

(19) 대한민국특허청(KR)  
(12) 특허공보(B1)

(51) Int. Cl.<sup>4</sup>  
F02D 5/02

(45) 공고일자 1985년02월23일  
(11) 공고번호 특1985-0000119

|           |  |           |               |
|-----------|--|-----------|---------------|
| (21) 출원번호 | 특1980-0001563  | (65) 공개번호 | 특1983-0002993 |
| (22) 출원일자 | 1980년04월16일  | (43) 공개일자 | 1983년05월31일   |
| (71) 출원인  | 가부시기 가이샤 히다찌 세이사쿠쇼 요시야마 히로기찌<br>일본국 도쿄도 지요다구 마루노우찌 1쵸메 5반 1고 |           |               |
| (72) 발명자  | 후루하시 도시오<br>일본국 이바라기췁 미도시 호리췌 993의 1                         |           |               |
| (74) 대리인  | 한규환  |           |               |

심사관 : 맹선호 (책자공보 제1039호)

(54) 전자식 엔진 제어방법

요약

내용 없음.

대표도

도1

명세서

[발명의 명칭]

전자식 엔진 제어방법

[도면의 간단한 설명]

제1도~제23도는 본 발명의 제어방법을 기화기를 갖춘 내연기관에 적용한 경우의 제1실시예를 나타낸 도면으로서,

제1도는 엔진의 드로틀 챔버의 단면도.

제2도는 점화장치의 개략도.

제3도는 백기가스 환류장치의 시스템도.

제4도는 제어시스템의 전체 구성 블록도.

제5도는 제4도의 제어회로의 프로그램 시스템도.

제6도는 ROM내에 기억된 프로그램 내용을 나타내는 메모리의 구성도.

제7도는 초기 프로그램의 상세한 플로우차아트.

제8도는 감시 프로그램의 상세한 플로우차아트.

제9도는 엔진 시동시의 점화시기를 나타내는 특성도.

제10도는 엔진 시동시의 연료공급량을 나타내는 특성도.

제11도는 엔진 시동시의 공기공급량을 나타내는 특성도.

제12도는 EGRCAL과 FISC 프로그램의 상세도.

제13도는 제5도의 프로그램(224)의 상세한 플로우차아트도.

제14도는 타스크 스케줄러(TASK SCHEDULER)의 상세한 플로우 차아트.

제15도는 타스크 레벨 프로그램의 기동요구 발생동작을 설명하기 위한 설명도.

제16도는 EXIT 프로그램의 상세한 플로우차아트.

제17도는 프로그램의 처리순서의 설명도.

제18도는 레벨 프로그램의 상태변화를 나타내는 도면.

제19도는 배경 조브(BACKGROUND JOB)의 상태변화를 설명하기 위한 도면.

제20도는 레벨 제로(0) 타스크의 프로그램의 상세한 플로우차아트.

제21도는 레벨 1타스크의 프로그램의 상세한 플로우차아트.

제22도는 IRQ 발생회로도.

제23도~제28도까지는 본 발명의 제어방법을 연료분사형의 내연기관에 적용한 경우의 제 1실시예를 나타낸 도면으로서,

제23도는 엔진 제어장치의 개략구성을 나타내는 단면도.

제24도는 제어장치의 전체구성 블록도.

제25도는 초기 프로그램의 상세한 플로우차아트.

제26도는 엔진 시동시의 연료공급량을 나타내는 특성도.

제27도는 EGRCAL과 FISC 프로그램의 상세한 플로우차아트.

제28도는 레벨 1 타스크의 프로그램의 상세한 플로우차아트.

제29도는 제4도의 CABG(162), FSC(176) 및 제24도의 IFSC 레지스터 및 EGRC 레지스터에 관련한 상세한 블록회로도.

제30도는 제4도 및 제24도의 IGNC(168), (1138)에 관련한 상세한 블록회로도.

제31도는 제4도 및 제24도의 이산형 임펄스회로(DIO)(128)에 관련한 상세한 회로도.

제32도는 제24도의 INJD 레지스터에 관련한 상세한 블록회로도.

제33도는 제32도에 있어서의 회로 동작을 설명하기 위한 파형도이다.

#### [발명의 상세한 설명]

본 발명은 내연기관의 전자식 제어방법에 관한 것으로서, 특히 중앙처리장치(이하 CPU라 함)에 의한 디지털 연산처리에 의하여 내연기관을 제어하는 제어방법에 관한 것이다.

CPU를 사용하여 소정의 프로그램에 따라 엔진을 제어하는 방법 및 장치는 미국 특허 3,969,614 및 4,163,282에 게시되어 있다. 이러한 종래의 기술에 의하면, 그 제어대상마다 센서로 부터의 정보인출에서부터 엔진을 제어하고자 하는 모든 기능에 대한 제어량의 설정까지의 일련의 프로그램을 엔진의 상태에 따라 실행하고 있었다. 이와 같은 제어프로그램으로서는 예컨대, 연료량 제어프로그램이나 점화시기 제어 프로그램 및 엔진 시동 프로그램 등이 있었다. 연료량 제어프로그램이나 점화시기 제어프로그램 등은 예컨대, 엔진 회전 일정 크랭크각에 동기하여 실행되며, 상기 프로그램의 연산결과인 연료공급량이나 점화시기 진각량을 펄스출력회로에 출력함으로써 상기 프로그램의 타스크 처리를 종료시키고 있었다.

이와 같이 엔진의 어떤 상태에서 프로그램을 실행하게 되면, 제어시스템인 CPU의 부하상태가 엔진의 동작상태에 따라 크게 변동한다.

따라서 엔진상태가 급변하는 상태에서는 CPU의 부하가 급변하는 상태에 따라서 현저히 커져 CPU의 서어비스(serviceability : 하드웨어 혹은 소프트웨어의 착오검출, 진단 및 수리의 용이성에 대한 정도)가 저하한다. 예컨대, 엔진속도가 저속에서 중속, 그리고 고속으로 변화하는 경우, 종래의 시스템에서는 CPU에의 타스크 처리의 실행회수인 연산요구회수가 엔진속도에 따라 증대하여, CPU는 이 요구에 응할 수 없는 경우가 있을 수 있다.

한편, 저속시에는 CPU의 부하율이 저하하여 CPU의 능력이 쓸모없이 된다. 이와 같이 종래의 장치 및 방법에서는 제어계 전체로서 CPU의 능력에 상응하는 서어비스를 얻을 수 없다는 불편이 있었다.

또 상기한 특허에서는 복수의 제어프로그램중 하나의 프로그램 내용을 변경할 필요가 생겼을 경우나 또는 새로운 프로그램을 추가하는 경우에는, 일부분만의 변경으로는 안되며, 결국 복수제어프로그램의 전체를 재작성하지 않으면 안된다는 불편이 있었다.

본 발명의 목적은 엔진의 상태변화에 영향을 받지 않고, 항상 높은 제어성능을 얻을 수 있는 엔진의 제어방법을 제공하는데 있다.

본 발명은 CPU가 행하는 복수의 엔진 제어내용을 그 CPU의 처리기능에 따라 몇개의 타스크로 나누어, 각각 처리기능이 엔진의 제어성에 미치는 영향에 따라 즉, 중요성에 따라 그 기동주기를 결정하고, 그 기동주기에 따라서 타스크의 프로그램을 실행한다. 이와 같이 하면, 엔진의 제어성에 있어서 높은 서어비스성을 필요로 하는 타스크 서어비스의 비율을 크게 높일 수 있고, 그리고 CPU의 부하상태는 엔진의 상태 즉, 엔진 회전수 등이 변동하더라도 거의 변화하지 않는다.

따라서 CPU의 능력을 충분히 발휘한 제어성능이 뛰어난 제어를 할 수 있게 된다.

또 본 발명은 복수의 엔진 제어내용(제어프로그램)의 실행을 관리하기 위한 관리프로그램을 갖추고, 또 복수의 제어프로그램중의 하나의 프로그램 내용을 변경할 경우 또는 새로운 프로그램을 추가하는 경우에는 다른 제어프로그램에 하등의 영향을 주지 않고 해당 프로그램의 내용을 변경 또는 추가하면 되므로, 프로그램 내용의 변경이 대단히 용이하다.

본 발명의 기타 목적 및 이점은 이하의 도면에 따른 설명에 의하여 명백해질 것이다.

우선, 본 발명은 기화기를 갖춘 내연기관에 적용한 경우의 제1실시예에 관해 제1도~제22도를 참조하여

설명한다.

제1도는 엔진의 드로를 챔버의 단면도이다. 먼저 이 드로틀챔버 주변에 설치된 각 솔레노이드의 동작에 의하여 드로를 챔버에 공급되는 연료량 및 바이페스 공기량의 제어를 설명한다. 가속페달(도시치 않음)에 의하여 저속측 드로틀밸브(12)의 개방정도가 제어되고, 이에 따라 공기청정기(도시생략)로부터 엔진의 각 실린더에 대한 공급공기량에 제어된다. 이 저속 드로틀밸브의 개구를 크게함에 따라 저속측 벤츄리(34)의 통과공기량이 증대하면, 이 저속측 벤츄리의 부압(負壓)에 의존하는 다이어프램(도시생략)에 의하여 고속측 드로틀밸브(14)가 개구된다. 이로써 흡입공기량 증가에 따른 저항의 증대가 경감한다. 이와 같이 하여 드로틀밸브(12), (14)의 제어하에 엔진에 공급되는 공기의 유량은, 부압센서(도시생략)에 의하여 아날로그량으로써 변환한다. 이 아날로그량 및 기타 후술하는 센서로부터의 신호에 의거하여 제1도의 각 솔레노이드밸브(16), (18), (20), (22)의 개방정도가 제어된다.

다음에 연료공급량의 제어에 관하여 설명한다. 연료탱크로부터 공급된 연료는 도관(24)으로부터 메인제트관(26)을 거쳐 도관(28)으로 유입된다. 또한 도관(24)의 연료는 메인 솔레노이드 밸브(18)를 거쳐 도관(28)으로 유입된다. 따라서 도관(28)으로 유입된 연료는 메인 솔레노이드 밸브(18)의 개구가 커지면 커질수록 많아지며, 이들 연료는 다시 엔진 에멀션 튜브(30)에서 공기와 혼합되고 메인 노즐(32)을 통과하여 벤츄리(34)에 공급된다. 고속드로틀밸브(14)의 개방시에는 메인노즐(32)에 연통한 노즐(36)에서도 벤츄리(38)에 연료가 공급된다. 한편, 메인 솔레노이드 밸브(18)와 동시에 슬로우(slow) 솔레노이드 밸브(16)도 제어되어 이 슬로우 솔레노이드 밸브(16)가 개방되면, 공기 청정기를 거친 공기는 개방구(40)로부터 도관(42)으로 공급된다. 한편 도관(28)으로부터의 연료는 슬로우 에멀션 튜브(44)를 거쳐 도관(42)으로 공급된다. 따라서 도관(42)의 연료량은 슬로우 솔레노이드 밸브(16)로부터의 공기량이 증대하는 만큼 적어진다. 이 도관(42)의 연료와 공기의 혼합물은 구멍(46)으로부터 드로틀 챔버에 공급된다.

연료 솔레노이드 밸브(20)는 연료량을 증가시키기 위한 밸브로서 엔진 시동을 위하여 연료량을 증가시키거나 엔진을 난기시키기 위하여 연료를 증가시키는데 사용된다. 도관(24)과 연통된 구멍(42)으로부터 유입된 연료는 연료 솔레노이드 밸브(20)의 개방정도에 따라, 드로틀 챔버로 통하는 도관(50)에 유입된다.

공기 솔레노이드 밸브(32)는 엔진에 공급되는 공기량을 제어하는 밸브로서, 공기청정기로부터의 공기가 개방구(52)로부터 공기 솔레노이드 밸브(22)로 공급되고, 그 개방의 정도에 따라 드로틀챔버로 통하는 도관(54)에 유입된다.

제1도의 슬로우 솔레노이드 밸브(16)와 메인 솔레노이드 밸브(18)에 의하여 공연비를 제어하여, 연료 솔레노이드 밸브(20)에 의해 연료량을 증가시킨다. 또 슬로우 솔레노이드 밸브(16)와 메인 솔레노이드 밸브(18) 및 공기 솔레노이드 밸브(22)로써 무부하시의 엔진속도를 제어한다.

