

For the world best IT education

시발(始發) 시리즈

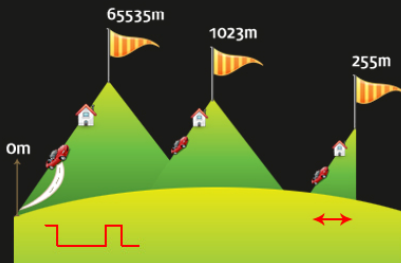


모델링에 기반한

임베디드 로봇 제어(하)

<ATmega128>

서동진·김채승 공저



난해하고 어려운 마이컴의 정복 방법 제시!

1. 포트와 핀을 쉽게 받아들이자! ⇨ 대한민국의 항구와 부두
2. 플립플롭으로 만들어진 레지스터의 정체와 비트의 설정은 ⇨ CodeWizardAVR과 칼라 교재로
3. 타이머카운터에 대한 이해 ⇨ 다양한 높이의 산 등산(관리사무소, 휴게소, 정상)을 비유하여..

여러분은 최단 시간에 최소 반복 학습으로 마이컴 제어에 필요한 주기(Period)와 PWM 생성에 자신감을 갖게 됩니다.

SUSONG 수송미디어

www.geo1004.com

목 차 (상)

□ 제1장: 마이컴(ATmega128) 개요

- 1절: 특징 및 구조
- 2절: 레지스터
- 3절: 메모리
- 4절: 시스템 클럭
- 5절: 리셋 및 슬립 모드
- 6절: Hexa 파일과 구동
- 7절: 실습

□ 제2장: 모델링

- 1절: 마이컴 이해 방법
- 2절: 등장인물과 활동공간
- 3절: 개발 기능 정의
- 4절: 로봇 회로 및 구조
- 5절: LCD 이해
- 6절: 주석문과 변수명 정의
- 7절: 실습

□ 제3장: 입출력 제어

- 1절: 제어 용어 이해
- 2절: I/O 포트 이해
- 3절: I/O 포트 추가 기능
- 4절: 레지스터 정의
- 5절: LCD 제어 기초
- 6절: 쉬프트 레지스터 활용
- 7절: 실습

□ 제4장: 인터럽트

- 1절: 인터럽트 개요
- 2절: 인터럽트 벡터
- 3절: MCU 제어 레지스터
- 4절: 인터럽트 제어 레지스터
- 5절: 상태 레지스터
- 6절: 마이컴과 플립플롭회로
- 7절: 실습

□ 제5장: 타이머카운터

- 1절: 타이머카운터 개요
- 2절: 8비트 TC 레지스터
- 3절: 8비트 TC 동작과 출력
- 4절: 16비트 TC 레지스터
- 5절: 16비트 TC 동작과 출력
- 6절: 위치독 타이머
- 7절: 실습

목 차 (하)

□ 제6장: 로봇 모터 제어

- 1절: 모터 일반
- 2절: DC 모터 제어
- 3절: 서보 모터 제어
- 4절: 스텝핑 모터 제어
- 5절: 자율 주행 로봇
- 6절: PID 제어
- 7절: 실습

□ 제7장: 리모컨 활용

- 1절: 측정기 사용법
- 2절: 오실로스코프 활용
- 3절: 적외선 수신기
- 4절: 삼성 리모컨 코드 이해
- 5절: 코드 수신 C 문법
- 6절: 리모컨 활용
- 7절: 실습

□ 제8장: 통신

- 1절: UART 개요
- 2절: UART 레지스터
- 3절: 송신과 수신
- 4절: MFC 프로젝트 생성
- 5절: MFC 화면 설계
- 6절: MFC 제어 구현
- 7절: 실습

□ 제9장: A/C변환기

- 1절: ADC 이해
- 2절: 관련 레지스터
- 3절: 변환 결과
- 4절: 온도 센서
- 5절: 거리 센서
- 6절: 아날로그 비교기
- 7절: 실습

□ 제10장: LCD제어(EEPROM)

- 1절: LCD 문자 표현
- 2절: LCD 제어 레지스터
- 3절: LCD 제어 함수
- 4절: LCD 표준 출력
- 5절: EEPROM 제어
- 6절: EEPROM 읽기 쓰기
- 7절: 실습

□ 부록

- 1절: Lock 비트 설정
- 2절: Fuse 비트 설정
- 3절: 한글 LCD 사용법
- 4절: 영문 찾아보기
- 5절: 한글 찾아보기
- 6절: 레지스터 목록

1 마이컴 이해 방법

4 비유 적용

③ 타이머/카운터 비유

- TOP & (㉠) value: 다양한 높이의 산
- TCNT register: 현재 등산 높이값(정수)
- up count: 등산
- down count: 하산
- OCR & OVF interrupt: 대피소(산장)
- TCNT initial value: 차량 이동(관리사무소)
- (㉡) : 주사기 약의 농도, 제어를 위한 적정 주기를 생성

