(19) 대한민국특허청(KR)
(12) 특허공보(B1)

(51) Int. Cl.[5]
A61L 2/24

(45) 공고일자 1995년01월09일
(11) 공고번호 특1995-0000062

(21) 출원번호	특1986-0700690	(65) 공개번호	특1987-0700153
(22) 출원일자	1986년10월04일	(43) 공개일자	1987년03월14일
(86) 국제출원번호	PCT/US 86/000258	(87) 국제공개번호	WO 86/04698
(86) 국제출원일자	1986년02월04일	(87) 국제공개일자	1986년08월14일

(30) 우선권주장 698434 1985년02월05일 미국(US)
(71) 출원인 더 스코파스 테크놀로지 컴패니 인코오포레이티드 로렌스 에스. 릭클스
미합중국 뉴욕 10165 뉴욕 이스트 42번 스트리트 60

(72) 발명자 제퍼리스, 레이몬드 피. 3세
미합중국 펜실바니아 19087 웨인 힐크레스트 로우드 276
엥글러, 필립 브이
미합중국 뉴욕 10591 테리타운 매플스트리트 21
로젠블라트, 아론 에이
미합중국 뉴욕 10591 뉴욕 웨스트 76번 스트리트 32

(74) 대리인 유영대, 나영환

심사관 : 공민호 (책자공보 제3840호)

(54) 가스 살균 시스템

요약

내용 없음.

대표도

도1

명세서

[발명의 명칭]

가스 살균 시스템

[도면의 간단한 설명]

이하에서는 본 발명을 첨부 도면을 참조하여 보다 상세히 설명하고자 한다.

제 1 도는 본 발명에 따른 전체 가스 살균 시스템의 블록 다이아그램이다.

제 2 도는 본 발명에 따른 가스 살균 시스템의 살균처리실, 밸브 및 펌프 블록에 대한 블록 다이아그램이다.

제 3 도는 가스 살균 시스템의 전자 제어 회로에 대한 블록도이다.

제 3a 도는 제 3 도의 전자 제어기에 사용된 어드레스 및 그 어드레스에 의해 제어되는 대응 성분 또는 신호에 대한 표이다.

제 4 도는 다양한 시스템 클럭 주파수 및 시스템 인터럽트가 유도되는 양태를 보여주는 블록도이다.

제 5 도는 제어기 표시등 및 제어 스위치를 보여주는 가스 살균 시스템에 대한 제어 배전반의 한 실시예에 대한 전면도이다.

제 6 도는 본 발명에 따른 가스 살균 시스템에 대한 상태도이다.

제 7 도는 본 발명에 따른 가스 살균 시스템에 대한 제 6 도의 상태도에 대응하는 상태 출력 매트릭스이다.

제 7a 및 7b 도는 제 6 도의 상태도를 구성하기 위한 시퀀싱 프로그램에 대한 플로우챠트이다.

제 8 도는 본 발명에 따른 가스 살균 시스템에 대한 안전 인터로크(interlock)구성의 블록도이다.

제 9 도는 본 발명에 따른 가스 살균 시스템의 전자 제어기의 메모리에 상주하는 소프트웨어에 대한 기능 순서도이다.

제 10 도는 가스 살균 시스템용 소프트웨어의 시간조정(timed)기능에 대한 순서도이다.

제 11 도는 가스 살균 시스템용 소프트웨어의 시간조정 기능중 하나에 대한 순서도이다.

제 12 도는 본 발명에 따른 가스 살균 시스템용 전자 제어회로의 데이터 메모리의 메모리 맵이다.

제 13 도는 본 발명에 따른 가스 살균 시스템용 전자 제어회로의 소프트웨어의 시간조정 기능중 다른 하나에 대한 플로우챠트이다.

제 14 도는 본 발명에 따른 가스 살균 시스템용 소프트웨어의 또 다른 시간 조정 기능에 대한 플로우챠트이다.

제 15 도는 본 발명에 따른 가스 살균 시스템용 소프트웨어의 또 다른 시간 조정 기능에 대한 플로우챠트이다.

제 16 도는 제어 유니트의 시간 조절 기능을 리세트하기 위한 제어 유니트에 작성된 프로그램에 대한 플로우챠트이다.

제 17 도는 본 발명에 따른 시스템으로부터 입력 데이터를 판독하기 위한 제어 유니트에 작성된 프로그램에 대한 플로우챠트이다.

제 18 도는 부품 고장의 경우 타임아웃 경보를 제공하기 위한 제어 유니트에 작성된 프로그램에 대한 플로우챠트이다.

제 19 도는 부품 고장의 경우 추가의 경보를 제공하기 위한 제어 유니트에 작성된 프로그램에 대한 플로우챠트이다.

제 20 도는 시스템의 제어된 부품으로 데이터를 기록하기 위한 제어 유니트에 작성된 프로그램에 대한 플로우챠트이다.

제 21 도는 제어된 시스템으로부터 아날로그 입력 데이터를 판독하기 위한 제어 유니트에 작성된 프로그램에 대한 플로우챠트이다.

제 22 도는 시스템의 다양한 시간 조정 기능을 제공하기 위한 제어 유니트에 작성된 프로그램에 대한 일반적인 플로우챠트이다.

제 23 도는 제 22 도의 프로그램의 일부에 대한 플로우챠트이다.

제 24 도는 시스템 출력을 제어하기 위한 제어 유니트에 작성된 프로그램에 대한 플로우챠트이다.

[발명의 상세한 설명]

[발명의 배경]

본 발명은 한정된 처리실로 가스를 전달하여 물질 및 물품을 살균하기 위한 시스템에 관한 것이며, 구체적으로 물품을 살균하는데 살균 가스를 사용하는 시스템, 예컨대 외부 물질에 의해 오염될 수 있는 기구 및 계기와 같은 의료 장치를 살균 가스로 살균하는 시스템에 관한 것이다. 본 발명의 시스템은 또한 필요에 따라 비의료용 물품 및 물질에도 사용될 수 있다. 본 발명의 시스템은 특히 살균유효량의 가스를 제공하도록 반응하는 두 성분을 본 발명의 장치에 의해 현장에서 조합시키는 가스 살균 시스템에 관한 것이다. 이로써 살균용 가스를 형성하도록 반응하는 성분들이 별도로 적재되므로 사고의 가능성은 최소화된다.

구체적으로 말하자면, 본 발명은 살균용 가스로서 이산화염소를 사용하는 시스템에 관한 것이다. 이산화염소 가스는 불안정할뿐만 아니라 인체에 유독하다. 예를 들면, 이산화염소는 시간이 경과함에 따라 그것의 구성 분자로 분해되므로, 용이하게 운반할 수 없다. 따라서, 이산화염소 가스를 운반하는 것은 바람직하지 못하다. 더욱이, 이산화염소 가스는 약각 폭발성이 있고 또한 접촉 분해를 일으키는 경향이 있다. 그러나 이산화염소 가스를 형성하도록 반응하는 성분들(예컨대, 염소산 나트륨 및 염소가스)는 비교적 용이하게 운반될 수 있고 살균용 이산화염소 가스를 제공하도록 현장에서 반응할 수 있다.

종래의 시스템은 통상 살균제로서 산화 에틸렌 가스를 사용해 왔다. 예로서 sybron사의 의약 생산부에서 제조된 castle 4040 산화 에틸렌살균제를 들 수 있다. 종래의 시스템에서는 산화 에틸렌이 살균용 가스로서 사용되었지만 이산화염소가 바람직한 살균제이다.

또한, 과거에 사용된 시스템은 보통 설계가 간단하고 장치의 신뢰도 유지와 사고에 대한 안전보호를 위한 고성능의 수단을 구비하고 있지 않았다. 아울러, 이러한 시스템은 용장성(redundancy)이 크지 않았으므로 시스템의 부품이 고장난 경우 살균 공정을 계속하기에 앞서 고장을 수리하는데 수동 조작이나 서비스원이 필요하였다.

[발명의 개요]

본 발명의 제 1 목적은 세균 박멸성, 포자 박멸성, 진균 박멸성 및 비루스 박멸성을 가진 가스를 사용하여 물품을 살균하는 살균 시스템을 제공하는 것이다.

본 발명의 제 2 목적은 반응에 의해 살균용 가스를 제공하는 2 이상의 성분을 유효량의 살균가스를 제공하도록 본 발명의 장치내의 적소에서 반응시키는 살균 시스템을 제공하는 것이다.

본 발명의 제 3 목적은 살균제가 이산화염소 가스인 살균 시스템을 제공하는 것이다.

본 발명의 제 4 목적은 적절한 용장성과 시스템의 신뢰도 및 안전을 유지하기 위한 수단을 가진 가스 살균시스템을 제공하는 것이다.

본 발명의 제 5 목적은 응용이 광범위하고 프로그래밍된 마이크로프로세서에 의해 제어되는 가스 살균 시스템을 제공하는 것이다.

본 발명의 한 실시예에 의하면, 본 발명의 상기 목적 및 그밖의 목적은 다음과 같은 물품을 가스로 처리하기 위한 시스템에 의해 달성되는 바, 그 시스템은 제 1 성분을 수용하기 위한 제 1 수단 ; 제 2 성분을 수용하기 위한 제 2 수단 ; 함께 반응하여 가스를 형성하는 제 1 및 제 2 성분 ; 가스를 형성하기 위해 두 성분을 함께 반응시키기 위한 수단 ; 물품을 한정된 처리실 수단에서 처리하도록 가스를 처리실 수단에 공급하기 위한 밸브 수단 ; 가스를 처리실 수단으로부터 제거하기 위한 수단 ; 상기 반응용 수단, 공급용 수단 및 제거용 수단을 제어하기 위한 것으로서, 물품을 가스에 의해 처리하고 그후 처리실 수단내의 대기를 허용가능한 표준 안전도 범위내로 만들기 위해 가스를 처리실 수단으로부터 제거하는 사이클을 정의하는 일련의 연속 상태를 통해 그 장치를 순환시키도록 예정된 순서의 단계들을 실행하는 컴퓨터 수단을 구비한 전자 제어기 수단을 포함한다.

본 발명의 다른 실시예에 있어서는, 처리하고자 하는 물품을 수용하기 위한 처리실 수단 ; 처리실 수단에 결합되어 가스를 처리실 수단에 공급하기 위한 밸브, 소정의 시간 간격후 가스를 처리실 수단으로부터 제거하기 위한 수단 및 상기 밸브 수단과 제거용 수단을 제어하기 위하여 처리실 수단으로부터 측정된 파라미터중 임의의 파라미터와 관련된 복수의 전기 신호를 수신하기 위한 전자 제어 수단을 포함하여 가스를 처리실 수단에 공급하기 위한 수단을 포함하며, 상기 전자 제어 수단은 소정의 명령 시퀀스에 따라 복수의 상태를 통하여 장치를 순환시키기 위한 컴퓨터 수단을 구비하고, 그 컴퓨터 수단은 장치의 고장에 응답하여 복수의 정의된 고장 상태중 한 상태에 대해 장치의 동작을 어보트(abort)시키기 위한 수단을 구비하는, 물품을 가스로 처리하기 위한 시스템이 제공된다.

본 발명의 또 다른 실시예에 의하면, 물건을 가스로 처리하기 위한 시스템이 제공되는바, 그 시스템은 처리하고자 하는 물품을 수용하기 위한 처리실 수단 ; 가스를 처리실 수단에 공급하기 위하여 처리실 수단에 결합된 밸브 수단, 소정의 시간 간격후에 처리실 수단으로부터 가스를 제거하기 위한 수단, 및 상기 밸브수단 및 제거용 수단을 제어하기 위하여 처리실 수단으로부터 측정된 파라미터중 임의의 파라미터와 관련된 복수의 전기 신호를 수신하는 전자 제어 수단을 포함하여 가스를 처리실 수단에 공급하기 위한 수단을 포함하며, 이때 상기 전자 제어 수단은 소정의 명령 시퀀스에 따라 복수이 상태를 통하여 장치를 순환시키기 위한 컴퓨터 수단을 구비하며, 상기 컴퓨터 수단은 메모리 수단을 포함하고, 추가로 밸브 수단의 개방 또는 폐쇄 조건을 나타내는 밸브 수단으로부터의 입력신호를 수신하기 위한 수단 및 밸브 수단을 선택적으로 개방 또는 폐쇄시키기 위하여 출력신호를 밸브 수단에 전송하기 위한 수단을 구비하며, 상기 입력 및 출력신호의 영상신호는 메모리 수단에 기억되고, 마스크 수단이 메모리 수단에 기억되며 입력 및 출력신호의 영상신호들을 비교하여 입력 및 출력 영상신호들이 일치하지 않는 경우 마스크 수단에서 한 비트의 세팅에 응답하여 경보 신호를 발생시킨다.

본 발명의 다른 목적, 특징 및 장점은 이하에 상세히 설명하고자 한다.

[전체 시스템]

이하 도면을 참조하면, 제 1 도는 전체 가스 살균 시스템을 도시한 것이다. 그 시스템은 살균실(10), 마이크로프로세서로 제어되는 것이 바람직한 전자 제어 회로(100), 밸브 및 펌프 블록(20) 및 디스플레이(110)로 구성된다. 살균실(10)내의 적당한 센서에 의해 발생되며 온도, 습도 및 살균실(10)내의 살균용 가스의 농도에 관계되는 신호를 포함하는 센서 입력(5)은 살균실(10)으로부터 제어회로(100)으로 공급된다. 그 센서 입력은 상기 측정된 살균실 파라미터에 관계되는 아날로그 신호와 특정의 디지털 신호, 예컨대 살균실 온도가 이하에서 보다 상세히 설명될 소정의 값에 도달하는 시기를 나타내는 신호를 둘다 포함한다. 개시 사이클(START CYCLE) 스위치(S1)은 시스템의 동작을 개시시키고 어보트 리세트(ABORT-RESET) 스위치(S2)(이하에 상세히 설명함)는 어보트 모드가 얻어질 경우, 즉 고장 또는 경보 조건이 발생할 경우 시스템 상태를 정해진 조건으로 재순환시키는데 사용된다. 밸브 및 펌프 블록(20)의 동작은 이하에서 보다 상세히 설명할 것이며, 분리된 성분들로부터 현장에서 생성되는 이산화염소 가스(22), 수증기(23) 및 질소(24)의 공급원을 포함한다. 밸브 및 펌프 블록은 또한 도시한 바와 같이 대기 배기된다. 밸브 및 펌프 블록(20)은 다수의 순차 제어 밸브 및 적절한 시기에 살균실에 필요한 조건을 제공하기 위한 진공 펌프를 포함한다. 바람직한 살균용 가스인 이산화염소의 불안정성 및 잠재적 유독성 때문에 반응시 이산화염소 가스를 형성하는 성분들을 운반하는 것이 바람직하다. 예를 들면, 그 성분들은 아염소산 나트륨 Na_2ClO_2 및 염소 가스 Cl_2 일 수 있다.

시스템의 부품들을 제어하기 위해 적절한 제어 신호(7)이 전자 제어 회로(100)에 의해 밸브와 펌프 블록(20) 및 처리실(10)에 공급된다. 또한 제어된 부품으로부터의 피드백 신호(8)은 제어회로(100)으로 궤환되므로 제어기는 시스템의 상태를 모니터할 수 있으며 신호(14)는 시스템의 상태를 오퍼레이터에게 알리기 위하여 디스플레이 배전반(100)에 결합된다.

아울러, 카트리지 센스 신호(12)가 가스 성분 카트리지가 시스템에 결합되었음을 지시하도록 부착된 가스 카트리지(Cl_2 성분 카트리지)로부터 공급된다.

[일반 기능]

제 2 도는 밸브 및 펌프 블록(20)의 구성을 보다 상세히 도시한 것이다. 밸브 펌프 블록(20)은 일련의 밸브(V1,V2,V3,V4,V5,V6,V7,V8,V9 및 V10). 펌프(P1, P2), 공기 필터(13), 배기된 이상화염소 가스를 해독하기 위한 해독기(22)(이상의 부품은 상기 계류중인 출원에 설명된 바와 같이 구성됨) 및 적절한 수증기, 질소, Cl_2, 가스, 공기 및 아염소산 나트륨의 공급원으로 구성된다. 제 2 도에 도시한 바와 같이, 밸브중 어떤 것은 단순히 순서적이지만, 다른 것은 측정된 공정 변수, 예컨대 가스 농도, 습도 및 압력의 값 중 선택된 변수에 응답하여 제어된다. 안전상의 이유로, 각 밸브(V)에는 그 밸브의 개방 조건(예 ;

LS2o)또는 폐쇄 조건(예 : LS2c)을 나타내도록 두 개의 리미트 스위치(LS)가 압입되어 있다. 첨부된 소프트웨어 리스트에서, 개방된 리스트 스위치는 LSOx로 표시되고 폐쇄된 리미트 스위치는 LSCx로 표시된다. 양 스위치들은 사이클이 어보트되지 않도록 전체 사이클동안 적절한 시기에 적소에 있어야 한다. 아울러, 디스플레이 배전반상에는 제 5 도에 도시한 바와 같이 다수의 조명등이 제공되는데, 이는 살균 사이클의 진행 또는 임의의 고장 조건의 발생을 지시해준다. 한 사이클은 처리실 도어(11)이 폐쇄된 후 순간적으로 개시 사이클(START-CYCLE) 스위치(S1)를 누름으로써 조작자에 의해 시작될 수 있다(제 1 도 참조). 그후 사이클은 전자 제어기(100)의 마이크로프로세서 메모리에 기억된 프로그램에 따라 자동적으로 진행한다. 이 과정은 이하에서 보다 상세히 설명될 것이다.

또한, 용장성을 제공하기 위하여, 다수의 수동식 제어 밸브, 예컨대 밸브(V9) 및 (V10)이 밸브(V3) 및 (V8)이 개방되지 않은 경우에 제공된다. 이러한 밸브들은 살균용 가스가 실내에 있을 때 밸브(V3) 및 (V8)이 개방되지 못하는 경우에 잠재적인 유독성 가스가 해독기(22)를 거쳐 제거될 수 있도록 서비스 요원에 의해 수동적으로 조작될 수 있다. 또한 가스가 수동 조작 밸브를 통해 인출될 수 있도록 보조 진공 펌프가 제공된다.

[살균 사이클]

살균 사이클은 마이크로프로세서 제어하의 인터록된 일련의 이벤트(event ; 사건) 및 결과적인 액션(action ; 작용)이다. 이 시퀀스의 단계들은 제 6 도의 상태도 및 제 7 도의 상태 출력 매트릭스에 상세히 제시했다. 이 단계들은 시퀀싱 프로그램에 의해 수행되는데, 그것에 대한 플로우챠트는 제 7a 및 제 7b 도에 도시되어 있고 그것의 상세한 설명은 부록으로 포함된 프로그램 리스트에 게재되어 있다. 2가지 형태의 이벤트는 시퀀스, 즉 독립 및 종속 이벤트도중에 발생한다. 일부 독립 이벤트는 외부 이벤트로서 제어된 밸브(예 , 리미트 스위치)로부터 제어기로의 접촉 입력신호를 포함하며, 제 3 도에 XOx 내지 X3x로 표시했다. 각각의 접촉 입력신호는 8비트 워드중 1비트이고 이러한 제어 입력 신호를 수집한 것을 본문에서는 일반적으로 디지탈 입력(DIN)으로 언급했다. 독립 이벤트는 또한 측정된 아날로그 공정값(AIN), 이를테면 압력, 온도, 습도 및 살균 가스 농도에 대응하는 신호의 수신을 포함한다. 그 측정값 신호는 제어기에 의해 수행되는 논리 비교 연산과 관련된다. 다른 독립 이벤트는 내부적으로 선언되고, 그 결과 제 5 도에 도시한 디스플레이 배전반상의 지시등이 켜지게된다. 제어기는 단 하나의 TRUE(참) 또는 FALSE(거짓) 결과로 독립 이벤트의 논리 조합인 종속 이벤트를 평가한다. 종속 이벤트가 참이 될 경우, 대응하는 액션이 수행된다. 즉, 제어 시스템은 제 7 도의 상태 출력 매트릭스에 의해 한정된 새로운 처리 상태로 이동된다. 한편, 종속 이벤트가 참이 아닌 경우에는, 제어기는 그것의 메모리에서 처리 상태를 홀드시키고 종속 이벤트를 재평가하기 전에 50밀리초의 주기동안 대기한다. 시스템이 고장난 경우, 그 시스템은 이하에 상세히 설명되는 바와 같이 즉시 적절한 어보트(ABORT) 상태로 자동적으로 전이된다. 이 과정은 사이클이 완결 또는 어보트될 때까지 계속된다.

[안정성 고려]

살균 시스템에는 여러 밸브들과 기타 부품들의 정확한 동작을 확보하도록 다수의 체크 시스템이 구비되어 있다. 이하에서 보다 상세히 설명되는 바와 같이, 제어기의 주요 타이밍 프로그램에 의해 작성된 인터로크 소프트웨어는 6250 마이크로초마다 모든 밸브의 정확한 위치를 확인한다. 경보조건은 밸브가 명령받은 상태에 있지 않은 경우에는 언제나 선언된다. 이러한 인터로크의 동작은 밸브 작동이 일어난 후 인터로크 체킹이 계속되고 각 처리 단계에서 상이한 고장 프로그래밍(ABORT 상태)로 유도될 수 있다는 점에서 전형적인 릴레이 논리 또는 프로그램가능한 논리 제어기와 다르다. 한 밸브의 정확한 상태는 작동이 확인된 후에 메모리를 이해하고 이해된 조건은 6250마이크로초마다 체크된다. 제 8 도는 이 체킹 루우틴을 수행하는데 필요한 안전-인터로크 부품에 대한 블록도이다. 작동을 수반하는 초기의 이벤트-타임아웃 조건 또는 임의의 후속 상태 체크를 통과하지 못하게되면 살균 사이클의 비정상적인 종료가 일어나게 된다. 사이클의 안전한 종료를 위한 제어 액션의 시퀀스는 살균 사이클의 모든 점에 대하여 정의되고 임의의 비정상적인(ALARM)처리 조건의 경우에 즉시 개시된다. 본 발명에 따른 이러한 면밀한 상태 체킹은 시스템 사이클의 어떤 지점에서 리미트 스위치가 우회하는 경우 스위치가 부적절한 위치에 있도록 결정되어 시스템으로 하여금 어보트(ABORT)상태로 들어가게 하기 때문에, 현장에서 인터로크 스위치의 고의적인 우회를 방지한다.

제 8 도에 도시한 바와 같이, 안전 인터로크 시스템은 전자 제어기의 내부 메모리(ROM)에 저장된 시퀀싱 프로그램(120)을 포함한다. 시퀀싱 프로그램(120)은 그것에 부착된 메모리에 상주하는 프로그램의 리스트에서 SEQ로서 정의되고 이 프로그램에 대한 플로우챠트는 제 7a 및 7b 도에 도시했다. 접점상태 체크 프로그램(122) 및 시퀀스 프로그램의 특정 지점에 의해 결정되는 일련의 마스크(124)로 메모리에 기억된다. 접점 상태 체크 프로그램은 부록에서 CSC로 정의되고 그에 대한 플로우챠트는 제 18 및 제 19 도에 도시했다. "개방" 리미트 스위치 접점(127)(밸브가 개방될 때 폐쇄되고 밸브가 개방될 때 개방됨)과 "폐쇄" 접점(129)(밸브가 폐쇄될 때 폐쇄되고 밸브가 개방될 때 개방됨)로부터의 실제 입력 신호의 메모리에 기억된 영상인 입력(126) 뿐만 아니라 다른 부품, 예컨대 살균실(10), 도어(11)로부터의 입력이 제공된다. 또한 일련의 접점 출력(125)는 시퀀싱 프로그램의 특정상태에 의해 제공된다. 접점 상태 체크 프로그램(122)는 접점 입력과 접점 출력(125)를 비교한다. 입력이 출력에 의해 설정된 소정의 값과 다를때마다, 대응하는 비트가 마스크(124)에서 턴온되는 경우에만 경보 조건이 선언된다. 이러한 안전설비는 즉시 부정확한 밸브 위치를 검출한다. 하드웨어로 구성된 워치도그 타이머(watchdog timer)(132)는 마이크로프로세서 제어기가 고장일 경우 타이머가 타임아웃될 때 전자 스위치(134)를 개방시키므로써 밸브(130)으로의 모든 출력을 디스에이블(disable)시켜 여분의 안전도를 제공하므로, 컴퓨터 고장의 경우에 밸브 및 펌프 블록(20)내의 임의의 밸브들의 작동을 방지하는데 이용된다.

제 7a 및 7b 도는 시퀀싱 프로그램 SEQ에 대한 플로우챠트이다. 그 시퀀싱 프로그램은 메인 태스크지명 프로그램(Main Dispatching Program)으로 명명되는 또다른 프로그램으로부터 유입된다. 상기 메인 태스크지명 프로그램은 본질적으로 적절한 시간간격에서 발생된 플래그에 대하여 체크하고 특징이 수행될 시기를 결정하는 것이다. 제 7a 도에 도시한 바와 같이, 시퀀싱 프로그램 유입시 시스템의 현재 상태는(180)에 나타낸 바와 같이, 메모리로부터 회복된다. 제 12 도에 도시한 바와 같이 현재 상태는 내

부 CPU RAM내의 레지스터(210a)에 기억된다. 내부 CPU RAM의 조직은 차후에 제 12 도와 관련하여 상세히 설명될 것이다. (182)에서는 상태가 최대 상태수치를 초과하였는지의 여부를 판별하는 체크가 이루어진다. 초과된 경우, (184)에서 제 6 도와 관련하여 상세히 후술하는 바의 어보트(ABORT) 상태로 들어간다. 그렇지 않은 경우에는 제 7b 도에 플로우챠트가 도시된 프로그램 ST로 들어가므로써 (186)에서 다음 상태에 대한 조건이 수행된다.

제 7b 도에 도시한 바와 같이, 프로그램 ST는 먼저(188) 및 (189)에 도시한 것처럼 단 하나의 참 또는 거짓의 결과로 각 종속 이벤트를 평가한다. 각 종속 이벤트는 다수의 독립 이벤트의 논리 조합이며, 이들은 각각 종속 이벤트가 참인 경우 특정되어야 한다. 종속 이벤트가 참이 아닌 경우, (190)에서 마이크로프로세서 내부 RAM(제 12 도 참조)내의 기억 장소에 있는 홀드 플래그(FØ)가 세트된다. 그렇지 않은 경우에는 (192)에서 다음 상태가 세트되고(193)에서 새로운 어보트 상태가 필요한 경우 새로운 어보트 상태가 세트되지만 유입되지는 않는다.

(194)에서, 선행하는 이벤트에 대한 타임아웃은 그 타임아웃으로 말미암아 경보 조건이 발생되지 않아서 어보트 상태에 도달될 수 있도록 디스에이블되어야 한다. 타임아웃은 프로그램이 작성된 타이머에 의해 제공되는데, 그 타이머는 타이머에 의해 정의된 예정된 시간애에 특정 액션, 예컨대 밸브의 이동에 대하여 모니터한다. 특정 액션이 발생된 경우, 타이머가 계속 작동되기 때문에 타임아웃이 디스에이블되어야 한다. 제 18 도에 도시한 바와 같이, 타임아웃을 디스에이블시키기 위해서는, 내부 RAM내의 타이머 카운터 인에블 레지스터(TCEN)(207)(제 12 도)의 한 플래그가 소거된다. 이러한 방법으로, 타이머가 작동을 끝낼 때 타이머용 플래그가 타이머 카운터 플래그 레지스터(TCFL)(206)(제 12 도)로 세트되는 경우, 경보가 발생되지 않을 것이다. 타임아웃 경보가 발생되는 경우, 제 18 도에 도시한 바와 같이 비트(TMOF)는 상태 레지스터에서 세트된다.

(195)에서, 마스크는 소거된다. 즉, 발생될 특정 이벤트에 대응하느 비트들이 "무정의(don't care)" 조건으로 세트되므로, 접점 출력에서의 대응하는 비트의 변화는 접점 상태 체크 프로그램에 의한 경보 조건을 개시하지 않는다. 이 시점에서, (196)으로 도시한 바와 같이 액션이 수행될 수도 있다. 이어서, 액션에 대한 타임아웃 계수치는 차후에 보다 상세히 설명되는 바와 같이 타이머 레지스터(200)(제 12 도) 중 적절한 것에 부하된다. 이어서, 액션 타임아웃 플래그는 (197)로 도시한 바와 같이 현재 모니터된 액션의 적절한 발생을 모니터하도록 인에이블된다. 이어서, 홀드 플래그(FØ)는 (198)에서 소거되고(SEQR)로 표시한 지점, 즉 제 7a 도의 플로우챠트로 복귀된다.