제2도는 점화장치를 나타내며, 증폭기(62)를 거쳐 전력 트랜지스터(64)에 펄스전류가 공급되며 이 전류에 의하여 트랜지스터(64)는 온(ON)된다. 이로 인하여 축전지(66)로부터 점화코일(68)에 1차코일 전류가 흐른다. 입력펄스 전류의 하강으로 트랜지스터(64)는 오프(off)상태가 되고 점화코일(68)의 2차코일에 고전압을 발생시킨다. 이 고전압은 엔진회전에 동기하여 배전기(70)를 거쳐 엔진의 각 실린더에 있는 점화플러그(72)의 각각에 공급된다.

제3도는 배기가스 완류(이하 EGR라 함) 시스템을 설명하기 위한 것으로서, 부압원(80)의 일정한 부압이 압력제어밸브(84)를 거쳐 제어밸브(86)에 가해진다. 압력제어밸브(84)는 트랜지스터(90)에 인가되는 반복펄스의 온듀티 사이클에 따라 부압원(80)의 일정부압을 대기(88)로 개방하는데 대한 비율을 제어하여 제어밸브(86)에 인가되는 부압을 제어한다. 따라서 제어밸브(86)에 인가되는 부압은 트랜지스터(90)의 듀티사이클을 근거로 정해진다. 이 압력제어밸브(84)의 제어부압에 의하여 배기관(92)으로부터 흡기관(82)까지의 EGR량이 제어된다.

제4도는 제어시스템의 전체 구성도이다. 이 제어시스템은 (102)와, 읽기전용메모리(104)(이하 ROM이라 함)와, 랜덤 액세스 메모리(106)(이하 RAM이라 함)와, 입출력회로(108)들로서 구성되어 있다.

상기 CPU(102)는 ROM(104)내에 기억된 각종의 프로그램에 따라 입출력회로(108)로부터의 입력데이터를 연산하여, 그 연산결과를 다시 입출력회로(108)에 되돌려보낸다. 이러한 연산에 필요한 일시적인 데이터 기억은 RAM(106)을 사용한다. CPU(102), ROM(104), RAM(106), 입출력회로(108)간의 각종 데이터를 주고 받는 것은 데이터버스와, 제어버스와 어드레스 버스 등으로 구성되는 버스 라인(110)에 의하여 행해진다. 입출력회로(108)에는 제1아날로그-디지틀 컨버터(122)(이하 ADC1이라 함)와, 제2의 아날로그-디지틀 컨버터(124)(이하 ADC2라 함)와, 각도 신호처리회로(126)와 1비트 정보를 입출력하기 위한 이산형 입출력회로(128)(이하 DIO라 함)의 입력수단을 갖는다.

ADC1에는 축전지 전압 검출센서(132)(이하 VBS라 함)와 냉각수온도 센서(134)(이하 TWS라 함)와 대기온도센서(136)(이하 TAS라 함)와, 조정전압발생기(138)(이하 VRS라 함)와, 드로틀 각도센서(140)(이하  $\theta_{THS}$ 라 함)와,  $\lambda$ 센서(142)(이하  $\lambda S$ 라 함) 등의 출력이 멀티플렉서(162)(이하 MPX라 함)에 인가되고, MPX(162)에 의하여 그 중의 하나를 선택하여 아날로그-디지틀 컨버터(164)(이하 ADC라 함)에 입력시킨다. ADC(164)의 출력인 디지틀 값은 레지스터(166)(이하 REG라 함)에 기억된다.

또 부압센서(144)(이하 VCS라 함)는 ADC2(124)에 입력되어 아날로그-디지틀 컨버터(172)(이하 ADC라 함)를 거쳐 디지틀 변환되어 레지스터(124)(이하 REG라 함)에 기억된다. 각도센서(146)(이하 ANGS라 함)로부터는 기준 크랭크각 예를 들면, 180도 크랭크각을 나타내는 신호(이하 RES라 함)와, 미소각 예를 들면, 1도 크랭크각을 나타내는 신호(이하 POS라 함)가 출력되어 각도신호 처리회로(126)에 인가되고, 여기서 파형 정형된다. DIO(128)에는 무부하 스위치(148)(이하 IDLE-SW라 함)와 톱기어스위치(150)(이하 TOP-SW라 함)와 시동스위치(152)(이하 START-SW라 함)의 출력이 입력되고 있다.

다음에 CPU(102)의 연산결과에 의거한 펄스출력회로 및 제어대상에 관하여 설명한다. 공연비 제어장치(165)(이하 CABCR라 함)는 이 실시예에서는 펄스신호의 듀티사이클을 바꾸어 슬로우 솔레노이드

밸브(16)와 메인 솔레노이드 밸브(18)를 제어하는 것이다.

CABC(165)의 듀티사이클을 증가시켜, 메인 솔레노이드 밸브(18)에 의한 연료공급을 감소시키기 때문에 CABC(165)로부터의 출력신호는 인버터(163)를 거쳐 메인 솔레노이드 밸브(18)에 인가된다. 한편, 솔로 우 솔레노이드 밸브(16)는 CABC(165)로부터의 펄스신호의 듀티 사이클이 증가함에 따라 연료공급량을 증대시킨다. CABC(165)는 반복펄스 주기를 기억하는 레지스터(이하 CABP라 함)와 동일펄스신호의 듀티사이클을 기억하는 레지스터(이하 CABD라 함)가 설치되어 있으며, CPU(102)내로 이들 데이터가 개별적으로 기억된다.

점화펄스 발생회로(168)(이하 IGNC라 함)는 점화시기 데이터를 기억하는 레지스터(이하 ADV라 함)와 점화코일 1차전류 통전시간을 제어하는 레지스터(이하 DWL라 함)를 포함하고 있으며, 이들 데이터는 CPU(102)내에 기억된다. 이 IGNC(168)의 출력펄스는 점화장치(170)에 인가된다. 이 점화장치(170)의 상세한 설명은 제2도에 나타내는 바와 같으며, 출력펄스는 제2도의 증폭기(62)에 인가된다.

연료량 증가펄스 발생회로(176)(이하 FSC라 함)는 펄스의 듀티사이클을 제어하여 제1도의 연료 솔레노이드 밸브를 제어하는 것으로서, 반복주기를 기억하는 레지스터(이하 FSCP라 함)와 동일펄스의 듀티사이클을 기억하기 위한 레지스터(이하 FSCD라 함)를 갖고 있다. EGR량 제어펄스 발생회로(178)(이하 EGRC라 함)에는 펄스반복주기의 데이터를 기억하기 위한 레지스터(이하 EGRC라 함)와 펄스신호의 듀티사이클의 데이터를 기억하기 위한 레지스터(이하 EGRD라 함) 등이 설치되어 있으며, 반복펄스가 AND 게이트(184)를 거쳐 공기 솔레노이드 밸브(22)에 인가되며, 이 AND 게이트(184)에는 D10(128)의 출력 D101의 신호가 인가되며, 이 D101 신호가 저레벨일때 AND 게이트(184)는 동작상태로 되어 공기 솔레노이드밸브(22)가 제어된다.

한편, D101이 고레벨일때는 AND 게이트(186)가 동작상태로 되어 EGR 장치(188)를 제어한다. EGR 장치(188)의 기본구성은 제3도에서 이미 설명한 바와 같다. D10(128)는 상술한 바와 같이 1비트 신호의 임출력회로로, 입력 또는 출력을 결정하기 위한 데이터를 기억하는 레지스터(이하 DDR이라 함)(192)와, 출력되는 데이터를 기억하는 레지스터(194)(이하 DOUT라 함)를 갖고 있다.

이 D10(128)에 의해 연료펌프(190)를 제어하기 위한 신호 D100가 출력된다.

제5도는 제4도 제어회로의 프로그램 시스템이다. 키스위치(도시생략)에 의하여 전원이 온 되면, CPU(102)는 스타아트 모우드로 되어 초기프로그램의 INITIALIZ(204)을 실행한다. 다음에 감시프로그램의 MONIT(106)을 실행한다 하여 배경조부(BACKGROUND JOB)(208)를 실행한다.

이 배경조부로서 예컨대, EGR량의 계산타스크(이하EGRCAL TASK라 함)나 연료 솔레노이드 밸브와 공기 솔레노이드 밸브의 계산타스크(이하 FISC라 함)를 실행한다. 이 타스크의 실행중에 개입중단요구(이하 IRQ라 함)가 발생하면 IRQ의 개시시스템(222)에서 IRQ 요구분석 프로그램(224)(이하 IRQ ANAL이라 함)을 실행한다. 이 IRQ ANAL의 프로그램은 다시 ADC1의 종료 개입중단 처리(이하 ADC1 END IRQ라 함) 프로그램(226)과, ADC2의 종료 개입중단처리(이하 ADC2 END IRQ라 함) 프로그램(228)과, 일정기간 경과 개입중단처리(이하 INTV IRQ라 함) 프로그램(230)과, 엔진정지 개입중단처리(이하 ENST IRQ라 함) 프로그램(232) 등으로 이루어져 후술하는 각 타스크의 기동이 필요한 타스크에 각각 기동요구(이하 QUEUE라 함)를 낸다. 이 IRQ ANAL 프로그램(224)내의 각 프로그램 ADC1 END IRQ(226), ADC2 END IRQ(228)와, INTV IRQ(230)의 각 프로그램에 의하여 QUEUE가 출력되는 각 타스크는 레벨 "0"타스크군(252), 레벨 "1"타스크군(254), 레벨 2 타스크군(256)이나, 레벨 3 타스크군(258), 또는 그 각 타스크군을 구성하는 타스크이다. 또 ENST IRQ 프로그램(232)에 의하여 QUEUE가 발생하는 타스크는 엔진 정지시의 처리타스크(262)(이하 ENST TASK라 함)이다. 이 ENST TASK(262)가 실행되면 다시 제어시스템은 스타아트 모우드로 되어 개시시스템(202)으로 되돌아간다.

타스크 스케줄러(242)는 QUEUE가 발생하고 있는 타스크군이나 또는 실행중단 타스크군중의 레벨이 높은 타스크군(여기서는 레벨 "0"를 최고로 한다)에서 실행하도록 타스크군의 실행순서를 결정한다. 타스크군의 실행이 종료하면, 종료보고 프로그램(260)(이하 EXIT라 함)에 의하여 종료보고된다. 이 종료보고에 의하여 실행대기중인 타스크군의 가장 레벨이 높은 타스크군을 다음에 실행한다. 실행중단 타스크군이나 QUEUE가 발생하고 있는 타스크군이 없어지면 타스크 스케줄러(242)로부터 다시 배경조부(208)가 실행된다. 다시 레벨 "0" 타스크군으로부터 레벨 "3" 타스크군의 어느 것인가를 실행중에 IRQ가 발생하면 IRQ 처리프로그램의 개시시스템(222)으로 되돌아간다.

제1표에 각 타스크의 기동과 그 기능을 나타낸다.