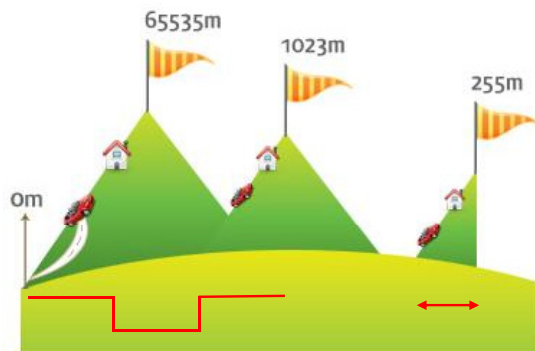


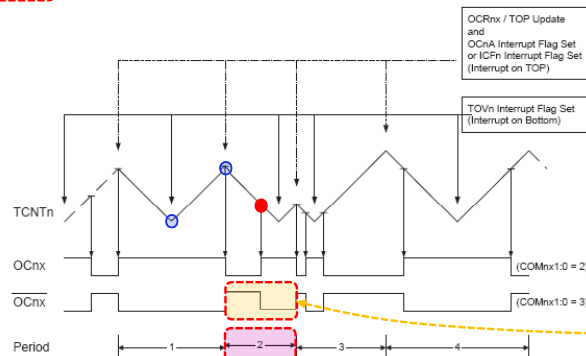
Figure 53.1 Phase Correct PWM Mode, Timing Diagram

The Timer/Counter Overflow Flag (TOVn) is set each time the counter reaches BOTTOM. When either OCRnA or ICRn is used for defining the TOP value, the OCnA or ICFn flag is set accordingly at the same timer clock cycle as the OCRnx Registers are updated with the double buffer value (at TOP). The interrupt flags can be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the compare registers. If the TOP value is lower than any of the compare registers, a compare match will never occur between the TCNTn and the OCRnx. Note that when using fixed TOP values, the unused bits are masked to zero when any of the OCRnx Registers are written. As the third period shown in Figure 53 illustrates, changing the TOP actively while the Timer/Counter is running in the phase correct mode can result in an unsymmetrical output. The reason for this can be found in the time of update of the OCRnx Register. Since the OCRnx update occurs at TOP, the PWM period starts and ends at TOP. This implies that the length of the falling slope is determined by the previous TOP value, while the length of the rising slope is determined by the new TOP value. When these two values differ the two slopes of the period will differ in length. The difference in length gives the unsymmetrical result on the output.

It is recommended to use the phase and frequency correct mode instead of the phase correct mode when changing the TOP value while the Timer/Counter is running. When using a static TOP value there are practically no differences between the two modes of operation.

In phase correct PWM mode, the compare units allow generation of PWM waveforms on the OCnx pins. Setting the COMnx1:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COMnx1:0 to 3 (See Table 60 on page 134). The actual OCnx value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OCnx). The PWM waveform is generated by setting (or clearing) the OCnx Register at the compare match between OCRnx and TCNTn when the counter increments, and clearing (or setting) the OCnx Register at compare match between OCRnx and TCNTn when



$$f_{OCnxPCPWM} = \frac{f_{clk_I/O}}{2 \cdot N \cdot TOP}$$



이해

마이컴에서 타이머카운터 이해의 중요성

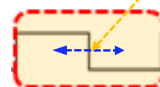
1. 마이컴 제어의 핵심

2. Figure 53에서와 같이 타이머 카운터 원리를 이해하여 제어 목적을 달성하기 위한 레지스터 설정 지식 필요

- DC 또는 Servo 모터 제어
- 리모컨 신호 수신
- 디지털 시계 생성 등

3. 마이컴 제어를 위한 필수 이해 사항

- period: 주기(시간)로 10ms는 0.01sec
- PWM: 전체 주기에서 High 비율 크기 (듀티비, Duty Ratio)



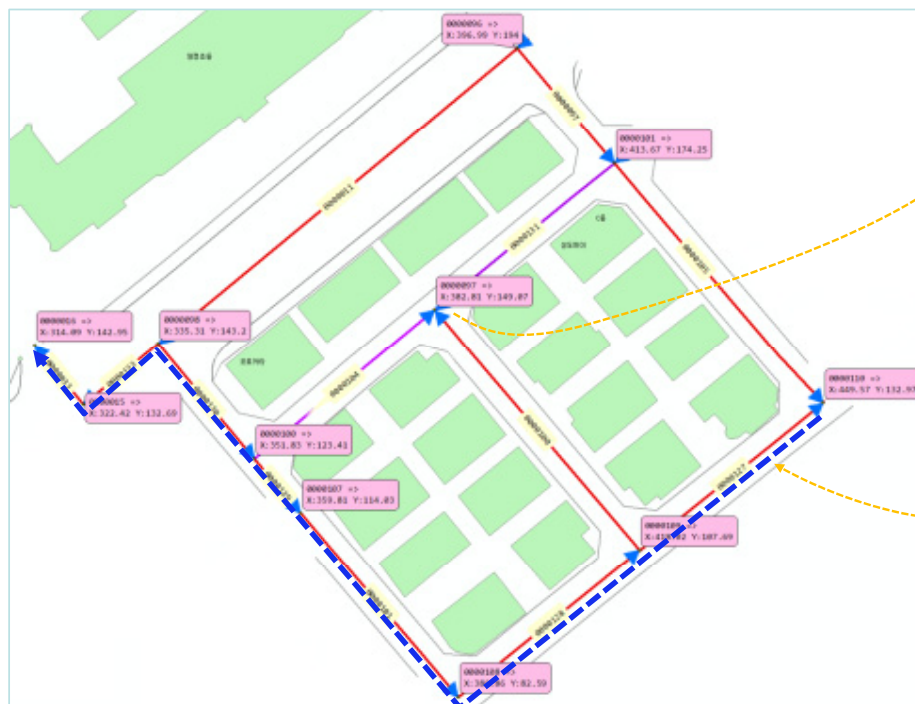
출처: ATmega128 Datasheet(2467R-AVR-06/08), p. 129