(200a)에서, 경보 또는 타임아웃 조건이 발생하였는지를 판별하기 위한 시험이 행하여진다. 경보 또는 타임아웃이 발생될 경우, (201)에서 현재 상태는 현재 어보트 상태로 세트된다. 이어서, 홀드 플래그(FØ)는 (202)에서 그 세팅 여부를 판별하도록 체크된다. 세트된 경우, 이어서 모든 서브루틴이 그것으로부터 유입되는 배경 또는 메인 태스크지명 프로그램으로의 복귀가 이루어진다. 플래그(FØ)가 세트되지 않은 경우에는, 시스템은 다음 상태로 계속되고 플래그(FØ)가 세트된 경우에만 퇴거하도록 시퀀싱 프로그램에서 유지된다.

제 19 도는 점점 상태 체크 프로그램을 보다 상세히 도시한 것이다. 도시한 바와 같이, 접접 입력에 대응하는 접점 입력 상태는 시스템 마이크로프로세서의 내부 RAM의 적절한 기억장소에 기억된다. 메모리 기억장소는 표시된 바와 같다(제12도 참조). 접점 출력 상태 비트에 대해서도 동일한 양상이 존재하며, 그 비트는 특정 상태에 대해서 발생한 이벤트를 상세히 나타낸다. 마찬가지로 내부 RAM에 기억된 마스크 MSK0-MSK3는 접점 상태 체크 프로그램에 의해 평가된다. 만일 접점 입력이 접점 출력으로부터 변하게 되면, RAM내의 한 기억장소(제 12 도 참조)인 상태 레지스터(204)의 한 비트를 세팅하므로써 경보 조건이 발생되지만, 이것은 마스크내의 대응 비트가 턴온되는 경우에만 행하여진다. 대응 출력의 변화가 발생될 수 있음을 나타내도록 대응 비트가 오프된 경우에는, 경보가 발생되지 않을 것이며 접점 출력이 이하에 보다 상세히 설명된 바와 같이 출력버퍼에 기록되어 경보를 작동시키는 일없이 적절히 제어되거나 시퀀스된 부품, 예컨대 밸브 또는 펌프를 작동시키게 될 것이다.

또한, 추가의 안전 설비가 시스템에 제공된다. 상술한 바와 같이, 서비스 요원에 의해 동작되는 수동식 밸브(V9) 및 (V10) 및 보조 펌프(P2)는 밸브(V8) 및 (V3)와 주 펌프(P1)이 적절히 동작하는 경우에 제공되어, 용장성을 제공한다. 아울러, 제 2 도에 도시한 바와 같이, 안전 설비가 제공되어 살균실(10)에서 과도한 온도 및 압력이 생길 가능성을 방지한다. 열 작동 스위치(11a)는 과도한 온도를 검출하도록 실내의 히이터(HTØ1)과 직렬로 제공된다. 예를들어, 히이터(HTØ1)이 턴오프되지 못하는 경우, 온도 조절 스위치(11a)가 과도한 온도를 감지하여 회로의 작동을 중지시킬 것이다.

한편, 과도한 압력이 살균실내에 생기는 경우, 압력 이완 밸브(9)가 살균실(10)에서의 가스 배출을 위해 제 2 해독기(22a)를 통해 대기에 제공된다.

또한, 살균용 염소가스를 시스템에 공급하는 밸브(V4)와 직렬로 체크밸브(15)가 제공된다. 체크밸브(15)는 밸브(V4) 및 (V4a)가 폐쇄되지 못할 경우에 질소용기로부터 질소가스가 염소가스 용기를 가압할 가능성을 방지한다. 체크밸브(15)는 염소가스가 염소가스 용기로부터 유출될 수 있게하여 밸브(V4) 및 (V4a)가 폐쇄되지 않을 경우에 질소가스가 염소가스 용기로 흘러들어가는 것을 막는다.

[오퍼레이터 대화(interaction)]

본 발명에 따른 시스템의 장치 및 살균 사이클은 최소의 오퍼레이터 개입 및 최대의 안전성을 제공한다.

제 5 도는 각종 표시 등을 도시한 본 발명에 사용되는 디스플레이 배전반의 한 실시예를 도시한 것이다. 어떤 등은 예시할 목적으로 제공된 것이며 사용되지 않는다. 살균 사이클은 처리실(10)의 도어(11)이 적절히 폐쇄될 때까지 개시될 수 없다. 도어 개방(DOOR-OPEN)등(LT1)이 제 7 도의 상태 1에서의 "1" 상태로부터 상태 2에서의 "0"상태로 상태를 변화시키는(LT01)으로 도시한 바와 같이 꺼지게 되고, 사이클 대기(READY-FOR-CYCLE)등 (LT11)이 켜지게 된다. 제 5 도를 참조하라. 사이클을 개시하기 위하여, 오퍼레이터는 준비되었을 때 사이클 개시(START-CYCLE)(S1) 스위치를 누르기만 한다. 그후, 부하 제거(REMOVE-LOAD) 등(LT17)이 켜짐으로써 사이클이 종료되거나 경보조건이 사이클을 중지시킬 때까지는 오퍼레이터

개입이 불필요하다. 후자의 경우에는 고장을 나타내는 경보등 중 하나가 켜질 것이다. 오퍼레이터는 어느 등이 켜져 있는지를 주목하여 필요한 액션을 취한 다음 시스템을 정의된 조건으로 다시 순환시켜서 가능하다면 고장 조건을 피할 준비가 될 때 어보트 리세트(ABORT-RESET)(S2) 스위치를 누른다. 예를들어, 고장 일소(PURGE-FAIL) 등(LT5)가 질소 탱크가 완전 소비될 가능성으로 인하여 켜지는 경우에, 탱크는(S2)를 누르기 전에 고장 조건을 진단하여 복구시키려는 시도가 있어야 한다. 사이클을 어보트시키는 차후의 액션은 예정되어 있고 자동적으로 이루어진다. 어떠한 추가 오퍼레이터 개입도 필요치 않다. 또한, 부품이 고장난 경우 다른 부품, 즉 펌프 또는 밸브가 고장난 부품을 대신할 수 있도록 시스템에 용장성이 제공되므로 시스템은 고장상태로부터 벗어날 수 있다.

[제어회로 설계]

전자 제어회로(100)의 전체 설계는 제 3 도에 도시되어 있다. 그 제어기는 인텔사에 의해 제조된 8031, 8051 또는 8751형 마이크로프로세서 CPU(102)를 이용하는 것이 좋은데, 그 이유는 이 프로세서가 비트 어드레스 가능한 데이터에 대한 논리변수 연산을 수행할 수 있기 때문이다. CPU(102)는 자기 내장식 랜덤 억세스 메모리(RAM) 및 리드 온리 메모리(ROM)을 포함한다. 또한, 그 제어기는 외부 ROM(104) 및 비소멸성 새도우 RAM(SRAM)(XICOR사에 의해 제조된 X2210형일 수도 있으며 이하에 설명되는 바와 같이 전원 차단후 임계 데이터를 저장함)을 포함할 수도 있다. 그밖에, 제어기는 수정클럭(108), 디지털 입력(DIN)을 수신하기 위한 입력 래치(113), A/D변환기(114) 및 아날로그 입력(AIN)용 필터(114a), 디지털 출력(DOU)용 출력 래치(117) 및 워치도그 타이머(WDT)(112)를 포함한다. 후자의 타이머는 제 8 도와 관련하여 상술한 바와같이 마이크로프로세서의 고장시 밸브로의 모든 출력이 비작동 상태가 되도록 구성된다.

아날로그 디지털 변환기(114) 및 아날로그 필터(114a)에 대한 아날로그는 아날로그 입력을 측정된 가스 농도, 온도, 습도 및 압력 파라미터로부터 디지털 데이터로 변환시킨다.

중앙처리기(102)는 어드레스/데이터 버스(16)에 결합되는데, 이는 RAM(106), ROM(104) 및 버스 트랜시버(105)를 결합한다. 어드레스 래치(103)는 CPU(102)로부터의 라인(107)에 의해 인에이블되어 다른 버스(103)는 CPU(102)로부터의 라인(107)에 의해 인에이블되어 다른 버스(109), 즉 판독/기록 및 어드레스 버스에 어드레스를 래치한다. 버스(109)는 DIN 래치(113), A/D변환기(114), 타임 스탬프 클럭(119) 및 DOU래치(117)로 하여금 살균 시퀀스 프로그램 실행시, 즉 CPU(12)가 각종 밸브 리미트 스위치로부터 입력 데이터를 호출하여 DIN래치(113)이 어드레스 지정될 때 적절한 시점에서 어드레스 지정되도록 한다. A/D변환기(114) 또는 DOU래치(117)는 다른 시점에서 어드레스 지정될 것이다.

두 개의 디코더, 즉 판독 인에이블 디코더(120) 및 기입 인에이블 디코더(112)는 버스(109)에 결합되어 래치(113),(117) 및 A/D변환기(114)가 판독 또는 기록하도록 한다. 디코더들의 제어를 위해 적절한 판독/기록 명령이 라인(126)상에 제공된다.

또한, 데이터 버스(124)는 입력 및 출력 래치와 A/D변환기로부터 데이터를 읽어내거나 그것들에 데이터를 읽어들이기 위해 제공된다.

또한, 데이터 버스 인에이블 라인(125) 및 RAM명령 라인(127)을 포함하는 몇 개의 추가 제어 라인이 이용된다. 라인(125)는 매우 짧은 기간 동안과, 입력 및 출력 동작이 수행되고 있는 도중에만, 예컨대 밸브제어를 위해 출력정보를 DOU래치(117)에 기록하고 있는 입출력(I/O) 서브루우틴(예컨대, 서브루우틴 WCO(접점 출력 기록), RCI(접점 입력 판독) 및 RAI(아날로그 입력 판독 ; 부록 참조)중에만 버스 트랜지버(105)를 인에이블시킨다. 이러한 방법으로, 밸브 및 펌프 블록의 여러 밸브들을 작동시키기 위한 데이터 버스(124)상의 데이터는 제한된 경우만을 제외하고 밸브로 전송 될 수 없다. 이것은 부가적인 시스템 안전도를 제공한다. 또한, 버스 트랜시버 XCVR(105)는 양방향성이고 그 데이터 전송 방향은 도시한 바와 같은 판독 및 기록 라인중 하나에 의해 제어된다.

RAM명령 라인(127)은 고장이 영구적으로 로그(log)되고 다른 임계데이터가 전원 고장의 경우에 저장될 수 있도록 신호를 섀도우 RAM(106)에 송출한다.

또한, 리세트 라인(129)는 기록 인에이블 디코더(122)와 위치도그 타이머(112)사이에 제공되고, 인에이블 라인(130)은 타이머(112)와 DOU래치(117)사이에 제공된다. 전술한 바와 같이 타이머(112)는 적절한 시스템 동작을 위해 CPU(102)를 모니터한다. 정상적인 경우, CPU(102)는 항상 위치도그타이머를 라인(129)를 거쳐 리세트시킨다. CPU의 기능이 잘못된 경우, 리세트 신호는 적시에 나타나지 못하고 타이머(112)가 타임아웃되어 라인(130)상의 출력 인에이블 신호를 제거한다. 이 신호의 제거는 모든 DPU래치(117)의 출력을 디스에이블시켜 CPU가 고장난 경우 밸브 작동을 방지하게 된다. 따라서, 상술한 시스템에 가일층 안전도가 제공되는 것이다.

제어기의 소자들이 제 3 도에 도시한 바와 같이 데이터 버스(116), (124)에 결합되므로, 이들은 마이크로프로세서에 의해 호출될 수 있는 메모리 어드레스로써 할당되었다. 제 3a 도는 참고로 이들 어드레스의 배열을 도시한 것이다. 상술한 바와 같이, SRAM(106) 및 DOU래치(117)과 같은 장치는 그들이 포함한 데이터가 마이크로프로세서 포트 라인의 비트들이 적절히 시퀀싱될 때만 변화될 수 있도록 제공된다. 이것은 일부의 마이크로프로세서 고장 모드가 메모리 내용 또는 밸브 위치의 바람직하지 못한 변화를 일으키는 것을 방지하는 안전 설비이다.

모든 프로세서 및 프로그램 시간조정은 5.9904㎒의 주파수를 갖는 것이 바람직한 기본 클럭 발진기(108)로부터 유도된다. 제 4 도는 사용된 여러 주파수들간의 관계를 나타낸다. 제 3 도에 도시한 바와 같이, 정밀클럭(119)를 시스템에 부가할 수도 있는데, 이는 처리 데이터를 타임스탬핑(time-stamping)하기 위한 클럭 칼렌다(clock-calender)를 제공하도록 데이터/어드레스 버스 또는 직렬 데이터 통신 라인(118)을 통해 판독될 수 있다.

제 4 도에 도시한 바와 같이, 기본 프로세서 타이밍은 CPU내부의 결정 제어식 클럭(108)에 의해 제공된다. 클럭(108)의 주파수는 CPU의 내부 카운터 단(130), (132)에 의해 1/12로 분할되어 499,220Hz의 CPU

어드레스 래치 인에이블(ALE)신호를 제공하게 된다. ALE신호는 어드레스들이 버스(109)상에 위치할 수 있도록 어드레스 래치(103)를 스트로우브시키는데 사용되고 또한 A/D변환기(114)의 동작을 제어한다.

신호 ALE는 또 다른 내부 분할기 단(134), (136)에도 결합된다. 분할기 단(134)는 직렬 데이터 전송을 위해 내부 카운터 단(138)에 의해 1200비트/초의 신호로 분할되는 타이머 1로 표시된 신호를 제공하며, 그것은 시스템 데이터를 직렬 라인(118)을 거쳐 원격 위치로 전송하도록 임의로 제공될 수 있다.

카운터 단(135)은 인터럽트 TIMER0를 제공한다. TIMER0는 6250 마이크로초마다 천이를 제공하여 주 시간 조정 기능 프로그램 TMR0가 모든 접점 입력 및 아날로그 입력을 판독하고 6250 마이크로초마다 모든 접점 출력을 기록하도록 한다. 이 프로그램 및 동작 시스템의 다른 프로그램의 동작은 이하에 보다 상세히 설명될 것이다.

이어서 TIMER0인터럽트는 각각 50 밀리초, 1초 및 1분의 주기로 발생하는 TIC, SEC 및 MIN으로 표시된 각 프로그램 실행 신호를 제공하도록 프로그램 TMR0소프트웨어 카운터 단(142), (144), (146)에 의해 추가로 분할된다. 이와 관련해서는 이하에 상세히 설명될 것이다.

제 3a 도는 어드레스 버스(109)상에 어드레스를 할당하는 것에 대하여 상세히 나타낸 것이다. 도시한 바와 같이, 버스(109)는 16비트 버스이다. 내부 CPU RAM은 어드레스 공간 00-FF로써 할당되고 버스(109)상의 비트 A0 내지 A7은 RAM의 기억장소를 정의한다. 내부 RAM은 도시한 바와 같이 비트 A0 내지 A15로 정의되는데, 비트A12-A15는 항상 0이다. 0000으로부터 0FFF까지의 어드레스가 사용된다. 기타 부품들, 외부 ROM(104), 외부 RAM(106), 클럭(119), A/D변환기(114), DIN래치(113), DOU래치(117) 및 위치도그 타이머(112)가 제 3a도에 지시된 어드레스로 지정된다. 도시된 바와 같이, 다양한 리미트 스위치 및 다른 접점 입력으로부터 DIN래치와 여러밸브 및 펌프등에 이르는 DOU래치는 각각 네 개의 8비트 워드를 래치할 수 있다. 디지털 입력 DIN 및 디지털 출력 DOU는 각기 8비트인 네 개의 워드로 각각 세분되고, 각 집단의 8비트는 모두 제 3a도에 지시된 각 어드레스에 의해 동시에 액세스된다.

제 3 도에 도시된 바와 같이, 아날로그 압력, 온도 습도 및 이산화염소 가스의 농도 파라미터는 각각의 센서(114c)로부터 각각의 증폭기(114 d, e, f 및 g)로 공급된다. 살균용 이산화염소 가스가 살균실로부터 배기되고 있을 때 이에 따른 시스템의 안전도를 제공하기 위해, 이산화염소 가스의 농도값이 정확히 측정되는 것이 중요하다. 따라서, 가스농도 신호용 증폭기(114g)는 살균실이 배기되고 있는 시간동안 제어신호 Y37에 의해 고증분 모드로 변화된다. 이 방식에서, A/D변환기(114)는 입력 농도 아날로그 신호를 다수의 정량화된 값과 비교하여 실제 농도를 더욱 정확히 표시하게 된다. 증폭기(114g)는 이 이외의 다른 모드 시점에서 저증분 모드로 남아있을 것이다. 예를 들어, 이산화염소 농도가 적당한 살균농도를 결정하기 위해 살균실내에서 측정되는 경우, 일층 높은 농도값이 측정되므로 A/D변환기(114)가 아날로그 농도값에 대응하는 정확한 디지털 신호를 제공한다. 그러므로, 증폭기(114g)가 저증분 모드로 유지될 수 있다. 이 증폭기(114g)는 "0"에서 "1"로 변하는 신호(Y37)에 의해서 고증분 모드로 교환될 수도 있다.

시스템 데이터-베이스는 외부 영역과 내부 영역으로 구분된 것으로써 생각될 수도 있다. 외부 데이터-베이스는 디지털 입력 DIN으로 구성되는 접점 입력(CCI), 디지털 출력 DOU으로 구성되는 접점 출력(CCO) 및 아날로그 입력(AIN)을 포함한다. 외부 데이터-베이스의 영상은 6250마이크로초마다 호출되는 TIMER0 프로그램(TMR0)의 서브루틴에 의해 내부 RAM내의 기억장소를 포함하는 내부 데이터 베이스에서 유지된다. 즉, 6250 마이크로초마다 모든 접점입력 및 아날로그 측정값이 DOU래치에서 부하된 접점 출력의 화상 제어기 내부 데이터-베이스에서 판독되고 저장된다는 것이다. CPU(102)의 내부 데이터 RAM에 대한 메모리 맵인 제 12 도를 참조해 보면, 접점 입력의 화상이 변수 CCI0내지 CCI3으로써 저장되고, 필터된 아날로그 입력이 변수ADI0 내지 ADI7로써 저장된다. 접점 출력은 변수 CC00-CC03으로써 기억된다. 입력데이터를 사용하는 프로그램은 단지 이러한 장소로부터 그 데이터를 퇴거할 수 있으며 입력장치로부터 직접 퇴거되는 것은 아니다. 그러므로, 프로그램은 입력 및 출력의 화상에 대하여 동작하는데 불과하다. 또한, 내부 데이터 베이스는 다수의 레지스터 뱅크 RB0-RB3을 포함한다. RB0에서 50밀리초타이머 TICK(50ms), 1초 타이머 TSEC(1초) 및 1분 타이머 TIMN(1분)을 비롯한 다수의 타이머(205)가 제공된다. 이 타이머들은 시간조정 기능 간격을 시스템 메인 태스크 지명 프로그램에 의해 그 간격에서 수행되는 기능을 계획하는데 제공된다. 50밀리초 TTMx마다 호출되느 레지스터 뱅크 RB3 내의 다수의 타이머(200)를 비롯하여, TICK타이머는 50밀리초 후에 타임 아웃되고 상태(STATUS) 레지스터(204)의 플래그 TICF를 모든 50밀리초 시간조정 기능을 개시하도록 메인 태스크지명 프로그램에 의해 사용되도록 세트시킨다. 이러한 타이머들은 예컨대 시스템 밸브에 대한 타임아웃 상태를 모니터하기 위해 호출되는 것이 바람직하다.

매초 STMx마다 호출되는 RB3 내의 다수의 타이머(200)를 비롯하여, TSEC타이머는 유사하게 1초후에 타임아웃되고 상태(STATUS) 레지스터(204)의 플래그(SECF)를 모든 1초 시간조정 기능을 개시하도록 메인 태스크지명 프로그램에 의해 사용되도록 세트시킨다. 이와 마찬가지로, 매분 MTMx마다 호출되는 RB3내의 다수의 타이머(200)를 비롯하여, 상태(STATUS) 레지스터(204)의 플래그(MINF)를 1분 시간조정 기능을 개시하도록 메인 태스크지명 프로그램에 의해 사용되도록 세트시킨다. 또한, 데이터 메모리는 시퀀스 프로그램에 프로그램에 의해 사용된 전류 상태 및 어보트(ABORT) 상태를 기억하기 위해 RB2의 레지스터들을 포함한다. 또한, 전술한 타이머용으로 시퀀스 상태 레지스터(8204), TCEN 및 TCFL 레지스터(207, 208)과 제어 밸브를 개방 또는 폐쇄시키도록 하기 위한 제어 레지스터 CTRL을 포함한다. 제어 레지스터의 4비트는 측정된 온도, 습도, 압력 및 가스농도 파라미터에 대응하여, 시스템의 네 개의 제어 로우프를 제어하는데 사용된다. 접점 입력과 출력상태를 비교할 경우 "무정의"상태를 허락하도록 비트 마스크(260)의 어레이가 내부 데이터-베이스에 제공된다. 데이터 요소의 추가 설명은 본 명세서에 첨부된 제어기 프로그램 소오스 리스트를 참고하기로 한다.

특히, CPU(102)의 내부 RAM은 다음과 같이 구성될 수도 있다. 256(FF)메모리 장소가 시간조정 기능 레지스터(메모리 장소 00부터 07)의 50밀리초, 1초 및 1분 타이머 ; 수신 버퍼 RBUF와 전송 버퍼 TBUF를 제어하기 위한 임의의 통신 프로그램 레지스터(메모리 장소 08부터 0F) ; 메인 태스크지명 프로그램 레지스터(메모리 장소 10부터 17) ; 타이머(205)(18부터 1D)에 의해 50밀리초, 1초 및 1분 간격으로 이행되

는 타이머(200) ; 카운터(1E 및 1F) ; 상태 비트(204)(20) ; 제어 비트(206)(21) ; 타이머 인에이블 비트 TCEN(22) ; 타이머 플래그 비트 TCFL(23) ; 일련의 입력용 마스크(260)(24-27) ; 접점 출력 화상 CC00-CC03(28-2B) ; 접점 입력 영상 CC10-CC13(2C-2F) ; 아날로그 입력 AD10-AD17(30-37) ; 및 온도, 압력, 농도 및 습도와 같은 측정된 처리 변수에 대한 세트 지점(38-3B)으로 구성된다. 내부 RAM의 나머지는 통신 버퍼(40부터 5F), 시스템 선택(60부터 7F) 및 내부 마이크로프로세서 레지스터와 기억장치(80부터 FF)로 지정되며, 이들은 당해 기술분야에 공지되어 있다.(Interl Corp. 에 의해 발행된 1982년 5월자 문서 번호 제 210359-001 호, Microcontroller User's Manual을 참조). 전체 시스템 프로그램이 CPU(102)의 내부 ROM에 포함되었다 하더라도, 외부 ROM이 또한 부가적인 프로그래밍 용량을 허용하도록 제공될 수도 있다. 제 12 도에는 비트에 의한 STATUS, CTRL, TCEN 및 TCFL의 내용이 도시되어 있다.

[상태 시퀀스]

살균 사이클의 진행은 제 5 도에 도시된 디스플레이 배전반상의 진행(PROGRESS)등으로부터 결정될 수 있다. 정상 사이클동안 고장등이 결코 켜져서는 안된다. 정상이든지 어보트되든지, 사이클 및 고장 데이터는 둘다 비휘발성 랜덤 액세스 메모리 또는 새도우 RAM(SRAM)내에 보존될 것이다. 예를 들면, 지정된 수의 사이클, 예를 들어 3회의 사이클후에, 가스카트리지가 방전될 것이고 대체될 것이다. 카트리지가 소모한 사이클의 수에 관련된 데이터가 이 메모리에 기억된다. 또한, 소정수의 사이클 또는 반복된 고장후에, 시스템은 보전이 될 때까지 디스에이블될 것이다. 이것은 현장에서 우회될 수 없는 안정 특성이며, 이 데이터는 또한 비휘발성 메모리에도 저장된다.

진술한 바와 같이, 제 6 도는 살균 시스템의 시퀀싱 프로그램의 동작을 정의하는 상태도이다. 제 7 도는 여러 진행상태에 대해 제 5 도에 도시된 디스플레이등 뿐만 아니라 제 2 도에 나타낸 부품들의 상태를 동정한 것이다. 시스템의 동작은 이하에 일층 상세하게 설명할 것이다.

시스템은 항상 개시 상태인 상태 "0"에서 시작한다. 이 상태동안, 제어회로(100)내의 마이크로프로세서의 모든 출력 라인이 밸브 및 펌프블록(20)의 모든 밸브를 초기에 무능화시키도록 세트된다. 단시간의 지체후에, 밸브(V7)은 제 7 도에서 상태 "0"동안 반대편(VV07)을 나타내는 "1"에 의해 도시된 바와 같이, 공기를 실내로 유입시키도록 개방된다. 게다가, 이 상태동안 제어 회로(100)은 마이크로프로세서의 모든 출력 부분의 상태를 메모리에 저장한다.

상태 "0" 및 "1"에서는, 살균실(10)에 대한 도어가 개방위치에 존재한다. 일단 도어가 폐쇄되면, 상태(2)가 시작된다. 제 6 도에 도시된 바와 같이 이것은 시스템이 그것의 사이클을 시작할 준비가 되었다는 것을 의미한다. 또한 제 7 도에 도시된 바와 같이, 상태(2)에서 밸브(V1-V6)가 폐쇄되고, 밸브(V7)이 개방상태로 유지되며, 밸브(V8)은 폐쇄된다. 디스플레이등(LT1-LT6)이 꺼지고, 등(LT11, 사이클대기(READY FOR CYCLE)이 켜지며, 등(LT12-LT17)이 꺼진다. 이와 대응하는 리미트 스위치(LS)는 그에 관련된 밸브의 조건, 예를 들어 폐쇄된 밸브(V2)의 경우 리미트 스위치(LS2c)가 형성되고 리미트 스위치 LS2o이 개방되는 조건에 의해 정해진 위치에 존재한다. 상기 지적된 바와 같이, 두 개의 리미트 스위치가 각 밸브상에 제공되는데, 다시 말해서, 개방위치에 대해 하나, 폐쇄위치에 대해 하나가 제공되며, 이것은 시스템의 안정성을 보장한다. 양 리미트 스위치는 그들의 적당한 위치에 존재해야 하며, 그렇지 않을 경우 고장이 발생할 것이다. 살균실(10)에 대한 도어가 개방될 경우, 일단 개시 상태가 통과되면 시스템은 상태(1)에 존재한다. 따라서 제 7 도에 도시된 바와 같이, (LT1)만이 켜지고 다른 등들은 꺼진다.

살균실 도어가 폐쇄되고 시스템이 상태(2)에 있다고 가정한다면, 개시 사이클(START CYCLE)스위치(S1)을 누를 경우 시스템은 상태(3)으로 이동한다. 이 지점에서, 밸브(V7)이 제 6 도의 상태 3에 대한 난에 "0"으로 표시된 바와 같이 폐쇄되고, 사이클 진행(CYCLE IN PROGRESS)등인 (LT12)가 켜진다. 제 2 도에 나타낸 바와 같이, 밸브(V7)이 개방될 경우, 살균실(10)을 필터(13)을 통해 대기로 배기한다. 그래서, 필터된 외부 공기의 실내로의 흐름은 밸브(V7)이 폐쇄될 때 중지된다.

만일, 도어가 상태(2)에서 개방된다면, 상태(1)로 즉시 복귀된다.

일단 상태(3)에서 (V7)이 폐쇄될 경우, 리미트 스위치(LS7c)의 폐쇄상태와 리미트 스위치(LS7o)의 개방상태로 지시된 바와 같이, 상태(4)로 들어갈 것이다. 만일 밸브(V7)이 정해진 시간내에 폐쇄되지 않는다면 데이터 메모리의(RB3)내의 TIC 타이머(TTMx)중 하나에 의해 이행된 타임아웃에 의해서 결정되는 바와 같이, 상태(29)인 어보트(ABORT)-1이 시작될 것이다. 또한, 폐쇄되어야 하는 밸브가 개방된 경우와 같은 경보조건이 발생한 경우, 경보 조건이 발생될 것이며, 고장 지점이 디스플레이 배전반에 나타날 것이며, 기능장애가 발생했다는 것을 오퍼레이터에게 알리게 된다.

일단 상태(3)에서 살균실 도어가 개방될 경우 제 6 도에 도시된 바와 같이 사이클이 어보트될 것이다.