[제 1 표]

| 레벨  | 프로그램명          | 기능                                  | 기동(다이빙)                      |
|-----|----------------|-------------------------------------|------------------------------|
| -   | IRQ ANAL       | IRQ의 요구분산과 바스스크루 또는 다스 크의 기동요구의 발생  | IRQ                          |
| -   | TASK SCHEDULER | 실행타스크군 또는 바스스크의 결정                  | IRQ ANAL의 종료 또는 EXIT의 종료     |
| -   | EXIT           | 실행타스크군의 종료요구                        | 각 타스크군의 종료                   |
| 0   | AD1 IN         | ADC1으로부터의 인출                        | INTV IRQ 10m sec 또는 ADC1 END |
|     | AD1 ST         | ADC1의 기동                            | INTV IRQ 10m sec             |
|     | AD2 IN         | ADC2로부터 출력인출                        | INTV IRQ 10m sec 또는 ADC2 END |
|     | AD2 ST         | ADC2의 기동                            | INTV IRQ 10m sec             |
|     | RPM IN         | 연결·속도의 인출                           | INTV IRQ 10m sec             |
| 1   | CARBC          | 공인비세이틀 위하여 류터사이클 계산                 | INTV IRQ 20m sec             |
|     | IGNCAL         | 전화시기 계산                             | INTV IRQ 20m sec             |
|     | DWLCAL         | 동전개시시기의 계산(점화로인을 통과하는 L자전류의 기간을 계산) | INTV IRQ 20m sec             |
| 2   | LAMBDA         | λ 제어                                | INTV IRQ 40m sec             |
| 3   | HOSEI          | 보정의 계산                              | INTV IRQ 100m sec            |
| -   | FISC           | 연료 흡대노이드 밸브의 공기 흡대노이드 밸브의 선점량 계산    | 비경조브                         |
| -   | EGRCAL         | EGR 부양제어밸브의 선점치계산                   | 비경조브                         |
| -   | INITIALIZ      | 입출력회로에의 초기치 설정                      | START 또는 RESTART             |
| ... | MONIT          | START-SW의 간사와 연도점프의 기동              | START 또는 RESTART             |
| ... | ENST TASK      | 연도점프점지와 IGN의 리세트                    | ENST IRQ                     |

제1표는 제5도의 시스템을 관리하기 위한 프로그램으로서, IRQ ANAL, TASK SCHEDULER와 EXIT가 있다. 이들의 프로그램(이하 OS라 함)은 제6도와 같이 ROM(104)의 어드레스(A000) 내지 어드레스(A300)에 유지되어 있다. 또, 레벨 "0" 프로그램으로서 AD1 IN, AD1 ST, AD2 IN, AD2 ST RPM IN의 각 프로그램이 있고, 이는 10(msec)마다 발생하는 INTV IRQ에 의하여 기동된다. 레벨 "1" 프로그램으로서 CARBC, IGNCAL, DWLCAL 프로그램이 있고, 이들은 20(m sec)마다 발생하는 INTV IRQ에 의하여 기동된다. 레벨 "2" 프로그램으로서 LAMBDA 프로그램이 있고, 이는 40(m sec)마다 발생하는 INTV IRQ에 의하여 기동된다. 레벨 "3" 프로그램으로서 HOSEI 프로그램이 있고, 100(m sec)마다 발생하는 INTV IRQ에 의하여 기동된다. 또 배경조브로서 EGRCAL과 FISC 프로그램이 있다. 레벨 "0" 프로그램은 PROG1으로서, 각각 제6도의 ROM(104)의 어드레스 A700내지 어드레스 AAFF에 기억되어 있다.

레벨 "1" 프로그램은 PROG2로서, ROM(104)의 어드레스 AB00에서 ABFF에 기억되어 있다. 레벨 "2" 프로그램은 PROG 3로서 ROM(104)의 어드레스 AE00에서부터 AEFf까지 기억되어 있다. 레벨 "2" 프로그램은 PROG 4로서 ROM(104)의 어드레스 AF00에서부터 AFFF에 기억되어 있다.

또 배경조브 프로그램은 B000에서 B1FF에 기억되어 있다 또한 상기 프로그램 PROG1에서 PROG4까지의 각 프로그램의 스타아트 어드레스의 리스트(이하 SFTMR이라 함)가 어드레스(B200에서 B2FF까지)에 기억되고, PROG1에서 PROG4까지의 각 프로그램 기동주기를 나타내는 값(이하 TTM이하함)이 어드레스 B300에서 B3FF에 기억되고 있다.

기타의 데이터는 필요에 따라 제6도의 ROM의 어드레스 B400에서 B4FF에 기억된다. 계속하여 연산을 위한 데이터 ADV MAP나 AF MAP, EGR MAP를 각각 기억하고 있다.

제5도에서의 초기 프로그램(204)을 제7도를 사용하여 상세히 설명한다. 스택(282)에서 IRQ 발생시의 대기영역을 설정한다. 다음에 스택(284)에서 RAM(106)을 모두 클리어 한다. 스택(286)에서 입출력회로(108)의 레지스터내에 초기치의 설정을 행한다. 이 초기치의 설정으로서 예컨대, 엔진의 실린더수나 각도 센서(146)의 초기치나 D10(128)의 DDR의 설정, INTV IRQ발생을 위한 타이머의 설정, ENST IRQ발생을 위한 검출기간의 설정과 엔진 회전수 검출을 위한 계속시간의 설정이 있다.

스택(288)에서 ADC1이 트리거되게 하고, 그리고 ADC1 END IRQ를 위한 금지를 해제한다. 이 경우 AD1ST 프로그램의 스타아트, 어드레스인 제6도의 어드레스 A700로 점프한다. 이에 의해 제4도의 ADC1(122)의 MPX(162)의 입력의 하나인 축전지 전압검출센서 VBS(132)의 출력이 선택되어 ADC(164)에 입력된다.

스택(290)으로 되돌아가서, 여기서 ADC1 END IRQ를 기다린다. ADC(164)의 동작이 완료하여 REG(166)에 디지를 값이 입력되면 스테이터스 레지스터 STATUS(198)에 ADC(164)의 동작완료가 보고되어, ADC1 END IRQ가 CPU(102)에 입력된다. 이에 의해 프로그램 AD1 IN이 실행되어 축전지 전압센서(132)의 출력이 취입된다.

스택(292)에서 VBS(132)에서부터 λS(142)의 값을 모두 인출하였는지를 확인한다. 이 경우의 VBS(132)의

입력의 인출이 종료되므로 스텝(288)으로 되돌아간다. 이 스텝(288)에서, 다시 AD1 ST프로그램이 기동되고, MPX(162)는 다음의 입력으로써 TWS(134)의 출력을 선택한다. TWS(134)의 출력의 아날로그-디지틀 변환이 끝나면 다시 스텝(292)에서, 프로그램 AD1 IN인출이 실행되고, 레지스터 REG(166)내에 기억되어 있는 TWS(134)의 출력의 디지틀 값이 인출되어 ROM(104)의 DATA영역에 기억된다. 스텝(292)에서, 다시 스텝(288)으로 되돌아간다. 이와 같이 스텝(288)에서부터 스텝(292)까지의 루우프를 반복실행으로 VBS(132) 내지  $\lambda S(142)$ 까지의 출력의 디지틀 값이 차례로 인출되고,  $\lambda S(142)$ 의 출력치의 인출이 끝나면 스텝(294)으로 진행한다. 스텝(294)에서 시동의 점화시기를 연산한다.

이 점화시기  $Q_{ADV}(START)$ 는 엔진의 냉각수의 온도 TW의 함수로써 연산한다. 이 함수를 제9도에 나타낸다. 제9도의 특성에 따라  $Q_{ADV}(START)$ 를 연산하여, 이 연산결과는 제4도 ADC(164)의 레지스터 ADV(166)에 입력된다.

스텝(296)에서 시동시의 공기 솔레노이드 밸브(22)의 개방정도를 연산한다. 이 연산은 제11도에 나타내는 특성에 의거하여 행해지고 그 연산출력은 레지스터 EGRD에 세트된다. 또 공기 솔레노이드 밸브의 개방 정도에 대한 고정치가 EGRP에 세트된다. 제11도에서 엔진 기동을 위한 공기 솔레노이드 밸브(22)의 밸브 개방정도는 EGRP에 기억되고 정지에 대한 세트치의 비이다.

스텝(298)에서 연료솔레노이드 듀티계수의 초기치가 연산된다. 이 연산치는 제10도에 의하여 행해지는 것이며, 레지스터 FSCP의 세트치에 대한 레지스터 FSCD의 세트치의 비율이 제10도의 특성도로서 도시되어 있다.

레지스터 FSCP와 FSCD에 데이터를 세트함에 따라, 제5도의 프로그램 INITIALIZ(204)의 실행이 종료되고, 제8도에 나타내는 MONIT프로그램(206)이 실행된다. 프로그램(206)은 스텝(302)에서부터 개시되고 스텝(302)에서 제4도의 시동스위치(152)가 온상태인지의 여부를 DI0(128)의 DI05 입력을 감시함으로써 결정한다. 만일 시동 스위치(152)가 온 상태이면 DI0(128)의 제5비트 DI05가 고논리 "H"로 된다. 또 반대로 오프가 되어 있으면 저논리 "L"로 된다. 지금 엔진이 시동전이라고 하면 시동 스위치는 오프이며, 스텝(302)으로부터 스텝(312)으로 진행되고 여기서 프로그램이 시동후인지의 여부를 판단한다. 이러한 판단은 예컨대, 시동표식(flag)이 되어 있는지의 여부를 판단한다. 이 시동표식은 스텝(308)에서 세트된다. 이 시동표식은 제15도의 RAM내의 소정위치에 세트된다. 시동전은 시동표식이 세트되지 않으므로 스텝(312)에서 부정(NO)되고 다시 스텝(302)으로 되돌아간다.

시동스위치(152)가 온될때까지 스텝(302)과 스텝(312)사이에서 루우틴 루프가 반복된다. 이러한 경로를 밟으면서 시동스위치의 감시를 행한다.

시동 스위치가 온되고 스텝(302)에서의 판단이 긍정(YES)으로 되면 스텝(304)으로 진행한다. 여기서 이미 시동스위치(152)가 온되어 있었던 것인지의 여부를 판단한다. 스텝(304)으로 진행한 경우는 시동스위치의 온을 검출한 직후여서 스텝(304)에서 판단이 부정(NO)으로 된다. 스텝(304)의 판단도 시동 표식이 되어 있는지 아닌지의 여부에 의해서 판단된다. 시동 표식이 나와있지 않은때는 판단을 부정(NO)으로 하여 스텝(306)으로 진행시켜 시동을 위한 준비가 된다.

예컨대, 본 실시예에서는 연료펌프(190)의 기동을 위하여 DI0(128)의 DOUT레지스터(194)의 제"0" 비트에 고논리 "H"를 세트한다.

이에 의하여 연료펌프가 작동된다.

다음에 스텝(306)에서 DOUT레지스터(194)의 제1비트를 저논리 "L"로 한다. 이에 의해 EGRC회로(178)의 출력으로 공기 솔레노이드 밸브(22)가 제어된다. 실제로는 DOUT레지스터(194)의 제"0"비트와 제1비트의 세트는 동시에 행해진다.

스텝(308)에서 INTV IRQ의 금지 해제가되어 점화출력 금지의 해제가 된다. 이 INTV IRQ의 금지해제는 예컨대, 제4도의 MASK레지스터(200)의 제4비트(제22도의 플립-플롭(739))를 고논리 "H"로 함으로써 행한다. 다시 스텝(308)에서 시동 표식을 낸다. 이 시동표식은 이미 시동스위치가 온 임을 나타내고 있어 스텝(304)과 스텝(312)에서 이 표식을 판단에 사용한다.

스텝(310)에서 입출력회로(108)의 출력부인 CAB(165)와 IGNC(168)와 FSC(176)와 EGRC(178)등에 기동을 걸기 때문에 후술하는 모우드 레지스터에 고논리 "H"를 세트한다. 이에 의하여 각 제어장치에 펄스 출력이 보내진다.

스텝(310)으로부터 스텝(302)으로 되돌아가 스텝(302)에서 시동스위치(152)가 온인지 여부가 판단된다. 시동 중에는 스위치가 온이어서 긍정(YES)으로 되면 스텝(304)으로 진행한다. 이 스텝(304)에서 시동표식을 점검하고 표식이 되어 있으면 이미 시동 중인 것으로 하여 스텝(302)으로 되돌아간다.

이와 같이 시동 모우터가 구동되고 있는 동안은 스텝(302)의 긍정(YES)과 스텝(304)의 긍정(YES)에 의하여 이루어진 루우프를 반복 수행한다.

엔진이 시동되면 시동 스위치(152)가 오프되므로 스텝(302)의 판단은 부정(NO)되면 스텝(312)으로 진행한다. 스텝(312)에서 시동표식이 점검되고 시동 표식이 되어 있으므로 스텝(314)으로 진행한다. 이 스텝(314)에서 ENST IRQ의 금지 해제가 행해져 이 스텝(316)이후 엔진의 정지는 이 ENST IRQ에서 검출할 수 있다.

다음에 배경조브의 프로그램(208)으로 진행한다. 이 프로그램을 제12도에 상세하게 나타낸다.

제12도에서, 스텝(410)에서 무부하스위치(148)가 온인지의 여부가 판단된다. 만일 온이면 배기가스 환류는 행해지지 않는다.