3 개발 기능 정의

2 프로젝트 정의

① 경로 주행 로봇 개요

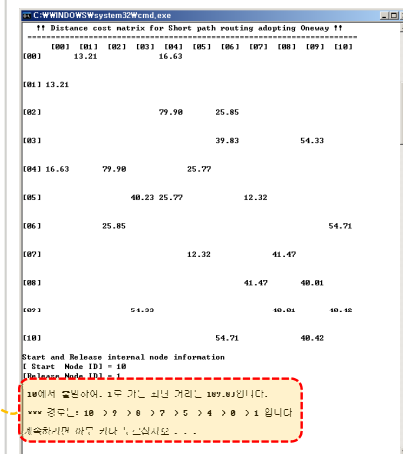
- 도로 주행: 출발점에서 도착점까지
- 도로망 구조: 주행 직선 도로, 교차로(사거리, 좌 삼거리, 우 삼거리, (㉠))
- 주행 중 승객 감지: 로봇 전방 10cm~70cm 이내의 승객(충돌체) 검색
- 정보 수집: 출발 시간, 도착 시간, 교차로 마다 온도, 산소 및 인체감지 정보



참고

로봇 주행 경로 계산

C로재의 노드 및 링크 정보와 최단경로 계산 구현 내용을 아래의 사례와 같이 활용하여 프로젝트에 적용합니다.



3 개발 기능 정의

3 교재 프로젝트

④ 각 장별 실습 내용

| 분류 | 기능명 | 대표 기능(단위 함수) | 각 장의 7절. 실습 |
|------|----------|---|-------------|
| 템플릿 | 단위 함수 | 프로젝트 구현을 위한 단위 함수 템플릿 | 2장 |
| 기본기능 | 디지털 시계 | delay() 함수를 이용한 시간, 분, 초 알고리즘 구현 인체 감지 센서 정보 확인 | 3장 |
| | 메뉴 생성 | 순서도(flowchart)를 활용한 버튼 메뉴 만들기 사용자 시간 설정: 인터럽트 및 폴링 방식 | 4장 |
| | 타이머 설정 | 32.768KHz에 의한 1초 생성으로 디지털 시계 완성 | 5장 |
| | 로봇모터제어 | 도로 구조와 충돌체를 감안한 PWM 제어 | 6장 |
| | 리모컨 응용 | 리모컨 수신 코드에 의한 로봇 수동 제어 | 7장 |
| | PC 자료 전달 | 로봇 제어 명령체계에 따른 Packet 정보 송수신 PC에서 무선 통신에 의한 정보 송수신 환경 구축 | 8장 |
| | 정보 수집 | ADC를 활용: 온도, 거리, 산소 센서 정보 수집 | 9장 |
| 분석기능 | 근접 노드 분석 | C교재 과제 내용 활용 | C교재 |
| | 최단 거리 분석 | C교재 내용 활용 | C교재 |
| 종합 | 소스 종합 | 순서도(flowchart)에 기반한 단위 함수의 통합 템플릿 | 10장 |
| 기타 | 한글 표현 | 한글 LCD 사용법 | 부록 #3 |

참고

구현 방법 및 7절. 실습

구현 방법은 저자가 교재 내용에서 정의된 모든 내용을 개발하여 설명합니다. 또한 독자의 개발 능력 향상을 위해 각 장의 7절. 실습에서는 대표 기능을 3장에서 9장까지 마이컴 제어 주제별로 따라하기에 의해 독자가 직접 개발해 볼 수 있도록 단계적 실습을 제시해드립니다.

10장 7절. 실습에서는 각 단위 함수를 개략 순서도를 활용하여 통합하는 템플릿 프로그램을 제시해드립니다.