(V7)이 폐쇄되고 상태(4)가 시작되었다고 가정한다면, 살균실 히터(HT01)이 상태(4)에 대한 (HT01)반대편 난에서 "1"로 지시된 바와 같이 켜진다. 만일 실내의 온도가 타임아웃 기간내에 충분한 값으로 증가할 경우, 상태(5)가 시작될 수 있다. 만일 그렇지 않을 경우, 어보트(ABORT)-1인 상태(29)가 시작되고, 그 후 스위치(S2)를 누를 경우 상태(2)로 복귀된다. 온도스위치(T1)(제 2 도)이 소정의 온도에 도달하는 실내의 대기온도에 의해 작동될 경우 안전 동작 온도가 도달된다. 그후에, 제 7 도의 "HT01"반대편 난에 부호 "C"로 표시한 바와 같이, 실내의 온도는 사이클중에 필요에 따라 히터를 켜고 끄는 작업에 의해 제어된다.

일단 상태(5)가 시작되면, 살균실(10)의 대기 내용물이 배기될 수 있도록 진공 펌프(P1)을 개시시키는 것에 대비하여, 밸브(V1)이 개방된다. 다시, 밸브(V1)이 타임아웃 기간내에 개방하지 않을 경우, 어보트(ABORT)-1인 상태(29)가 시작된다.

진공 밸브(V1)이 타임아웃 기간내에 개방할 경우, 상태(6)이 시작된다. 이 지점에서, 진공 펌프(P1)이 개시되고, 등(LT13)은 배기가 진행중임을 나타낸다. 타이머가 개시되어 펌프가 유지되는 시간을 결정한다. 일단 상태(6)에 있으면, 사이클내의 이 지점에서 살균이 진공하에 있기 때문에 살균실 도어(11)이

더 이상 개방되지 않을 수 있다.

상태(6)에서, 실내의 압력은 그것이 PEVAC로 정의된 액면가보다 작거나 같도록 충분히 감소되었는지를 결정하기 위해 체크된다. 만일 압력이 PEVAC보다 작다면, 그때 상태(7)은 시작되고 밸브(V1)이 폐쇄된다.

실내의 압력이 배기시간이 경과한 후에 PEVAC보다 크고 적당한 진공값보다 작게 나타나면 상태(29)가 시작된다. 이와 마찬가지로 만일 밸브(V1)이 특정시간내에 폐쇄되지 않는다면, 상태(7)로부터 상태(29)로 들어간다.

밸브(V1)이 상태(7)에서 폐쇄된 후에 상태(8)에서 리크(leak)-홀드 시험이 시작된다. 만일 리크-홀드 시간후의 압력이 액면가 PLEAK보다 작다면 상태(9)가 시작되고, 그렇지 않을 경우 어보트 상태(29)가 시작된다.

상태(9)에서, 수증기가 실내로 유입될 수 있다. 즉, 밸브(V6)이 제 6 도의 "C"로 표시한 바와 같이 제어된 개방 상태이며, 습도가 정해진 시간에 특정값에 도달했는지의 여부가 결정된다. 액면 습도 HNOM이 특정시간내에 도달되지 않았다면, 어보트(ABORT)-2인 상태(30)이 시작될 것이다. 배기가 완료된 후에, 등(LT13)이 꺼지고, 진행완료(FILL IN PROGRESS)를 나타내는 등(LT14)가 켜진다. 완료(FILL)는 실내로 비살균용 가스, 예를 들면 증기 및 질소가스가 공급되었음을 의미한다. 이 지점에서, 시스템은 상태도의 새로운 지점에서 시작되며, 이 지점에서 기능장애로 인해 시스템이 다른 어보트 상태인 상태(30)로 복귀된다. 어보트(ABORT)-2(상태 30)에 대한 각종 밸브 및 디스플레이의 상태가 제 7 도에 도시되어 있다.

상태(9)에서 습도 타이머가 타임아웃된다. 만일 습도값이 액면가 HNOM보다 크다면, 상태(10)이 시작되고, 그렇지 않을 경우 상태(30)이 시작되고 사이클이 어보트된다.

상태(10)에서, 습도값이 소정의 기간동안 모니터되는 습도 지속 시험이 수행된다. 만일 습도값이 소정시간동안 유지되지 않을 경우, 상태(30)이 시작되고, 그렇지 않다면 상태(11)이 시작된다. 밸브(V2) 및 (V8)이 개방되고, 밸브(V5)가 이어서 밸브(V6)과 함께 제어된다.

밸브(V5)에 의해 질소가 시스템에 들어오게 된다. 이 지점에서, 밸브(V2)가 개방되더라도, 함께 제어된 (V4) 및 (V4A)가 폐쇄되기 때문에 이산화염소가 실내로 들어올 수 없다.

상태(11)에서, 밸브(V2)가 그것이 개방되었는지의 여부를 결정하도록 체크된다. 만일 특정 시간내에 개방되지 않을 경우, 상태(30)이 시작된다. 만일 밸브(V2)가 특정 시간내에 개방된다면, 상태(12)가 시작되고, 밸브(V4) 및 (V4A)가 제어되며, 실내로 이산화염소가 들어오게 된다. 타이머가 개시되고, 그 동안에 실내의 이산화염소 가스의 농도값이 측정된다. 상술한 바와 같이, 이산화염소는 현장에서 두가지 성분인 Cl_2 가스와 아염소산나트륨의 반응에 의해 발생될 수 있다. 염소 가스는 당분야에 공지된 바와 같이, 접속 포트를 거쳐 시스템에 결합될 수 있는 용기내에 이용된다. 제 2 도에 도시된 바와 같이, 아염소산나트륨의 용기가 밸브(V2)와 (V4)사이에서 시스템에 결합된다. 상태(12)에서 (LT14)가 꺼지고, 살균 진행(STERILIZATION IN PROGRESS)등인 (LT15)가 켜진다.

일단 상태(12)에서 측정된 가스 농도가 예정된 기간내에 액면 농도 CNOM보다 크거나 같은 농도에 도달한다면, 상태(13)이 시작된다. 허용 가능한 살균용 가스의 농도는 예를 들면 1.0mg/L에서 약 300mg/L 사이에 있어야 한다. 그렇지 않을 경우, 새로운 어보트 상태로서 어보트(ABORT)-3인 상태(31)이 시작된다. 살균용 이산화염소 가스가 실내에 존재하기 때문에 새로운 조건이 살균실에 존재하므로, 새로운 어보트 상태가 필요하다. 이것은 고장의 경우에 수반될 일련의 다른 과정을 필요로 한다. 따라서, 새로운 어보트 상태가 제공된다.

상태(13)에서, 가스-유지 시험이 시작된다. 가스 농도가 소정의 시간 GTMR동안 CNOM보다 크거나 같을 경우, 상태(14)가 시작되고, 그렇지 않을 경우 상태(31)이 시작되면 사이클이 어보트된다.

상태(14)에서, 실내의 온도가 측정된다. 이 온도가 최소 온도 TMIN보다 크고 최대 온도 TMAX보다 작을 경우, 상태(15)가 시작되고 살균 타이머가 개시된다. 이 온도가 적당하지 않으면, 상태(31)이 시작되며 어보트가 발생한다. 통상의 작동 온도는 약 30℃이다.

상태(15)동안, 살균이 진행된다. 습도 조절용 밸브(V6)은 여전히 제어되어 개방되어 있고, 밸브(V4) 및 (V4A)도 제어되고 개방된다. 경보 조건, 예를 들어 임의의 조건이 변화한 경우, 즉 밸브가 적당한 상태로 유지되지 않을 경우가 발생한다면, 상태(31)이 시작된다. 통상 수 시간일 수 있는 살균 시간 STMR이 경과한 후만이 상태(16)이 시작된다.

상태(16)에서, 밸브(V4), (V4A) 및 (V6)이 폐쇄되고(만약 이들이 필요한 타임아웃 기간내에 폐쇄되지 않는다면, 상태(31)이 시작됨), 밸브(V3)이 제어상태에 있으며, 밸브(V8)이 또한 개방된다. 상태(17)에서, 등(LT15)가 꺼지고 등(LT16)이 켜진다. 살균 타이머가 타임아웃되고 밸브(V4), (V4A)가 폐쇄되었을 경우, 등(LT15)가 꺼진다. 등(LT16)은 퍼어징 진행중임을 나타낸다. 상태(17)동안, 실내의 가스는 밸브(V3) 및 (V8)과, 이산화염소를 무해 물질로 변환시키는 제 2 도에 DUMP(22)로 표시한 해독기(22)를 통해 제거된다. 상기 계류중인 특허출원 제 601,443호에 기재된 바와 같이, 해독작용은 티오황산 나트륨과 같은 환원제를 통하여 배기된 과산화 염소를 통과시키므로써 수행될 수도 있다. 해독된 가스는 진공펌프(P1)에 의해 밸브(V8)을 거쳐 제거된다. 밸브(V3) 및 (V8)이 타임아웃 기간내에 개방되지 않는다면, 어보트(ABORT)-3인 상태(31)이 시작된다. 상태(17)동안, 살균실(10)이 배기되는 동안의 시간의 양을 제어하는 배기 타이머가 개시된다. 상태(18)은 밸브(V3) 및 (V8)이 둘다 소정의 기간내에 개방될 경우에만 시작된다.

상태(18)에서, 일단 배기 타이머가 시간 ETMR을 지나서 타임아웃되었다면, 상태(19)가 시작되고 밸브(V3) 및 (V8)은 폐쇄된다. 이 밸브(V3) 및 (V8)이 폐쇄될 경우 상태(20)이 시작된다.

제 6 도에 도시된 바와 같이, 경보조건이 발생하거나 또는 밸브(V3) 또는 (V8)이 정해진 시간내에 폐쇄

되지 않는다면, 상태(31)이 시작된다.

상기(20)에서, 밸브(V5)가 제어상태에 있다. 이로써 필요한 경우 질소가스가 시스템내로 들어와서, 일단 밸브(V3)이 상태(22)에서 재개방되면 해독기(22)를 거쳐서 밸브(V2)뒤에 남아 있는 살균 가스가 제거되도록 시스템을 준비시킨다. 상태(20)에서, 압력이 체크된다. 만일 이 압력이 최대 압력 PMAX보다 클 경우, 밸브(V5)가 상태(21)에 폐쇄되어 질소 공급을 중단한다. 만일 압력이 PMAX보다 작다면, 새로운 어보트 상태로서 어보트(ABORT)-4인 상태(32)가 시작된다.

상태(21)에서는 밸브(V5)가 규정된 타임아웃 기간내에 폐쇄되었는지를 결정하도록 체크된다. 만일 폐쇄되지 않았다면 상태(32)로 들어가서 그 사이클은 어보트(ABORT)된다. 상태(22)에서, 시스템내의 잔존 살균 가스가 해독기(22)에 의해 해독되고 밸브(V3) 및 (V8)이 재개방되어 가스가 제거된다. 상기 밸브(V3) 및 (V8)이 충분한 시간동안 개방되면 상태(23)이 시작되지만 단지 밸브(V3) 및 (V8)이 개방되는 경우에 한한다. 상태(23)에서는 (DESORB) 타이머로 표시된 또 다른 타이머가 작동된다. 이로써 살균실내의 물질중에 흡수된 살균가스를 기간 DTMR동안 제거 또는 탈착시킨다.

밸브(V3) 및 (V8)이 개방에 실패하면, 어보트-5, 상태(33)이 시작된다. 이러한 상황에서는 시스템으로부터 살균가스가 제거될 수 있도록 오퍼레이터가 밸브(V9) 및 (V10)을 수동으로 작동시킨다. 상기 밸브(V9) 및 (V10)의 수동 작동성은 밸브 기호 위쪽의 T자형태로 제 2 도에 도시되어 있다. 만일 밸브(V9) 및 (V10)이 수동 개방되면, 상태(33), 어보트-5가 자동으로 시작된다.

상태(23)이 성공적으로 도달되고 DESORB타이머가 DTMR시간후에 타임 아웃될 경우 상태(24)로 들어간다. 이러한 지점에서는 밸브(V2), (V3) 및 (V8)이 폐쇄되고 이들 밸브들이 폐쇄되었는가를 결정하기 위한 체크가 행해진다. 이어서, 상태(25)가 시작되고 그동안 가스 유지시험이 수행된다. 만일 가스농도가 시간주기 GHTM이내에서 허용가능한 값 CMIN보다 작거나 같다면, 상태(26)이 시작된다. 안정도의 허용 가능한 수준은 예컨대, 이산화염소 0.5ppm이하이어야 한다. 그렇지 않을 경우에는 스위치(52)의 조작에 의해 상태(20)으로의 복귀가 행해지기 전에 모의(dummy) 상태(35)로 들어간다. 이것은 밸브를 개방하는 지연시간을 제공한다.

상태(25)에서는 제거단계중의 이산화염소 가스 농도를 측정하는 동안 증폭기가 고증분 모드로 되도록 하기위해 증폭기(14g)(제 3 도 참조)의 증분이 변경된다. 이것은 GCI(증분 변경 제어)상태(25)에 "1"로 표시된다. 이것은 이미 언급한 바와 같이, 시스템의 초과 안정도를 제공하는 제거단계동안 농도를 더욱 정확하게 측정한다. 또한, 상태(25)에서 카운터 CNT(제 12 도의 RB3 참조)가 감소된다. 이 카운터는 카운터(CNTO)의 초기 계수치에 의해 결정되는 정해진 시간동안 시스템을 상태(25), (20), (21), (22), (23) 및 (24)를 통해 상태(35)를 거쳐 순환하게 한다. 따라서, 농도 CMIN이 시간 GHTM이내에 도달되지 않을 경우, 또는 카운터 CNT가 0에 도달되지 않을 경우에는 언제든지 상태(35)로 들어갈 것이다. 농도가 CMIN 이하 이고 CNTØ 가 Ø일 경우 상태(25)로부터 상태(26)이 시작될 것이다. 이것은 살균실내의 농도 센서가 고장일 경우 시스템 안정도를 보장하도록 제공된다. 여러번의 사이클을 통해 상태(35)를 통과하므로써 가스농도가 감소되므로, 비록 농도 센서가 가스 농도가 CMIN이하임을 가리킨다 하더라도 시스템은 농도를 허용 가능한 안정도로 감소시키는데 필요한 회수의 시간을 통해 자동적으로 순환시킬 것이다. 이것은 매우 중요한데, 왜냐하면 농도센서가 고장났거나 이러한 부가적인 안정성이 제공되지 않았을 경우 시스템은 가스 농도가 비록 실제로는 그렇지 않다 하더라도, 허용가능한 안정도내에 있음을 가리키기 때문이다.

상태(26)에서는 카운터가 체크되고, 시스템 사이클을 매회 적어도 단계(26)까지 증가시킨다. 예컨대, 계수치가 3이하일 경우 상태(28)로의 점프가 이루어진다. 만일 3보다 크거나 같다면 상태(27)로 들어간다. 상태(28)에서는 밸브(V5)가 계속 제어되고 이어서 계수치가 증가된다. 이때 질소가스는 살균실로 유입될 수 있다.

사이클 계수가 3보다 크거나 같을 경우 상태(27)로 직접 들어간다. 상태(27)에서는 밸브들(V2), (V3), (V4) 및 (V8)이 개방되고, 모든 잔존 가스가 시스템으로부터 폐기되고 카트리지내의 Cl_2가스도 폐기된다.

충분한 시간이 경과되면, 예를 들어 폐기 지속시간 DHTM이 경과되면, 상태(28)이 시작된다. 상태(28)로부터, 살균실내의 압력이 대기압의 5%이내에 있을때까지 검출되는 동안 시스템은 상태(37)로 들어간다. 이 시점에서 등(LT17), 제거 로우드가 커진다. 또한 상태(38)이 시작되고 등(LT11)이 켜지면 스위치(S2)의 작동으로 말미암아 상태(1)로 복귀될 수 있다. 시스템이 상태(27)을 통과할 경우 오퍼레이터는 가스 카트리지를 대체해야 함을 인지할 수 있을 것이다.

제 6 도 및 제 7 도에서 지시된 바와 같이, 어보트 상태(29) 및 (30)이 시작된 후에는 스위치(S2)가 눌려진 다음 상태(2)로의 복귀가 이루어져 상태(2)의 조건이 설정된다.

어보트 상태(31)에서 상태(20)으로의 복귀가 이루어지고 스위치(S2)가 눌려지면 상태(20)의 조건이 설정된다. 어보트 상태(32)에서는 상태(19)로의 복귀가 이루어지며 상태(19)의 조건이 설정된다. 또, 어보트 상태(33), (34) 및 (36)에서는 각각 상태(23), (25) 및 (37)로의 복귀가 이루어진다. 상태(38)에 도달하면, 오퍼레이터는 사이클을 완료하였음을 알리는 지시를 수신하며 등(LT17)이 켜진다. 구획실 도어를 개방되게 하기 위해 스위치(S2)가 작동되고 상태(1)이 시작된다. 어보트 상태에 도달할 경우 적절한 고장등이 켜진다. 어보트 상태로부터 상태(20), (23) 또는 (26)으로의 복귀가 이루어질 때 시스템은 여러 상태들을 통해 대개 순서적으로 순환하도록 진행한다.

[일반적인 소프트웨어 기능]

시퀀싱 프로그램을 이상에 기술한 바 있다. 일반적으로, 살균 시스템제어기에 대한 소프트웨어는 인터럽트 구동된다. 인터럽트가 발생할때까지는 항상 메인 태스크지명 프로그램에 의해 배경 태스크가 작동한다. 인터럽트시에는 몇가지 가능한 이벤트 소스들로부터 소프트웨어 제어가 적절한 인터럽트 처리 프로그램으로 전송된다. 이것은 제 9 도에 예시되어 있다.

제 9 도에는 메인 태스크지명 프로그램(300)이 도시되어 있다. 이 프로그램은 또한 첨부된 프로그램 리

스트 부록에서 찾아볼 수 있다. 본질적으로, 50밀리초, 1초 또는 1분 기능을 가리키는 타이머 플래그의 존재에 대한 상기 프로그램 모니터가 수행되어야 한다. 이들 플래그가 제 12 도의 상태 레지스터(STAT)(204)에 저장된다. 플래그가 발생할 때, 프로그램(300)은 적절한 타이머 프로그램(318), (300) 또는 (322)로 점프한다. 타이머 프로그램이 우선 순위에 입각하여 수행되므로써 먼저 1분 기능이 수행되고 이어서 50밀리초(T50) 기능이 수행된다.

인터럽트 이벤트에는 4개의 소스, 즉 파워업, 타이머, 통신 및 파워다운 소스가 있다. 상기 파워업, 파워다운 및 통신을 외부 하드웨어 인터럽트이고, 반면에 타이머 인터럽트 TMR0은 프로그램 제어하의 내부 하드웨어 인터럽트이다. 파워업 소스를 제외하고는 각 인터럽트 처리 프로그램이 그것의 태스크 기능을 개시하기 전에 CPU스택내의 작동 프로그램 문맥을 절감하고 그 문맥은 인터럽트된 프로그램이 재소모되기 전에 재기억된다. 타이머 인터럽트 처리기(TMR0)는 모든 다른 비-인터럽트 프로그래밍 기능을 시퀀스한다. 기술된 바와 같이, 상기 처리기는 제 12 도의 상태 레지스터(204)를 통해 시간 조정된 태스크가 작동하고 있음을 의미하는 하나 이상의 플래그(MINF 312, SECF 314, TICF 316)을 통과시켜 상기 프로그래밍 기능을 달성한다. 메인 태스크지명 프로그램(300)은 플래그를 테스트해서 선택된 기능을 (318-322)로 도시된 바와 같이 실행시킨다. 이러한 방법은 추가의 인터럽트 액션을 가능하게 하는 반면에 하위 우선순위 기능을 완료시킨다. 1분, 1초 및 50밀리초(TIC)간격으로 수행되는 기능의 일부는 각기 제 9 도의 (318), (320) 및 (322)로 나타낸 바와 같다. 이하에서는 각각의 인터럽트 이벤트의 카테고리하에서 수행될 태스크를 더욱 상세하게 설명하고자 한다.

[메인 태스크지명 프로그램]

본질적으로, 메인 태스크 지명 프로그램은 타이머 플래그를 찾아서 하나가 발견되면 적절한 서브루우틴을 호출한다(제 9 도 참조). 메인 태스크지명 프로그램은 첨부된 프로그램 목록에서 찾아볼 수 있다.

[파워업]

(310)에 도시된 바와 같이 파워업시에는 프로세서 스택, 레지스터 뱅크 및 다른 기능이 초기화되어야 한다. 이러한 인터럽트 기능은 프로세서 문맥의 절감을 필요로하지 않는다. 대신에, 사전 프로세스가 정보가 전기적으로 재프로그래밍 가능한 메모리 SRAM(106)으로부터 판독될 것이며, 클럭(119)가 리세트되고 그 프로세스는 전술한 어느 상태에서부터든지 재차 개시될 것이다. 워치도그 타이머가 리세트되고 이어서 메인 태스크지명 프로그램(300)으로 제어 작업이 옮겨갈 것이다.

파워업 루틴은 프로그램 명칭 INIT하에 프로그램 부록에 게재하였다.

[전원 고장]

전원고장 프로그램이 구현되는 것이 바람직하다. 제 9 도에 도시된 바와 같은 상기 프로그램에 대한 한 실시예는 (312)에서의 임계 메모리 내용을 SRAM(106)에 기억하는데, 여기서 그 데이터는 전원이 재기억될 때까지 보유된다. 전원 고장 인터럽트는 5볼트 논리선이 4.55볼트 이하로 하강할때는 언제든지 발생하도록 설계될 수 있으며, 4.75볼트로의 회복은 파워업 동안 활용될 수 있다. 전원 고장 프로그램은 첨부된 프로그램 부록에 게재하였다.

[통신]

통신 특징(COM)이 본 발명에 따른 시스템에 선택적으로 구비될 수 있다. 통신 프로그램은 문자가 직렬 출력 버퍼로부터 제거되거나 또는 직렬 입력 버퍼로 들어갈때마다 작동될 수 있다. 이러한 프로그램의 기능은 문자가 전송될 때 전송 버퍼로 문자를 전송하고 또한 문자가 수신될 때 수신버퍼로부터 문자를 제거하는 것이다. 2개의 선입선출(FIFO) 큐(queue)가 입력 및 출력데이터 스트림을 유지하도록 제공될 수 있다. 상기 통신 프로그램은 종료 또는 제어 문자의 존재에 대하여 입출력 데이터 스트림을 시험한다. 종료 문자의 경우에는 플래그가 설정된다. 당분야에 공지된 프로그램은 제어회로에 연결된 전형적인 직렬 인터페이스 장치에 대한 제어문자를 처리하기 위해 제공될 수 있다. 예를 들면, 전화선로를 통해 기록하기 위한 정보를 프린터 또는 디스플레이 장치로 전송하는 것이 바람직할 수 있다. 당분야에 공지된 다른 프로그램은 표준 모뎀 제어기능, 즉 RS232C명령을 처리하도록 사용될 수 있다. 하드웨어 입출력선은 필수적인 모뎀 제어 신호에 대해서도 제공될 수 있다. 통신 프로그램은 프로세서 문맥을 절감하고 재기억한다.

[시간 기능]

제어기에서의 시간 기능은 TIMER0 타이머(매 6250 마이크로초)에 의해 개시되는 기능 50 밀리초마다 개시되는 기능, 매초마다 개시되는 기능 및 매분동안 작동하는 기능과 같은 4가지 레벨로 발생한다. 제 10 도에 더욱 명백하게 도시된 바와 같이 이들 레벨 사이에서는 마이크로프로세서 데이터 베이스에서 정의된 데이터 영역을 통해 데이터가 교환된다. TMR0 프로그램은 또한 제어기에 연결된 입출력장치를 접근시킨다. 매초동안 작동되는 프로그램 제어기능(CTR)은 제 24 도에 도시된 바와 같이 비트 하나하나마다 CTRL 레지스터의 4개의 하위 비트로 인에이블될 때 밸브 명령을 제어 레지스터의 4개의 상위 비트로 전송한다.

제 4 도에 도시된 바와 같이, 타이머 인터럽트(TMR0)는 6250 마이크로초(6.25밀리초)의 간격으로 발생한다. 각각의 인터럽트에서는 TMR0 프로그램이 들어가고 모든 시간 기능이 계획된다. 프로세서의 기본적인 주기가 대략 2 마이크로초이므로 차후의 인터럽트가 있기 전에 3120 명령 사이클이 경과할 것이다. 이러한 시간중 일부는 데이터 수집 및 인터럽트 기능을 수행하기 위해 매회의 타이머 인터럽트시에 사용되는데, 예를 들면 아날로그 입력 및 데이터 입력이 CPU의 내부 RAM에서 판독 및 기억된다. 이것은 제 9 도에(330)으로 도시되어 있다. 타이머 인터럽트 직후에 프로세서 문맥은 적절한 레지스터에서 절감되고 이어서 인터럽트 타이머 TIMER0가 리세트 및 재개시될 것이다. 50 밀리초, 1초 및 1 분의 간격으로 발생하는 프로그램 기능은 전술한 바와 같이 각각의 시간 간격이 경과될때마다 플래그를 통과시키므로써 (332)에 도시된 바와 같이 계획될 것이다. 다음, 데이터 입력, 상태 체크 및 출력이 수행된다. 끝으로, 사진

프로그램 문맥이 재기억되고 인터럽트 복귀가 실행된다. 만일 임의의 시간지정 이벤트가 발생할 경우, 그것들은 메인 태스크지명 프로그램에 의해 순차적으로 수행될 것이다. 그렇지 않을 경우에는 메인 태스크지명 프로그램이 재개시될 것이다.

각각의 타이머에 대해 실행되는 베이직 타이머 프로그램이 제 22 도의 플로우챠트에 도시되어 있다. 도시된 바와 같이 우선 타이머가 단축되고, 그 타이머가 타임아웃, 즉 계수치 0에 도달했는지의 여부를 결정하기 위한 체크가 이루어진다. 만일 타이머가 되었다면, 대응하는 타이머 플래그가 제 12 도에 도시된 TCFL 레지스터(208)에서 세트된다. 그렇지 않을 경우 대응하는 플래그가 소거되고, 이어서 다음 타이머에 대한 프로그램이 실행된다. 그리고 모든 타이머가 완료되면 메인 태스크지명 프로그램에 대한 복귀가 이루어진다.

단축 타이머 기능은 제 23 도에 도시하였다. 도시된 바와 같이, 타이머가 단축되었을 때 시간이 종료되면 플래그가 TCFL 레지스터에 세트된다. 이어서, 적절한 타이머 레지스터(200)에 현재 계수치가 기억된다.

[1. TIMER0 타이머 (TMR0)]

6250 마이크로초마다 발생하는 최하위 레벨의 시간기능은 인터럽트 TIMER0에 의해 시작된다. 이것은 각종 시간기능에 대한 플로우챠트인 제 10 도의 최상측 부분에 표시되어 있다. 프로세서 문맥을 질감한 후에는 TMR0 인터럽트 프로그램의 제 1 기능이 (400)에 표시된 바와 같이 타이머를 리세트 및 재개시시킨다. 이것은 서브루우틴(RRT)에 의해 수행된다. 제 10 도에서, 소정의 기능을 구현하기 위한 대응하는 프로그램이 플로우챠트 기호상에 표시되어 있으며, 첨부된 부록에 게재되어 있다. TMR0 프로그램은 시간-임계 기능이다. 타이머가 재개시되면, 제어기에서의 모든 접점 입력이(410) 및 (412)에 도시된 대응하는 메모리다. 이것은 논리적 처리를 크게 촉진시킨다. 이러한 기능을 구현하기 위한 서브루우틴은 제17도에 도시되어 있고, 또한 서브루우틴(RCT)로서 첨부된 프로그램 부록에 제시되어 있다. 또한 접점 출력 정보가(CC00-CC03)에서 이러한 메모리내에 할당되고 제10도에 (425)로 표시되어 있다. 이어서 인터럽트 프로그램은 이러한 메모리 영역에 기억된 비트 마스크(415)를 사용하여 접점 입력 및 출력 상태 비트의 마스킹된 비교를 수행한다. 이것은 (420)으로 도시된다. 만일 어떤 비트가 그들의 대응하는 소정의 출력과 부합되지 않으면, "무정의" 상태동안 마스크될 때 경보조건이 (430)으로 도시된 바와 같이 상태 레지스터(204)(제12도)내에 비트를 세팅하므로써 설정된다.