따라서 스텝(3412)으로 진행하여 여기서 공기솔레노이드 밸브(22)를 DI0의 레지스터 DOUT(194)의 제1비트를 저논리 "L"로 함으로써 지정된다. 이 때문에, 레지스터 EGRD에 세트된 값에 따라 제1도의 공기 솔

레노이드 밸브(22)가 제어된다. 바이패스 통로의 공기유량을 제어하는 공기솔레노이드 밸브(22)는 특정한 운전 상태에 따라 제어된다. 즉, 기온이 낮은 겨울철의 운전이나, 엔진이 식어있는 상태에서의 시동 운전 및 엔진에 부하가 걸려있는, 즉 자동차 에어컨 사용중에 운전하는 경우에는 바이패스 통로의 공기유량을 증가시킨다. 스텝(414)에서 냉각수 온도에 따라 공기 솔레노이드 밸브(22)의 듀티가 레지스터 EGRD에 설정된다. 또한, 수온 TW에 따른 연료 솔레노이드 밸브로부터의 연료 공급량이 레지스터 FSCD에 세트된다. 이들 값에 따라 연료 솔레노이드 밸브(20)와 공기 솔레노이드 밸브(22)가 제어되어 무부하시의 엔진이 제어된다. 이 스텝(416)이 종료함으로써 스텝(410)으로 진행하여 이 플로우를 반복실행한다.

한편, 무부하 스위치(148)가 온이 아닌 상태에서는 공기 솔레노이드 밸브(22)는 사용되지 않고, 대신 배기가스 환류가 행해진다. 이 때문에 EGR량을 제어하는 EGR장치(188)가 구동되고 이 장치를 구동하기 위하여 스텝(418)에서 DIO의 DOUT레지스터(194)의 제1비트가 고논리 "H"로 세트되며, 레지스터 EGRD에 세트된 값으로 EGR장치(188)가 구동된다. 다음에 스텝(420)에서 레지스터 FSCD에 "0"이 세트되어 연료솔레노이드(15)의 구동을 정지한다. 스텝(422)에서, 냉각수온의 일정치 예컨대, TA°C보다 높은지의 여부를 판단하여 만일 높으면 EGR작동을 하지 않게 한다. 이 때문에 스텝(426)에서 EGR CUT를 행하기 위한 값을 세트한다. 또 스텝(422)에서 냉각수온이 일정치(TA°C) 보다 낮은 값이면 스텝(424)으로 진행한다. 스텝(424)에서 냉각수온이 TB°C보다 낮은지의 여부를 판단한다. 만일 낮을 경우 역시 EGR작동을 하지 않게 함으로 스텝(426)에서 EGR CUT를 행하는 값을 세트한다. 이들의 값은 스텝(430)에서 EGRD레지스터에 세트된다.

한편, 냉각수온 TW가 TB°C보다 높고 TA°C보다 낮은 경우 EGR을 행한다. 이때의 EGR량은 부압센서(144)의 출력 VC와 엔진 회전속도(N)에 따라 결정된다.

출력 VC와 N에 의한 EGR량의 맵(MAP)은 제6도 ROM의 어드레스 B700~B7FF에 세트되어 있고, 이 맵으로부터 검색하여 EGR량을 정한다. 스텝(428)에서 검색을 하여 이 값을 스텝(430)에서 레지스터 EGRD에 세트한다. 이 레지스터 EGRD의 세트값에 따라 제4도에 나타낸 장치(188)는 구동된다.

제12도에 도시한 플로우차아트에서는 스텝(430) 또는 스텝(416)의 종료에 따라 다시 스텝(410)으로 되돌아간다. 이와 같이 함으로써, 계산기는 공기 솔레노이드 밸브(22)를 제어하기 위한 스텝(410)으로부터(416)까지의 플로우차아트나 또는 스텝(418)으로부터 스텝(430)까지의 스텝의 플로우차아트를 항상 실행할 수가 있다. 따라서 IRQ등의 발생이 생기지 않는 것으로 하면 개시시스템(202)으로부터 출발한 프로그램은 INITIALIZ(204), MONIT(206)을 거쳐 배경조브(208)의 FISC 또는 EGRCAL를 항상 실행을 계속하게 된다. MONIT(206)이나 배경조브(208)는 개입중단 요구(IRQ)를 발생시켜 그 처리를 중단할 수 있게 되어 있고 IRQ에 의한 처리가 끝나면 다시 상기 프로그램의 실행을 재개한다.

다음에 제5도에서 IRQ의 발생에 의거한 처리에 대한 설명을 한다. IRQ의 요인 분석의 프로그램(224)은 ADC1 END IRQ(226)의 처리와, ADC2 END IRQ(228)의 처리와, INTV IRQ(230)의 처리와, ENST IRQ(232)의 처리등으로 구성되어 있다. 여기서 각 프로그램(226), (228), (230), (232)의 각각을 실행하기 위해서는 먼저, IRQ의 요구내용이 무엇인가를 조사한다. 이를 위하여 제4도의 STATUS레지스터(198)의 내용이 조사된다. 이 STATUS레지스터의 내용을 봄으로써 IRQ의 발생용인이 판명된다. 이 발생요인에 따라 상기 각 프로그램(226), (228), (230), (232)을 실행하고 이에 의해 TASK(252), (254), (256), (258), (256)내의 실행이 필요한 TASK에 기동요구(QUEUE)를 낸다.

단, IRQ의 발생을 많게하면 관리 프로그램(이하 OS라함)의 실행시간이 많아져 실질적인 엔진 제어를 위한 연산시간을 취할 수 없게되는 결정이 있다. 따라서 이 실시예에서는, ADC2 END IRQ(228)은 INITIALIZ(204) 또는 MONIT(206)의 실행중에만 발생시키고 그 외는 발생시키지 않는다. 즉 MONIT(206)의 제8도에 도시한 스텝(314)에서 제4도의 MASK레지스터(200)(제22도의 플립플롭(766))에 ADC2 END IRQ의 금지명령 저논리 "L"를 세트한다. 또 ADC1 END IRQ(226)는 처음부터 발생시키지 않는 즉, 개시시스템(202)에서 모든 개입중단이 금지되도록 입출력 회로의 일반적 리세트신호로 IRQ발생금지 상태가 되도록 MASK레지스터를 세트한다. 그후 금지해제의 명령을 내지 않도록 함으로써 ADC1 END IRQ를 금지상태로 한다.

STATUS레지스터(224)의 프로그램에 대한 구체적인 예를 제13도에 나타낸다. IRQ의 입구 개시시스템(222)으로부터 스텝(500)에서 ADC1 END IRQ가 발생하지 않았음이 판단되면 스텝(502)으로 진행한다. 여기서 IRQ의 발생요구가 ADC2 END IRQ인지의 여부를 판단하여 긍정(YES)일 경우 스텝(516)에서 타스크 레벨 "0" 타스크의 프로그램에 기동요구를 낸다.

이것은 제15도에 도시한 RAM(106)내의 타스크 콘트를 워드 TCWO의 b6에 "1"의 표식을 세트한다. 그리하여 TASK SCHEDULE(242)로 진행한다. ADC2 END IRQ가 발생하는 것은 이 실시예에서는 제5도의 INITIALIZ(204)의 프로그램이 실행중이기 때문이다. 그 이외의 상태에서는 ADC2 END IRQ는 금지된다. 스텝(502)의 판단이 "부정(NO)"인 경우 스텝(504)으로 진행한다.

스텝(504)에서 IRQ의 요인이 일정 주기로 발생하는 INTV IRQ인지의 여부를 판단한다.

"긍정(YES)"인 경우 스텝(506)으로 진행한다. 스텝(506)에서부터 스텝(514)까지는 타스크 레벨 "0"에서부터 타스크 레벨 "3"까지의 프로그램의 기동 타이밍인지의 여부를 판단하는 기능을 갖는다. 먼저 타스크 레벨 "0"을 조사한다. 타스크 레벨 "0"의 타스크 콘트를 워드 즉, 제15도의 TCWO의 b0~b5까지의 카운터는 0을 +1만큼씩 증가시킨다. 이와 같이 본 실시예에서는 가산법을 채용했으나 감산법을 채용할 수도 있다. 스텝(508)에서 TCWO의 카운터 0의 값과 제15도의 타스크 기동 타이머 TTMO를 비교한다. 여기서 TTMO에는 "1"이 인가되어 있다. 여기서 INTV IRQ는 10msec마다 발생하는 것으로 하고 있으므로 TTMO에 "1"이 인가되어 있다고 하는 것은 타스크 레벨 "0" 프로그램(제5도의 252)은 10msec마다 기동됨을 나타낸다. 제15도의 카운터 CNTR0와 TTMO를 스텝(508)에서 비교하여 일치하는 경우 긍정되며 이 경우 스텝(510)으로 진행하여 타스크 콘트를 워드 TCWO의 b6에 표식 "1"을 세트한다.

이 실시예에서는 각 TCWO의 b6는 그 타스크의 기동요구의 표식 "1"이 세트된다. 스텝(510)에서 TCWO의 b6

에 표식 "1"을 세트하므로 이 TCW0의 b0~b5에 세트된 카운터 CNTR0를 클리어 한다.

스텝(512)에서 TASK 레벨 "0"으로부터 TASK 레벨 "1"의 기동 타이밍 검색이 실행된다. 스텝(514)에서 TASK 레벨 "3"이 종료 되었는지를 판단한다. 즉 n=4인지를 판단한다. 이 경우 n=10이므로 스텝(506)으로 되돌아 간다. 스텝(506)에서 TASK 레벨 "1"의 프로그램의 TASK 콘트롤 워어드인 제15도의 RAM(106)내의 TCW1의 카운터 CNTR1의 내용을 +1 증가시킨다. 스텝(508)에서 제15도의 ROM(104)의 TTM1과 비교한다. 이 실시예에서는 TTM1의 내용은 "2"이다. 즉, TASK 레벨 "1"의 기동 타이밍은 20msec이다. 지금 카운터 CNTR1의 내용이 "1"이라고 가정하면 스텝(508)의 판단은 "부정"되고, 즉 TASK 레벨 1 TASK(254)는 기동 타이밍이 아님이 판단되어 스텝(512)으로 진행한다. 여기서 다시 검색되는 TASK 레벨이 갱신되고 다음은 TASK 레벨 "2"로 된다.

마찬가지로 TASK 레벨 "3"까지 끝나면 스텝(512)에서 n=4가 되고 스텝(514)에서 n=MAX의 조건이 만족된다. 그리고 TASK 스케줄러(242)로 진행한다. 스텝(504)에서 INTV IRQ가 부정되면 스텝(518)으로 진행한다. 여기서 엔진 정지를 나타내는 ENST IRQ 인지의 여부를 판단한다. 스텝(504)에서 "부정"된 경우 반드시 ENST IRQ일 것이므로 스텝(518)을 생략하고 스텝(520)으로 진행해도 된다. 스텝(520)은 엔진정지에 의거한 특별한 프로그램으로 연료펌프를 정지시키고 다시 정확계, 연료계의 모든 출력계의 신호를 리세트 상태로 하고 제5도의 개시스텝(202)으로 되돌아 간다.

제14도는 TASK 스케줄러(242)의 상세한 플로우차트로서, 스텝(530)에서 TASK 레벨 "n"의 실행이 필요한가를 판단한다. 맨 처음에는 n=0이므로, TASK 레벨 "0"의 프로그램의 실행의 필요에 따라 판단된다. 즉, 우선 순위가 높은 TASK의 순으로 기동 요구의 존재를 조사해 나간다. 그것은 TASK 콘트롤 워어드 TCW0의 b6와 b7을 검색함으로써 판단할 수 있다. b6은 기동요구 표식으로서 "1"이 세트되어 있으면 기동요구가 "있음"을 알 수 있다. 또 b7에는 실행중을 나타내는 표식으로서 여기에 "1"이 세트되어 있으면 실행중인 현시점에서는 중단되어 있음을 나타낸다. 따라서 b6과 b7중 적어도 어느 한쪽에 "1"이 세트되어 있으면 실행이 필요하게 되어 스텝(538)으로 진행한다.

