 512Byte
- 입출력 20 핀

본 논문에서는 엔코더 카운트의 모든 과정을 소프트웨어로 처리했을 뿐 아니라 PIC16C55 하나 당 세 개의 엔코더를 처리하여 기존의 엔코더 각각에 하나씩의 엔코더 회로를 연결 시킬 때보다 몇배의 효율을 기대할 수 있었다. 우선 연결 사항을 살펴보면 다음과 같다. 20개의 입출력 단자 중에 6개의 단자는 3개의 모터로부터 엔코더 신호 A상과 B상을 받아들이는 데 사용된다. 그리고 나머지 14개 단자중 12개 단자는 주 제어 프로세서에 COUNT 정보를 넘겨 주는 데 사용되고, 남은 2개 단자는 주 프로세서에 모터 식별에 대한 정보를 넘겨 준다. 전체 회로도도 다음과 같다.

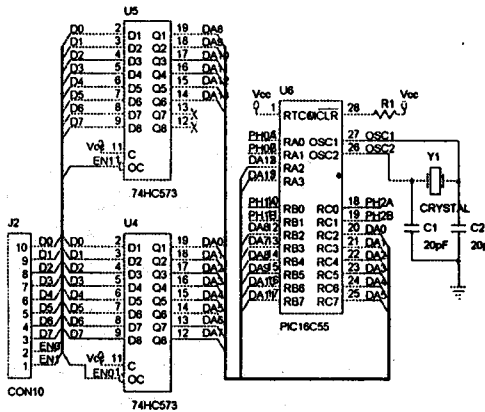


그림 1. 전체 엔코더 회로도

그림1에서 볼 수 있듯이 3개의 엔코더로부터 RA0, RA1, RB0, RB1, RC0, RC1으로 펄스가 입력된다. 각 포트의 하위 비트로 입력단을 위

치 시킨 것은, 후에 데이터 처리를 위해 쉬프트 시켜야 하는 번거로움을 없애기 위해서이다. RA2, RA3 단자는 주 프로세서로 데이터가 전해질 때, 몇번째 엔코더인지에 대한 정보를 실어 보낸다. 이때 두 개 단자 모두 1인 경우에는 busy상태임을 나타내어 잘못된 값을 읽어가는 것을 막았다. 또한 카운트에 대한 정보는 래치를 통해 주 프로세서로 전달 되어지는데 RB포트와 RC포트의 12개 단자를 이용하여 4096까지의 카운트가 가능해 지며, 바로 전의 카운트 값과의 차와 비교하여 2048 이상의 차이가 발생하면 역 방향으로 회전하였음을 알 수 있다. 프로그램의 전체 구성은 그림2와 같다.

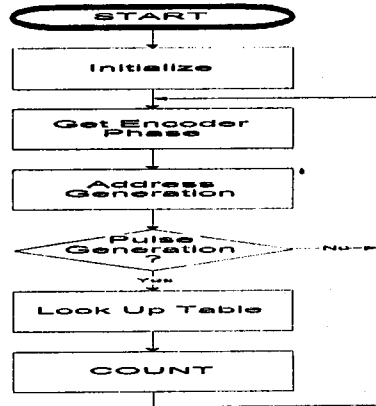


그림 2. 프로그램 플로차트

우선 PIC의 각 포트의 입출력 설정 및 초기화가 이루어진다. 이 후 엔코더 펄스 처리 과정에 앞서서 인식 비트에 BUSY 신호를 보내어서 처리 과정 중에 잘못된 값을 읽어가지 않도록 하였다. 엔코더로부터 입력된 A상과 B상으로 딜레이 된 펄스와 함께 어드레스를 설정하게 되고 테이블의 어드레스 값과 비교하여 방향을 판단하게 되며 UP/DOWN 카운트를 실행하게 된다. 이 때 입력된 상과 딜레이 된 상의 XOR 비교에서 펄스가 발생할 때만 카운트가 되게 하므로써 모터 정지 시에는 카운트가 무추도록 하였다. 이러한 일련의 과정을 거쳐 펄스 카운트 값을 엔코더 인식 비트와 함께 래치를 통해 주 제어 프로세서로 보내어 준다. 주 프로세서에서는 받아들인 카운트 정보를 통해 엔코더의 회전각과 방향을 알아낼 수 있다.

3. 실험 및 결과

본 논문에서는 앞에서 제시한 바와 같이PIC를 이용한 엔코더 계측을 구현하기 위하여 다음과 같이 시스템을 구성하였다.