타입아웃 경보는 또한 TMR0 프로그램에 의해 구현된다. 제 18 도에 도시된 바와 같이 서브루우틴(CSC2)와 첨부된 프로그램 부록은 어떻게 타임아웃이 결정되는가를 보여준다. 타이머가 타임아웃되면, 예를 들어 밸브가 적시에 폐쇄되었는지 또는 개방되었는지의 여부를 결정하기 위한 타이머가 타임아웃되면, 플래그가 타이머 플래그 레지스터(TCFL)에서 세트될 것이다. 플래그 세팅이 고장시 어보트를 필요로 할 경우, 예를 들어 적시에 폐쇄하려는 밸브의 고장이 어보트 상태를 초래하면, 플래그가 타이머 인에이블 레지스터(TCEN)에서 설정되어야 한다. 이것은 경보 조건이 설정되어야 하는 타임아웃 경보 프로그램을 알리는데, 상기 경보 조건은 상태 레지스터로 적재될 것이다. 이것은 또한 시퀀싱 프로그램에 의해 어보트 상태로의 전이를 유발할 것이다.

다음 전류 접점 출력상태가 그 메모리 화상으로부터 (435)에 도시된 바와 같이 프로그래밍(WC0)에 의해 출력 접점 래치내로 적재된다. 끝으로, (440) 및 (445)에 도시된 바와 같이 현재 아날로그 입력로그 입력 데이터(445)가 판독되어(RAI) 지수적으로 여과되고(FILTER), 비트 어드레스가능 스페이스 외측의 수정 메모리 위치에 기억된다(제 21 도 참조), 8개의 타이머 인터럽트가 50 밀리초를 소요한다. 따라서, 총 8개의 아날로그 입력들중 잘 여과된 아날로그 입력 스캔(가스농도, 압력, 온도 및 습도에 대응하는 4개의 제어 루우프를 위해 사용되는 4개의 입력만)이 매회 50 밀리초의 프로그램이 엔터되는 시간에 이용가능하다. 그러므로, 50 밀리초마다 RAI프로그램은 각각의 채널에 대해 8개씩, 64개의 입력 샘플들을 얻는데, 이 각 채널에 대한 8개의 샘플들이 각 채널에 대해 단일 아날로그 값을 얻도록 평균화된다. 이어서, 메인 태스크지명 프로그램에 대한 복귀가 이루어진다. 제 11 도의 플로우챠트에는 TIMER0 프로그램이 요약되어 있다.

[2. TIC 타이머(T50)]

TIC 기능은 50 밀리초마다 수행되는데, 이것을 시퀀싱(SEQ) 프로그램의 수행을 포함한다. 수행되는 제 1 기능은 (500)으로 도시된 바와 같은 위치 도그 타이머를 리세트하는 것인데, 그 이유는 이러한 타이머가 적시에 리세트되지 않을 경우 모든 밸브 출력이 제 8 도를 참조하여 기술된 바와 같이 디스에이블되기 때문이다. 다음 모든 티크 타이머(TIM)가 (510)에서 감분되며 그들의 계수가(512)에서 기억되고 대응하는 상태 플래그들이 제12도의 레지스터(TCFL)(208)내로 (520)에서 세트 또는 소거된다. 또한 TCFL 레지스터(208)(제12도 참조)내의 타임아웃 플래그의 세팅은 대응하는 비트의 상태가 도시된 바와 같은 시퀀싱 프로그램에 의해 타이머 카운터 인에이블 레지스터(TCEN)(207)에서 결정될 것을 필요로 한다. 이러한 방식으로 대응하는 TCEN 비트가 세트되지 않으면, 타이머 플래그가 작동할 때 제어기가 ABORT 상태로 들어가게 하지 않도록 알린다. 예를 들면, 살균 타이머가 대략 4시간 후에 타임아웃될 때 ABORT상태로 들어가지 않아야 한다. 그러나, 밸브 타임아웃을 위해서는 타이머가 타임아웃될 경우 어보트할 필요가 있으며 밸브가 적시에 개방 또는 폐쇄되지 않아야 하므로 대응하는 TCEN 비트가 시퀀싱 프로그램에 의해 세트되고 경보를 발생시킨다. 만일 밸브를 적시에 폐쇄하면, 대응하는 TECN 비트가 디스에이블되어 경보가 전혀 발생되지 않는다. TICK 타이머가 감분되면 하나의 상태에서 전술한 다음 상태까지의 진행을 제어하는 메인 시퀀싱 로직(515)(SEQ)은 아직 존재하지 않는 특정된 상태 조건을 위한 보관 기능에 기인하여 더 이상 진행하지 않을때까지 수행된다. 이어서, 출력이 (530)에서 메모리(CC0)내의 접점 출력화상으로 적재된다. 즉, 제어하고자 하는 적절한 밸브 또는 히터에 대한 출력 데이터가 메모리에 기억된다. 이어서, TMR0 프로그램 서브루우틴 WC0는 출력 화상을 그 다음 통로상의 제어된 장치로 기록할 것이다. TIC 기능 프로그램은 제 13 도의 플로우챠트에 요약되어 있다.

[3. 제 2 타이머]

매초 모든 1초 타이머들은 단계(550)에서 감분되며, 그 대응하는 상태 비트들이 세트 또는 소거(555)된다. 이것은 ABORT가 타이머 플래그의 발생시점에서 발생하는지의 여부에 따라 플래그 비트(TCFL) 및 적절한 타이머 카운터 인에이블 비트(TCEN)의 세팅을 포함한다. 결국, 시퀀싱 프로그램(515)로부터 설정점(557)을 채택하는 제어 프로그램(559)(CTR)은 내부 RAM의 접점 출력 레지스터내로 계속 적재하기 위해 제어장치에 대한 새로운 출력상태를 CONTROL 레지스터 내로 적재한다. 다음 상태가 TMR0프로그램을 통해 통과할 동안 이들 출력이 제어장치로 공급된다. 제 8 도에 도시된 바와 같이, 시간식 기능이 명령 MIN, SEC 및 TICK에서 발생한다. 1초 프로그램 TIS에 대한 플로우챠트는 제 14 도에 도시했다.

제 14 도에 도시된 바와 같이 1초 타이머 프로그램에 대한 제 1 기능은 상태 레지스터(제 12 도 참조)내에서의 1초 플래그(SECF)의 소거를 포함한다. 이어서 제 23도 에 도시된 바와 같이 제 14 도의 (600)에서 모든 1초 타이머가 감분된다. 이어서, 프로그램 TIS가 (602)에서의 시퀀싱 프로그램으로부터 루우프 상태를 얻으며 특정 루우프에 대한 제어 레지스터(2060)내의 대응하는 제어 비트가 (604)에서 인에이블되는지를 결정한다. 상기 각각의 루우프는 4개의 측정된 아날로그 프로세스 변수, 압력, 온도, 습도 및 가스농도중 하나에 대응한다. 이는 제 24 도에 도시되어 있다. 지시된 바와 같이, 제어 레지스터(206)의 하위 4개의 비트는 4개의 루우프 상태에 해당한다. 상기 루우프가 인에이블될 경우, 그 값은 측정된 입력값, 예컨대 가스농도 또는 압력을 (606)에 도시된 바와 같이 시퀀싱 프로그램으로부터의 기억된 설정점 값으로부터 감산하므로써 결정된다. 만일 이 값이 0보다 클 경우 제어 레지스터내의 4개의 상위 비트들중 대응하는 한 비트가 (607)에서 설정된다. 만일 제어 레지스터 비트가 0이면, 대응하는 제어 레지스터 비트가 (608)에 도시된 바와 같이 소거된다.

(610)에서는 프로그램이 다음 루우프를 얻으며 그 루우프에 대해 단계 A-X를 반복한다. 이어서 후속하는 2개의 루우프가 얻어지며, 이들 2개의 루우프에 대해 단계 A-X가 연속적으로 반복된다. 모두 4개의 루우프가 수행될 때 메인 태스크지명 프로그램에 대한 복귀가 이루어진다.

아날로그 입력 데이터, 설정점, 제어 레지스터, 제어 프로그램(CTR), 출력 적재 프로그램(CTL) 및 접점 출력 CCO간의 상호 관계는 제 24 도에 도시되어 있다. 도시된 바와 같이 프로그램 CTR은 아날로그 입력 데이터 ADI, 설정점 SP 및 메모리로부터의 제어 레지스터(CTRL) 상태를 검색한다. 이어서, 제어 레지스터에 대한 새로운 상태가 제 14 도의 순서도에 따라서 결정되고 그 새로운 상태는 제어 레지스터내로 적재된다. 프로그램 CTL은 밸브 및 히터를 메모리내의 적절한 접점 출력 레지스터로 제어하기 위해 적절한 출력을 적재한다. TMR0 프로그램중에 이들 출력은 프로그램 WCO에 의한 제어장치에 연결된다. 제 10 도 및 20 도를 참조하라.

[4. 분 타이머(TIM)]

1분 간격으로 임의의 일괄처리시간 클럭(119)가 (610)에 도시된 바와 같이 갱신될 수 있다. 이러한 클럭은 적절한 인쇄 또는 디스플레이 장치에 의해 프로세스 상태의 디스플레이를 개시하도록 사용될 수 있다. 모든 1분 타이머가 (620)에서 감분되고, 그들의 대응 상태 비트가 (630)에서 세트 또는 소거된다. TIM프로그램은 제 15 도의 플로우챠트에 요약되어 있다.

본 발명에 따른 가스 살균 시스템에 대한 소프트웨어의 샘플 부록을 이하에 첨부하였다.