스텝(538)에서 b7의 표식을 판단하여, b7이 "1"이면 실행중단 중이므로 스텝(540)에서 중단하고 있던 실행을 재개한다. b6과 b7양쪽에 표식이 세트되어 있어도 스텝(538)의 판단이 긍정되면 중단중인 프로그램이 다시 기동된다. b6만이 "1"인 경우, 레벨의 기동요구 표식 즉, b6를 스텝(542)에서 클리어하여 스텝(544)에서 b7의 표식(이하 RUN표식이라 함)을 세트한다. 스텝(542)과 (544)는 TASK 레벨의 기동요구 상태에서부터 실행상태로 진행했음을 나타낸다. 스텝(546)에서 TASK 레벨프로그램의 스타아트 어드레스를 검색한다. 이것은 제15도에 도시한 ROM(104)내에 각 TASK 레벨의 TCW에 대응시켜 세트된 스타아트 어드레스 테이블 TSA에 의하여 구해지며 이 스타아트 어드레스로 점프함으로써 그 TASK의 실행이 행해진다.

제14도로 되돌아가, 스텝(530)에서 "부정"이라고 판단됐을 경우, 이 경우는 검색TASK 레벨의 프로그램에는 기동요구가 나와있지 않고, 또한 실행중단도 아님을 나타내고 있다. 이 경우 다음의 TASK 레벨의 검색으로 옮긴다.

즉, TASK 레벨의 n이 n+1로 되어 레벨이 하나 이동한다. 여기서 n이 MAX 즉, 여기서는 4인지를 결정하고, 4가 아니면 다시 스텝(530)으로 진행한다. 이를 반복하여, n=4가 되면 스텝(536)으로부터 배경조브의 개입중단 프로그램으로 되돌아간다.

즉, 스텝(536)에서는 TASK 레벨 0 내지 3까지의 모든 프로그램의 실행이 필요없음이 판명되므로 IRQ의 발생전 배경조브의 개입중단 프로그램으로 복귀한다.

제15도는 상술한 TASK 콘트롤 워어드 TCW와 ROM내의 TASK 기동 주기를 나타내는 TTM, TASK 스타아트 어드레스 테이블의 관계를 나타낸 것이다. TASK 콘트롤 워어드의 0~3에 대응하여 ROM내에 TASK 기동 주기 TTM이 있어 INTV IRQ마다 TCW의 카운터 CNTR이 연속갱신되어 각 TASK의 TTM과 일치함으로써, 그 TCW의 b6에 표식이 세트된다. 이 표식에 의하여 다음에 내의 TASK 스타아트 어드레스 TSA로부터 2TASK의 스타아트 어드레스가 검색되어 그 스타아트 어드레스로 점프함으로써 프로그램 1~4의 선택된 프로그램이 실행된다.

TASK 프로그램의 실행중에는 RAM(106)내의 프로그램에 대응된 TCW의 b7에 표식이 세트된다. 이 표식이 세트되어 있는 동안은 실행중임을 판단할 수 있다. 이와 같이 하여 제5도의 TASK 스케줄러(242)의 프로그램이 실행된다. 그리고 예컨대, 0~3까지의 프로그램(252~258)중의 어느 하나가 실행된다.

이 TASK의 실행중에 IRQ가 발생하면 다시 그 TASK를 중단시켜 IRQ의 처리를 한다. 지금 IRQ가 발생하지 않는다고 하면 실행중의 TASK의 처리는 드디어 종료된다. 이에 의해 종료 보고를 하기 위하여 EXIT(260)로 진행한다. 이 EXIT프로그램의 상세한 것을 제16도에 나타낸다. 이 프로그램은 종료TASK를 찾아내기 위한 스텝(562)과 (564)로 구성된다. 이 스텝(562), (564)에서 먼저 TASK 레벨의 "0"부터 검색하여 종료된 TASK 레벨을 찾아낸다.

이에 의해 스텝(568)을 진행하여 여기서 종료한 TASK의 콘트롤 워어드의 b7의 RUN표식을 리세트 한다. 이에 의해 프로그램의 실행이 완전히 끝난 것이 된다. 그리고 다시 TASK 스케줄러로 되돌아가 다음의 실행프로그램이 결정된다. 프로그램의 실행과 중단을 제17도로 다시 설명한다. 지금 가장 우선도가 낮은 배경조브(208)를 실행하고 있었다고 한다면, 시간(t<sub>1</sub>)에서 INTV IRQ가 발생하면 관리 프로그램(이하 OS프로그램이라함)으로 실행이 옮겨진다. 그리고 여기서 제5도의 STATUS레지스터(224) 및 TASK 스케줄러(242)의 실행에 의하여 예컨대, 레벨 "0" TASK(252)가 실행요구 됐음을 알게된다. 그리고 시간(t<sub>2</sub>)에서의 b7에 "1"이 세트되고 또한 b6가 클리어되며 레벨 "0" TASK(252)가 실행된다. 또 동시에 레벨 "1"인 프로그램도 기동 요구가 나왔다고 하면 TCW1의 b6에도 표식 "1"이 세트된다.

그러나 먼저 우선도가 높은 레벨 0 TASK(252)를 실행한다. 시간(t<sub>2</sub>)에서 이 레벨 0 TASK의 실행을 종료하면 다시 프로그램으로 되돌아가 TCW0의 b7의 표식을 클리어하여 다음의 레벨인 레벨 "1"의 TASK

실행요구의 검지와 실행준비가 되어  $t_4$ 에서 실행이 시작된다. 여기서 레벨 "1"의 타스크를 처리하는데 시간이 걸리면 다시 다음의 INTV IRQ가 발행하고,  $t_5$ 에서 레벨 1의 프로그램을 중단하여 OS프로그램으로 되돌아 간다. 그리고 OS 프로그램을 실행하여 이 INTV IRQ로 기동해야 할 프로그램 예컨대, 레벨 "0" 타스크(252)에 기동요구를 낸다. 그리고 우선도가 가장 높은 레벨 "0" 타스크(252)를 시간( $t_6$ )에 실행한다.

즉, TCW<sub>0</sub>의 b7에 표식 "1"을 세트하고 그리고 b6의 표식을 클리어하여 TSA0부터 레벨 "0" 타스크(252)의 스타아트 어드레스를 검색하고 레벨 "0" 타스크(252)를 실행한다. 시간( $t_7$ )에서 레벨 "0" 타스크(252)의 실행을 끝내고 프로그램으로 돌아가 EXIT에서 TCW<sub>0</sub>의 b7을 클리어한다.

그리고 OS프로그램의 타스크 스케줄러(242)에서 기동요구 또는 중단 프로그램을 검색한다. 이 타스크 스케줄러(242)에 의하여 레벨 "1" 타스크(254)의 중단을 검지하여 실행을 다시 수행한다. 레벨 "1" 타스크의 재기동은 시간( $t_5$ )에서 CPU(102)의 내용이 대기되어 있던 것을 시간( $t_8$ )에서 다시 CPU로 되돌림으로써 행해진다. 레벨 "1" 타스크(254)를 재기동시켜 시간( $t_9$ )에서 그 실행을 종료하면 OS프로그램으로 되돌아가 EXIT프로그램을 실행함으로써 프로그램(254)의 종료 보고가 된다.

즉, TCW<sub>1</sub>의 b7의 표식이 클리어된다. 계속하여, 타스크 스케줄러(242)의 실행에 의하여 기동 요구가 나와 있는 프로그램이나 실행 중단인 프로그램이 없는지의 여부를 검색하여 예컨대, 프로그램(256)의 TCW<sub>2</sub>의 b2에 표식 "1"이 세워져 있는가를 확인한다. 이에 의하여 레벨 "2"타스크(256)의 스타아트 어드레스인 TSA<sub>2</sub>를 검색하여 프로그램(256)의 스타아트 어드레스로 점프하고 프로그램의 레벨 "2" 타스크(256)를 실행한다. 이때 전술한 바와 같이 TCW<sub>2</sub>의 b<sub>6</sub>표식을 클리어 하여 b7의 표식을 세트한다. 그리고 시간( $t_{11}$ )에서 프로그램의 레벨 "2" 타스크(256)의 실행이 종료되면 OS 프로그램으로 되돌아가 프로그램에 의하여 TCW b7의 표식을 클리어 하여 종료보고 한다.

그리고 다시 기동 요구와 실행 중단의 검색을 행한다. 이 검색에 의하여 기동 요구나 실행 중단의 프로그램이 없다면 배경조브(208)의 대기 영역으로 부터 CPU에 각 데이터를 재공급하여 처리되게 한다. 그리고  $t_{13}$ 에서 INTV IRQ가 들어가면 CPU의 내용을 다시 대기영역에 전번시켜 OS 프로그램으로 되돌아가 IRQ의 요구분석을 하여 다시 프로그램에 기동 요구를 건다.

이상의 설명으로 부터 알 수 있는 바와 같이, 각 프로그램의 실행중 개입 중단이 발생하면 그 프로그램을 중단하고, 프로그램으로 되돌아가 가장 높은 레벨의 프로그램부터 실행한다. 이와 같이 함으로써 보다 중요한 프로그램에 우선적으로 처리 시간을 할당한다.

또 각프로그램을 우선도에 따라 각 레벨로 나누어, 각 레벨에 포함되는 각각의 처리 프로그램을 연속적으로 실행시켜 각각의 레벨의 프로그램에 대하여 그 전체적인 처리의 종료를 보고시키고 있으므로, OS프로그램이 차지하는 비율이 적어도 되고 처리효율이 향상한다. 또 이와 같이 함으로써 동일 레벨내의 프로그램에 대하여 상호간에 개입 중단이 발생하는 일이 없으므로 개입 중단을 위한 대기 영역을 각 레벨을 프로그램에 대응시켜 설정함으로써 실행이 가능하므로 대기 영역이 대폭적으로 감소한다.

제5도 및 제17도에서 설명한 각 레벨의 프로그램 즉 레벨 "0" 타스크(252) 레벨 "1" 타스크(254), 레벨 "2" 타스크(256), 레벨 3타스크(258)의 실행상태를 제18도에 나타낸다. IDLE상태는 기동 대기의 상태이고 프로그램에 대한 기동 요구가 나가지 않는 상태이다. 다음에 기동 요구가 나가면, TCW의 b6에 표식에 세트되어 기동이 필요함을 표시한다. IDLE상태로 부터 QUEUE상태로 이동하는 시간은 각 프로그램레벨에 따라 정해진다. 또 QUEUE상태로 되었어도 실행되는 순서는 우선도에 따라 정해진다.

프로그램이 실행 상태로 들어가는 것을 OS프로그램에서 TCW의 b6의 표식이 클리어되어 b7에 표식이 세트되기 때문이다. 이로써 프로그램의 실행이 시작된다. 이러한 상태가 RUN상태이다. 그리고 실행이 끝나면 TCW의 b7표식이 클리어 되어 종료 보고를 완료한다. 이로써 RUN상태는 끝나고 다시 IDLE상태가 되어 다음의 기동요구가 나오는 것을 기다린다. 그러나 프로그램의 실행중 즉, RUN중에 IRQ가 발생되면 그 프로그램은 실행을 중단하지 않으면 안된다. 이 때문에 CPU의 내용이 대기되어 실행이 중단된다.

이 상태가 READY상태이다. 다음에 개입중단 프로그램에 다시 실행되게한 상태가 되면 대기 영역에 대기해 있던 내용을 다시 CPU에 보내어 실행이 재개되게 한다. 즉 READY상태로 부터 다시 RUN상태로 되돌아 간다. 이와 같이 각 레벨 프로그램은 제18도의 4가지의 상태를 반복한다. 제18도는 대표적인 흐름도이지만, READY상태에서 TCW의 b6에 표식 "1"이 세트될 가능성이 있다. 이것은 예컨대 기동중단 중에 프로그램의 다음 기동 요구 타이밍이 되어 버렸을 경우이다. 이때는 b7 표식이 우선되어 먼저 중단중인 타스크 프로그램을 실행한다. 이로 인하여 b7표식이 리세트되고 b6표식에 의하여 IDLE상태를 거치지 않고, QUEUE상태가 된다.