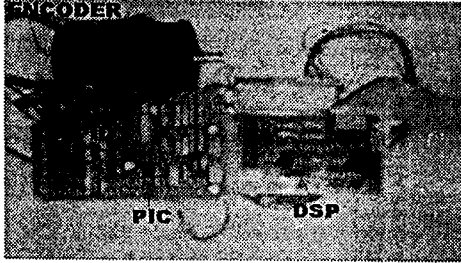


그림 3. 전체 외관

위의 그림에서 엔코더는 Tamakawa사의 DC모터에 장착된 1회전당 1000펄스를 발생시키는 인크리멘탈 방식의 엔코더를 사용했으며, 엔코더의 A상과 B상의 펄스는 PIC16C55로 연결되어 있다. 실험 환경에서 PIC는 DC모터 3개의 엔코더 정보를 처리할 수 있게 하였고 모든 처리 로직은 프로그램으로 구현하였다.

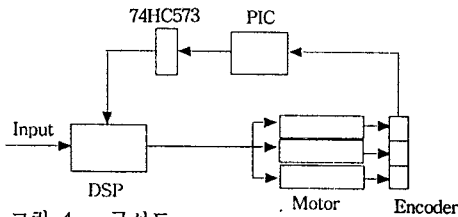


그림 4. 구성도

입력되어진 엔코더 펄스는 PIC내에서 프로그램으로 체제로직과 카운터로직을 만들고 카운트 되어진 값은 외부 장착된 TTL(74HC573)을 통해 모터 제어 로직이 있는 주 제어 프로세서(DSP)로 전달되어 진다. 주프로세서는 1ms주기로 PIC에서 송출된 3개의 엔코더 정보를 받아 관리하고 모터 제어를 실행한다. 이 때 PIC에서 계수하는 카운터 값은 자체의 워드길이 및 DSP와의 인터페이스 문제로 12비트에 해당하는 값만큼만 관리할수 있기 때문에 최소 12us보다 빠른 속도로 카운터 값을 읽어야 하고,본 실험에서는 모터의 제어 주기인 1ms로 설정하였다. PIC프로그램은 구현이 용이한 C언어(MPLABC)를 이용하였으며, 출력된 실행 파일은 Pro-Engine을 이용하여 PIC에 프로그래밍하였다. 여기서, 알고리즘 수행 시간과 처리 가능한 모터 rpm의 관계는 알고리즘 수행 시간이 엔코더 펄스 주기보다 빨라야하므로 아래식과 같이 나타낼 수 있고,

$$T_p \leq T_c/n$$

- T_p : PIC 알고리즘 수행 시간(s)
- T_c : 엔코더 펄스 주기(s)
- n : 체배수

측정 가능한 회전수를 구하면,

$$N = 60 \times P_c / Q_e = 60 / (T_e \cdot Q_e) \leq 60 / (n \cdot T_p \cdot Q_e)$$

- P_c : 1초당 엔코더 펄스 수
- Q_e : 1회전당 엔코더 펄스 수
- N : 분당 회전 수 (rpm)

따라서, 측정가능한 최대 rpm은 n 과 T_p 와 Q_e 에 반비례 함을 알 수 있다. 그러나 n 과 Q_e 값은 정해져 있으므로, 보다 고속의 모터를 측정하기 위해선 T_p 즉, PIC 알고리즘 수행 시간을 가능한 줄여야 한다는 것을 알 수 있다. 본 실험에서는 1체배, 회전당 1000펄스 엔코더를 사용하고 알고리즘 수행 시간이 12us로 5000rpm까지의 측정이 가능하다.

이러한 방식의 사용으로 정보를 모터 제어시 필요한 1ms 이내로 구현할 수 있고, 또한 다수의 모터 구동시 엔코더를 계수하는데 많은 하드웨어 소자를 사용함으로써 인한 노이즈의 영향 및 가격 상승의 요인을 줄이는 데 효율이 있으리라 기대된다.

4. 결론

본 논문에서는 기존의 엔코더 계수시 일어나는 하드웨어의 증가 및 노이즈 영향을 줄이기 위해 마이크로프로세서인 PIC16C55를 이용하여 소프트웨어로 계수하는 방법을 제시 하였고 그 유용성을 증명하였다. 이러한 기술은 모터를 구동하는 분야, 특히 다수의 모터를 이용하는 분야인 로봇공학 및 공장자동화 분야에서 활용이 기대된다. 향후 PIC의 계수 범위 및 속도 향상을 도모하여 좀더 넓은 범위의 엔코더 계수에의 응용에 대한 연구가 필요하다.

[참고 문헌]

- [1] 아경산업 자동화 연구소, "서보모터 제어 이론과 실습", 성안당, pp39-49
- [2] 이보희, "이족보행로봇 IWR의 걸음세 구현", 인하대 박사 논문, pp56-58, 1996
- [3] Microchip, PIC16/17 Microcontroller Data Book, pp2(5)-2(202), 1995