[명칭(SC1 살균 제어회로용 프로그램)]

```
************************************************************
CONDTANT DEFINITIONS
************************************************************
MCHAN    EQU    07H          ; MAX A/D CHAN NUMBER
CHMSK    EQU    07H          ; A/D CHANNEL MASK
BNK0      EQU    00H           ; RB0
BNK1      EQU    08H           ; RB1
BNK2      EQU    10H           ; RB2
BNK3      EQU    18H           ; RB3
STATE    EQU    R6            ; CURRENT STATE
ABORT    EQU    R7            ; ABORT STATE
SSTA      EQU    0               ; SRAM OFFSET FOR STATE
SABO      EQU    2               ; SRAM OFFSET FOR ABORT
SCNT      EQU    4               ; SRAM OFFSET FOR COUNT
SMAX      EQU    38             ; MAX. VALID STATE
VDLY      EQU    8               ; VALVE DELAY(400MSEC)
HDLY      EQU    2                ; HEATER DELAY(2 MIN)
TVAC      EQU    30             ; EVAC TIME(30 MIN)
LKHT      EQU    5               ; LEAK HOLD TIME(5 MIN)
```

```
PVAC      EQU    242            ; EVAC PRESSURE(95% FS)
PRLK      EQU    223             ; PRESS. LEAK LIM.(80% FS)
HUMT      EQU    30             ; HUMIDIF. TIME(30 MIN.)
HNOM      EQU    207            ; NOM. HUM. LEVEL(81% FS)
HUMH      EQU    90              ; HUM. HOLD TIME(90% MIN)
TLOW      EQU    0               ; MIN. STERIL. TEMP.(0% FS)
TMAX      EQU    255            ; MAX. STERIL. TEMP.(100%)
CNCT      EQU    15              ; CONC. TIME(15 MIN.)
CNOM      EQU    64             ; NOM. STERIL. CONC.
CONH      EQU    100            ; GAS HOLD TIME(100 MIN)
TSTR      EQU    200            ; STERIL. TIME(200 MIN)
TEVC    EQU    30              ; EVAC. TIME(30 MIN)
PNZT      EQU    15             ; N2 PRESS. TIME(15 MIN)
DSRB      EQU    30             ; DESORB. TIME(30 MIN)
TLGH      EQU    15             ; LOW GAS HOLD TIME(15)
CNTM      EQU    5              ; MIN. NO. OF PURGE CYCLES
CMIN      EQU    25              ; MIN. CONCENTRATION(10%)
PATM      EQU    12             ; ATM PRESS.(5% FS)
PMAX      EQU    28             ; MAX OPER PRESS. (11% FS)
TDMP      EQU    15             ; DUMP HOLD TIME(15 MIN)
PSP1      EQU    60               ; PRESSURE SETPOINT
TSP1      EQU    60               ; TEMPERATURE SETPOINT
HSP1      EQU    60               ; HUMIDITY SETPOINT
CSP1      EQU    60               ; CONCENTRATION SETPOINT
************************************************************
EXTERNAL DEVICE ADDRESSES
************************************************************
;*             EXTERNAL SHADOW RAM
SRAM      XDATA    2000H        ; SHADOW RAM ADDRESS
;
; *         ANALOG INPUTS
IN0       XDATA     6000H        ; CHAN-0 ADDRESS(PRESS.)
IN1       XDATA     6001H         ; CHAN-1 ADDRESS(TEMP.)
IN2       XDATA     6002H         ; CHAN-2 ADDRESS(HUM.)
IN3       XDATA     6003H         ; CHAN-3 ADDRESS(CONC.)
IN4       XDATA     6004H         ; CHAN-4 ADDRESS
IN5       XDATA     6005H         ; CHAN-5 ADDRESS
IN6       XDATA     6006H         ; CHAN-6 ADDRESS
IN7       XDATA     6007H         ; CHAN-7 ADDRESS
;
; *     CLOCK PORT
CLK     XDATA      4000H        ; CLOCK ADDRESS
;
; *     CONTACT INPUTS
```

```
X0      XDATA         0C000H      ; CCI-0 ADDRESS
X1      XDATA         0C001H      ; CCI-1 ADDRESS
X2      XDATA         0C002H      ; CCI-2 ADDRESS
X3      XDATA         0C003H      ; CCI-3 ADDRESS
;
; *     SWITCHES
SU1     XDATA       0C004H       ; SWITCH ADDRESS
;
; *     CONTACT OUTPUTS
Y0     XDATA      0E000H          ; CCO-0 ADDRESS
Y1     XDATA      0E001H          ; CCO-1 ADDRESS
Y2     XDATA      0E002H          ; CCO-2 ADDRESS
Y3     XDATA      0E003H          ; CCO-3 ADDRESS
;
; *    WATCHDOG TIMER
WDT    XDATA    0E004H         ; WATCHDOG RESET ADDRESS
************************************************************
DATA-BASE ALLOCATIONS
************************************************************
DSEG
ORG         05H              ; TIME COUNTERS
TICK         DS           1               ; TICK COUNT
TSEC        DS           1               ; SEC. COUNT
TMIN        DS           1               ; MIN. COUNT
ORG       0CH                ; SIO BUFFER POINTERS
RPUT       DS           1               ; RCV PUT POINTER
RTAK       DS           1               ; RCV TAKE POINTER
TRUT       DS           1               ; XMT PUT POINTER
TTAK       DS           1               ; XMT TAKE POINTER
ORG         1BH              ; TIC TIMERS
TTM0       DS           1               ; TTIMER-0
TTM1       DS           1               ; TTIMER-1
ORG         1AH              ; SECOND TIMERS
STM0       DS            1              ; STIMER-0
STM1       DS            1              ; STIMER-1
ORG         1CH              ; MINUTE TIMERS
MTM0      DS           1               ; MTIMERS-0
MTM1      DS           1               ; MTIMERS-1
ORG          1EH              ; COUNTERS
CNT0       DS            1              ; COUNTR-0
CNT1       DS            1              ; COUNTR-1
BSEG                            ;
ORG           20H             ; INTERNAL BIT SPACE
STAT     DATA     20H             ; STATUS BYTE
```

```
CTRL    DATA       21H            ; CONTROL BYTE
TCEN    DATA       22H            ; TIMER/COUNTER ENABLES
TCFL    DATA       23H            ; TIMER/COUNTER FLAGS
MSK0    DATA       24H           ; OUTPUT MASK REGISTER
MSK1    DATA       25H           ; OUTPUT MASK REGISTER
MSK2    DATA       26H           ; OUTPUT MASK REGISTER
MSK3    DATA       27H           ; OUTPUT MASK REGISTER
ORG        28H         28  ; IMAGED I/O BITS
CCO0    DATA28      28H         ; OUTPUT PORT 1
CCO1    DATA29      29H         ; OUTPUT PORT 1
CCO2    DATA2A      2AH        ; OUTPUT PORT 2
CCO3    DATA      2BH           ; OUTPUT PORT 0
CCI0    DATA      2CH             ; INPUT PORT 0
CCI1    DATA      2DH             ; INPUT PORT 1
CCI2    DATA      2EH             ; INPUT PORT 3
CCI3    DATA      2FH             ; INPUT PORT 3
DSEG
ORG        30H             ; ANALOG DATA IMAGE
ADI0    DS         1                    ; PRESS. INPUT
ADI1    DS         1                    ; TEMP. INPUT
ADI2    DS         1                    ; HUM. INPUT
ADI3    DS         1                    ; CONC. INPUT
ADI4    DS         1                    ; CHANNEL 4 INPUT
ADI5    DS         1                    ; CHANNEL 5  INPUT
ADI6    DS         1                    ; CHANNEL 6 INPUT
ADI7    DS         1                    ; CHANNEL 7  INPUT
ORG      38H               ; INTERNAL DATA AREA
STP0    DS         1                   ; PRESS. SETPOINT
STP1    DS         1                   ; TEMP. SETPOINT
STP2    DS         1                   ; HUM. SETPOINT
STP3    DS         1                   ; CONC. SETPOINT
ORG      3CH               ; BATCH TIME CLOCK
TIME   DS         1                   ; BATCH TIME
************************************************************
DATA DEFINITIONS
[STATUS & CONTROL]
************************************************************
;            STATUS
TICF         BIT      STAT.0         ; TICK FLAG
SECF         BIT      STAT.1        ; SECOND FLAG
MINF        BIT      STAT.3         ; MINUTE FLAG
RCVF        BIT      STAT.4        ; RCV FLAG
XMTF       BIT      STAT.5         ; XMT FLAG
TMOF       BIT      STAT.6         ; TIMEOUTN FLAG
```

```
ALMF        BIT      STAT.7        ; ALARM FLAG
;
;              CTRL
CEN0        BIT     CTRL.0         ; PRESS. LOOP ENABLE
CEN1        BIT     CTRL.1         ; TEMP. LOOP ENABLE
CEN2        BIT     CTRL.2         ; HUM. LOOP ENABLE
CEN3        BIT     CTRL.3         ; CONC. LOOP ENABLE
CTR0        BIT     CTRL.4         ; PRESS. LOOP OUTPUT
CTR1        BIT     CTRL.5         ; TEMP. LOOP OUTPUT
CTR2        BIT     CTRL.6         ; HUM. LOOP OUTPUT
CTR3        BIT     CTRL.7         ; CONC. LOOP OUTPUT
;              TECN
TECN0      BIT     TCEN.0          ; TT0 ENABLE
TECN1      BIT     TCEN.1          ; TT1-ENABLE
TECN2      BIT     TCEN.2          ; ST0 ENABLE
TECN3      BIT     TCEN.3          ; ST1 ENABLE
TECN4      BIT     TCEN.4          ; MT0 ENABLE
TECN5      BIT     TCEN.5          ; MT1 ENABLE
TECN6      BIT     TCEN.6          ; MT2 ENABLE
TECN7      BIT     TCEN.7          ; MT3 ENABLE
;
;            TCFL
TFL0       BIT     TCFL.0            ; TT0 TIMEOUT
TFL1       BIT     TCFL.1            ; TT1 TIMEOUT
TFL2       BIT     TCFL.2            ; ST0 TIMEOUT
TFL3       BIT     TCFL.3            ; ST1 TIMEOUT
TFL4       BIT     TCFL.4            ; MT0 TIMEOUT
TFL5       BIT     TCFL.5            ; MT1 TIMEOUT
TFL6       BIT     TCFL.6            ; CT0 TIMEOUT
TFL7       BIT     TCFL.7            ; CT1 TIMEOUT
************************************************************
DATA DEFINITIONS
[OUTPUT PORTS]
************************************************************
;          OPORT_0
LT01       BIT      CC00.0           ; DOOR-OPEN
LT02       BIT      CC00.1           ; EVAC-FAIL
LT03       BIT      CC00.2           ; FILL-FAIL
LT04       BIT      CC00.3           ; STERIL-FAIL
LT05       BIT      CC00.4           ; PURGE-FAIL
LT06       BIT      CC00.5           ; LOAD-UNSTERILE
LT07       BIT      CC00.6           ; SPARE
LT08       BIT      CC00.7           ; TEST-FAIL
;
```

```
;         OPORT_1
LT11     BIT      CC01.0            ; READY-FOR-CYCLE
LT12     BIT      CC01.1            ; CYCLE-IN-PROGRESS
LT13     BIT      CC01.2            ; EVAC-IN-PROGRESS
LT14     BIT      CC01.3            ; FILL-IN-PROGRESS
LT15     BIT      CC01.4            ; STERIL-IN-PROGRESS
LT16     BIT      CC01.5            ; PURGE-IN-PROGRESS
LT17     BIT      CC01.6            ; REMOVE-LOAD
LT18     BIT      CC01.7            ; SPARE
;
;         OPORT_2
VV01     BIT     CC02.0           ; OPEN-MAIN-VAC-VALVE
VV02     BIT     CC02.1           ; OPEN-MAIN-GAS-VALVE
VV03     BIT     CC02.2           ; OPEN-MAIN-DUMP-VALVE
VV04     BIT     CC02.3           ; OPEN-GAS-CTRL-VALVE
VV05     BIT     CC02.4           ; OPEN-N2-CTRL-VALVE
VV06     BIT     CC02.5           ; OPEN-STEAM-CTRL-VALVE
VV07     BIT     CC02.6           ; OPEN-BREAK-VALVE
VV08     BIT     CC02.7           ; OPEN-DUMP-VAC-VALVE
;
;         OPORT_3
PP01    BIT      CC03.0         ; TURN-P1-ON
HT01   BIT      CC03.1          ; TURN-H1-ON
SPR1    BIT       CC03.2        ; SPARE
SPR2    BIT       CC03.3        ; SPARE
SPR3    BIT       CC03.4        ; SPARE
SPR4    BIT       CC03.5        ; SPARE
ADZC    BIT       CC03.6       ; A/D ZERO CALIB.
LGG1    BIT       CC03.7       ; CONC. HIGH GAIN SWITCH
*************************************************************
DATA DEFINITIONS
[INPUT PORTS]
*************************************************************
;         IPORT_0
LSC1     BIT      CCI0.0         ; V1-CLOSED
LSC2     BIT      CCI0.1         ; V2-CLOSED
LSC3     BIT      CCI0.2         ; V3-CLOSED
LSC4     BIT      CCI0.3         ; V4-CLOSED
LSC5     BIT      CCI0.4         ; V5-CLOSED
LSC6     BIT      CCI0.5         ; V6-CLOSED
LSC7     BIT      CCI0.6         ; V7-CLOSED
LSC8     BIT      CCI0.7         ; V8-CLOSED
;
;        IPORT_1
```

```
LS01    BIT     CCI1.0          ; V1-OPEN
LS02    BIT     CCI1.1          ; V2-OPEN
LS03    BIT     CCI1.2          ; V3-OPEN
LS04    BIT     CCI1.3          ; V4-OPEN
LS05    BIT     CCI1.4          ; V5-OPEN
LS06    BIT     CCI1.5          ; V6-OPEN
LS07    BIT     CCI1.6          ; V7-OPEN
LS08    BIT     CCI1.7          ; V8-OPEN
;
;       IPORT_2
DSC1    BIT     CCI2.0          ; DOOR-SW-CLOSED
DSC1    BIT     CCI2.1          ; TEMP-SW-CLOSED
DSC1    BIT     CCI2.2          ; MAN-SW1-CLOSED
DSC2    BIT     CCI2.3          ; MAN-SW2-CLOSED
DSC1    BIT     CCI2.4          ; SPARE
DSC2    BIT     CCI2.5          ; SPARE
DSC3    BIT     CCI2.6          ; SPARE
DSC4    BIT     CCI2.7          ; SPARE
;
;       IPORT_3
SI05    BIT     CCI3.0          ; SPARE
SI06    BIT     CCI3.1          ; SPARE
SI07    BIT     CCI3.2          ; SPARE
SI08    BIT     CCI3.3          ; SPARE
SI09    BIT     CCI3.4          ; SPARE
SI10    BIT     CCI3.5          ; SPARE
SI11    BIT     CCI3.6          ; SPARE
SI12    BIT     CCI3.7          ; SPARE
*************************************************************
MASK   BIT DEFINITIONS
*************************************************************
;       MASK-REG-0
MVC1    BIT    MSK0.0           ; V1-CLOSED-MASK
MVC2    BIT    MSK0.1           ; V2-CLOSED-MASK
MVC3    BIT    MSK0.2           ; V3-CLOSED-MASK
MVC4    BIT    MSK0.3           ; V4-CLOSED-MASK
MVC5    BIT    MSK0.4           ; V5-CLOSED-MASK
MVC6    BIT    MSK0.5           ; V6-CLOSED-MASK
MVC7    BIT    MSK0.6           ; V7-CLOSED-MASK
MVC8    BIT    MSK0.7           ; V8-CLOSED-MASK
;
;       MASK-REG-1
MV01    BIT     MSK1.0          ; V1-OPEN-MASK
MV02    BIT     MSK1.1          ; V2-OPEN-MASK
```

```
MV03    BIT     MSK1.2          ; V3-OPEN-MASK
MV04    BIT     MSK1.3          ; V4-OPEN-MASK
MV05    BIT     MSK1.4          ; V5-OPEN-MASK
MV06    BIT     MSK1.5          ; V6-OPEN-MASK
MV07    BIT     MSK1.6          ; V7-OPEN-MASK
MV08    BIT     MSK1.7          ; V8-OPEN-MASK
;
;          MASK-REG-2
MDC1    BIT     MSK2.0          ; DS-CLOSED-MASK
MDC1    BIT     MSK2.1          ; TS-CLOSED-MASK
MDC1    BIT     MSK2.2          ; SW1-CLOSED-MASK
MDC2    BIT     MSK2.3          ; SW2-CLOSED-MASK
************************************************************
INTERRUPT VECTORS
************************************************************
CSEG
ORG         0000H
RSTV    LJMP      INIT          ; RESET VECTOR
;
ORG       000BH
TINT:   LJMP      TMR0        ; TIMER0 VECTOR
;
ORG       0013H
PINT:   LJMP      PWRF         ; PWR FAIL VECTOR
;
ORG       001BH
TM1V:   RETI                       : TIMER1 VECTOR
;
ORG       0023H
SIOV :  LJMP    SIOHND       ; SERIAL DATA VECTOR
************************************************************
POWER FAIL HANDLER
************************************************************
PWRF:   CLR     P1.6            ; STORE SRAM DATA
SETB    P1.6              ; DISABLE STORE
RETI                          ; INTERRUPT RETURN
;
************************************************************
TIMER INTERRUPT HANDLER
************************************************************
ORG       0030H
TMR0 :  PUSH      PSW           ; SAVE PROC. STATUS
PUSH      ACC               ; SAVE ACCUMULATOR
PUSH      DPL                ; SAVE  DP(L)
```

```
PUSH      DPH                ; SAVE  DP(H)
MOV       PSW, #BNK0   ; USE RB0
CLR        EA                   ; DISABLE INTERRUPTS
ACALL     RRT                ; RESET AND RESTART TIMERS
ACALL     RCI                 ; READ CONTANT INPUTS
ACALL     CSC                ; CONTACT STATUS CHECK
ACALL     WCO               ; WROTE CONTACT OUTPUTS
ACALL     RAI                 ; READ ANALOG INPUTS
SETB       EA                  ; RESTORE INTERRUPS
TRTN :    POP        DPH            ; RESTORE DP(H)
POP        DPL                 ; RESTORE DP(L)
POP        ACC                ; RESTORE ACCUMULATOR
POP        PSW                ; RESTORE PROC. STATUS
;
RETI                              ; RETURN FROM TIMER0 INT.
;
************************************************************
IMR0 SUBROUTINES
************************************************************
RRT :   CLR        TR0                  ; STOP TIMER0
MOV   A, #LOU(-3120+7)    ; LOAD COUNT(L)
ADD   A, TL0                    ; CORRECT FOR OVERRUN
MOV   TL0, A                   ; RELOAD COUNTER(L)
MOV   A, #HIGH(-3120+7)    ; REPEAT FOR COUNT(H)
ADDC   A, TH0                  ; GET CORRECTED HIGH BYTE
MOV    TH0, A                   ; LOAD COUNTER(H)
SETB   TR0                      ; RESTART TIMER
CLOCK: DJNZ    TICK, CLK3       ; IF 50-MSEC
MOV    TICK, #8                ; RELOAD TICK COUNT
SETB   TICE                     ; SET 50-MSEC FLAG
DINZ    TSEC, CLK2           ; IF 1-SEC
MOV    TSEC, #20             ; RELOAD TSEC COUNT
SETB    SECF                    ; SET 1-SEC FLAG
DJNZ     TMIN, CLK1          ; IF 1-MIN
MOV     TMIN, #60            ; RELOAD TMIN COUNT
SETB    MINE                    ; SET 1-MIN FLAG
SJMP    CLK4                    ; END
CLK1 :  CLR      MINF                ; ELSE, CLR MIN, FLAG
SJMP     CLK4                   ; END
CLK2 :  CLR      SECF                ; ELSE, CLR SEC, FLAG
SJMP     CLK4                   ; END
CLK3 :  CLR      TIFC                ; ELSE, CLR TIC, FLAG
CLK4 :  NOP                          ; END
RET                              ; RETURN FROM TIMER PROG.
```

```
RCI :    MOV     DPTR, #X0          ; POINT CONTACT INPUTS
MOV     R0, #CC10              ; POINT DATA-BASE IMAGE
MOV     R4, #4                    ; FOR R4 :=4 DOWNT  or 0 DO
CI1 :    CLR      R1, 4                    ; ENABLE I/O
MOVX     A, QDPTR            ; GET INPUT
SETB     P1, 4                      ; DISABLE I/O
MOV     QR0, A                  ; STORE IT IN DATA-BASE
INC       DPTR                     ; POINT NEXT INPUT
INC       R0                         ; POINT NEXT NEXT STORAGE
DJNZ     R4, CI1                  ; END
RET                                    ; RETURN
CSC :  CLR      ALMF                  ; CLEAR ALARM FLAG
MOV     A, CCO2                ; GET VALVE OUTPUTS
XRL      A, CCI1                   ; COMPARE WITH LS0 INPUTS
ANL      A, MSK1                ; MASK OFF V0 DON'T CARES
MOV      R2, A                    ; SAVE PARTIAL RESULT
MOV      A, CCO2                ; GET VALVE OUTPUTS
CPL       A                           ; MAKE CLOSED NORMAL
XRL       A, CCI0                ; COMPARE WITH LSC INPUTS
ANL      A, MSK0                ; MASK OFF VC DON'T CARES
ORL      A, R2                      ; ADD PREV. RESULT
JZ        CSC2                       ; IF MISMATCH
SERB     ALMF                    ; SET ALARM FLAG
CSC2 :  NOP                                ; END
MOV    A, TCFL                 ; GET TIMEOUTS
ANL     A, TCEN                 ; TEST IF ENABLED
JZ       CSC4                        ; IF(TM0. AND. TEN)
SETB    TM0F                    ; SET TIMEOUT FLAG
SJMP    CSC5                     ; END
CSC4 :   CLR     TM0F               ; ELSE, CLEAR TIMEOUT FLAG
CSC5 :   NOP                            ; END
RET                                    ; RETURN
UC0 :   MOV     DPTR, #Y0        ; POINT CONTACT OUTPUTS
MOV     R0, #CC00           ; POINT DATA-BASE IMAGE
MOV     R4, #4                    ; FOR R4 : = 4 DOWNTO 0 DO
C01 :    MOV     A, QR0              ; GET OUTPUT DATA
CPL      A                           : INVERT IT FOR OUTPUT
CLR      P1, 4                     ; ENABLE I/O
MOVX   QDPTR, A             ; LOAD OUTPUT LATCH
SETB    P1, 4                      ; DISABLE I/O
INC      DPTR                    ; POINT NEXT OUTPUT
INC      R0                         ; POINT NEXT DATA
DJNZ    R4, C01                ; END
RET                                    ; RETURN
```

```
RAI :    MOV    DPIR, #IN0          ; POINT FIRST ANALOG CHAN
MOV    R0, #AD10             ; POINT FIRST ANALOG DATA
MOV    R4, #8                    ; FOR R4 : = 8 DOWNTO 0 DO
RA1 :   CLR     P1, 4                    ; ENABLE I/O
MOVX :  A, #DPIR              ; GET ANALOG DATA
SETB    P1, 4                     ; DISABLE I/O
ACALL   FILTER                ; FILTER ANALOG DATA
MOV     QR0, A                  ; LOAD IT INTO DATA BASE
INC      DPTR                     ; POINT NEXT CHANNEL
INC      R0                         ; POINT NEXT DATA
DJNZ     R4, RA1                ; END
RET                                    ; RETURN
FILTER :  MOV   B, #020H          ; LOAD FILT. CONST. CB
MUL   AB                         ; B, A : =0.125 R(1)
PUSH   B                          ; SAVE PROD(H)
PUSH   ACC                    ; SAVE PROD(L)
MOV    B, #0E0H           ; LOAD (1-CB) CONST.
MOV    A, QR0                ; GET X(I-1)
MUL    AB                        ; B, A : =0.875* X(I-1)
MOV    R2, B                  ; SAVE HIGH BYTE
POP     B                         ; LOAD PROD(L)
ADD     A,B                    ; ADD LOW BYTES
XCH     A, R2                  ; GET HIGH BYTE
POP      B                         ; LOAD PROD(H)
ADDC    A, B                  ; A, R2 IS FILTERD DATA
RET :                                    ; RETURN
************************************************************
SCHEDULED TIME FUNCTIONS
************************************************************
T50 :    CLR       TICF              ; CLEAR TICK FLAG
MOV      RSE, #BNK2    ; USE RB2
ACALL    RWT                ; RESET WATCHDOG TIMER
ACCALL   DTT               ; DECREMENT TICK TIMERS
ACCALL   SEQ               ; PERFORM SEQUENCE LOGIC
ACCALL   CTL               ; LOAD CONTROL OUTPUTS
RET                                ; RETURN TO DISPATCHING
T1K :   CLR        SECF            ; CLEAR 1-SEC FLAG
MOV      PSW, #BNK2   ; USE RB2
ACCALL   DST               ; DECREMENT SECOND TIMERS
ACCALL   CTR               ; PERFORM CONTROL ACTIONS
RET
T1M :   CTR     MINF                ; CLEAR 1-MIN FLAG
MOV      PSW, #BNK2     ; USE RB2
ACCALL   UBC                ; UPDATE BATCH CLOCK
```

```
ACCALL   DMT              ; CECREMENT MINUTE TIMERS
RET
RUT :   MOV   DPTR, #WDT    ; POINT WATCHDOG TIMER
CLR    A                        ; CLEAR ACCUMULATOR
CLR    P1, #                    ; ENABLE I/O
MOVX   QDPTR, A         ; RESET WATCHDOG TIMER
SETB    P1.4                  ; DISABLE I/O
RET
UBC :   MOV    KO, #TIME      ; POINT TIME(L)
CLR     C                        ; CLEAR CARRY
XCH     A, QRO                ; GET TIME(L)
INC      A                       ; INCREMENT IT
XCH     A, QRO               ; UPDATE TIME(L)
INC      RO                      ; POINT TIME(H)
XCH     A, QRO                ; GET TIME(H)
ADDC    A, %O               ; PROPAGATE CARRY
XCH     A, QRO                ; UPDATE TIME(H)
RET
************************************************************
CONTRO CALCULATIONS
************************************************************
CTR :   MOV    RO, #STPO      ; POINT SETPOINT
MOV    R1, # ADIO         ; POINT DATA
CLR     C                        ; CLEAR CARRY
MOV    A, QRO               ; GET PRESS SETPOINT
SUBB   A, QR1               ; SUBTRACT MEAS. PRESS
JNC     CT2                    ; IF MV> SP
SETB   CTRO                 ; INCREASE OUTPUT
SJMP   CT3                    ; END
CT2 :   CLR     CTRO               ; ELSE, DECR. OUTPUT
CY3 :   NOP                            ; END
INC      RO                       ; POINT NEXT SETPOINT
INC      R1                       ; POINT NEXT MEASUREMENT
CLR      C                        ; CLEAR CARRY
MOV     A, QRO               ; GET TEMP. SETPOINT
SUBB    A, QR1               ; SUBTRACT MEAS. TEMP
JNC      CT4                    ; IF MV> SP
CLR     CTR1                  ; DECREASE OUTPUT
SJMP    CT5                    ; END
CT4 :   SETB    CTR1              ; ELSE, INCR. OUTPUT
CT5 :   NOP                            ; END
INC       RO                       ; POINT NEXT SETPOINT
INC       R1                       ; POINT NEXT MEASUREMENT
CLR       C                        ; CLEAR CARRY
```

```
MOV     A, QR0                ; GET HUM. SETPOINT
SUBB    A, QR1                ; SUBTRACT HUM. MEAS.
JNC      CT6                    ; IF MF> SP
CLR      CTR2                  ; DECREASE OUTPUT
SJMP     CT7                   ; END
CT6 :   SETB    CTR2               ; ELSE, INCREASE. OUTPUT
CT7 :   NOP                            ; END
INC      R0                        ; POINT NEXT SETPOINT
INC      R1                        ; POINT NEXT MEASUREMENT
CLR      C                         ; CLEAR CARRY
MOV     A, QR0                ; GET HUM. SETPOINT
SUBB    A, QR1                ; SUBTRACT CONC. MEAS.
JNC      CT8                      ; IF MV> SP
CLR      CTR3                  ; DECREASE OUTPUT
SJMP     CT9                   ; END
CT8 :   SETB    CTR3               ; ELSE, INCR. OUTPUT
CT9 :   NOP                            ; END
RET                            ; RETURN
************************************************************
SOFTWARE TICK TIMERS (50MSEC)
************************************************************
DTT :   MOV    R0, #TTM0     ; POINT FIRST TICK TIMER
MOV    A,QR0             ; GET LAST COUNT
JZ       TT1                    ; IF COUNT <> 0
DEC      A                     ; DECREMENT ACC.
MOV    QR0, A            ; UPDATE COUNT
JZ       TT1                    ; IF NOT TIMEOUT
CLR      TFL0                ; CLEAR FLAG
SJMP    TT2                 ; END
TT1 :   SETB    TFL0               ; ELSE, SET FLAG
TT2 :   NOP                        ; END
MOV    R0, #TTM1       ; POINT SECOND TICK TIMER
MOV    A,QR0             ; GET LAST COUNT
JZ       TI4                    ; IF COUNT <> 0
DEC      A                     ; DECREMENT ACC.
MOV    QR0, A            ; UPDATE COUNT
JZ       TT4                    ; IF NOT TIMEOUT
CLR      TFL1                ; CLEAR FLAG
SJMP    TT5                 ; END
TT4 :   SETB    TFL1               ; ELSE, SET FLAG
TT5 :   NOP                        ; END
RET                            ; RETURN
************************************************************
SOFTWARE SECOND TIMERS
```

```
************************************************************
DST    MOV    R0, #STM0   ; POINT FIRST SEC. TIMER
MOV    A,QR0             ; GET LAST COUNT
JZ       ST1                   ; IF COUNT <> 0
DEC     A                     ; DECREMENT ACC.
MOV    QR0, A            ; UPDATE COUNT
JZ       ST1                   ; IF NOT TIMEOUT
CLR      IFL2                  ; CLEAR FLAG
SJMP    ST2                 ; END
ST1 :   SETB    ST2              ; ELSE, SET FLAG
ST2 :   NOP                          ; END
MOV    R0, #STM1     ; POINT NEXT SECOND. TIMER
MOV    A,QR0             ; GET LAST COUNT
JZ       ST4                   ; IF COUNT <> 0
DEC     A                     ; DECREMENT ACC.
MOV    QR0, A             ; UPDATE COUNT
JZ       ST4                    ; IF NOT TIMEOUT
CLR      TFL3                  ; CLEAR FLAG
SJMP    ST5                  ; END
ST4 :   SETB    TFL3             ; ELSE, SET FLAG
ST5 :   NOP                           ; END
RET                                ; RETURN
************************************************************
SOFTWARE MINUTE TIMERS
************************************************************
DMT :  MOV    R0, #MTM0      ; POINT FIRST MIN TIMER
MOV    A,QR0                  ; GET LAST COUNT
JZ       MT1                       ; IF COUNT <> 0
DEC     A                         ; DECREMENT ACC.
MOV    QR0, A                ; UPDATE COUNT
JZ       MT1                       ; IF NOT TIMEOUT
CLR      TFL4                    ; CLEAR FLAG
SJMP    MT2                     ; END
MT1 :   SETB    MT2                 ; ELSE, SET FLAG
MT2 :   NOP                               ; END
MOV    R0, #MTM1           ; POINT SECOND MIN TIMER
MOV    A,QR0                   ; GET LAST COUNT
JZ       MT4                       ; IF COUNT <> 0
DEC     A                         ; DECREMENT ACC.
MOV    QR0, A                ; UPDATE COUNT
JZ       MT4                       ; IF NOT TIMEOUT
CLR     TFL5                    ; CLEAR FLAG
SJMP    MT5                     ; END
MT4 :   SETB    MT5                 ; ELSE, SET FLAG
```

```
MT5 :   NOP                              ; END
RET                                    ; RETURN
*************************************************************
SOFTWARE COUNTERS
*************************************************************
DCT0 : MOV    R0, #CNT0          ; POINT FIRST COUNTER
MOV    A,QR0                    ; GET LAST COUNT
JZ       DC1                        ; IF COUNT <> 0
DEC     A                           ; DECREMENT ACC.
MOV    QR0, A                   ; UPDATE COUNT
JZ       DC1                        ; IF NOT ZERO
CLR     TFL6                      ; CLEAR FLAG
SJMP    DC2                      ; END
DC1 :   SETB    TFL6                ; ELSE, SET FLAG
DC2 :   NOP                             ; END
RET                                    ; RETURN
DCT1 :  MOV    R0, #CNT1        ; POINT SECOND COUNTER
MOV    A,QR0                    ; GET LAST COUNT
JZ       DC3                        ; IF COUNT <> 0
DEC     A                           ; DECREMENT ACC.
MOV    QR0, A                   ; UPDATE COUNT
JZ       DC3                        ; IF NOT ZERO
CLR      TFL7                     ; CLEAR FLAG
SJMP    DC4                      ; END
DC3 :   SETB    TFL7                ; ELSE, SET FLAG
DC4 :   NOP                             ; END
RET                                    ; RETURN
*************************************************************
COUNTROL OUTPUTS
*************************************************************
CTL :    MOV    C, CTR0              ; GET OUTPUT-0
ANL     C, CEN0                 ; ALLOW IF ENABLED
MOV    VV05, C                ; OUTPUT TO V5
MOV    C, CTR1                 ; GET OUTPUT-1
ANL    C, CEN1                  ; ALLOW IF ENABLED
MOV    HT01, C                 ; OUTPUT TO H1
MOV    C, CTR2                 ; GET OUTPUT-2
ANL    C, CEN2                  ; ALLOW IF ENABLED
MOV    VV06, C                ; OUTPUT TO V6
MOV    C, CTR3                 ; GET OUTPUT-3
ANL    C, CEN3                  ; ALLOW IF ENABLED
MOV    VV04, C                ; OUTPUT TO V4
RET
*************************************************************
```

```
POWER-ON INITIALIZATION
************************************************************
INIT :   MOV     SP, #060H          ; INITIALIZE STACK POINTER
MOV     PSW, #BNK0         ; USE RB0
CLR      A                          ; CLEAR ACCUMULATOR
MOV     R0, #2                  ; POINT LOWEST RAM LOC.
MOV     R1, #126               ; FOR R1 : =126 DOWNTO 0 DO
ILP :    MOV     RR0, A              ; CLEAR MEMORY LOC.
INC      R0                         ; POINT NEXT LOCATION
DJNZ    R1, ILP                  ; END
MOV    TICK, #8               ; INITIALIZE TICK COUNT
MOV    TSEC, #20              ; INITIALIZE SEC. COUNT
MOV    TMIN, #60              ; INITIALIZE MIN. COUNT
MOV    PSW, #BNK1          ; USE RB1
MOV    RPUT, #40H           ; INITIALIZE RPUT POINTER
MOV    RTAK, #40H           ; INITIALIZE RTAK POINTER
MOV    TPUK, #50H           ; INITIALIZE TPUT POINTER
MOV    TTAK, #50H           ; INITIALIZE TTAK. POINTER
MOV    PSW, #BNK2          ; USE RB2
MOV    STATE, #0             ; STATE : =0
MOV    ABORT, #0            ; ABORT : =0
MOV    SCON, #052H         ; SET SERUAL PORT BITS
MOV    TMOD, #061H        ; SET TIMER MODES
MOV    87H, #00H               ; SET SMOD : = 0 IN PCON
MOV    1P, #002H               ; SET INTERRUPT PRIORITIES
MOV    1E, #096H               ; ENABLE INTERRUPTS
MOV    TL0, #LOW(-3120)  ; LOAD COUNT(L)
MOV    TH0, #HIGH(-3120)  ; LOAD COUNT(H)
MOV    TH1, #-13               ; SET BAUD RATE(1200)
MOV    A, #0FFH              ; SET ACCUM. ALL 1'S
CLR     P1.4                       ; ENABLE I/O
MOV    DPTR, #Y0            ; POINT Y0 OUTPUTS
MOVX   RDPTR, A            ; CLEAR Y0
MOV    DPTR, #Y1            ; POINT Y1 OUTPUTS
MOVX   RDPTR, A            ; CLEAR Y1
MOV    DPTR, #Y2            ; POINT Y2 OUTPUTS
MOVX   RDPTR, A            ; CLEAR Y2
MOV    DPTR, #Y3            ; POINT Y3 OUTPUTS
MOVX   RDPTR, A            ; CLEAR Y3
SETB    P1.4                      ; DISABLE I/O
ACALL  RWT                     ; RESET WATCHDOG TIMER
MOV    TIME, #0               ; CLEAR TIME(L)
MOV    TIME+1, #0           ; CLEAR TIME(H)
SETB    TR1                       ; START BAUD CLOCK
```

```
SETB    TR0                         ; START TIMER
SJMP    MAIN                        ; START MAIN PROGRAM
;
TEST :  RET                                 ; TEST COMPUTER FUNCTIONS
*************************************************************
SEQUENCING PROGRAM
*************************************************************
SEQ :   NOP                                     ; REPEAT
MOV     PSW, #BNK2              ; USE RB2
MOV     A, STATE                  ; COMPARE. MAX, STATE
ADD     A, #NOT(SMAX)         ; COMPARE. MAX, STATE
JNC      SQ1                            ; IF INVALID STATE
MOV     A, #31                       ; TAKE STATE #31
MOV     STATE, A                  ; SET STATE TO #31
SJMP     SQ2                           ; END
SQ1 :    MOV    A, STATE              ; ELSE, USE CURRENT STATE
SQ2 :    NOP                                    ; END
RL       A                                 ; MAKE IT 4-BYTE-
RL       A                                 ; ADDRESS OFFSET
MOV     DPTR, #JMPTBL       ; OFFSET IN JUMP TABLE
JMP     RA+DPTR                     ; PERFORM STATE
SEQR :  MOV   C, ALMF                 ; GET ALARM FLAG
ORL     C, TMOF                     ; OR WITH TIMEOUT FLAG
JNC      SQ3                            ; IF(ALM. OR. TMO)
MOV     A, ABORT                  ; GET ABORT STATE
MOV   STATE, A                      ; SET STATE TO ABORT
CLR     F0                                ; CLEAR HOLD FLAG
SQ3 :   NOP                                     ; END
JNB      F0, SEQ                        ; UNTIL HOLD
RET                                         ; RETURN
*************************************************************
MAIN DISPATCHING PROGRAM
*************************************************************
MAIN : NOP                                      ; DO FOREVER
JNB      MINF, MN1                  ; IF 1-MIN TIME
LCALL   T1M                           ; DO 1-MIN FUNCTIONS
MN1 :   JNB     SECF, MN2               ; IF 1-SEC TIME
LCALL   T1X                           ; DO 1-SEC-FUNCTIONS
MN2 :   JNB     TICF, MN3               ; IF TICK TIME
LCALL   T50                           ; DO TICK FUNCTIONS
MN3 :   JNB     RCVF, MN4               ; IF RCV TIME
LCALL   RCV                           ; DO RCV FUNCTIONS
MN4 :   JNB     XMTF, MN5               ; IF XMT TIME
LCALL   XMT                           ; DO XMT FUNCTIONS
```

```
MN5 :  LCALL   TEST                        ; ELSE, PERFORM TESTS
SJMP    MAIN                              ; END
GTCT : MOV    A, #1                        ; READ SRAM
RET
RCV :   CLR     RCVF                        ; RESET RCV FLAG
RET
XMT :   CLR    XMTF                         ; RESET XMT FLAG
RET
SIOHND : RET                                ; SERIAL I/O HANDLER
$INCLUDE(STATES. STC)
;
END
************************************************************
STATE JUMP TABLE
************************************************************
JMPTBL :   LJMP    STATE0
DB       0
LJMP    STATE1
DB       0
LJMP    STATE2
DB       0
LJMP    STATE3
DB       0
LJMP    STATE4
DB       0
LJMP    STATE5
DB       0
LJMP    STATE6
DB       0
LJMP    STATE7
DB       0
LJMP    STATE8
DB       0
LJMP    STATE9
DB       0
LJMP    STATE10
DB       0
LJMP    STATE11
DB       0
LJMP    STATE12
DB       0
LJMP    STATE13
DB       0
LJMP    STATE14
```

```
DB      0
LJMP    STATE15
DB      0
LJMP    STATE16
DB      0
LJMP    STATE17
DB      0
LJMP    STATE18
DB      0
LJMP    STATE19
DB      0
LJMP    STATE20
DB      0
LJMP    STATE21
DB      0
LJMP    STATE22
DB      0
LJMP    STATE23
DB      0
LJMP    STATE24
DB      0
LJMP    STATE25
DB      0
LJMP    STATE26
DB      0
LJMP    STATE27
DB      0
LJMP    STATE28
DB      0
LJMP    STATE29
DB      0
LJMP    STATE30
DB      0
LJMP    STATE31
DB      0
LJMP    STATE32
DB      0
LJMP    STATE33
DB      0
LJMP    STATE34
DB      0
LJMP    STATE35
DB      0
LJMP    STATE36
```

```
DB       0
LJMP    STATE37
DB       0
LJMP    STATE38
DB       0
;
STATE0 :  MOV     STATE, #1 ; STATE : =1
MOV     ABORT, #1    ; ABORT : =1
MOV     STAT, #00H   ; RESET STATUS
MOV     CTRL, #00H   ; RESET CONTROLS
MOV     TCEN, #00H   ; RESET ALARM
MOV     TCFL, #00H   ; RESET TIMEOUT FLAGS
MOV     MSK0, #00H   ; RESET CLOSED MASKS
MOV     MSK1, #00H   ; RESET OPEN MASKS
MOV     MSK2, #00H   ; RESET MIgC  MASKS
MOV     MSK3, #00H   ; RESET MISC MASKS
MOV     CC00, #00H   ; RESET ALARM LIGHTS
MOV     CC01, #00H   ; RESET RUN LICHTS
MOV     CC02, #00H   ; RESET ALL VALVES
MOV     CC03, #00H   ; RESET MISC. OUTPUTS
CLR      F0                  ; CLEAR HOLD FLAG
LJMP     SEQR            ; RETURN
;
STATE1 :  JNB    DSC1, S11      ; IF DOOR CLOSED
MOV     STATE, #2       ; STATE : =2
MOV     ABORT, #29    ; ABORT : =29
CLR      LT01                ; DOOR-OPEN(OFF)
SETB     LT11               ; READY-FOR-CYCLE(ON)
CLR      F0                   ; CLEAR HOLD FLAG
SJMP     S12                 ; END
S11 :       SETB    LT01             ; ELSE, DOOR-OPEN(ON)
CLR     LT11                ; READY-FOR- CYCLE(OFF)
SETB    F0                  ; SET HOLD FLAG
S12 :       NOP                           ; END
LJMP   SEQR               ; RETURN
;
STATE2 :  JNB    SWC1, S21             ; IF START-CYCLE(PUSHED)
MOV     STATE, #3              ; STATE : =3
MOV     ABORT, #29            ; ABORT : =29
CLR      LT11                        ; READY-FOR-CYCLE(OFF)
SETB     LT12                       ; CYCLE-IN-PROGRESS(ON)
MOV     CNT0, #CNTM        ; LOAD MIN. COUNT
CLR      TFL6                        ; CLEAR COUNT FLAG
CLR      MVC7                      ; CLEAR VC7 MASK
```

```
CLR      MVO7                    ; CLEAR VO7 MASK
CLR      VVO7                    ; CLOSE-BREAK-VALVE
MOV    TTMO, #VDLY          ; LOAD TIMEOUT DELAY
CLR     TFLO                     ; RESET TIMEOUT FLAG