제6장 : 로봇 모터 제어

로봇 모터 제어

모터 일반

DC 모터 제어

서보 모터 제어

스텝핑 모터 제어

자율 주행 로봇

PID 제어

* 실습

2 DC 모터 제어

7 DC 속도 제어

① 이동 조건과 거리에 따라 적절한 속도로 이동하도록 제어 필요

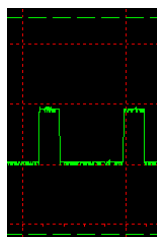
- DC 모터에 공급되는 전류에 비례
- DC 모터 전자 코일의 저항이 일정하기에 전압을 변화시키면 같은 비율로 전류가 변화됨

② PWM(Pulse Width Modulation) 제어

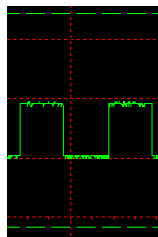
- 모터에 공급되는 전원을 동일한 주기 펄스에서 On/Off 비율을 제어하는 방식
- 펄스의 듀티(Duty Ratio)비를 바꾸어 모터에 공급되는 평균 전압을 변화시키는 방식

$$\text{듀티비}[\%] = \frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100$$

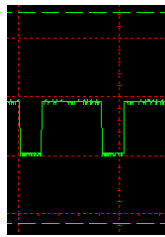
- On 시간이 길어질 경우: 모터에 전달되는 평균 전압이 높아짐



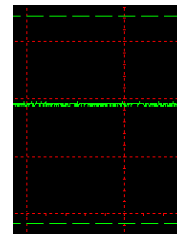
25%



50%



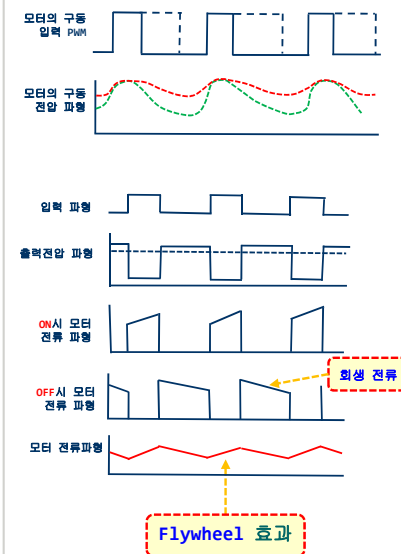
75%



100%

③ PWM에 모터 반응

- (㉠): 마이크로 프로세서를 이용한 빠른 주파수에 대해 DC 모터의 기계적 반응을 갖지 않은 특성
 - 모터에 흐르는 전류는 Off로 되어 있는 동안에도 쉬지 않고 흐르고 있는 것처럼 보임
 - 평균 전류는 On시의 전류와 회생 전류의 합으로 구해지게 됨
- 다이오드: Off로 되어 있는 동안 모터의 코일에 축적된 에너지를 전류로 흘리는 작용
 - 회생 전류가 만들어지도록 역할



3 서보 모터 제어

2 제어 원리

① 제어선을 통해 목표 각도 전달

- 제어선(주황색)에 전달되는 펄스의 (ⓐ) 시간으로 정해짐

② 펄스의 주기

- 일반적으로 10~20ms 주기로 서보 모터에 PWM 신호를 인가

③ 펄스에 따른 각도

| | | | | |
|-------|---|--------|---|--------|
| 0° | ~ | 90° | ~ | 180° |
| 0.5ms | | 1.32ms | | 2.10ms |

④ 세부 동작

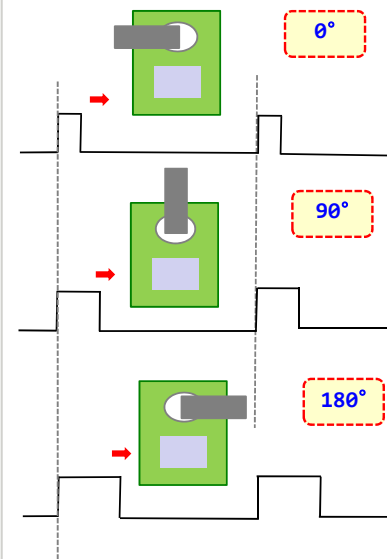
- 사양 확인
- 토크와 관계
 - 토크는 클수록 힘이 좋지만 속도는 저하됨
 - 내부의 기어비에 따름. 속도가 빠르면 토크가 약함. 토크를 좋게 하면 속도가 느림

3 RC 서보 외형

① PWM 신호선: 주황색

② 전원(Vcc): 빨강색

③ 접지(GND): 검지



4 스텝핑 모터 제어

3 유니 폴라 구동 방식

① 1상 여자 방식

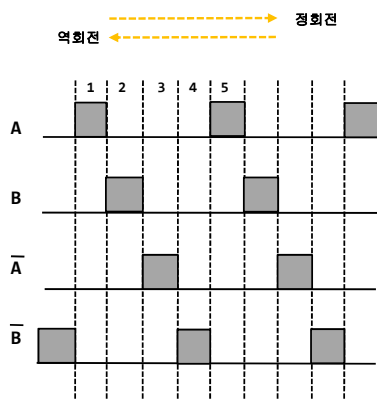
- 원리: 코일의 1개의 상에만 전류를 순차적으로 흐르게 하는 방식
- 장단점: 소비 전력이 작지만 감쇄 진동이 커서 동작이 불안정해지기 쉬움

