



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2018년08월03일  
(11) 등록번호 10-1884925  
(24) 등록일자 2018년07월27일

(51) 국제특허분류(Int. Cl.)  
H04L 25/02 (2006.01)  
(21) 출원번호 10-2013-7025628  
(22) 출원일자(국제) 2012년02월22일  
심사청구일자 2017년02월15일  
(85) 번역문제출일자 2013년09월27일  
(65) 공개번호 10-2014-0014202  
(43) 공개일자 2014년02월05일  
(86) 국제출원번호 PCT/US2012/026010  
(87) 국제공개번호 WO 2012/121881  
국제공개일자 2012년09월13일  
(30) 우선권주장  
13/041,535 2011년03월07일 미국(US)  
(56) 선행기술조사문헌  
US20060123176 A1\*  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
마이크로칩 테크놀로지 인코포레이티드  
미국 85224-6199 아리조나 챠들러 웨스트 챠들러  
블러바드 2355  
(72) 발명자  
마호 조엘  
미국 애리조나 85296, 길버트, 사우스 라 아블레  
타 143  
(74) 대리인  
특허법인세진

전체 청구항 수 : 총 20 항

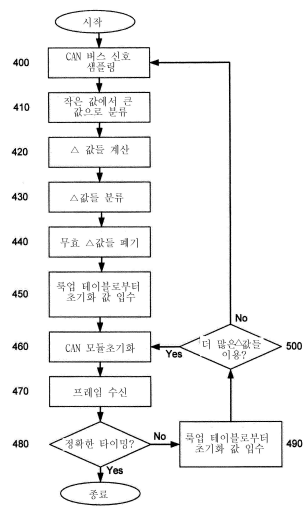
심사관 : 김대성

(54) 발명의 명칭 CAN 버스 모듈을 구비하고, 자동 속도 검출을 하는 마이크로컨트롤러

(57) 요약

마이크로컨트롤러에서 CAN(Controller Area Network) 모듈을 초기화하는 방법에서, CAN 신호의 복수의 하강 또는 상승 구간들 사이의 기간들을 측정하는 단계; 상기 기간들을 분류하는 단계; 상기 분류된 기간들의 인접 구간들 사이의 차이 값들을 구하는 단계; 상기 차이 값들을 분류하는 단계; 상기 분류된 차이 값들로부터 제1 차이 값을 선택하고, 상기 제1 차이 값으로부터 제1 주파수를 구하는 단계; 상기 선택된 주파수를 이용하여 상기 CAN 모듈을 초기화하는 단계; CAN 신호 프레임의 수신하는 단계; 에러가 발생했는지 여부를 판단하는 단계; 에러가 발생한 경우 다음 주파수를 선택하고, 유효한 CAN 주파수가 구해질 때까지 초기화를 반복하는 단계가 제공될 수 있다.

대표도 - 도4



## 명세서

### 청구범위

#### 청구항 1

마이크로컨트롤러에서 CAN(Controller Area Network) 모듈을 초기화하는 방법으로서,

CAN 신호의 연속 하강 또는 연속 상승 에지들 사이의 시간들을 측정하는 단계;

제1 분류 리스트를 제공하기 위해 상기 측정된 시간들을 분류하는 단계;

상기 제1 분류 리스트의 인접한 측정 시간들 사이의 차이 값들을 결정하는 단계;

제2 분류 리스트를 제공하기 위해 상기 차이 값들을 분류하는 단계;

상기 제2 분류 리스트로부터 제1 차이 값을 선택하고, 상기 제1 차이 값으로부터 제1 비트 전송률 주파수를 결정하는 단계;

상기 선택된 제1 비트 전송률 주파수를 이용하여 상기 CAN 모듈을 초기화하는 단계;

CAN 신호 프레임을 수신하는 단계;

에러가 발생했는지의 여부를 판단하는 단계; 및

에러가 발생하면, 다음 주파수를 선택하고 유효한 CAN 주파수가 발견될 때까지 상기 CAN 모듈의 초기화를 반복하는 단계를 포함하는 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

#### 청구항 2

제1항에 있어서,

상기 시간들을 측정하기 위해 외부 CAN 라인을 상기 마이크로컨트롤러의 캡처 모듈(capture module)의 입력과 결합시키는 단계를 더 포함하는 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

#### 청구항 3

제1항에 있어서,

상기 시간들을 측정하기 위해 CAN 신호를 상기 마이크로컨트롤러의 캡처 모듈의 입력에 내부적으로 공급하는 단계를 더 포함하는 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