이와 같이 각 레벨 프로그램 즉, 레벨 타스크(252), (254), (256), (258)은 각각 제18도의 어느 한 상태에 있다. 또 각 레벨프로그램이 READY상태로 되는 것은 각 프로그램마다에 존재한다고 할 수 없고 각 레벨 프로그램 마다에 존재한다. 즉, 예컨대 레벨 "1" 타스크 프로그램은 제1표에 나타난 CARCB프로그램이나, IGNCAL 프로그램이나, DWLAL 프로그램으로 구성되어 있다. 그러므로 레벨 "1" 타스크 프로그램에 대하여 READY상태가 존재하지만 상기 각 CARCB, IGNCAL, DWLAL, 프로그램에 대하여서는 READY상태가 존재하는 것이 아니므로, 각 레벨 프로그램에 대응하여 대기영역을 갖도록 하면된다. 제15도의 RAM(106)의 대기영역(602)은 각 레벨에 대응한 크기만큼 설정되어 있다. 제19도는 배경조브의 상태변화를 나타내는 도면으로서, 실행중에 RUN상태이고, IRQ가 발생하면 CPU의 내용을 대기시켜 READY상태가 된다. 또, 개입 중단 처리가 종료하면 대기해 있던 내용을 CPU로 보내 실행재개 상태 즉, RUN이 된다. IRQ의 발생으로 CPU의 내용을 대기시키는 경우 제15도의 RAM내에 나타난 대기영역(602)에 대기된다.

제20도는 레벨 "0"타스크의 프로그램이다. 이 프로그램은 제1표에 나타난 바와 같이 10m sec마다 기동요구가 나온다. 스텝(650)에서 ADC1의 데이터의 인출을 행하고 스텝(654)에서 ADC1의 다음의 데이터를 인출하기 위한 기동요구를 한다. 또, 652의 스텝은 시동전에 ADC END IRQ를 실행시키도록한 것이고, 시동

전인 표시가 나와있을 경우 시동전이기 때문에 RT1 즉, 개입중단 프로그램으로 되돌아간다. 이 프로그램은 제5도의 INITIALIZ(204)이다. 스텝(656)에서 ADC2의 데이터를 인출하고 다시 스텝(658)에서 ADC2의 다음의 데이터 인입을 위한 기동을 세트되게 한다. 스텝(660)에서 엔진 속도를 인출 한다. 이들의 스텝이 종료되면 OS의 EXIT프로그램을 실행하여 TCW의 b7의 표시를 클리어 한다.

제21도는 레벨 1타스크의 프로그램이고, 스텝(672)에서 시동중인지의 여부를 판단한다. 시동중인 경우 시동중인 연료를 공급하고 또한 점화기시를 결정함으로 계산할 필요가 없고, 스텝(678)으로 점프한다. 스텝(674)에서 연료를 계산하기 위한 CARBCAL 프로그램을 실행하고, 다음에 스텝(676)에서 IGNCAL 프로그램을 실행한다. 이들은 각각 테이블 검색으로 행해진다. 스텝(678)에서 DWALCAL 프로그램을 실행하여, 이것으로 통전 시간을 계산한다. 레벨 "2"의 LAMBDA 프로그램은  $\lambda$ 의 보정 프로그램이고, 레벨 "3"의 HOSEI 프로그램은 각 보정 계수의 프로그램이다. 보정 계수로서는 예컨대, 대기온도나 수온등의 보정 계수를 구하는 프로그램이고, 이들의 매개변수는 특정의 시간상수가 크므로 긴 시간의 간격으로 매개변수를 계수하면 된다.

또 본 발명은 상술한 바와 같이 모든 연산제어가 엔진의 회전수에 무관하게 행해지도록 INTV IRQ를 발행하고 있다. 이와 같은 IRQ의 발생회로의 구성을 제22도에 나타낸다. 도면에서 레지스터(735)는 CPU로 부터 타이머 개입중단 주기를 설정하는 데이터(예를들면, 10msec)가 데이터 버스(752)를 통하여 세트되고, 동시에 카운터(736)에는 클럭펄스의 CLOCK이 입력된다. 그리고 이 카운트수와 상기 레지스터에 세트된 내용은 비교기(737)에서 비교되고, 레지스터(735)의 내용과 카운터(736)의 내용이 같아졌을 때, 비교기(737)에서 출력이 발생되어, 플립플롭(738) 및 (740)을 세트한다.

이와 동시에 AND회로 (747)의 출력으로 카운터(736)와 플립플롭(738)을 리세트되게 한다. 한편, 플립플롭(739)이 세트되어 있으면 타이머 개입중단 신호 IRQ는 AND회로 (748), OR회로(751)를 거쳐 출력된다. 여기서 상기 플립플롭(739)은 상기 IRQ신호를 불필요로 할때(예를들면, 엔진 시동중), IRQ를 마스크(MASK)하기 위한 플립플롭이다. 그리고 이때는 CPU로 부터 플립플롭(739)에 대하여 리세트 지령을 보낸다.

또 상기한 타이머 개입중단에 대하여 엔진의 회전이 운전중에 고장 또는 조작상의 잘못으로 정지되는 등의 엔진 고장시 엔진고장(ENST) 개입중단 요구는 다음과 같은 구성에 의하여 발생된다. 이 구성은 타이머 개입중단의 경우와 동일하며 레지스터(741), 카운터(742), 비교기(743), AND회로 (749), (750), 플립플롭(744), (745), (746)으로 구성되고, 동작도 상술한 경우와 동일하다. 단, 카운터(742)에 입력되는 신호는 엔진이 회전하고 있을 때에 발생하는 신호로서, 제4도에 도시한 센서(146)로 부터 발생하는 크랭크의 기준각 신호 REF이며, 예를들면 4기통인 엔진 일때는 크랭크가 180° 회전할때마다 발생하는 신호이다. 그리고 카운터(742)는 상기 RES신호가 인가되고 있는 동안 리세트 되기 때문에 엔진 고장 개입중단 신호는 발생하지 않고, 엔진이 정지하면 REF신호가 정지되어 리세트가 해제되기 때문에 엔진 고장 개입중단신호가 발생한다.

이상 설명한 바와 같은 타이머 개입중단 IRQ는 상술한 제5도의 플로우차트에 나타낸 바와 같이, 타스크의 기동을 트리거하여, 타스크의 우선 처리가 행해진다. 즉 CPU가 개입중단 요구를 받으며 그 개입중단 요구의 판정을 행한다. 그리고 이것이 타이머 개입중단이면 우선도가 높은 순으로 레벨이 분류된 타스크군(252), (254), (256), (258)의 기동을 트리거하여 타스크 스케줄러(242)에 의해 선택한 타스크의 실행이 행해진다. 타스크 실행이 종료되면 EXIT에서 종료 보고되고, 다시 다음의 타이머 개입 중단과 타스크 스케줄리에 의하여 다음의 타스크가 선택된다. 한편, 개입중단이 엔진 고장 개입 중단인 경우는 연료펌프를 오프로 함과 동시에 점화장치를 리세트하여 임출력 제어회로를 모두 디스에이블로 한다.

ADC1 END IRQ와 ADC2 END IRQ도 동일하며, ADC1의 사이켄스 동작이 종료하면 플립플롭(764)에 "1"이 세트된다. 그리고 CPU로 부터 버스라인(752)을 통하여 플립플롭(762)에 "1"이 세트되면 AND게이트(770)가 이네이블되고 OR게이트 IRQ를 거쳐 CPU에 ADC1 END IRQ의 서비스를 요구한다. 그러나 플립플롭(762)에 "1"이 세트되어 있지 않을 경우는 ADC1 END IRQ는 금지된다. ADC2에 관해서도 동일하게 적용되며, ADC2의 사이켄스의 종료로 플립플롭(768)에 "1"이 세트된다.

이때 플립플롭(766)에 "1"이 세트되어 있으면, ADC2 END IRQ는 AND게이트(772)와 OR 게이트(751)에 의하여 발생하나 플립플롭(766)에 "1"이 세트되어 있지 않으면, AND게이트(772)가 디스에이블 상태로 되므로, ADC2 END IRQ는 발생하지 않는다. 따라서 플립플롭(739), (745), (762), (766)에 대하여 "1"을 세정한 것만의 IRQ가 발생하여 "0"을 세트하면 IRQ의 발생을 금지하는 상태가 된다.

이상 설명한 바와 같이 본 발명에 의하면 제1표에 나타내는 프로그램의 처리기능에 따라 그 우선도가 결정된다. 그리고 우선도에 따른 소정 파라미터의 시간 간격으로 기동 요구가 출력된다. 이와 같이 엔진제어의 주요 타스크를 엔진 회전속도에 관계없이 일정한 시간간격으로 기동되기 때문에 엔진 상태가 변화해도 CPU의 부하가 거의 변동하지 않고 항상 고성능인 제어가 가능하다.

또, 제5도의 실시예에 의하면 각 레벨 프로그램마다 기동주기를 변화시키고 있으므로 제어효율이 더욱 향상한다. 또, 동일 레벨인 프로그램에 대해서는 그 프로그램의 종료시마다 EXIT보고시키고 있으므로 동일 레벨내에서 다음의 프로그램으로 진행하는 데 하등의 OS 프로그램의 실행을 요하지 않는다. 따라서 처리시간이 빨라진다. 또 이로 인하여 동일 레벨내의 프로그램 상호간의 개입중단이 서로 걸리지 않으므로 대기영역의 설정을 각 타스크 레벨에 따라 행하면 되므로 영역이 작아도 된다.

엔진이 시동이 된 후, ADC1END 개입중단이나 ADC2 END 개입중단이 마스크되므로, OS 프로그램의 실행시간을 단축할 수가 있어 처리효율을 향상시킨다. 제15도에 나타내는 실시예에 의하면, TCW는 각 레벨에 대하여 설정하면되고, 제1표의 각 프로그램에 대하여 설정할 필요가 없으므로 TCW의 영역이 작아도 된다. 그리고 TCW에는 기동 요구를 나타내는 표시 b6와 실행중임을 나타내는 표시 b7이 세트되어 있으므로 제17도와 제18도에 나타낸 바와 같이 프로그램의 실행과 중단 그리고 재개가 순조롭고, 또한 우선도가 높게 할 수 있고, CPU의 제어효율이 향상한다.

다음에 본 발명을 연료분사형의 내연기관에 적용한 경우의 제2 실시예에 관해 제23도~제33도를 참조하

여 설명한다. 제1 실시예와 제2 실시예와는 내연 기관의 형이 다를뿐 이므로 다음에 기술하는 상이점 이외는 동일하다.

특히 본 발명의 본질적 제어방법에 관해서는 거의 동일하다. 제23도에는 엔진 계통 전체의 제어장치가 도시되어 있다. 도면에서 흡입 공기는 공기청정기(1002), 드로틀 챔버(1004), 흡기관(1006)을 거쳐 실린더(1008)에 공급된다. 실린더(1008)에서 연소한 가스는 실린더(1008)로부터 배기관(1010)을 통하여 대기중에 배출된다.

드로틀 챔버(1004)에는 연료를 분사하기 위한 분사기(1012)가 설치되어 있고, 이 분사기(1012)로부터 분출한 연료는 드로틀 챔버(1004)의 공기 통로내에서 분무되면서 흡입공기와 혼합되어 혼합기를 형성하는데, 이 혼합기는 흡기관(1006)을 통하여 흡기 밸브(1020)의 밸브를 열어줌으로써 실린더(1008)의 연소실에 공급된다.