② 2상 여자 방식

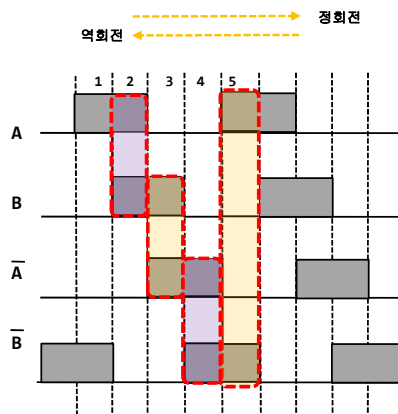
- 원리: 2개의 상에 동시에 전류를 흐르게 하는 방식으로 1상에 비해 2배의 전류가 흐름
- 장단점: 회전력이 커지며 감쇄 진동이 적어 넓은 주파수 범위에서 사용 가능하나 진동이 심함

③ 1-2상 여자 방식

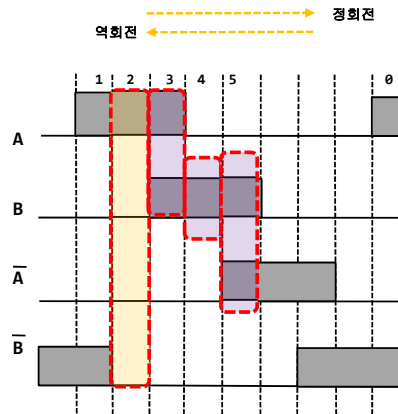
- 원리: 2상 여자 방식과 1상 여자 방식이 반복되는 방식
- 장단점: 소비 전력은 1상 여자 방식의 1.5배 이지만 회전력은 2상 여자 방식과 같아짐
1개의 펄스 당 회전하는 회전각이 1상 및 2상 여자 방식에 비해 (③)이 되어 분해능 높음



(a) 유니폴라 1상여자



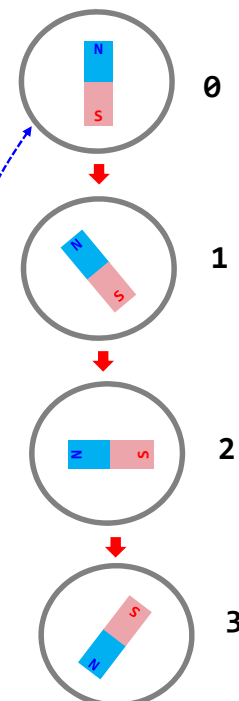
(b) 유니폴라 2상여자



(c) 유니폴라 1-2상여자

1-2상 여자 사례

| Color | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|---|---|---|---|---|---|---|---|
| Brown | | + | + | + | | + | + | + |
| Red | + | + | | + | + | + | | + |
| Orange | - | - | | | | | | |
| Yellow | | - | - | - | | | | - |
| Pink | | | | - | - | - | | |
| Blue | | | | | | - | - | - |



5 자율 주행 로봇

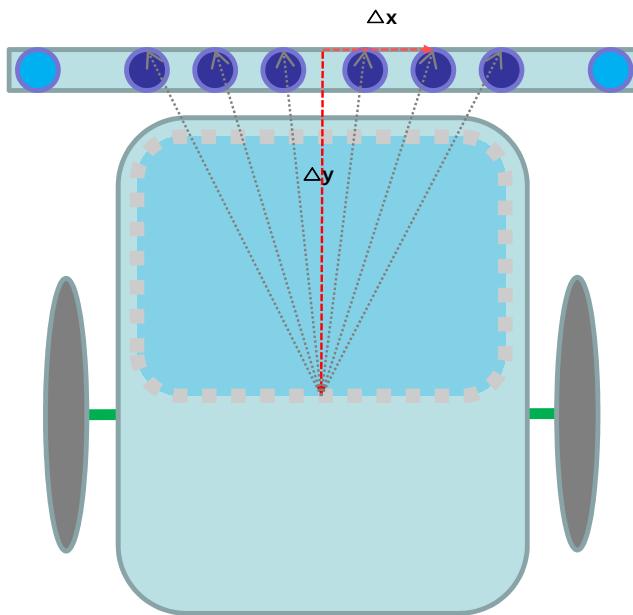
4 로봇 자세 각도와 교차로 인식에 의한 제어

① 방법

- 로봇의 중심을 기준으로 기울어진 각도 계산: 6개 포토 트랜지스터
- 교차로 인지를 위한 별도의 센서 추가 설계: 양쪽 1개씩 2개 포토 트랜지스터

② 자세에 따른 각도 계산법

- 발광, 수광 6개 센서를 이용한 라인 판별에 따른 x, y 축 변화값 확인
- 각도 계산: $(\textcircled{a})(\Delta x/\Delta y); \Rightarrow -nn.nn \sim +nn.nnn$