SETB    TENO                    ; ENABLE TIMEOUT ALARM
CLR      FO                         ; CLEAR HOLD FLAG
SJMP     S23                       ; END
S21 :       JB       DSC1, S22              ; ELSE, IF DOOR-OPEN
MOV     STATE, #1              ; STATE : =1
MOV     ABORT, #29           ; ABORT : =29
CLR      FO                         ; CLEAR HOLD FLAG
S22 :       SETB    FO                      ; ELSE, SET HOLD FLAG
S23 :       NOP                                ; END
LJMP     SEQR                    ; RETURN
;
STATE3 :  JNB    LSC7, S31              ; IF V7 CLOSED
MOV     STATE, #4              ; STATE : =4
MOV     ABORT, #29            ; ABORT : =29
CLR      TENO                       ; CLEAR TIMEOUT ENABLE
SETB     MVC7                     ; SET VC7 MASK
SETB     MVC7                     ; SET VC7 MASK
SETB     HTO1                      ; TURN HEATER ON
MOV  MTMO, #HDLY           ; LOAD HEATER TIMEOUT
CLR      TFL4                       ; RESET TIMEOUT FLAG
SETB     TEN4                      ; ENBLE TIMEOUT ALARM
CLR       FO                          ; CLEAR HOLD FLAG
SJMP     S33                        ; END
S31 :       JB       DSC1, S32              ; ELSE, IF DOOR OPEN
MOV     A, ABORT               ; GET ABORT STATE
MOV     STATE, A               ; STATE : = ABORT-1
SETB    LTO1                     ; DOOR-OPEN(ON)
CLR      FO                         ; CLEAR HOLD FLAG
SJMP     S33                       ; END
S32 :       SETB    FO                      ; ELSE, SET HOLD FLAG
S33 :       NOP                                ; END
LJMP     SEQR                    ; RETURN
;
STATE4 :  JNB      TSC1, S41              ; IF HEATER ON
MOV     STATE, #5              ; STATE : =5
MOV     ABORT, #29            ; ABORT : =29
CLR      TEN4                       ; CLEAR TIMEOUT ENABLE
SETB     MTC1                     ; SET TEMP SU MASK
MOV   STP1, #TSP1             ; LOAD TEMP.SETPOINT
SETB     CEN1                       ; ENABLE TEMP. CONTROL
```

```
CLR     MVC1                          ; CLEAR VC1 MASK
CLR     MVO1                          ; CLEAR VO1 MASK
SETB    VVO1                          ; OPEN V1
MOV   TTMO, #VDLY                  ; LOAD TIMEOUT DELAY
CLR     TFLO                            ; RESET TIMEOUT FLAG
SETB    TENO                          ; ENABLE TIMEOUT ALARM
CR2      FO                               ; CLEAR HOLD FLAG
SJMP    S43                           ; END
S41 :       JB      DSC1, S42                    ; ELSE, IF DOOR OPEN
MOV     A, ABORT                    ; GET ABORT STATE
MOV     STATE, A                     ; STATE : = ABORT-1
SETB    LTO1                          ; DOOR-OPEN(ON)
CLR     FO                              ; CLEAR HOLD FLAG
SJMP    S43                           ; END
S42 :       SETB    FO                           ; ELSE, SET HOLD FLAG
S43 :       NOP     SEQR                      ; END
;           LJMP    SEQR                        ; RETURN
;
STATE5 :  JNB     LSO1, S51                  ; IF VAC VALVE OPEN
MOV     STATE, #6                   ; STATE : =6
MOV     ABORT, #29                  ; ABORT : =29
CLR     TENO                          ; CLEAR TIMEOUT ENABLE
SETB    MVC1                          ; SET VC1 MASK
SETB    MVO1                          ; SET VO1 MASK
SETB    PPO1                           ; TURN P1 ON
SETB    LT13                          ; EVAC-IN-PROGRESS(ON)
MOV   MTMO, #TVAC                ; LOAD EVAC TIME
CLR     TFL4                             ; RESET TIMEOUT FLAG
CLR     FO                                ; CLEAR HOLD FLAG
SJMP    S53                           ; END
S51 :       JB      DSC1, S52                    ; ELSE, IF DOOR OPEN
MOV     A, ABORT                    ; GET ABORT STATE
MOV     STATE, A                     ; STATE : = ABORT-1
SETB    LTO1                          ; DOOR-OPEN(ON)
CLR     FO                              ; CLEAR HOLD FLAG
SJMP    S53                           ; END
S52 :       SETB    FO                           ; ELSE, SET HOLD FLAG
NOP     SEQR                          ; END
S53 :       LJMP    SEQR                      ; RETURN
;
STATE6 :  JNB     TFL4, S62                   ; IF EVAC TIME
CLR     C                                ; CLEAR CARRY
MOV     A, ADIO                      ; GET PRESSURE
SUBB    A, #PVAC                     ; SUBTRACT PRESS, LIMIT
```

```
JC        S61                                ; IF P.LE.PVAC
MOV     STATE, #7                          ; STATE : =7
MOV     ABORT, #29                         ; ABORT : =29
CLR      MVC1                              ; CLEAR VC1 MASK
CLR      MV01                              ; CLEAR V01 MASK
CLR      VV01                              ; CLOSE V1
MV0     TTM0, #VDLY                        ; LOAD TIMEOUT
CLR      TFL0                              ; RESET TIMEOUT FLAG
SETB    TEN0                               ; ENABLE TIMEOUT ALARM
CLR      F0                                ; CLEAR HOLD FLAG
SJMP     S63                               ; END
S61 :       SETB     LT02                  ; ELSE, EVAC-FAIL(ON)
MOV     A, ABORT                           ; GET ABORT STATE
MOV     STATE, A                           ; STATE : = ABORT-1
CLR      F0                                ; CLEAR HOLD FLAG
SJMP     S63                               ; END
S62 :       SETB    F0                     ; ELSE, SET HOLD FLAG
S63 :       NOP                            ; END
S63 :       LJMP     SEQR                  ; RETURN
;
STATE7 :  JNB      LSC1, S71                ; IF V1 CLOSED
MOV     STATE, #8                          ; STATE : =8
MOV     ABORT, #29                         ; ABORT : =29
CLR      TEN0                              ; DISABLE TIMEOUT
SETB    MVC1                               ; SET VC1 MASK
SETB    MOV1                               ; SET V01 MASK
MOV  MTM0, #LKHT                           ; LOAD LEAK HOLD TIME
CLR      TFL4                              ; RESET TIMEOUT FLAG
CLR      F0                                ; CLEAR HOLD FLAG
SJMP     S72                               ; END
S71 :       SETB     F0                    ; ELSE, SET HOLD FLAG
NOP                                        ; END
S72 :       LJMP     SEQR                  ; RETURN;
;
STATE8 :  JNB      TFL4, S82                ; IF LEAK HOLD TIME
CLR      C                                 ; CLEAR CARRY
MOV      A, AD10                           ; GET PRESSURE
SUBB     A, #PRLK                          ; SUBTRACT LEAK LIMIT
JC        S81                              ; IF P.LE. PRLK
MOV     STATE, #9                          ; STATE : =9
MOV     ABORT, #30                         ; ABORT : =30
MOV     STP2, #HSP1                        ; GET HUM. SETPOINT
CLR      MV06                              ; CLEAR V06 MASK
CLR      MVC6                              ; CLEAR VC6 MASK
```

```
SETB    CEN2                                    ; ENABLE HUM. LOOP(V6)
MOV     MTM0, #HUMT                     ; LOAD HUM. TIMER
CLR      TFL4                                   ; RESET TIMEOUT FLAG
CLR      LT13                                ; EVAC-IN-PROGRESS(OFF)
SETB LT14                                   ; FILL-IN-PROGRESS(ON)
CLR    F0                                     ; CLEAR HOLD FLAG
SJMP     S83                                   ; END
S81 :       SETB     LT02                          ; ELSE, EVAC-FAIL(ON)
MOV     A, ABORT                        ; GET ABORT STATE
MOV     STATE, A                        ; STATE : = ABORT-1
CLR      F0                                    ; CLEAR HOLD FLAG
SJMP     S83                                ; END
S82 :       SETB     F0                              ; ELSE, SET HOLD FLAG
S83 :       NOP                                      ; END
LJMP     SEQR                             ; RETURN
;
STATE9 :  JNB      TFL4, S92                    ; IF HUM TIME
CLR         C                               ; CLEAR CARRY
MOV       A, ADI3                       ; GET HUMIDITY
SUBB      A, #HNOM                  ; SUBTRACT HUM. LEVEL
JC           S91                              ; IF HUM. GE. HNOM
MOV       STATE, #10                ; STATE : =10
MOV       ABORT, #30               ; ABORT : =30
MOV         MTM0,#HUME         ; LOAD HUM. HOLD TIMER
CLR         TFL4                          ; RESET TIMEOUT FLAG
CLR           F0                             ; CLEAR HOLD FLAG
SJMP        S83                             ; END
S91 :       SETB       LT03                          ; ELSE, FILL-FAIL(ON)
MOV     A, ABORT                        ; GET ABORT STATE
MOV     STATE, A                        ; STATE : = ABORT-2
SJMP     S93                                ; END
S92 :       SETB     F0                              ; ELSE, SET HOLD FLAG
S93 :       NOP                                      ; END
LJMP     SEQR                             ; RETURN
;
STATE10 : JNB      TFL4, S101                ; IF HUM. HOLD TIME
MOV     STATE, #11                  ; STATE : =11
MOV     ABORT, #30                 ; ABORT : =30
CLR      MVC2                           ; CLEAR VC2 MASK
CLR      MV02                           ; CLEAR V02 MASK
SETB    VV02                            ; OPEN V2
CLR     MVC8                            ; CLEAR VC8 MASK
CLR      MV08                           ; CLEAR V08 MASK
SETB    VV08                            ; OPEN V8
```

```
MOV  TTM0,#VDLY                    ; LOAD VALVE TIMEOUT
CLR    TFL0                              ; RESET TIMEOUT FLAG
SETB  TEN0                              ; ENABLE TIMEOUT ALARM
MOV    STP0, #PSP1                  ; GET PRESS. SETPOINT
CLR     MV05                             ; CLEAR V05 MASK
CLR     MV05                             ; CLEAR VC5 MASK
SETB   CEN0                              ; ENABLE PRESS. LOOP(V5)
CLR      F0                                 ; CLEAR HOLD FLAG
SJMP     S102                             ; END
S101 :      SETB     F0                             ; ELSE, SET HOLD FLAG
S102 :      NOP                                       ; END
LJMP     SEQR                            ; RETURN
;
STATE11 : MOV     C, LS02                    ; TEST V2 OPEN-
ANL      C, LS08                          ; AND V8 OPEN
JNC      S111                               ; IF(V2, AND, V8) OPEN
MOV     STATE, #12                   ; STATE : =12
MOV     ABORT, #31                  ; ABORT : =31
CLR      TEN0                              ; DISABLE TIMEOUT FLAG
SETB     MVC2                            ; SET VC2 MASK
SETB     MV02                            ; SET V02 MASK
SETB     MVC8                            ; SET VC8 MASK
SETB     MV08                            ; SET VC8 MASK
MOV     STP3, #CSP1                ; GET CONC. SETPOINT
CLR      MOV4                            ; CLEAR V04 MASK
CLR      MVC4                            ; CLEAR VC4 MASK
STEB    CEN3                           ; ENABLE CONC. LOOP(V4)
MOV     MTM0, #CNCT            ; LOAD CONC. TIMER
CLR     TFL4                               ; RESET TIMEOUT FLAG
CLR     LT14                              ; FILL-IN-PROGRESS(OFF)
SETB   LT15                              ; STERIL-IN-PROGRESS(ON)
CLR      F0                                 ; CLEAR HOLD FLAG
SJMP     S112                             ; END
S111 :      SETB     F0                             ; ELSE, SET HOLD FLAG
S112 :      HOP                                       ; END
LJMP     SEQR                           ; RETURN
;
STATE12 : JNB     TFL4, S122               ; IF CONC. TIME
CLR      C                                 ; CLEAR CARRY
MOV    A, AD13                       ; GET CONC.
SUBB   A, #CNOM                    ; SUBTRACT CONC. LEVEL
JC       S121                               ; IF CONC. GE. CNOM
MOV     STATE, #13                ; STATE : =13
MOV     ABORT, #31                ; ABORT : =31
```

```
MOV MTMO, #CONH                ; LOAD CONC. HOLD TIMER
CLR     TFL4                           ; RESET TIMEOUT FLAG
CLR     F0                               ; CLEAR HOLD FLAG
SJMP     S123                          ; END
S121 :      SETB     LT04                      ; ELSE, STERIL-FAIL(ON)
MOV     A, ABORT                   ; GET ABORT STATE
MOV     STATE, A                    ; STATE : = ABORT-3
CLR     F0                                ; CLEAR HOLD FLAG
SJMP     S123                         ; END
S122 :      SETB     F0                         ; ELSE, SET HOLD FLAG
S123 :      NOP                                  ; END
LJMP     SEQR                       ; RETURN
;
STATE13 : JNB     TFL4, S123          ; IF GAS HOLD TIME
CLR      C                             ; CLEAR CARRY
MOV    A, ADI3                     ; GET CONC.
SUBB   A, #CNOM               ; SUBTRACT CONC. LEVEL
JC       S131                          ; IF CONC. GE. CNOM
MOV     STATE, #14            ; STATE : =14
MOV     ABORT, #31           ; ABORT : =31
CLR     F0                            ; CLEAR HOLD FLAG
SJMP     S133                      ; END
S131 :      SETB     LT04                ; ELSE, STERIL-FAIL(ON)
MOV     A, ABORT             ; GET ABORT STATE
MOV     STATE, A              ; STATE : = ABORT-3
CLR     F0                          ; CLEAR HOLD FLAG
SJMP     S133                    ; END
S132 :      SETB     F0                    ; ELSE, SET HOLD FLAG
S133 :      NOP                              ; END
LJMP     SEQR                  ; RETURN
;
STATE14 : CLR      C                 ; CLEAR CARRY
MOV    A, ADI1              ; GET TEMP
SUBB   A, #TLOU          ; SUBTRACT MIN. TEMP.
JC       S141                    ; IF TEMP. GE. TMIN
CLR      C                       ; CLEAR CARRY
MOV    A, #TMAX         ; GET MAX. TEMP LEVEL
SUBB   A, ADI1             ; SUBTRACT TEMP.
JC       S141                  ; IF TEMP. LE. TMIN
MOV     STATE, #15     ; STATE : =15
MOV     ABORT, #31    ; ABORT : =31
MOV     MTMO, #TSTR  ; LOAD STERIL. TIMER
CLR     TFL4                   ; RESET TIMEOUT FLAG
CLR     F0                       ; CLEAR HOLD FLAG
```

```
SJMP     S142                ; END
S141 :      SETB     LT04           ; ELSE, STERIL-FAIL(ON)
MOV      A, ABORT        ; GET ABORT STATE
MOV      STATE, A         ; STATE : = ABORT-3
CLR      F0                      ; CLEAR HOLD FLAG
S142 :      NOP                           ; END
LJMP     SEQR              ; RETURN
;
STATE15 : JNB      TFL4, S151       ; IF STERIL. TIME
MOV      STATE, #16          ; STATE : =16
MOV      ABORT, #31         ; ABORT : =31
CLR      CEN0                    ; PRESS. LOOP(OFF)
CLR       CEN2                   ; HUM. LOOP(OFF)
CLR       CEN3                   ; GAS LOOP(OFF)
CLR       CTR0                   ; PRESS. OUTPUT(OFF)
CLR       CTR2                    ; HUM. OUTPUT(OFF)
CLR       CTR3                   ; GAS. OUTPUT(OFF)
CLR       VV06                  ; CLOSE V6
CLR       VV05                  ; CLOSE V5
CLR       VV04                  ; CLOSE V4
MOV  TTM0,#VDLY        ; LOAD TIMEOUT DELAY
CLR    TEL0                     ; RESET TIMEOUT FLAG
SETB  TEN0                     ; ENABLE TIMEOUT ALARM
CLR       F0                        ; CLEAR HOLD FLAG
SJMP     S152                  ; END
S151 :      SETB     F0                    ; ELSE, SET HOLD FLAG
S152 :       NOP                           ; END
LJMP     SEQR                ; RETURN
;
STATE16 : MOV     C, LSC4         ; TEST V4 CLOSED
ANL       C, LSC5               ; TEST V5 CLOSED
ANL       C, LSC6               ; TEST V6 CLOSED
JNC       S161                    ; IF(V4, V5, V6) CLOSED
MOV      STATE, #17        ; STATE : =17
MOV      ABORT, #31       ; ABORT : =31
CLR    TEN0                    ; DISABLE TIMEOUT ALARM
SETB   MVC4                ; SET VC4 MASK
SETB   MV04                ; SET V04 MASK
SETB   MVC5                ; SET VC5 MASK
SETB   MV05                ; SET V05 MASK
SETB   MVC6                ; SET VC6 MASK
SETB   MV06                ; SET V06 MASK
CLR      MVC3                ; CLEAR VC3 MASK
CLR      MV03                ; CLEAR V03 MASK
```

```
SETB    VV03                ; OPEN V3
CLR     LT15                  ; STERIL-IN-PROGRESS(OFF)
SETB    LT16                ; PURGE-IN-PROGRESS(ON)
MOV  TTM0,#VDLY      ; LOAD TIMEOUT DELAY
CLR      TEL0                ; RESET TIMEOUT FLAG
SETB   TEN0                ; ENABLE, TIMEOUT ALARM
CLR      F0                    ; CLEAR HOLD FLAG
SJMP     S162                ; END
S161 :      SETB     F0                ; ELSE, SET HOLD FLAG
S162 :      NOP                          ; END
LJMP     SEQR              ; RETURN
;
STATE17 : MOV     C, LS03        ; TEST V3 OPEN
ANL      C, LS08             ; AND V8 OPEN
JNC      S171                  ; IF(V3, AND.V8) OPEN
MOV     STATE, #18        ; STATE : =18
MOV     ABORT, #31       ; ABORT : =31
CLR    TEN0                   ; DISABLE TIMEOUT ALARM
SETB   MVC3                ; SET VC3 MASK
SETB   MV03                ; SET V03 MASK
SETB   MVC8                ; SET VC8 MASK
SETB   MV08                ; SET V08 MASK
MOV  MTM0, #TEVC   ; LOAD EVAC. TIMER
CLR    TFL4                  ; RESET TIMEOUT FLAG
CLR      F0                    ; CLEAR HOLD FLAG
SJMP     S172                ; END
S171 :      SETB     F0                ; ELSE, SET HOLD FLAG
S172 :      NOP                          ; END
LJMP     SEQR              ; RETURN
;
STATE18 : JNB      TFL4, S181       ; IF EVAC. TIME
MOV     STATE, #19          ; STATE : =19
MOV     ABORT, #31         ; ABORT : =31
CLR      MVC3                ; CLEAR VC3 MASK
CLR      MV03                ; CLEAR V03 MASK
CLR      VV03                ; CLOSE V3
CLR      MVC8                ; CLEAR VC8 MASK
CLR      MV08                ; CLEAR V08 MASK
CLR      VV08                ; CLOSE V8
MOV  TTM0,#VDLY        ; LOAD VALVE TIMER
CLR    TEL0                  ; RESET TIMEOUT FLAG
SETB  TEN0                  ; ENABLE TIMEOUT ALARM
CLR      F0                      ; CLEAR HOLD FLAG
SJMP     S182                ; END
```

```
S181 :      SETB     F0                  ; ELSE, SET HOLD FLAG
S182 :       NOP                          ; END
LJMP     SEQR               ; RETURN;
;
STATE19 : MOV     C, LSC3          ; TEST V3 CLOSED-
ANL      C, LSC8               ; AND V8 CLOSED
JNC      S191                   ; IF(V3, AND.V8) CLOSED
MOV     STATE, #20         ; STATE : =20
MOV     ABORT, #32         ; ABORT : =32
CLR    TEN0                     ; DISABLE TIMEOUT ALARM
SETB   MVC8                  ; SET VC8 MASK
SETB   MV08                  ; SET V08 MASK
MOV  STP0, #PSP1          ; GET PRESS. SETPOINT
CLR    MV05                   ; CLEAR V05 MASK
CLR    MVC5                   ; CLEAR VC5 MASK
SETB  CEN0                    ; ENABLE PRESS. CONTROL(V5)
MOV  MTM0,#PN2T        ; LOAD N2 PRESS. TIMER
CLR    TEL4                   ; RESET TIMEOUT FLAG
CLR      F0                     ; CLEAR HOLD FLAG
SJMP     S192                   ; END
S191 :      SETB     F0                  ; ELSE, SET HOLD FLAG
S192 :      NOP                          ; END
;
STATE20 : JNB     TFL4, S202       ; IF PEPRESS. TIME
CLR      C                       ; CLEAR CARRY
MOV    A, #PMAX           ; GET MIN. PRESS. LEVEL
SUBB   A, AD10                ; SUBTRACT PRESSURE
JC       S201                     ; IF PRESS. GE. PMAX
MOV     STATE, #21         ; STATE : =21
MOV     ABORT, #32         ; ABORT : =32
CLR     CEN0                    ; N2 LOOP(OFF)
CLR     CTR0                    ; N2 OUTPUT(OFF)
CLR     VV05                   ; CLOSE N2 VALVE
MOV  TTM0,#VDLY          ; LOAD VALVE TIMEOUT
CLR     TFL0                     ; RESET TIMEOUT FLAG
SETB   TEN0                    ; ENABLE TIMEOUT ALARM
CLR     F0                       ; CLEAR HOLD FLAG
SJMP     S202                   ; END
S201 :      SETB     LT05               ; ELSE, PURGE-FAIL(ON)
MOV     A, ABORT           ; GET ABORT STATE
MOV     STATE, A            ; STATE : = ABORT-4
CLR     F0                       ; CLEAR HOLD FLAG
SJMP     S203                   ; END
S202 :      SETB     F0                  ; ELSE, SET HOLD TIMER
```

```
S203 :      NOP                              ; END
LJMP     SEQR                 ; RETURN
;
STATE21 : JNB      LSC5, S211       ; IF V5 CLOSED
MOV     STATE, #22           ; STATE : =22
MOV     ABORT, #33           ; ABORT : =33
CLR      TEN0                        ; DISABLE TIMEOUT ALARM
SETB    MVC5                       ; SET VC5 MASK
SETB    MV05                       ; SET V05 MASK
CLR      MVC3                       ; CLEAR VC3 MASK
CLR      MV03                       ; CLEAR V03 MASK
SETB    VV03                       ; OPEN V3
CLR      MVC8                        ; CLEAR VC8 MASK
CLR      MV08                        ; CLEAR V08 MASK
SETB    VV08                        ; OPEN V8
MOV  TTM0,#VDLY             ; LOAD TIMEOUT DELAY
CLR    TEL0                          ; RESET TIMEOUT FLAG
SETB  TEN0                          ; ENABLE TIMEOUT ALARM
CLR      F0                             ; CLEAR HOLD FLAG
SJMP     S212                         ; END
S211 :      SETB     F0                        ; ELSE, SET HOLD FLAG
S212 :       NOP                               ; END
LJMP     SEQR                      ; RETURN
;
STATE22 : MOV     C, LSC3               ; TEST V3 OPEN-
ANL      C, LSC8                     ; AND V8 CLOSE
JNC      S221                           ; IF(V3, AND.V8) CLOSED
MOV     STATE, #23              ; STATE : =23
MOV ABORT, #33                  ; ABORT : =33
CLR    TEN0                           ; DISABLE TIMEOUT ALARM
SETB   MVC3                        ; SET VC3 MASK
SETB   MV03                        ; SET VC3 MASK
SETB   MVC8                        ; SET VC8 MASK
SETB   MV08                        ; SET V08 MASK
MOV  MTM0,#DSRB            ; LOAD DESORB TIMER
CLR    TEL4                          ; RESET TIMEOUT FLAG
CLR      F0                             ; CLEAR HOLD FLAG
SJMP     S222                         ; END
S221 :      SETB     F0                        ; ELSE, SET HOLD FLAG
S222 :      NOP                                ; END
LJMP     SEQR                      ; RETURN
;
STATE23 : JNB      TEL4, S231        ; IF DESORB TIME
MOV     STATE, #24           ; STATE : =24
```

```
MOV     ABORT, #34          ; ABORT : =34
CLR     MVC3                ; CLEAR VC3 MASK
CLR     MV03                ; CLEAR VC3 MASK
CLR       VV03               ; CLEAR VC3
CLR       MVC8               ; CLEAR VC8 MASK
CLR       MV08               ; CLEAR V08 MASK
CLR       VV08               ; CLEAR VC8
CLR       MVC2               ; CLEAR VC2 MASK
CLR       MV02              ; CLEAR V02 MASK
CLR       VVC2               ; CLEAR V2
MOV  TTM0,#VDLY          ; LOAD TIMEOUT DELAY
CLR    TEL0                ; RESET TIMEOUT FLAG
SETB  TEN0                ; ENABLE TIMEOUT ALARM
CLR       F0                  ; CLEAR HOLD FLAG
SJMP      S232               ; END
S231 :      SETB     F0                ; ELSE, SET HOLD FLAG
S232 :       NOP                        ; END
LJMP      SEQR              ; RETURN
;
STATE24 : MOV     C, LSC3         ; TEST V3 CLOSED-
ANL       C, LSC8           ; AND V8 CLOSED-
ANL       C, LSC2           ; AND V2 CLOSED
JNC       S241              ; IF(V2, V3, V8) CLOSED
MOV     STATE, #25         ; STATE : =25
MOV     ABORT, #35        ; ABORT : =35
CLR     TEN0                ; DISABLE TIMEOUT ALARM
SETB    MVC3               ; SET VC3 MASK
SETB    MV03               ; SET V03 MASK
SETB    MVC8               ; SET VC8 MASK
SETB    MV08               ; SET V08 MASK
SETB    MVC2               ; SET VC2 MASK
SETB    MV02               ; SET V02 MASK
SETB    LGG1               ; SWITCH TO HIGH GAIN
MOV  MTM0, #TLGH       ; START LOU GAS HILD
LCALL  DCT0               ; DECREMENT PURGE COUNT
CLR     TFL4                ; RESET TIMEOUT FLAG
CLR       F0                  ; CLEAR HOLD FLAG
SJMP      S242               ; END
S241 :      SETB     F0                ; ELSE, SET HOLD FLAG
S242 :      NOP                        ; END
LJMP      SEQR              ; RETURN
;
STATE25 : JNB     TFL4, S252      ; IF LOU-HOLD. TIME
CLR       C                   ; CLEAR CARRY
```

```
MOV    A, #CMIN             ; GET MAX. LEVEL
SUBB   A, ADI3                ; SUBTRACT CONC
ORL  C./TFL6                  ; OR CARRY UITH COUNT FLAG
JC    S251                        ; IF (CONC. LE. CMIN). AND. TFL6=1
MOV     STATE, #26          ; STATE : =26
MOV     ABORT, #36         ; ABORT : =36
CLR     CEN1                    ; DISABLE TEMP. CTRL
CLR     PP01                    ; TURN PUMP OFF
CLR     HT01                    ; TURN HEATER OFF
CLR     LGG1                    ; SET LOW GAIN
CLR     F0                        ; CLEAR HOLD FLAG
SJMP     S253                   ; END
S251 :      MOV     A, ABORT       ; ELSE, GET ABORT STATE
MOV     STATE, A           ; STATE : =35
CLR      F0                      ; CLEAR HOLD FLAG
SJMP     S253                 ; END
S252 :      SETB     F0                ; ELSE, SET HOLD FLAG
S253 :      NOP                           ; END
LJMP     SEQR               ; RETURN
;
STATE26 : LCALL GTCT            ; DECREMENT  &  GET CYCLE CNT
JNZ      S261                   ; IF LAST RUN
MOV     STATE, #27       ; STATE : =27
MOV     ABORT, #36      ; ABORT : =36
CLR    MVC2                  ; CLEAR VC2 MASK
CLR    MV02                  ; CLEAR VC2 MASK
SETB   VV02                 ; OPEN V2
CLR      MVC3                ; CLEAR VC3 MASK
CLR      MOV3                ; CLEAR V03 MASK
SETB   VV03                 ; OPEN V3
CLR      MVC4                ; CLEAR VC4 MASK
CLR      MV04                ; CLEAR V04 MASK
SETB      MV04              ; OPEN V4
CLR      MVC8                ; CLEAR VC8 MASK
CLR      MV08                ; CLEAR V08 MASK
SETB      VV08             ; CLEAR V8
MOV  MTM0, #TOMP    ; LOAD DUMP TIMER
CLR    TFL4                   ; RESET TIMEOUT FLAG
CLR      F0                     ; CLEAR HOLD FLAG
SJMP     S262                ; END
S261 :      MOV     STATE, #28         ; ELSE, STATE : =28
MOV     ABORT, #36           ; ABORT : =36
CLR     F0                           ; CLEAR HOLD FLAG
S262 :      NOP                                ; END
```

```
LJMP     SEQR                    ; RETURN
;
STATE27 : JNB      TEL4, S271       ; IF DUMP-TIME
MOV     STATE, #28          ; STATE : =28
MOV     ABORT, #36         ; ABORT : =36
CLR    VV02                       ; CLOSE V2
CLR    VV03                       ; CLOSE V3
CLR    VV04                       ; CLOSE V4
CLR    VV08                       ; CLOSE V8
CLR      F0                         ; CLEAR HOLD FLAG
SJMP     S272                      ; END
S271 :     SETB     F0                   ; ELSE, SET HOLD FLAG
S272 :     NOP                             ; END
LJMP     SEQR                   ; RETURN
;
STATE28 : MOV     STATE, #37         ; STATE : =37
MOV     ABORT, #36             ; ABORT : =36
CLR    MV05                        ; CLEAR V05 MASK
CLR    MVC5                        ; CLEAR VC5 MASK
SETB   CEN0                       ; PRESS. CONTROL(ON)
MOV  STP0, #PATM             ; SET ATM. SETPOINT
CLR      F0                            ; CLEAR HOLD FLAG
LJMP     SEQR                       ; RETURN
STATE29 : MOV     C, LSC1              ; TEST V1 CLOSED-
ANL      C, LSC2                   ; AND V2 CLOSED-
ANL      C, LSC3                   ; AND V3 CLOSED-
ANL      C, LSC4                   ; AND V4 CLOSED-
ANL      C, LSC5                    ; AND V5 CLOSED-
ANL      C, LSC6                    ; AND V6 CLOSED-
ANL      C, LSC7                    ; AND V7 OPEN-
ANL      C, LSC8                    ; AND V8 CLOSED-
ANL      C, SUC2                   ; AND SU2 CLOSED-
JNC      S291                          ; IF RESET
MOV     STATE, #2                ; STATE : =2
MOV     ABORT, #0                ; ABORT : =0
MOV     STATE, #00H            ; RESET STATUS
MOV    CC00, #00H              ; RESET ALARM LIGHTS
MOV    CC01, #01H              ; RESET RUN LIGHTS
CLR     F0                              ; CLEAR HOLD FLAG
SJMP    S292                         ; END
S291 :      MOV    CTRL, #00H           ; ELSE, RESET CONTROLS
MOV    TCEN, #00H             ; RESET ALARMS
MOV   MSK0, #00H              ; RESET CLOSED MASKS
MOV     MSK1, #00H            ; RESET OPEN MASKS
```

```
MOV     MSK2, #00H              ; RESET MISC, MASKS
MOV     CC02, #40H               ; RESET VALVES
MOV     CC03, #00H              ; RESET MISC. OUTPUTS
MOV    CC01, #00H               ; TURN CYCLE LIGHTS OFF
SETB   LT02                           ; EVAC-FAIL(ON)
SETB      F0                            ; SET HOLD FLAG
S292 :      NOP                                    ; END
LJMP     SEQR                         ; RETURN
;
STATE30 : MOV     C, LSC1              ; TEST V1 CLOSED-
ANL      C, LSC2                     ; AND V2 CLOSED-
ANL      C, LSC3                     ; AND V3 CLOSED-
ANL      C, LSC4                     ; AND V4 CLOSED-
ANL      C, LSC5                     ; AND V5 CLOSED-
ANL      C, LSC6                     ; AND V6 CLOSED-
ANL      C, LSC7                     ; AND V7 OPEN-
ANL      C, LSC8                      ; AND V8 CLOSED-
ANL      C, SUC2                     ; AND SU2 PUSHED-
JNC      S301                            ; IF RESET
MOV     STATE, #2               ; STATE : =2
MOV     ABORT, #0               ; ABORT : =0
MOV     STAT, #00H             ; RESET STATUS
MOV     CC00, #00H             ; RESET RUN LIGHTS
MOV     CC01, #01H              ; RESET RUN LIGHTS
CLR      F0                              ; CLEAR HOLD FLAG
SJMP     S302                          ; END
S301 :      MOV   CTRL, #00H             ; ELSE, RESET CONTROLS
MOV   TCEN, #00H                ; RESET ALARMS
MOV    MSK0, #00H               ; RESET CLOSED MASKS
MOV     MSK1, #00H              ; RESET OPEN MASKS
MOV     MSK2, #00H              ; RESET MISC, MASKS
MOV     CC02, #40H               ; RESET VALVES
MOV     CC03, #00H              ; RESET MISC. OUTPUTS
MOV    CC01, #00H               ; TURN CYCLE LIGHTS OFF
SETB   MLT03                          ; FILL-FAIL(ON)
SETB      F0                            ; SET HOLD FLAG
S302 :      NOP                                    ; END
LJMP     SEQR                         ; RETURN
;
STATE31 : MOV     C, LSC1               ; TEST V1 CLOSED-
ANL      C, LS02                     ; AND V2 OPEN-
ANL      C, LSC3                      ; AND V3 CLOSED-
ANL      C, LSC4                     ; AND V4 CLOSED-
ANL      C, LSC6                     ; AND V6 CLOSED-
```

```
ANL      C, LSC7                    ; AND V7 OPEN-
ANL      C, LSC8                    ; AND V8 CLOSED-
ANL      C, SUC2                    ; AND SU2 PUSHED-
JNC      S311                       ; IF RESET
MOV     STATE, #20                ; STATE : =20
MOV     ABORT, #31                ; ABORT : =32
MOV     STAT, #00H                ; RESET STATUS
MOV   MSK0, #0EFH                 ; SET ALL CLOSED MASKS
MOV   MSK1, #0EFH                 ; SET ALL OPEN MASKS
MOV     MSK2, #001H             ; SET MISC. MASKS
MOV    CC00, #00H                 ; RESET ALARM LIGHTS
MOV     CC01, #22H                  ; RESET RUN LIGHTS
CLR      F0                              ; CLEAR HOLD FLAG
SJMP     S312                          ; END
S311 :      MOV     STRL, #03H            ; ELSE, RESET CONTROLS
MOV     TCEN, #00H                ; RESET ALARMS
MOV     MSK0, #00H                ; RESET CLOSED MASKS
MOV     MSK1, #00H                ; RESET OPEN MASKS
MOV     MSK2, #00H                ; RESET MISC, MASKS
MOV     MSK2, #02H                ; RESET ALL VALVES
MOV     CC03, #01H                ; RESET MISC. OUTPUTS
SETB   LT04                          ; STERIL-FAIL(ON)
SETB      F0                           ; SET HOLD FLAG
S312 :       NOP                                  ; END
LJMP     SEQR                        ; RETURN
;
STATE32 : MOV     C, LSC5                ; TEST V5 CLOSE-
ANL      C, SUC2                      ; AND SU2
JNC      S321                           ; IF(V5 CLOSED  &  SU2 PUSHED)
MOV     STATE, #19                  ; STATE : =19
MOV     ABORT, #32                ; ABORT : =32
CLR      F0                               ; CLEAR HOLD FLAG
SJMP     S322                           ; END
S321 :      MOV     CC02, #02H            ; ELSE, RESET ALL VALVES
SETB     F0                              ; SET HOLD FLAG
S322 :      NOP                                    ; END
LJMP     SEQR                          ; RETURN
;
STATE33 : MOV     C, SUC2                ; TEST SU2
JNC      S331                           ; IF PUSHED
MOV     STATE, #23                ; STATE : =23
MOV     ABORT, #33                ; ABORT : =34
CLR      F0                              ; CLEAR HOLD FLAG
SJMP     S332                          ; END
```

```
S331 :      SETB     F0                          ; ELSE,  SET HOLD FLAG
S332 :      NOP                                    ; END
LJMP     SEQR                          ; RETURN
;
STATE34 : MOV     C, SUC2                ; TEST SU2
JNC      S341                            ; IF PUSHED
MOV     STATE, #25                ; STATE : =25
MOV     ABORT, #35               ; ABORT : =35
LCALL    DCT0                        ; DECREMENT PURGE COUNT
CLR      F0                              ; CLEAR HOLD FLAG
SJMP     S342                          ; END
S341 :      STEB     F0                          ; ELSE,  SET HOLD FLAG
S342 :      NOP                                    ; END
LJMP     SEQR                          ; RETURN
;
STATE35 : MOV     C, LSC1               ; TEST V1 CLOSED-
ANL      C, LSO2                       ; AND V2 OPEN-
ANL      C, LSC3                       ; AND V3 CLOSED-
ANL      C, LSC4                       ; AND V4 CLOSED-
ANL      C, LSC5                       ; AND V5 CLOSED-
ANL      C, LSC6                       ; AND V6 CLOSED-
ANL      C, LSC7                       ; AND V7 CLOSED-
ANL      C, LSC8                       ; AND V8 CLOSED-
JNC      S351                            ; IF RESET
MOV     STATE, #20               ; STATE : =20
MOV     ABORT, #32              ; ABORT : =32
MOV     STATE, #00H           ; RESET STATUS
MOV   MSK0, #0EFH             ; SET ALL CLOSED MASKS
MOV   MSK1, #0EFH             ; SET ALL OPEN MASKS
MOV   MSK2, #001H             ; SET MISC. MASKS
MOV   CC00, #00H               ; RESET ALARM LIGHTS
MOV   CC01, #22H                ; RESET RUN LIGHTS
MOV   STP0, #PSP1               ; LOAD PRESS. SETPOINT
SETB  CEN0                          ; ENABLE PRESSURE CONTROL
MOV   MTM0, #PN2T            ; SET PRESSURE TIMER
CLR     TFL4                          ; CLEAR TIMER FLAG
CLR      F0                             ; CLEAR HOLD FLAG
SJMP     S312                          ; END
S351 :      MOV     CTRL, #03H           ; ELSE, RESET CONTROLS
MOV     TCEN, #00H              ; RESET ALARMS
MOV     MSK0, #00H              ; RESET CLOSED MASKS
MOV     MSK1, #00H              ; RESET OPEN MASKS
MOV     MSK2, #00H              ; RESET MISC. MASKS
MOV     CC02, #02H              ; RESET ALL VALVES
```

```
MOV     CC03, #01H              ; RESET MISC. OUTPUTS
SETB      F0                            ; SET HOLD FLAG
S352 :      NOP                                   ; END
LJMP     SEQR                         ; RETURN
;
STATE36 : MOV     C, SUC2               ; TEST SU2
JNC      S361                           ; IF PUSHED
MOV     STATE, #26               ; STATE : =26
MOV     ABORT, #37              ; ABORT : =37
CLR      F0                             ; CLEAR HOLD FLAG
SJMP     S362                          ; END
S361 :      SETB     F0                        ; ELSE,  SET HOLD FLAG
S362 :      NOP                                   ; END
LJMP     SEQR                        ; RETURN
;
STATE37 : CLR      C                          ; CLEAR CARRY
MOV    A, #PATM                 ; GET ATM SETPOINT
SUBB   A, BD10                    ; SUBTRACT PRESSURE
JC       S371                           ; IF PRESS. GE. ATM
MOV     STATE, #38              ; STATE : =38
MOV     ABORT, #0               ; ABORT : =0
CLR     MVC7                        ; CLEAR VC7 MASK
CLR     MV07                        ; CLEAR V07 MASK
CLR     CEN0                         ; N2 LOOF(OFF)
CLR     CTR0                         ; N2 OUTP(OFF)
CLR     VV05                        ; CLOSE N2 VALVE
SETB    VV07                       ; OPEN V7
CLR     LT16                          ; PURGE-IN-PROGRESS(OFF)
SETB   LT17                          ; REMOVE-LOAD(ON)
CLR     F0                             ; CLEAR HOLD FLAG
SJMP     S372                        ; END
S371 :      SETB     LT05                   ; ELSE, SET HOLD FLAG
S372 :      NOP                                   ; END
LJMP     SEQR                       ; RETURN
;
STATE38 : JNB     SUC2, S381           ; IF SU2 PUSHED
MOV     STATE, #0               ; STATE : =0(RESET)
MOV     ABORT, #0              ; ABORT : =0
CLR      F0                             ; CLEAR HOLD FLAG
SJMP     S382                          ; END
S381 :       STEB     F0                       ; ELSE,  SET HOLD FLAG
S382 :      NOP                                   ; END
LJMP     SEQR                       ; RETURN
;
```