분사기(1012)의 출구 근방에는 드로틀 밸브(1014), (1016)가 설치되어 있다. 드로틀 밸브(1014)는 가속 페달과 기계적으로 연동하도록 구성되고 운전자에 의하여 구동된다. 한편, 드로틀 밸브(1016)는 다이아프램(1018)에 의하여 구동되도록 배치되고 공기유량이 적은 영역에서 전폐 상태가 되어 공기 유량이 증가함에 따라 다이아프램(1018)내의 부압이 증가하게 되므로 드로틀 밸브(1016)는 열리기 시작하며, 흡입저항의 증가를 억제한다. 드로틀 챔버(4)의 드로틀 밸브(1014), (1016)의 상류에는 공기통로(1022)가 설치되고, 이 공기통로(1022)에는 열전대식 공기유량 장치로 구성시킨 전기적 발열체(1024)가 배치되며, 공기유속과 발열체의 전열량과의 관계로부터 정해진 공기 유속에 따라 변화하는 전기 신호가 인출된다. 발열체(1024)는 바이패스 공기통로(1022)내에 설치되어 있으므로 실린더(1008)의 역화(back fire)발생시에 생기는 고온가스로부터 보호함과 동시에 흡입 공기 중의 먼지 등에 의하여 오염되는 것으로부터도 보호된다. 이 바이패스 공기통로(1022)의 출구는 벤츄리의 최대협소부의 근방에 형성되고, 그 입구는 벤츄리의 상류측에 형성되어 있다. 분사기(1012)에 공급되는 연료는 연료탱크(1030)로부터 연료펌프(1032), 연료댐퍼(1034) 및 필터(1036)를 거쳐 연료압력 레귤레이터(1038)에 공급된다. 한편, 연료압력 레귤레이터(1038)로부터는 분사기(1012)에 파이프(1040)를 거쳐 가압 연료가 공급되고, 그 분사기(1012)로부터 요연료가 분사되는 흡기관(1006)의 압력과 상기 분사기(1012)에서의 연료압력의 차가 항상 일정하게 되도록 연료압력 레귤레이터(1038)로부터 연료탱크(1030)에 피드백 파이프(1042)를 거쳐 연료가 되돌려지도록 되어 있다. 흡기밸브(1020)로부터 흡입된 혼합기는 피스톤(1050)에 의하여 압축되어 점화 플러그(1052)에서 발생하는 스파아크에 의하여 연소하며 이 연소는 운동 에너지로 변환된다. 실린더(1008)는 냉각수(1054)에 의하여 냉각되고, 이 냉각수의 온도는 수온센서(1056)에 의하여 계측되며, 이 계측치는 엔진 온도로서 이용된다. 점화 플러그(1052)에는 점화 코일(1058)로부터 점화 타미에 맞춰 고전압이 공급된다.

또 도시하지 않은 크랭크 축에는 엔진의 회전수에 따라 기준 크랭크각마다 또는 일정각도(예를들면 0.5도)마다 기준 각신호 및 위치신호를 내는 크랭크 각 센서가 설치되어 있다. 이 크랭크각 센서의 출력, 수온센서(1056)의 출력(1056 A) 및 발열체(1024)로부터의 전기 신호는 마이크로 컴퓨터등으로 구성되는 제어회로(1064)에 입력되고, 제어회로(1064)에서 연산처리되며, 이 제어회로(1064)의 출력에 의하여 분사기(1012) 및 점화코일(1058)이 구동된다.

이상의 구성에 의해 제어되는 엔진 장치에 있어서, 드로틀 챔버(1004)에는 드로틀의 드로틀 밸브(1016)를 지나 흡기관(1006)에 연통하는 바이패스통로(1026)가 형성되며, 이 바이패스통로(1026)에는 개폐 제어되는 바이패스 밸브(1062)가 설치되어 있다. 이 바이패스 밸브(1062)의 구동에는 상기 제어회로(1064)의 제어입력이 공급되어 개폐제어되도록 되어 있다.

이 바이패스 밸브(1062)는 드로틀 밸브(1016)를 우회하여 설치된 바이패스통로(1026)에 설치되고 펄스전류에 의하여 개폐제어가 된다. 이 바이패스 밸브(1062)는 밸브의 편의량에 의하여 바이패스통로(1026)의 단면적을 변경하며, 이 편의량은 제어회로(1064)의 출력에 의하여 구동장치가 구동 제어되게 한다.

즉 제어회로(1064)에서는 구동장치의 제어를 하기 위하여 개폐주기 신호가 발생되며, 구동장치는 이 개폐주기 신호에 의하여 바이패스 밸브(1064)의 편의량을 조절하기 위한 제어신호를 바이패스 밸브(1062)의 구동부에 부여하는 것이다.

제24도는 제어장치의 전체 구성도이다. 제24도에 있어서 제4도와 같거나 또는 동일한 구성품은 제4도의 경우와 동일한 참조번호가 병기되어 있다. 또 제4도와 다른 부품에 대하여서만 설명한다.

유량센서로 구성시킨 발열체(1024)(이하 AFS라함)의 출력은 ADC 2에 입력되며, 아날로그-디지를 변환회로(172)(이하 ADC라 함)를 거쳐 디지를 변환되어 레지스터(174)(이하 REG라함)에 세트된다.

다음에 CPU의 연산결과에 의거한 펄스 출력회로 및 제어대상에 관하여 설명한다. 분사기 제어회로(INJC 라함)는 연산결과에 의거한 펄스 출력회로로 변환하는 회로이다. 따라서 연료분사량에 상당한 펄스폭을 갖는 펄스가 INJC(1134)에서 만들어져 AND게이트(1136)를 거쳐 분사기(1012)에 인가된다.

점화펄스 발생회로(1138)(이하 IGNC라함)는 점화시기를 세트하는 레지스터(ADV 라함)와 점화코일의 1차 전류 통전개시 시간을 세트하는 레지스터(DWL라함)등을 가지며, CPU로부터 이들 데이터가 세트된다. 세트된 데이터에 의거하여 펄스를 발생시켜 제2도에서 상술한 증폭기(62)에 AND게이트(1140)를 거쳐 이 펄스를 인가한다.

바이패스 밸브(1062)의 밸브 개방 정도는 제어회로(이하 ISCC라함)(1142)로부터 AND게이트(1144)를 거쳐 공급된 펄스에 의하여 제어된다. ISCC(1142)는 펄스폭을 세트하는 레지스터 ISCD와 반복펄스주기를 세트하는 레지스터 ISCP를 갖고 있다.

제3도에 도시한 EGR제어밸브(86)를 교대로 제어하는 트랜지스터(90)를 제어하도록 EGR량 제어펄스 발생회로(178)(이하 EGRC라함)에는 펄스의 듀티 사이클을 나타내는 값을 세트하는 레지스터 EGRD와 펄스의 반

복주기를 나타내는 값을 세트하는 레지스터 EGRP를 갖고 있다. 이 EGRC의 출력펄스는 AND게이트(1156)를 거쳐 트랜지스터(90)에 인가된다.

또, 1비트의 입출력신호는 회로 D10 (128)에 의하여 제어된다. 입력신호로서는 IDLE-SW신호, TOP-SW신호, START-SW신호가 있다. 또 출력신호로서는 연료펌프를 구동하기 위한 펄스출력신호가 있다. 이 D10은 단자를 입력단으로서 사용할 것인가, 출력단자로서 사용할 것인가를 결정하기 위한 레지스터 DDR(192)와, 출력데이터를 래치하기 위한 레지스터 DOUT(194)가 설치되어 있다.

모우드 레지스터(1160)는 입출력회로(108)내부의 각종의 상태를 지령하는 명령기억 레지스터(이하 MOD라 함)인데 예컨대, 이 모우드 레지스터(1160)에 명령을 세트함으로써 AND 게이트(1136), (1140), (1144), (1156)를 모두 이 케이블 상태로 한다든가, 디스에이블 상태로 한다. 이와 같이 모우드 레지스터(1160)에 명령을 세트함으로써 INJCL나 IGNC 및 ISCC의 출력신호의 발생과 종료를 제어할 수가 있다.

각 TASK의 기동과 그 기능을 나타내는 상술한 제1표에서의 레벨 "1" TASK 프로그램 CARBC는 기화기를 사용하지 않은 본 실시예에서는 불필요한 것이고, 그 대신 프로그램 INJC가 대체되지 않으면 안된다. 이 프로그램 INJC는 연료분사량의 연산을 그 기능으로 하고 그 프로그램의 기동은 20msec마다 발생하는 개입중단요인 INTV IRQ에 의하여 행해진다. 기타의 프로그램에 관해서는 제1 실시예의 경우와 동일하다.

제1 실시예의 제7도에 나타난 INITIALIZ 프로그램과 서로 다른점에 관해서만 말하면, 제25도의 스텝(290)에서는 센서(1056)의 출력의 아날로그-디지틀 변환이 완료되면 다시 프로그램 AD11N이 실행되어 레지스터 REG(166)내에 세트되어 있는 수온 센서 TWS(134)의 출력의 디지틀 값을 인출하고 제1 실시예와 마찬가지로 ROM(104)의 DATA영역에 기억된다. 또 스텝(296')에서 시동시의 공기 바이패스 밸브(1062)의 개방정도의 연산을 행한다. 이 연산은 제1 실시예와 마찬가지로 제11도에 나타내는 특성에 의거하여 행해지며, 이 연산출력은 레지스터 ISCD에 세트된다.

스텝(298')에서 연료분사시간의 초기치가 연산된다. 이 연산치는 제26도에 의하여 행회지며 레지스터 INJD1에 설정된다. 제26도는 수온에 대한 연료분사량의 미리 설정된 값을 나타내는 것이다. 이 도면은 제1 실시예의 제10도에 대응하는 것이다.

배경조브의 프로그램은 제27도에 나타내는 플로우차트에 따라 실행된다. 제27도에 있어서, 스텝(410)에서 IDLE-SW(148)가 온인지의 여부를 판단한다. 만일 온이면 배기가스 환류를 행할 수 없다.

따라서 스텝(412)으로 진행하여 EGR 레지스터에 "0"을 세트한다. 스텝(414)에서 냉각수온에 따라 공기 바이패스 밸브(1062)의 듀티사이클을 구하고 스텝(416)에서 이 듀티사이클을 ISCD 레지스터에 세트한다. 이 세트치에 따라 엔진의 공기 바이패스량이 결정된다. 스텝(416)이 끝남에 따라 다시 스텝(410)으로 진행하며, CPU에 대한 IRQ의 서어비스의 요구가 나오지 않는한 이 페루우프에서 상기 스텝이 반복된다.

한편, IDLE-SW가 오프되면 ISC는 실행할 수 없다. 따라서, 스텝(420)에서 ISCD 레지스터에 "0"을 세트한다. 그리고 이 상태에서 EGR량의 연산이 행해진다. 이때문에 냉각수온 TW가 일정온도 TA°C보다 높은지의 여부가 판단된다. 만일 높은 경우 EGR을 CUT 상태로 하기 위하여 스텝(426)으로 진행하여 EGR 레지스터에 "0"을 세트한다.

또 TW가 TA보다 낮은 경우 스텝(424)으로 진행하여 일정온도 TB보다 낮은지의 여부를 판단하여 낮은 경우에도 EGR을 CUT한다. 따라서 스텝(426)으로 진행하여 EGR에 "0"을 세트한다. 스텝(422)의 온도레벨 TA는 상한의 온도를 나타내고, 한편 스텝(424)의 TB는 하한온도를 나타내며, 이 사이의 온도범위에서만 EGR작동을 한다. 따라서 이 사이의 범위에서는 스텝(428)으로 진행하며, 여기서 흡입공기량과 엔진회전속도 N으로부터 맵검색에 의하여 EGR량을 연산한다.

이에 대한 맵은 제6도의 ROM의 어드레스 B700-B7FF에 설명되어 있다. 이 검색치는 스텝(430)에서 EGR 레지스터에 세트된다. 이에 의하여 EGR 레지스터와 미리 세트되어 있는 EGRP 레지스터의 듀티사이클에 의한 값으로 EGR 밸브가 열려 EGR이 수행된다.

제27도에 나타낸 플로우 차트에서는 스텝(430) 또는 스텝(416)이 종료함에 따라 다시 스텝(410)으로 되돌아간다. 이와 같이 함으로써 계산기는 공기바이패스 밸브(1062)를 제어하기 위한 스텝(410)부터 스텝(416)까지의 플로우차트나, 또는 EGR량을 제어하기 위한 스텝(420)에서 스텝(430)까지의 스텝플로우차트를 항상 실행한다.

따라서 제1 실시예와 마찬가지로 IRQ 등의 발생이 생기지 않는 것이라 하면, 개시스텝(202)(제5도)으로부터 출발한 프로그램은 INITIALIZ(204), MONIT(206)을 거쳐 배경조브(208)인 ISCC0 프로그램 또는, EGRCON 프로그램을 항상 실행을 계속하는 것이 된다.