③ 제어

- 로봇의 기울어진 각도에 따른 모터 속도 제어 가능

6 PID 제어

5 컴퓨터에 의한 PID 제어

① 환경: 컴퓨터는 아날로그와 같이 연속적인 아날로그 제어량을 만들어낼 수 없음

컴퓨터 데이터의 입출력은 일정시간 간격으로 밖에 할 수 없음

② 해결: 미분 적분 연산이 필요하기에 샘플링 방식(이산값)에 적합한 PID 연산 방식

- **샘플링 방식의 피드백 제어(PID 제어) 기본식**: 제어 대상 변수를 계속적으로 측정하여 기준 입력과 비교하고 발생하는 오차를 감소시키기 위해 제어 법칙에 따라 제어 동작 수행

- 제어값(Manipulated Variable) 계산식

= 오차값, 오차값의 적분, 오차값의 미분에 비례

= 비례제어 + 적분제어 + 미분제어

= $(K_p \times \text{편차}) + (K_i \times \text{편차의누적값}) + (K_d \times \text{전회편차와의차})$

$$MV_n = MV_{n-1} + \Delta MV_n$$

$$\Delta MV_n = K_p(e_n - e_{n-1}) + K_i * e_n + K_d((e_n - e_{n-1}) - (e_{n-1} - e_{n-2}))$$

MV_n : 현재 조작량

MV_{n-1} : 전회 조작량

ΔMV_n : 현재 조작량

e_n : 현재의 오차

e_{n-1} : 전회의 오차

e_{n-2} : 전전회의 오차

③ 결론

- 해당 회와 이전 회의 편차값만 측정하면 조작량 계산이 가능하고 (@) 목표값에 도달

Answer @ -1을 곱한다.

3 [구현 설명]

⑨ 단계: fnSetMotorVel() 함수의 오른쪽 모터를 구현하자.

- 왼쪽 모터 제어 소스 아래에 전달되는 dGetRightVel 변수 값에 따라 모터의 정회전을 판단해야 하기에 if 문을 이용하여 0 보다 크면 정회전이 작으면 역회전이 위의 모두 아니면 모두 해제가 되도록 한다.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|--------|
| | COM1A1 | COM1A0 | COM1B1 | COM1B0 | COM1C1 | COM1C0 | WGM11 | WGM10 | TCCR1A |
| 읽기/쓰기 초기값 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|--------|
| | COM3A1 | COM3A0 | COM3B1 | COM3B0 | COM3C1 | COM3C0 | WGM31 | WGM30 | TCCR3A |
| 읽기/쓰기 초기값 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | R/W 0 | |

- if 문 내에 dGetRightVel 변수값에 0 보다 큰 경우에 소스를 구현하자. TCCR1A 3번 비트는 1을 TCCR3A 7번 비트만 0을 설정한다.

ucRMotorH와 ucRMotorL에 dGetRightVel의 값이 적용되도록 구현하자. ucRMotorH에는 dGetRightVel의 비트를 8비트 아래로 쉬프트하여 3비트만 마스크하여 적용해야 한다. OCR1CH와 OCR1CL에 ucRMotorH, ucRMotorL을 적용한다. OCR3AH와 OCR3AL에 모두 0을 적용한다.

- else if 문 내에 위와 반대 개념으로 구현한다. 전달된 값은 -이다. 어떻게 해결할 것인가? (㉠)
- else 문에 TCCR1A의 3번, TCCR3A 7번 비트를 모두 0으로 설정한다. 비교 매치값은 모두 0이 되도록 한다.

3 [답안: EX6_3_DC_PWM.c]

```

92 //=====80
93 // =====[Right Wheel]=====
94 // Right direction(+): + value OCR1C
95 if( dGetRightVel>0 )
96 {
97     TCCR1A |= 0x08;
98     TCCR3A &= 0x7F; // ~0x80;
99
100     ucRMotorH = ( dGetRightVel>>8) & 0x03 ;
101     ucRMotorL = ( dGetRightVel & 0xFF );
102     // OCR1C: Speed value
103     OCR1CH = ucRMotorH; OCR1CL = ucRMotorL;
104     // OCR3A: 0
105     OCR3AH = 0x00; OCR3AL = 0x00;
106 }
107 // Right direction(-): - value => + value OCR3A
108 else if( dGetRightVel<0 )
109 {
110     TCCR3A |= 0x80;
111     TCCR1A &= 0xF7; // ~0x80;
112
113     dGetRightVel *= -1; // "-" => "+"
114     ucRMotorH = ( dGetRightVel>>8) & 0x03 ;
115     ucRMotorL = ( dGetRightVel & 0xFF );
116     // OCR1C: (0)
117     OCR1CH = 0x00; OCR1CL = 0x00;
118     // OCR3A: Speed value
119     OCR3AH = ucRMotorH; OCR3AL = ucRMotorL;
120 }
121 else
122 {
123     TCCR3A &= 0x7F; // ~0x80; => off
124     TCCR1A &= 0xF7; // ~0x08; => off
125
126     ucRMotorH = 0;
127     ucRMotorL = 0;
128     OCR1CH = ucRMotorH; OCR1CL = ucRMotorL;
129     OCR3AH = ucRMotorH; OCR3AL = ucRMotorL;
130 }
131 }