이상에서는, 본 발명을 특정 실시예와 관련하여 설명하였지만, 본 발명의 청구범위에서 벗어나지 않는

다양한 변경예 및 개조예가 가능함은 명백할 것이다. 예를 들어, 당 분야에서 통상의 지식을 가진자라면, 본 명세서에 기술된 장치가 유독성 가스, 예컨대 표백용 가스, 훈증 소독기, 살균제등을 이용하는 시스템과 같은 각종 처리 시스템과 관련하여 사용하는데 적합하다는 것을 알 수 있을 것이다. 따라서, 본 명세서 및 첨부 도면은 본 발명을 제한하는 것이 아니라 예시의 목적으로 게재한 것이다.

(57) 청구의 범위

청구항 1

물품을 가스로 처리하기 위한 장치로서, 처리하고자 하는 물품을 수용하기 위한 처리실 수단(10) ; 가스를 상기 처리실 수단에 공급하기 위해 처리실 수단에 결합된 밸브 수단(20), 소정의 시간 간격후 처리실 수단으로부터 가스를 제거하기 위한 수단(20, P1, P2), 상기 밸브 수단 및 수단을 제어하기 위하여 상기 처리실 수단으로부터 측정된 파라미터들과 관련된 복수의 전기신호를 수신하는 전자 제어 수단(100)을 포함하여 가스(22)를 처리실 수단에 공급하기 위한 수단(20)을 포함하되, 상기 전자 제어 수단은 소정의 명령 시퀀스에 따라서 복수의 상태를 통해 상기 장치를 순환시키기 위한 컴퓨터 수단(102)를 포함하며, 상기 컴퓨터 수단은 상기 장치의 고장에 응답하여 복수의 정의된 고장 상태에 대하여 상기 장치의 동작 어보트(abort)시키기 위한 수단을 포함하는 장치.

청구항 2

제 2 항에 있어서, 상기 가스가 상기 물품을 상기 가스에 의해 살균시키기 위한 살균 가스인 장치.

청구항 3

제 2 항에 있어서, 상기 살균 가스가 2종 이상의 성분으로부터 발생되며, 가스의 제 1 성분을 수용하기 위한 제 1 수단, 가스의 제 2 성분을 수용하기 위한 제 2 수단, 상기 제 1 및 제 2 성분을 상기 살균 가스를 발생하도록 서로 반응시키기 위한 수단을 추가로 포함하며, 상기 제1 성분과 제2 성분을 반응시키기 위한 수단은 상기 복수의 측정된 파라미터중 선택된 파라미터의 크기에 응답하여 상기 컴퓨터 수단에 의해 제어되는 장치.

청구항 4

제 3 항에 있어서, 상기 비교적 안정한 가스를 상기 처리실 수단에 공급하기 위한 밸브 수단(V5)를 추가로 포함하는 장치.

청구항 5

제 3 항에 있어서, 상기 처리실 수단에 여과된 공기를 공급하기 위한 밸브 수단(V7)을 추가로 포함하는 장치.

청구항 6

제 3 항에 있어서, 상기 처리실내의 습도에 영향을 주도록 상기 처리실 수단에 수증기를 공급하기 위한 밸브 수단(V6)을 추가로 포함하는 장치.

청구항 7

제 3 항에 있어서, 상기 복수의 측정된 파라미터가 상기 처리실 수단내의 온도, 압력, 습도 및 상기 처리실 수단내의 상기 살균 가스의 농도를 포함하는 장치.

청구항 8

제 2 항에 있어서, 상기 살균 가스가 이산화 염소를 포함하는 장치.

청구항 9

제 3 항에 있어서, 상기 살균 가스가 이산화 염소를 포함하고, 상기 제 1 성분은 염소 가스를 포함하며, 상기 제 2 성분은 아염소산 나트륨을 포함하는 장치.

청구항 10

제 2 항에 있어서, 상기 제거용 수단이 진공 펌프 수단(P1, P2) 및 추가의 밸브 수단(V9)를 포함하는 장치.

청구항 11

제 2 항에 있어서, 상기 밸브 수단(20)이 제1 및 제 2 스위치 수단(LSo, LSc)을 구비하며, 상기 제 1 스위치 수단(LSo)은 상기 밸브 수단이 개방되는 시기를 나타내고 상기 제 2 스위치 수단(LSo)은 상기 밸브 수단이 폐쇄되는 시기를 나타내며, 상기 제 1 및 제 2 스위치 수단은 상기 제 1 스위치 수단이 폐쇄될 때 상기 제 2 스위치 수단이 개방되도록 반대의 상태로 존재하는 장치.

청구항 12

제 2 항에 있어서, 상기 전자 제어 수단은 메모리 수단(106, 104)을 포함하고, 상기 밸브 수단의 폐쇄 또는 개방 상태를 나타내는 상기 밸브 수단으로부터의 입력 신호를 수신하기 위한 수단(8)과 상기 밸브 수단을 선택적으로 개방 또는 폐쇄시키도록 출력 신호를 상기 밸브 수단에 전송하기 위한 수단(7)을 추가로 포함하며, 상기 입력 및 출력 신호의 영상 신호가 상기 메모리 수단에 기억되는 장치.

청구항 13

제 12 항에 있어서, 상기 컴퓨터 수단에 의해 인에이블(enabling)신호가 발생되는 시기외에는 상기 출력 신호가 상기 밸브 수단으로 전송되는 것을 디스에이블링(disabling)시키기 위한 수단을 추가로 포함하는 장치.

청구항 14

제 2 항에 있어서, 상기 컴퓨터 수단의 적절한 동작을 모니터하기 위한 수단을 추가로 포함하며, 상기 모니터 수단은 상기 컴퓨터 수단의 고장시에 상기 밸브 수단의 작동을 방지하도록 디스에이블링 신호를 발송하는 장치.

청구항 15

제 12 항에 있어서, 상기 메모리 수단(106, 104)에 기억된 마스크 수단(124)를 추가로 포함하며, 상기 컴퓨터 수단은 상기 입력 및 출력 신호의 상기 영상 신호들을 비교하여 상기 입력 영상 신호와 출력 영상 신호가 일치하지 않는 경우 상기 마스크내의 한 비트의 세팅에 응답하여 경보 신호를 발생하는 장치.

청구항 16

제 11 항에 있어서, 상기 제 1 및 제 2 스위치 수단의 상태를 모니터하기 위한 수단및 상기 제1 및 제2 스위치 수단이 적절한 상태로 존재하지 않을 경우 경보 신호를 발생하기 위한 수단을 추가로 포함하는 장치.

청구항 17

제 2 항에 있어서, 상기 밸브 수단은 상기 컴퓨터 수단으로부터의 명령에 응답하여 제 1 상태와 제 2 상태 사이에서 이동하며, 상기 밸브 수단이 소정의 시간 간격내에 상기 제 1 상태에서 제 2 상태로 이동하지 않는 경우 경보 신호를 발생하기 위한 타이머 수단(119)를 추가로 포함하는 장치.

청구항 18

제 2 항에 있어서, 일단 상기 정의된 고장상태중 한 상태에 도달되면, 상기 장치를 또 다른 정의된 상태로 순환시키기 위한 수단을 추가로 포함하는 장치.

청구항 19

물품을 가스로 처리하기 위한 장치로서, 제 1 성분을 수용하기 위한 제1 수단 ; 제 2 성분을 수용하기 위한 제 2 수단(상기 제 1 성분과 제 2 성분을 함께 반응할 때 상기 가스를 형성함) ; 상기 가스를 형성시키기 위해 상기 두 성분을 함께 반응시키기 위한 수단(V4) ; 상기 처리실 수단내의 상기 물품을 처리하기 위해 상기 가스를 상기 처리실 수단에 공급하기 위한 제 1 밸브 수단(V2) ; 상기 가스를 상기 처리실 수단으로부터 제어하기 위한 수단(P1, P2) ; 상기 반응용 수단, 공급용 수단 및 제어용 수단을 제어하기 위한 전자 제어기 수단(100)을 포함하되, 상기 제어기 수단은 상기 물품을 상기 가스에 의해 처리한 후에 상기 처리실 수단이 허용가능한 안전도 기준내에 있도록 상기 처리실 수단으로부터 상기 가스를 제거하는 사이클을 정의하는 일련의 연속 상태를 통하여 상기 장치를 순환시키도록 예정된 순서의 단계를 실행하는 컴퓨터 수단을 포함하는 장치.

청구항 20

제 19 항에 있어서, 상기 가스가 물품을 상기 가스에 의해 살균시키기 위한 살균 가스인 장치.

청구항 21

제 20 항에 있어서, 상기 컴퓨터 수단이 상기 반응용 수단, 공급용 수단 및 제거용 수단의 동작을 제어하기 위해 상기 처리실 수단으로부터 측정된 파라미터들과 관련된 복수의 전기 신호를 수신하기 위한 수단(114)를 포함하는 장치.

청구항 22

제 21 항에 있어서, 상기 반응용 수단은 상기 살균 가스를 발생시키도록 상기 제1 및 제 2 성분을 서로 반응시키기 위한 제2 밸브 수단(V4)를 포함하며, 상기 제 2 밸브 수단은 복수의 측정된 파라미터 중 선택된 파라미터의 크기에 응답하여 상기 컴퓨터 수단에 의해 제어되는 장치.

청구항 23

제 22 항에 있어서, 비교적 안정한 가스를 상기 처리실 수단에 공급하기 위한 밸브 수단(V5)를 추가로 포함하는 장치.

청구항 24

제 22 항에 있어서, 여과된 공기를 상기 처리실 수단에 공급하기 위한 밸브 수단(V7)을 추가로 포함하는 장치.

청구항 25

제 22 항에 있어서, 상기 처리실내의 습도에 영향을 주도록 상기 처리실 수단에 수증기를 공급하기 위해 밸브 수단(V6)을 추가로 포함하는 장치.

청구항 26

제 22 항에 있어서, 상기 복수의 측정된 파라미터가 상기 처리실 수단내의 온도, 압력, 습도 및 상기 처리실 수단내의 상기 살균 가스의 농도를 포함하는 장치.

청구항 27

제 20 항에 있어서, 상기 살균 가스가 이산화 염소를 포함하는 장치.

청구항 28

제 27 항에 있어서, 상기 살균 가스가 이산화 염소를 포함하고, 상기 제 1 성분은 염소가스를 포함하며, 상기 제 2 성분은 아염소산 나트륨을 포함하는 장치.

청구항 29

제 20 항에 있어서, 상기 제거 수단이 진공 펌프 수단(P1, P2) 및 추가의 밸브 수단(V1)을 포함하는 장치.

청구항 30

제 20 항에 있어서, 상기 밸브 수단은 제1 및 제2 스위치 수단을 구비하며, 상기 제 1 스위치 수단(LSo)은 상기 밸브 수단이 개방되는 시기를 나타내고 상기 제 2 스위치 수단(LSc)은 상기 밸브 수단이 폐쇄되는 시기를 나타내며, 상기 제 1 및 제 2 스위치 수단은 상기 제 1 스위치 수단이 폐쇄될 때 상기 제 2 스위치 수단이 개방되도록 반대의 상태로 존재하는 장치.

청구항 31

제 20 항에 있어서, 상기 전자 제어 수단이 메모리 수단(106, 104)를 포함하며, 상기 밸브 수단의 폐쇄 또는 개방 조건을 나타내는 상기 밸브 수단으로부터의 입력 신호를 수신하기 위한 수단 및 상기 밸브 수단을 선택적으로 개방 또는 폐쇄시키도록 상기 밸브 수단에 출력 신호를 전송하기 위한 수단을 추가로 포함하고, 상기 입력 및 출력 신호의 영상이 상기 메모리 수단에 기억되는 장치.

청구항 32

제 31 항에 있어서, 상기 컴퓨터 수단에 의해 인에이블링 신호가 발생되는 시기외에는 상기 출력신호가 상기 밸브 수단으로 전송되는 것을 디스에이블링시키기 위한 수단을 추가로 포함하는 장치.

청구항 33

제 20 항에 있어서, 상기 컴퓨터 수단의 적절한 동작을 모니터하기 위한 수단을 추가로 포함하며, 상기 모니터 수단은 상기 컴퓨터 수단의 고장시에 상기 밸브 수단의 작동을 방지하도록 디스에이블링 신호를 발송하는 장치.

청구항 34

제 31 항에 있어서, 상기 메모리 수단에 기억된 마스크 수단(124)를 추가로 포함하며, 상기 컴퓨터 수단은 상기 입력 및 출력 신호의 영상을 비교하여 상기 입력 영상과 출력 영상이 일치하지 않는 경우 상기 마스크 수단내의 한 비트의 세팅에 응답하여 경보 신호를 발생하는 장치.

청구항 35

제 30 항에 있어서, 상기 제1 및 제2 스위치 수단의 상태를 모니터하기 위한 수단 및 상기 제 1 및 제 2 스위치 수단이 적절한 상태로 존재하지 않을 경우 경보 신호를 발생하기 위한 수단을 추가로 포함하는 장치

청구항 36

제 20 항에 있어서, 상기 밸브 수단은 상기 컴퓨터 수단으로부터의 명령에 응답하여 제1 상태와 제2 상태 사이에서 이동하며, 상기 밸브 수단이 소정의 시간 간격내에 상기 제 1 상태로 이동하지 않는 경우 경보 신호를 발생하기 위한 타이머 수단을 추가로 포함하는 장치.

청구항 37

제 20 항에 있어서, 상기 컴퓨터 수단은 상기 장치의 고장에 응답하여 복수의 정의된 고장 상태중 하나에 대하여 상기 장치의 동작을 어보트(abort)시키기 위한 수단을 포함하며, 상기 선택된 고장 상태는 고장이 발생된 상기 사이클내의 상태에 종속되는 장치.

청구항 38

제 37 항에 있어서, 일상 상기 정의된 고장상태중 한 상태에 도달되면, 상기 장치를 또 다른 정의된 상태로 리세트(reset)시키기 위한 수단을 추가로 포함하는 장치.

청구항 39

물품을 가스로 처리하기 위한 장치로서, 처리하고자 하는 물품을 수용하기 위한 처리실 수단(10), 처리실 수단에 가스를 공급하기 위하여 처리실 수단에 결합된 밸브 수단(V2), 소정의 시간 간격후 가스를 처리실 수단으로부터 제거하기 위한 수단(P1, P2), 상기 밸브 수단 및 제거 수단을 제어하기 위하여 상기

처리실 수단으로부터 측정된 파라미터들과 관련된 복수의 전기 신호를 수신하는 전자 제어 수단을 포함하여 상기 처리실 수단에 가스를 공급하기 위한 수단(V2)를 포함하되, 상기 전자 제어 수단은 소정의 명령 시퀀스에 따라서 복수의 상태를 통하여 상기 장치를 순환시키기 위한 컴퓨터 수단을 포함하고, 상기 컴퓨터 수단은 메모리 수단을 포함하며, 상기 밸브 수단의 폐쇄 또는 개방 상태를 나타내는 상기 밸브 수단으로부터의 입력 신호를 수신하기 위한 수단 및 상기 밸브 수단을 선택적으로 개방 또는 폐쇄시키도록 상기 밸브 수단에 출력 신호를 전송하기 위한 수단을 추가로 포함하며, 상기 입력 및 출력신호의 영상 신호가 상기 메모리 수단에 기억되고, 상기 메모리 수단에 마스크 수단이 기억되며, 상기 컴퓨터 수단은 상기 입력 및 출력신호의 영상 신호를 비교하여 상기 입력 신호와 출력 영상 신호가 일치하지 않는 경우 상기 마스크 수단내의 한 비트의 세팅에 응답하여 경보 신호를 발생하는 장치.

청구항 40

제 39 항에 있어서, 상기 가스가 상기 물품을 상기 가스에 의해 살균시키기 위한 살균 가스인 장치.

청구항 41

제 40 항에 있어서, 상기 살균용 가스가 2종의 성분으로부터 발생되며, 제 1 가스 성분을 수용하기 위한 제 1 수단, 제 2 가스 성분을 수용하기 위한 제 2 수단, 상기 가스를 발생시키기 위해 상기 제 1 및 제 2 성분을 서로 반응시키기 위한 수단을 추가로 포함하며, 상기 반응용 수단은 상기 복수의 측정된 파라미터중 선택된 파라미터의 크기에 응답하여 상기 컴퓨터 수단에 의해 제어되는 장치.

청구항 42

제 41 항에 있어서, 비교적 안정한 가스를 상기 처리실 수단에 공급하기 위한 밸브 수단(V5)를 추가로 포함하는 장치.

청구항 43

제 41 항에 있어서, 상기 처리실 수단에 여과된 공기를 공급하기 위한 밸브 수단(V7)을 추가로 포함하는 장치.

청구항 44

제 41 항에 있어서, 상기 처리실내의 습도에 영향을 주도록 상기 처리실 수단에 수증기를 공급하기 위한 밸브 수단(V6)을 추가로 포함하는 장치.

청구항 45

제 41 항에 있어서, 상기 복수의 측정된 파라미터가 상기 처리실 수단내의 온도, 압력, 습도 및 상기 처리실 수단내의 상기 살균 가스의 농도를 포함하는 장치.

청구항 46

제 40 항에 있어서, 상기 살균 가스가 이산화 염소를 포함하는 장치.

청구항 47

제 41 항에 있어서, 상기 살균 가스가 이산화 염소를 포함하고 상기 제 1 성분은 염소 가스를 포함하며 상기 제 2 수단은 아염소산 나트륨을 포함하는 장치.

청구항 48

제 40 항에 있어서, 상기 제거용 수단이 진공펌프 수단(P1, P2) 및 추가의 밸브 수단(V1)을 포함하는 장치.

청구항 49

제 40 항에 있어서, 상기 밸브 수단은 제 1 및 제 2 스위치 수단을 포함하며, 상기 제 1 스위치 수단(LSo)은 상기 밸브 수단이 개방되는 시기를 나타내고 상기 제 2 스위치 수단(LSc)은 상기 밸브 수단이 폐쇄되는 시기를 나탸내며, 상기 제 1 및 제 2 스위치 수단은 상기 제 1 스위치 수단이 폐쇄될 때 상기 제 2 스위치 수단이 개방되도록 반대의 상태로 존재하는 장치.

청구항 50

제 40 항에 있어서, 상기 컴퓨터 수단에 의해 인에이블링 신호가 발생되는 시기외에는 상기 출력 신호가 상기 밸브 수단으로 전송되는 것을 인에이블링시키기 위한 수단을 추가로 포함하는 장치.

청구항 51

제 40 항에 있어서, 상기 컴퓨터 수단의 적절한 동작을 모니터하기 위한 수단을 추가로 포함하고, 상기 모니터 수단은 상기 컴퓨터 수단의 고장시에 상기 밸브 수단의 작동을 방지하도록 디스에이블링 신호를 발송하는 장치.

청구항 52

제 49 항에 있어서, 상기 제1 및 제2 스위치 수단의 상태를 모니터하기 위한 수단 및 상기 제 1 및 제 2 스위치 수단이 적절한 상태로 존재하지 않는 경우 경보신호를 발생하기 위한 수단을 추가로 포함하는 장치.

청구항 53

제 40 항에 있어서, 상기 밸브 수단은 상기 컴퓨터 수단으로부터의 명령에 응답하여 제 1 상태와 제 2 상태 사이에서 이동하며, 상기 밸브 수단이 소정의 시간 간격내에 상기 제 1 상태에서 제 2 상태로 이동하지 않는 경우에 경보신호를 발생하기 위한 수단을 추가로 포함하는 장치.

청구항 54

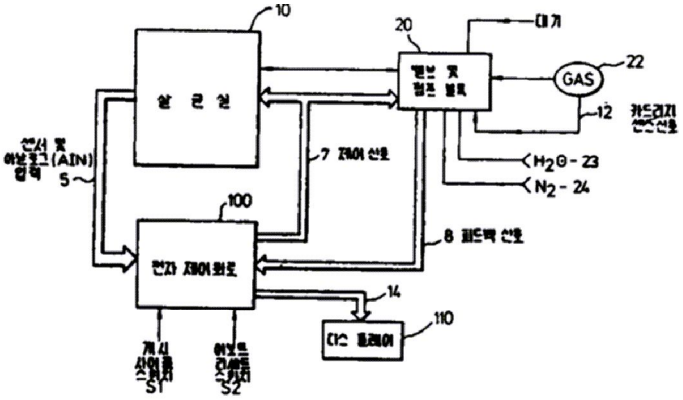
제 40 항에 있어서, 상기 컴퓨터 수단은 상기 장치의 고장에 응답하여 복수의 정의된 고정상태중 하나에 대하여 상기 장치의 동작을 어보트하기 위한 수단을 포함하며, 상기 선택된 고장 상태는 고장이 발생된 상기 사이클내의 상태에 종속되는 장치.