제28도는 레벨 "1"의 TASK 프로그램을 실행하는 플로우차트로서 스텝(672)에서 시동중인지의 여부를 판단한다. 시동중인 경우는 다른 프로그램에서 시동중의 연료나 정화시기가 결정되므로 이 프로그램군에서 계산할 필요가 없다. 따라서 제8도의 MONIT 프로그램의 스텝(308)과 (310)에서 INTV IRQ의 금지가 해제되어 SFTMR1 프로그램군이 기동했을 경우, 기동표식이 세트되어 있으므로 스텝(672)에서 "공정"으로 되면 스텝(678)으로 진행하고, 스텝(678)에서 점화코일의 일차전류 통전시간의 계산을 행하여 이 프로그램군의 연산을 종료한다. 시동중이 아닌 경우, 스텝(672)에서 스텝(674)으로 진행한다. 이 스텝(674)에서 INJC 프로그램을 실행한다. 여기에서는 SFTMR에서 구한 흡입공기량과 엔진회전속도 N으로부터 제6도에 나타내는 AF 맵에 의하여 연료분사량을 구한다. 이 분사량에 LAMBDA 프로그램에 의한 다른 계수(factors)로서 수정하고, 또 HOSE1 프로그램에 의한 그외의 보정을 가하여 그 값을 INJD 레지스터에 세트한다. 이 INJD의 프로그램을 실행 완료하면 스텝(676)에서는 IGNCAL 프로그램을 실행한다. IGNCAL 프로그램을 수행하므로 제6도에 도시한 맵으로부터 검색된 계수 QA와 NE에 의거하여 정화시기를 구하고 그 연산결과를 ADV 레지스터에 세트한다. IGNCAL 프로그램의 종료에 따라 스텝(678)으로 진행하여 DWL CAL 프로그램을 실행하고, 점화코일의 1차 코일전류 통전개시점을 구하여 그 값을 DWL 레지스터에 기억시킨다.

이상 INJC, IGNCAL, DWL CAL의 각 프로그램의 실행을 끝내고 종료보고를 위한 EXIT 프로그램으로 점프하여

종료보고가 된다.

제29도는 제4도의 CABD(165), FSC(176), EGRC(178) 및 제24도의 ISCC(1142)와 EGRC(178)의 상세도이다. 이 도면에서 CABD, FSCD, ISCD와 EGRD의 각 레지스터는 펄스폭을 나타내는 것으로서, 레지스터(802)에 해당한다. 또, CABP, FSCP, IFSCP와 EGRP에 해당하는 레지스터(806)가 있다.

지금 모우드 레지스터(1160)의 비트 bo에 고논리 "H"가 세트되어 있는 것으로 한다. 이때문에 AND게이트(1144)(도면 제24)와 (816)이 함께 동작상태이다. 카운터회로로 구성되는 타이머(804)는 AND게이트(816)로부터의 클록을 계수한다. 이 계수치 B는 레지스터(806)의 값과 비교기(810)에서 비교되어 계수치 B의 값이 레지스터(806)의 값이상이 되면 타이머(804)는 클리어된다. 따라서 타이머(804)는 레지스터(806)의 값 C에서 정하는 주기로 계수를 반복한다.

또, 타이머(804)의 계수치는 레지스터(802)의 값과 비교기(808)에서 비교된다. 이때, 레지스터(802)의 값 A가 타이머(804)의 계수치 B보다 큰 조건에서 플립플롭(812)을 세트하여 값 B가 A이상이 되는 조건에서 리세트된다. 이때문에 플립플롭(812)의 세트시간은 레지스터(802)의 값 A에 따라 정해진다. 이값 A를 크게함으로써 플립플롭(812)의 세트시간은 길어진다.

또 상술한 바와 같이 타이머(804)의 계수는 레지스터(806)의 세트치에 따른 주파수로 반복되므로, 플립플롭(812)의 세트출력은 레지스터(806)의 세트치에 의하여 반복하여 주기에 따라 반복 출력된다. 모우드 레지스터(1160)(제24도)의 bo비트가 고논리 "H"레벨이므로 AND게이트(1144)를 거쳐 출력된다. 모우드 레지스터(1160)의 bo를 저논리 "L"로 하면 게이트(1144)와 (816)는 디스에이블되며 플립플롭(812)의 출력은 정지되며 동시에 타이머(804)에의 입력도 정지된다.

따라서 제24도에 나타내는 모우드 레지스터에 CPU로부터 제어데이터를 세트함으로써 제29도의 회로 동작의 개시 또는 정지를 제어할 수가 있다. 제29도는 모우드 레지스터의 bo에 의하여 AND게이트(1144)와 (816)이 제어되는 예이나, bo는 제24도의 ISCC(1142)의 ISCD를 제어하는 비트이다. 제24도의 EGRC(178)도 제29도와 동일한 구성이나, (1142)의 동작의 시동과 정지는 모우드 레지스터의 bo비트로 제어되고 EGRC(178)는 b2비트로 제어된다.

제30도는 제4도 IGNC(168) 또는 제24도의 IGNC(1138)의 상세도이다. CPU로부터 점화코일의 1차전류 통전 개시시간을 제어하는 데이터가 DWL레지스터에 세트되어, 점화시기를 나타내는 데이터가 ADV레지스터에 세트된다. 지금 DWL레지스터(1168)의 세트치를 A. ADV레지스터(1169)의 세트치를 C라 한다.

지금 모우드 레지스터(1160)의 b1비트가 고논리 "H"이면, AND게이트(1156)와 (860)은 신호를 전달하는 상태가 되어 있고 POS펄스가 AND게이트(860)를 거쳐 카운터(850)에 입력된다. 이 카운터(850)의 카운트치는 이때문에 엔진크랭크각에 따라 증가하여 기준각을 나타내는 REF펄스에 의하여 클리어된다. 이 계수치를 B라 한다. 계수치가 작을때는 비교기(852)의  $A > B$ 의 출력이 OR게이트를 거쳐 플립플롭(856)에 공급되어 플립플롭(856)은 리세트상태가 된다.

따라서 AND게이트(1156)로부터는 펄스출력이 나오지 않는다. 카운터(850)의 계수치가 커지면 DWL레지스터(1168)의 세트치보다 카운터(850)의 계수치가 커지면 AND게이트(864)로부터의 출력에 의하여 플립플롭(856)은 세트된다. 이 세트출력이 게이트(1156)를 거쳐 점화장치에 인가되고, 점화코일에 1차전류가 흐른다. 계수치가 더 높아지면 비교기(854)의  $C \neq B$ 출력에 의하여 플립플롭(856)은 다시 리세트된다. 이로 인하여 AND게이트(858)으로부터 펄스출력은 개입중단되고 점화를 위한 스파이크가 발생된다.

제4도의 D10(128) 및 제24도의 D10(128)를 상세히 제31도에 도시한다. 이 도면에서 DDR는 D10의 입출력 포트 D100~DE07을 입력상태로 하느냐 출력상태로 하느냐를 결정하는 것이다.

DDR중 고논리 "H"가 세트된 비트로부터의 신호가 그에 대응한 3치상태(Tri State)의 구동기(872~886)에 가해져 그 3치상태의 구동기는 도통상태가 된다. 이에 의하여 DDR의 고논리 "H"의 비트에 대응한 DOUT의 비트는 대응하고 있는 질문상태를 거쳐 출력된다.

한편, 라인 D100~D107의 신호는 CPU로부터 버퍼증폭기(892~904)를 거쳐 자유로이 읽어낼 수가 있다. 3치상태의 구동기(872~886)중 부도통이 되어 있는 3치상태의 구동기에 대응하는 라인의 신호는 외부의 상태에 의존하므로 이 라인에 대해서는 외부의 상태를 읽어낼 수가 있다.

제32도는 제24도의 INJC(1134)의 상세도로서, 크랭크각 센서로부터의 REF펄스는 크랭크각의 상사점 직전의 일정각도(예를 들면, 80도라든지 90도)에서 발생한다. 크랭크각의 상사점(TDC)과 REF의 관계를 제33도의 A, B를 나타낸다.

지금 모우드 레지스터(1160)의 b4비트가 고논리 "H"가 되어 있다고 가정하므로 게이트(910), (912), (1136)(제24도)는 도전상태로 된다. 이때문에 카운터(904)의 계수치는 제33도에 나타내는 바와 같이 REF펄스마다 클리어된다. 레지스터(902)는 연료분사 개시점을 결정하는 값(A)을 CPU로부터 수신하여 기억하는 레지스터이다.

레지스터(902)의 값(A)은 계수치 B와 비교기(906)에 의하여 비교되고 플립플롭(908)으로 세트한다. 플립플롭(908)이 세트되어 있으면 게이트(1136)로부터 펄스가 보내져 분사밸브(1012)에 공급된다. 또 게이트(912)가 트리거되어 카운터로 이루어지는 타이머(916)는 클록펄스를 계수한다. INJD레지스터(914)는 제24도의 INJD레지스터이고, 이 레지스터의 세트치 C에 대응하는 시간동안 밸브는 열린다. 즉, 타이머(916)의 계수치 D가 C보다 작은 동안은 플립플롭(920)이 세트되어 있으나,  $C \neq D$ 인 조건에서 플립플롭(920)이 세트되고 AND게이트(1136)로부터의 분사용 펄스는 정지한다.

이와 같이하여 연료분사의 개시점과 밸브가 열리는 시간이 제어된다. 또 모우드 레지스터(1160)의 저논리 "0"(L)로 함으로써 게이트(1136)의 출력 및 모든 동작을 정지할 수가 있다. 이상과 같이 본 발명에 의하여 출력계의 동작의 시동과 정지를 CPU로 자유롭게 제어할 수가 있으므로 오동작이 생기지 않는다. 또 먼저 입출력회로의 입력회로를 동작시키기 위한 초기치 예컨대, 엔진회전속에 관한 측정을 위한 설정

치의 세트나 D10의 각 비트의 입출력의 지정등을 행하고 또 입출력회로(108)에서 CPU(102)에 대하여 서 어비스를 요구하기 위한 IRQ의 조건의 설정이나 입력회로의 IRQ의 금지해제를 행한다. 다음에 시동에 필요한 입력의 인출과 그를 근거로한 시동을 위한 세트치의 연산을 행한다. 이들의 세트치를 입출력회로(108)의 각 레지스터에 세트한 후, 시동기의 시동감시를 행한다. 시동기의 시동을 확인하고 입출력회로(108)의 출력측장치를 기동한다. 이와 같이 엔진을 기동하기 위한 준비를 미리 행하고 있으므로 시동에 관계없이 출력측 장치를 기동할 수 있고, 또 기동할 때까지 출력장치를 정지시키고 있으므로 엔진의 제어장치를 오동작시키는 일이 없다. 또 전력소비도 적고 입출력회로(108)나 그 출력을 증폭하기 위한 증폭기의 발열도 억제된다.

또 제8도의 MONIT프로그램의 스텝(308)에서 INTV IRQ의 금지를 해제하고 있으므로, 시동중 즉, 스텝(302)과 스텝(304)의 공정에 의하여 만들어지는 페루우프를 실행하는 중에 INTV IRQ를 수신하고 INTV IRQ에 의거하여 CPU(102)는 그보다 중요한 일을 행한다. 이에 의하여 입출력계의 각 레지스터 IDJD, ADV, DWL, ISCD, EGRD로부터의 새로운 입력에 의거하여 연산결과의 설정을 할 수가 있다. 시동모 우터의 시동에 의하여 축전지 전압이 변동하므로 그 변동에 따라 출력계의 레지스터의 설정치를 신속하게 변경시켜 주는 일은 대단히 중요하며 이것이 가능하게 된다.

본 발명에 의하면 연료분사의 연산을 타이머 개입중단으로 행하고 있으므로 연산장치의 부하가 엔진의 상태에 관계없이 안정해 있다.

(57) 청구의 범위

청구항 1

프로세서가 다수의 타스크를 수행함에 따라 출력신호를 발생시키고, 이들 타스크의 기능이 프로세서제어 되는 엔진운전에 영향을 미치게 하는데, 상기 타스크를 수행하는 실행을 통하여, 이들 타스크가 다수의 타스크 프로그램을 구비하게 한 전자식 엔진제어방법에 있어서, 타스크 프로그램을 실행하는 중에 프로 세서가 응답하는 연속적인 개입중단신호를 발생시키는 단계(a)와, 각각의 개입중단신호에 응답하여, 메모리내에 현재 타스크 프로그램을 실행할 것을 요구하는 실행요구신호를 기억시키는 단계(b)와, 단계(b)에서 실행요구신호의 기억이 종료되자마자 상기 메모리를 검색하여 실행요구 신호의 존재여부를 확인하고 단계(b)에서 실행요구신호를 기억시킨 선택된 타스크 프로그램의 실행을 개시하는 단계(c)등으로 이루어진 것을 특징으로 하는 전자식 엔진제어방법.

도면

도면1