```

교재 활용 방안

대학 교육기관

- ① 적정 시간: (상) 3시간 14회 수업, (하) 3시간 14회 수업
- ② 수업 방식
 - 교수: 교재의 **Keyword** 중심의 이론만 교육
교재 프로젝트에 응용 기능 추가 과제 제시
 - 학생: 실습 문제를 수업 시간 또는 수업 시간 외 자율적으로 구현

직업능력 교육기관

- ① 적정 시간: (상) 3시간 20회 수업, (하) 3시간 20회 수업
- ② 수업 방식
 - 강사: 교재의 **Keyword** 중심의 이론만 교육
교재 프로젝트에 응용 기능 추가 과제 제시
 - 학생: 실습 문제를 수업 시간 또는 수업 시간외 자율적으로 구현

개인 학습

- ① 자신의 능력에 따라 학습 방법 선택
 - geo1004.com의 음성 및 동영상 최대한 활용
- ② C언어 기초가 없는 분
 - 동저자 저술 “C언어 첫걸음”, “C언어 프로그래밍” 선행 학습
- ③ 제어 경험이 없는 분
 - 동저자 저술 “임베디드 제어 첫걸음” 선행 학습

중학생 및 고등학생(방과후 학교 또는 특활 활동)

- [중학생 및 고등학생(방과후 학교 또는 특활 활동)]
- ① “C언어 첫걸음”, “임베디드 제어 첫걸음” 선행 학습(필수)
 - ② geo1004.com의 슬로우 버전 음성 및 동영상을 활용한 학습

교재 활용 방안

학습 순서

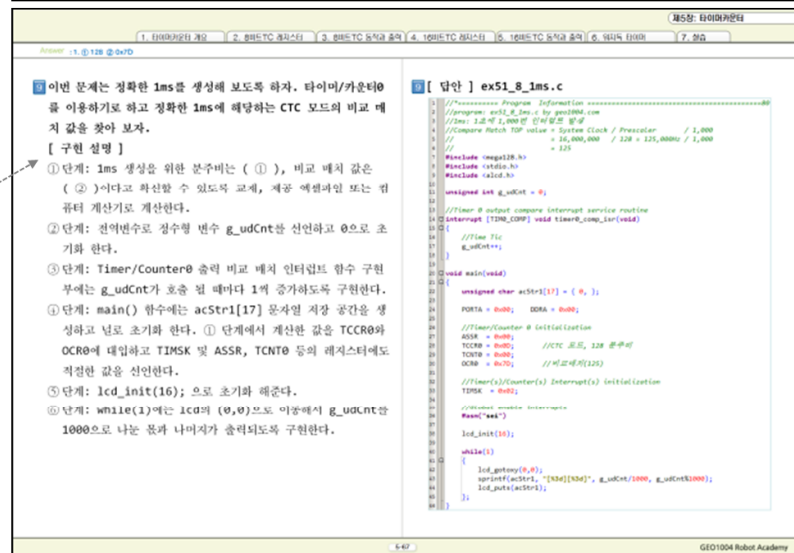
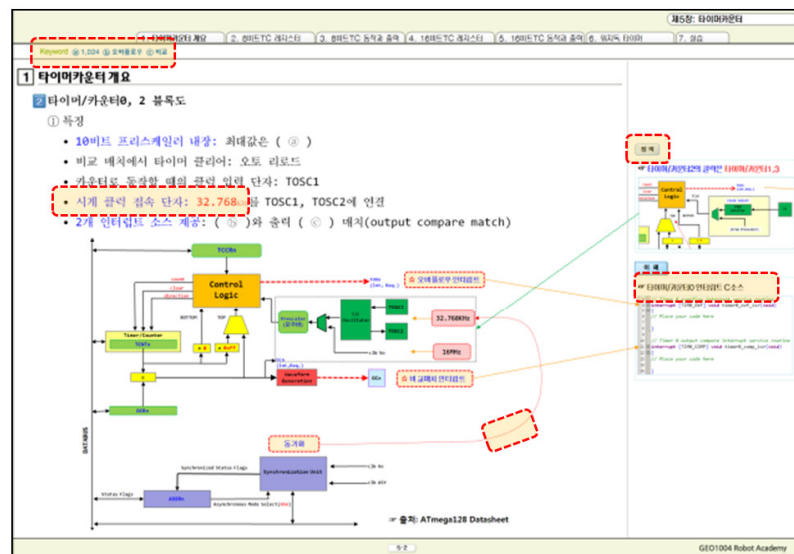
- ① 1안: 최초 학습자는 장 단위로 학습
1장 1절 => 1장 2절 => 1장 3절 => 1장 4절 => 1장 5절
=> 1장 6절 => 1장 7절(실습) => 2장 1절 ..
- ② 2안: 경험자는 1안에 다음과 같은 학습 방법도 가능
1장 1절 => 2장 1절 => 3장 1절 => ... => 10장 1절 => 1
장 2절 => 2장 2절