청구항 55

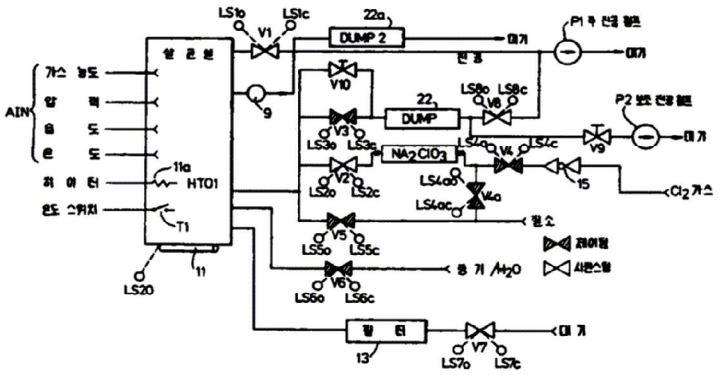
제 39 항에 있어서, 일단 상기 정의된 고장 상태중 한 상태에 도달되면, 상기 장치를 또 다른 정의된 상태로 순환시키기 위한 수단을 추가로 포함하는 장치.

도면

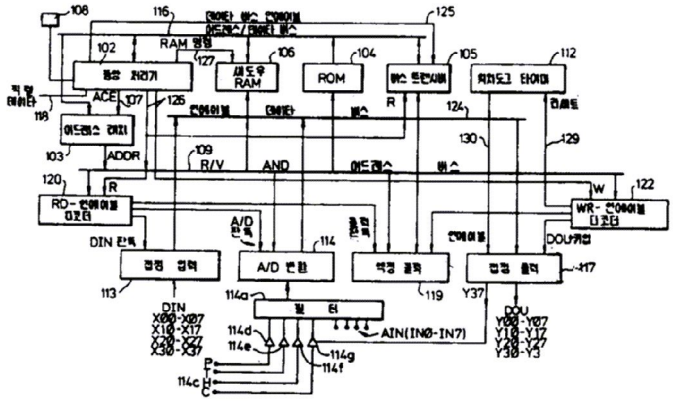
도면1



도면2



도면3



도면3a

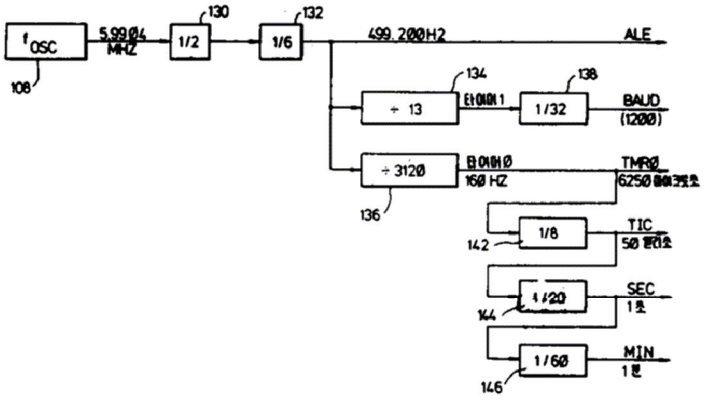
어드레스	설 명	A15 A14 A13 A12
00-FF	내부 RAM	
0000-0FFF	내부 ROM	0 0 0 0
1000-1FFF	외부 ROM	0 0 0 1
2000-203F	외부 SPAM	0 0 1 0
4000-400F	출 력	0 1 0 0
6000-6007	A/D 판 독	0 1 1 0
C000 C001 C002 C003	X00 - X07 X10 - X17 X20 - X27 X30 - X37 } DIN	1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0
E000 E001 E002 E003	Y00 - Y07 Y10 - Y17 Y20 Y27 Y30 Y37 } DOU	1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0
E004	워치 도그 - 리셋트	1 1 1 0

도면3b

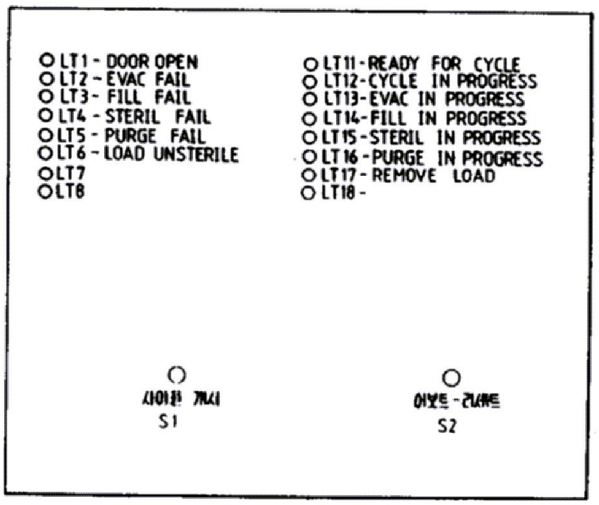
어드레스 버스 비트

A11 A10 A9 A8	A7 A6 A5 A4	A3 A2 A1 A0
———	A7 A6 A5 A4	A3 A2 A1 A0
A11 A10 A9 A8	A7 A6 A5 A4	A3 A2 A1 A0
A11 A10 A9 A8	A7 A6 A5 A4	A3 A2 A1 A0
0 0 0 0	0 0 A5 A4	A3 A2 A1 A0
0 0 0 0	0 0 0 0	A3 A2 A1 A0
0 0 0 0	0 0 0 0	0 A2 A1 A0
0	0	0 0 0 0
0	0	0 0 0 1
0	0	0 0 1 0
0	0	0 0 1 1
0	0	0 0 0 0
0	0	0 0 0 1
0	0	0 0 1 0
0	0	0 0 1 1
0	0	0 1 0 0

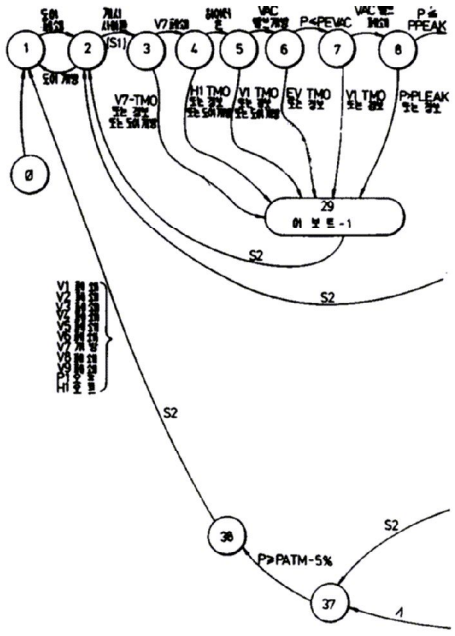
도면4


f OSC
5.9904 MHZ
108
130
1/2
132
1/6
499.200H2
ALE
134
÷ 13
타이머1
138
1/32
BAUD
(1200)
÷ 3120
타이머0
160 HZ
TMR0
136
142
1/8
TIC
144
1/20
SEC
1초
1/60
MIN
1분
146

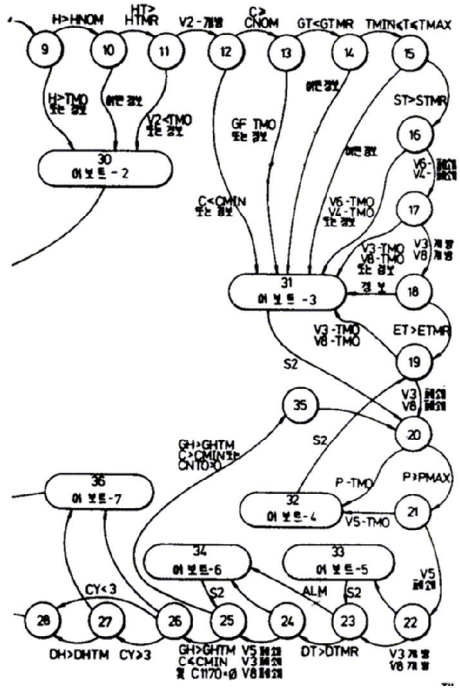
도면5


LT1 - DOOR OPEN
LT2 - EVAC FAIL
LT3 - FILL FAIL
LT4 - STERIL FAIL
LT5 - PURGE FAIL
LT6 - LOAD UNSTERILE
LT7
LT8
LT11 - READY FOR CYCLE
LT12 - CYCLE IN PROGRESS
LT13 - EVAC IN PROGRESS
LT14 - FILL IN PROGRESS
LT15 - STERIL IN PROGRESS
LT16 - PURGE IN PROGRESS
LT17 - REMOVE LOAD
LT18 -
S1
S2

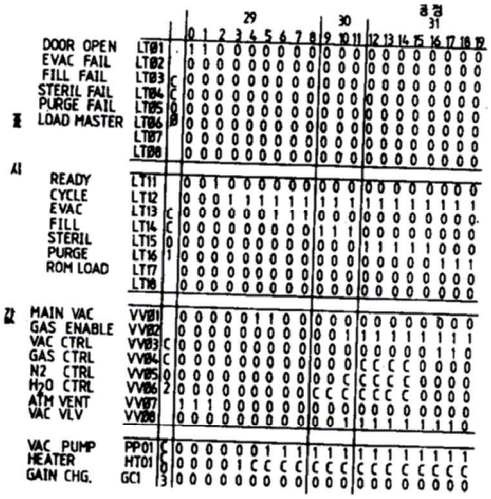
도면6a


1
2
3
4
5
6
7
8
P<PEVAC
P PPEAK
V7-TMO
H1 TMO
V1 TMO
EV TMO
V1 TMO
P>PLEAK
0
29
S2
S2
S2
38
P>PATM-5%
37
1

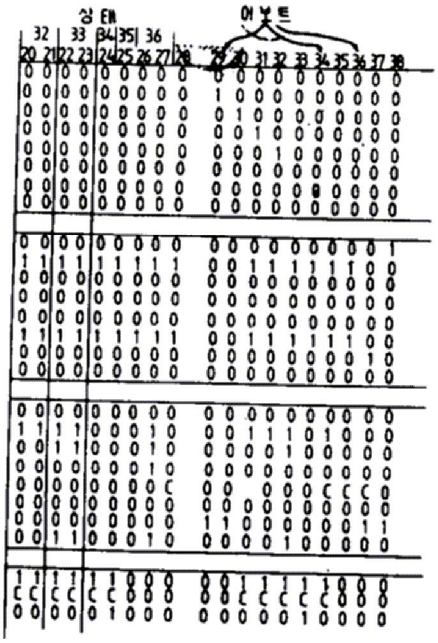
도면6b



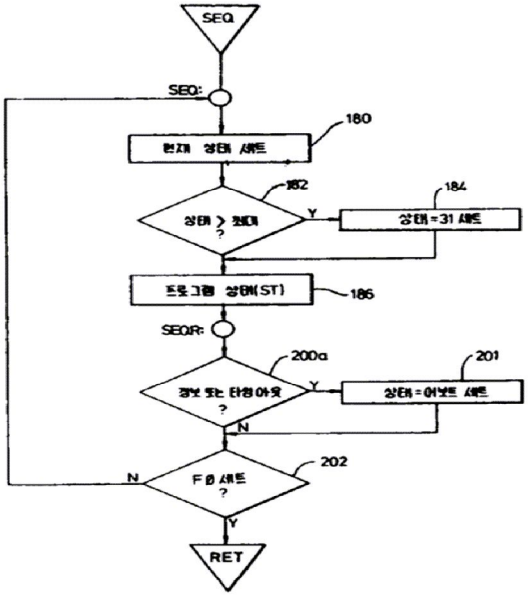
도면7a



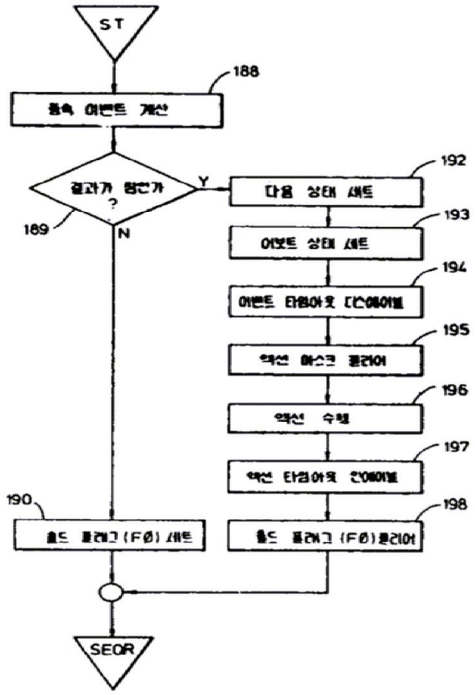
도면7b


상 태
어 보 트

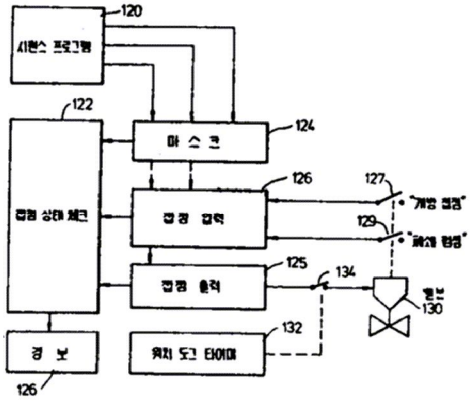
도면7c


SEQ
SEQ:
180
182
184
Y
상태=31 세트
185
SEQR:
200a
201
Y
상태=어보트 세트
N
202
N
Y
RET

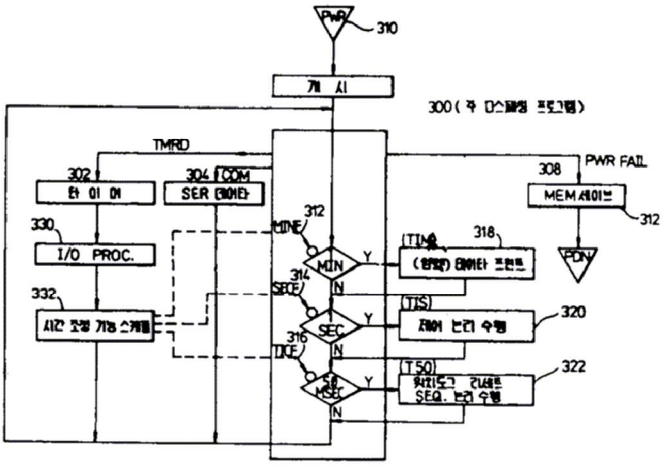
도면7d


ST
188
Y
N
189
192
193
194
195
196
197
198
190
SEQR

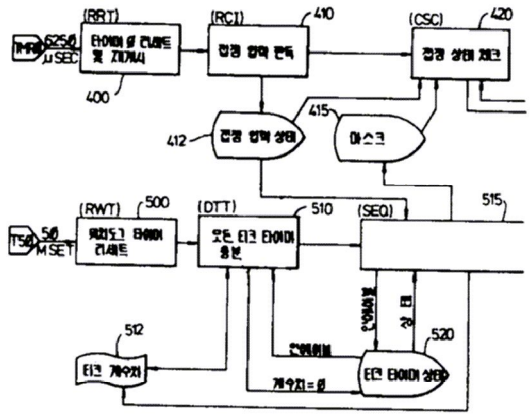
도면8


120
122
124
126
127
129
125
134
130
132
126

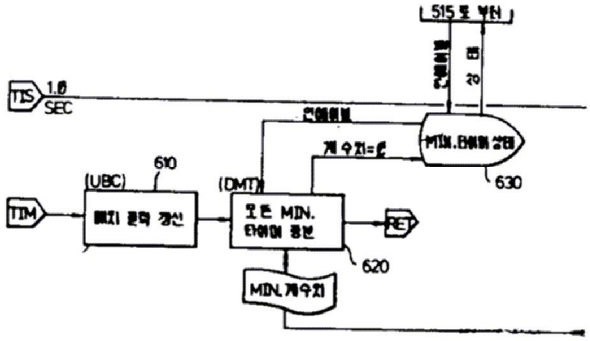
도면9


PWR
310
300
TMRD
302
304
308
PWR FAIL
312
330
I/O PROC.
312
MIN
318
314
SEC
316
320
322

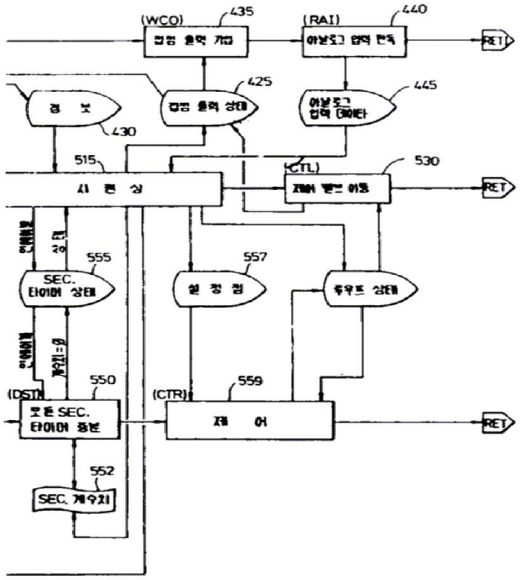
도면10



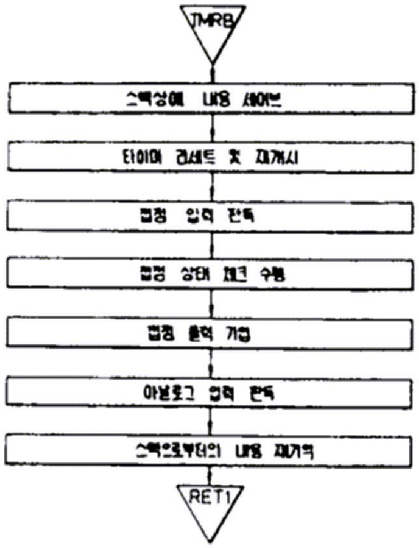
도면10a



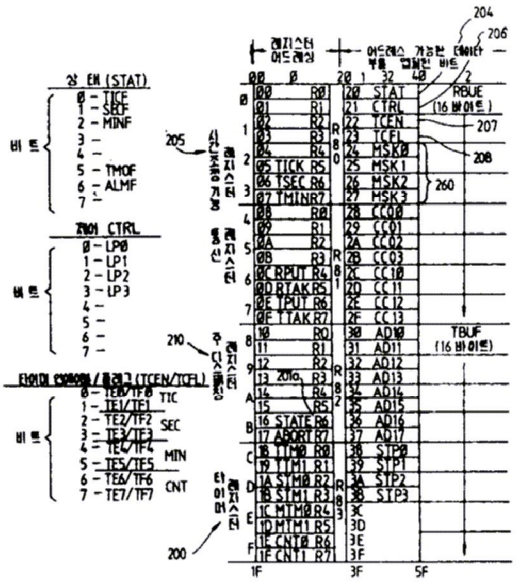
도면10b



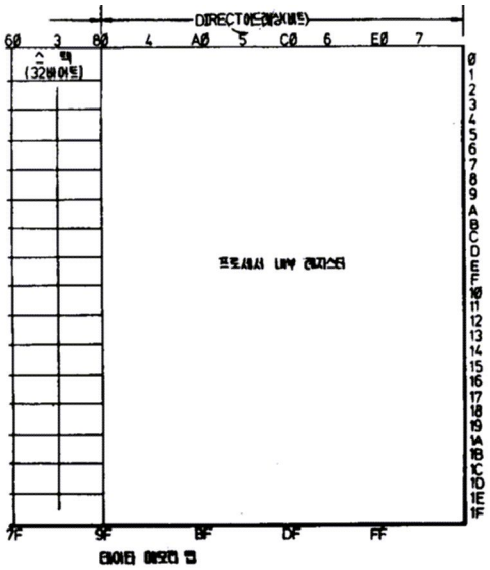
도면11


TMRB
스택상에 내용 POP
타이머 리셋트 및 재개시
펄정 입력 판독
펄정 상태 체크 수행
펄정 출력 개입
아날로그 입력 판독
스택으로부터의 내용 재기억
RET1

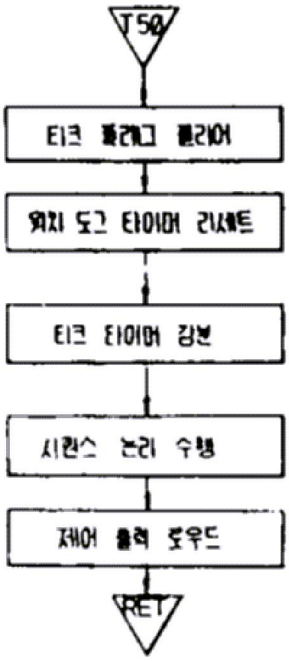
도면12a


204
206
207
208
260
205
210
200
TBUF
(16 바이트)
RBUE
(16 바이트)

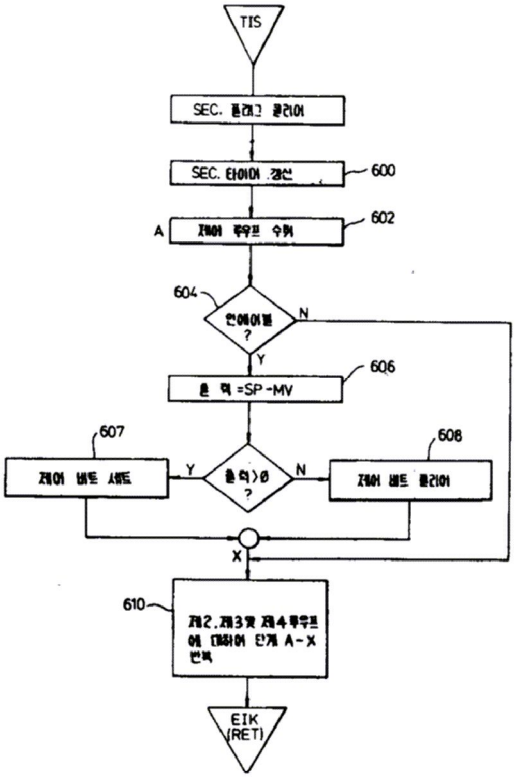
도면12b


DIRECT어드레싱(32바이트)
60 3 80 4 A0 5 C0 6 E0 7
(32바이트)
0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
1F 3F 5F 7F 9F BF DF FF

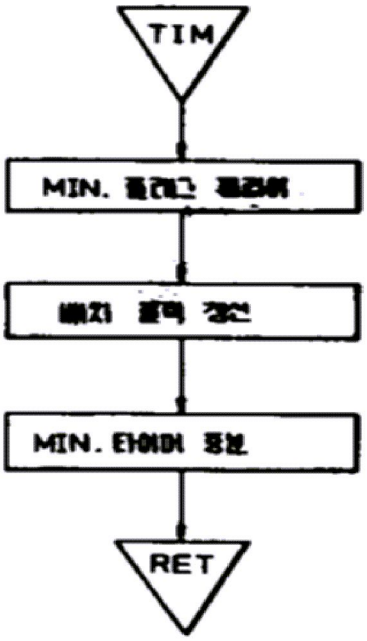
도면13


T50
타크 플래그 클리어
워치 도그 타이머 리셋트
타크 타이머 감분
시퀀스 논리 수행
제어 출력 포우드
RET

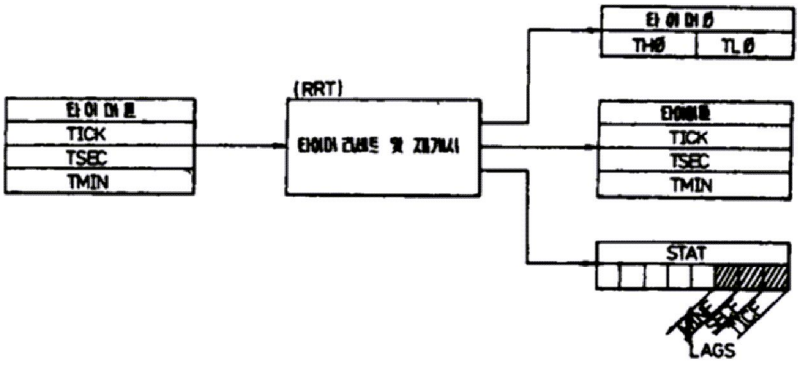
도면14


TIS
SEC. 플래그 클리어
SEC. 타이머 갱신
600
A
제어 루우프 수행
602
604
언에이블 ?
N
Y
606
=SP-MV
607
제어 비트 세트
Y
>0 ?
N
608
제어 비트 클리어
X
610
제2,제3및 제4루우프에 대하여 단계 A~X 반복
EIK (RETI)

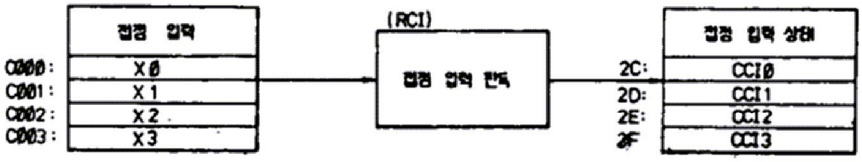
도면15


TIM
MIN. 플래그 클리어
배치 갱신
MIN. 타이머 초기
RET

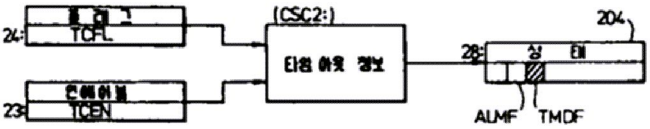
도면16



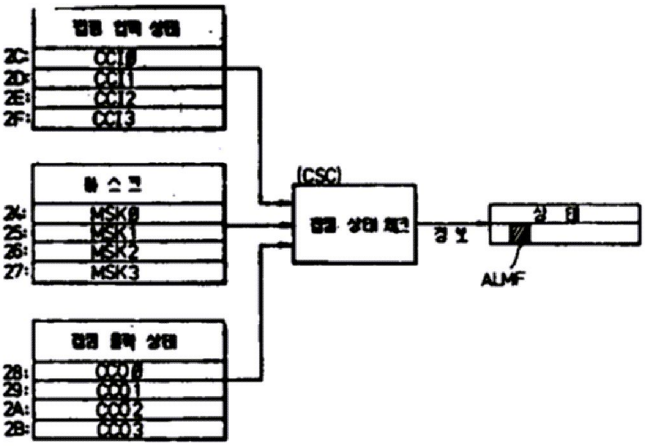
도면17



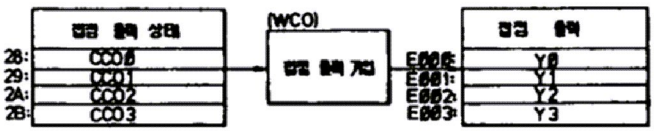
도면18



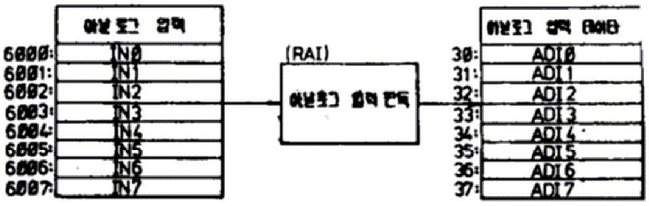
도면19



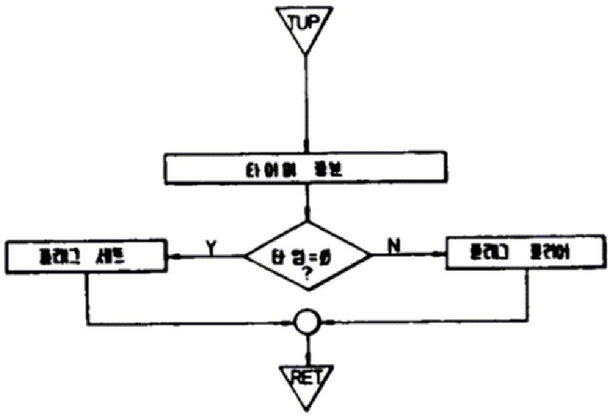
도면20



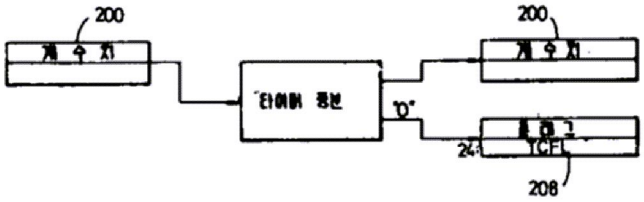
도면21



도면22



도면23



도면24