교재 내용 학습 방법

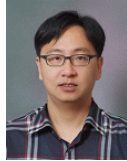
- ① 음성 설명 청강: 키워드 직접 교재 표기
- ② 블록 다이어그램: 저자 추가 기술 및 화살표의 지시 내용 중요
- ③ 중요 개념: 파란색, 빨간색 및 화살표
- ④ 정의 및 이해 내용 반드시 숙지
- ⑤ 이론 설명 및 학습: CodeVision AVR 인터페이스 활용

교재 실습 학습 방법

- ① 음성 또는 동영상 소스 구현 설명 청강
 - 키워드 직접 표기
 - 컴퓨터 계산기 또는 제공 엑셀 파일 활용
- ② 실습 구현
 - 1안: 처음에는 설명의 단계별 내용 참고 구현
 - 2안: 문제만 읽고 구현 가능할 때까지 반복
 - 1안이 불가능 한 경우: 답안 소스 분석
- ③ 교재 내용 습득 후
 - geo1004.com에 제시된 다양한 응용 과제를 구현



저자 소개



서동진(徐銅辰)

[학 력]

조선대학교 제어계측공학과 박사 졸업

Visiting Student Scholar(교환학생연구원): Robotics Institute of Carnegie Mellon University

전자부품연구원 지능로봇시스템연구센터 연구원

노틸러스 효성 로봇개발팀 선임연구원

[주요 학술 활동]

- ① 5편의 SCI급 논문을 포함한 8편의 국제학술지 논문 게재
- ② 9편의 국내학술지 논문 게재
- ③ 30편 이상의 국제학회 논문 발표
- ④ 30편 이상의 국내학회 논문 발표
- ⑤ 1건 저술
- ⑦ 소프트웨어 지적재산권 7건 이상 등록
- ⑧ 대한전자공학회 우수논문 발표상 외 다수 수상

인 사 말

독자분들께 !

반갑습니다.

지식인이자라면 자신의 지식을 정리해 그 결과를 책으로 출판할 꿈을 갖습니다.

본 교재는 처음 마이크로 컨트롤러 제어를 접하는 모든 분들에게 도움을 드리기 위해 기획 되었습니다. 공저자가 각 분야에서 20여년 간 기초 공학과 실무를 바탕으로 경험한 지식을 종합하여 완성하였습니다. 또한 교재의 목표를 감히 “For the world best IT education”이라고 했습니다.

독자분들이 책의 내용에서, 소스에서, 구성에서, 그 가치와 가능성을 확인해 주십시오. 영문화 되어 세계의 모든 독자분들이 인정할 만큼 Best로 검증되었을 때, 마이크로 컨트롤러를 실무적으로 입문하기 위해서 반드시 습득해야 할 책이 본 교재가 되었을 때에 “For”라는 단어를 삭제하도록 하겠습니다.

본 교재는 항상 독자와 함께하는 모델을 추구할 것임을 약속드리며 이 책이 출간되기까지 도와주신 모든 분들께 진심으로 감사 드립니다.

2011년 12월 저자 올림

[2011년] 수송미디어 출판 도서 소개



[2012년] 수송미디어 출판 예정 도서(가칭) 목록

1. 임베디드 제어 첫걸음: 저자 확정
2. 임베디드 제어 첫걸음(영문판, 중국어판): 역자 확정
3. 임베디드 제어 첫걸음(일본어판, 독일어판, 프랑스판): 저자 미확정
4. C언어 첫걸음(영문판, 중국어판): 역자 확정
5. C언어 첫걸음(일본어판, 독일어판, 프랑스판): 저자 미확정
6. 차륜 기반 구동 로봇(AngelRo E-500) 제어: 저자 확정
7. API 함수를 이용한 그래픽 프로그래밍: 저자 미확정
8. MFC 함수를 이용한 로봇 관제 프로그래밍: 저자 미확정
9. 스마트폰을 활용한 로봇 관제 프로그래밍: 저자 미확정
10. 내 기술을 클래스로 만드는 CPP 프로그래밍: 저자 확정
11. VBA를 활용한 초기자료 구축 기법: 저자 미확정
12. AutoCAD 기초: 저자 미확정
13. 인벤터를 활용한 로봇 기구 설계: 저자 확정

위의 교재 집필에 관심 있으신 분은 출판사로 연락 주세요.
출판사에서 다양한 자료(소스)와 디자인 등을 지원해 드립니다.

위의 교재 외에도 IT(컴퓨터, 전자, 전기, 기계) 분야의 교재 발간에 관심이 있으신 분은 협의 후 출판 가능합니다.



geo1004.com

값 18000원



ISBN 978-89-967116-3-6