#### 청구항 4

제1항에 있어서,

상기 차이 값들을 분류하는 단계 이후에 무효 주파수들을 초래하는 차이 값들을 폐기하는 단계를 더 포함하는 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

#### 청구항 5

제1항에 있어서,

상기 시간들의 측정에 타이머 및 캡처 유닛이 사용되는, 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

**청구항 6**

제3항에 있어서,

CAN 신호의 연속 하강 에지들 또는 연속 상승 에지들 사이에 소정 수의 시간들이 측정되는, 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

**청구항 7**

제1항에 있어서,

어떠한 유효 주파수도 발견되지 않으면 상기 방법이 반복되는, 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

**청구항 8**

제1항에 있어서,

상기 차이 값으로부터 주파수를 결정하는 단계는 룩업 테이블(look-up table)을 이용하여 수행되는, 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

**청구항 9**

제1항에 있어서,

상기 차이 값으로부터 주파수를 결정하는 단계는 주파수를 계산하기 위해 라운딩(rounding) 또는 트렁케이션(truncation)을 이용하는, 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

**청구항 10**

제1항에 있어서,

상기 마이크로컨트롤러는 상기 방법을 수행하도록 프로그램되는, 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

**청구항 11**

제1항에 있어서,

상기 마이크로컨트롤러는 상기 방법을 실행하는 상태 머신(state machine)을 포함하는, 마이크로컨트롤러에서 CAN 모듈 초기화 방법.

**청구항 12**

마이크로컨트롤러로서,

중앙 처리 장치(CPU);

CAN(Controller Area Network) 모듈;

타이머; 및

상기 타이머와 결합된 캡처 모듈을 포함하며;

상기 마이크로컨트롤러는, 제1항 내지 제11항 중 어느 한 항에 따른 방법을 수행하여 CAN 주파수를 자동으로 검출하도록 구성되는, 마이크로컨트롤러.

**청구항 13**

제12항에 있어서,

상기 마이크로컨트롤러는 또한 무효 주파수를 초래하는 차이 값들을 폐기하도록 구성되는, 마이크로컨트롤러.

**청구항 14**

제12항에 있어서,

상기 캡처 모듈은 상기 타이머와 캡처 레지스터를 결합하는 드라이버를 제어하는 에지 검출기를 포함하는, 마이크로컨트롤러.

**청구항 15**

제12항에 있어서,

상기 CAN 모듈의 입력부와 상기 캡처 모듈의 입력부 사이에 결합된 제어 가능한 드라이버를 포함하는, 마이크로컨트롤러.

**청구항 16**

제12항에 있어서,

상기 마이크로컨트롤러는 룩업 테이블을 이용하여 상기 차이값으로부터 주파수를 결정하도록 구성되는, 마이크로컨트롤러.

**청구항 17**

제12항에 있어서,

상기 마이크로컨트롤러는 라운딩 또는 트렁케이션을 이용하여 상기 차이값으로부터 주파수를 결정하도록 구성되는, 마이크로컨트롤러.

**청구항 18**

제12항에 있어서,

상기 마이크로컨트롤러는 상기 CAN 주파수를 결정하기 위해 소프트웨어에 의해 프로그램되는, 마이크로컨트롤러.

**청구항 19**

제12항에 있어서,

상기 CAN 주파수를 결정하기 위한 상태 머신을 포함하는, 마이크로컨트롤러.

**청구항 20**

제19항에 있어서,

상기 상태 머신을 제어하는 특수 기능 레지스터(special function register)를 포함하는, 마이크로컨트롤러.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 CAN(Controller Area Network) 모듈을 갖는 마이크로컨트롤러에 관한 것이다.

**배경 기술**

[0002] CAN(Controller Area Network)은 고 레벨의 보안을 갖는 분배식 실시간 제어를 효율적으로 지원하는 직렬 통신 프로토콜이다. CAN의 응용은 고속 디지털 통신 네트워크로부터 저비용 멀티플렉싱 배선까지의 범위를 갖는다. CAN은 실시간 응용을 위한 고 신뢰성(high integrity) 직렬 데이터 통신 버스이다. CAN은 초당 최대 1Megabits의 데이터율로 동작하고, 우수한 에러 검출 및 격리 능력(confinement capability)을 가지며, 원래는 차량에서 사용하기 위해 개발되었다. CAN 이면의 동기는 차량 전자장치들, 엔진 제어 장치, 센서, ABS(anti-skid brake system) 등의 정보처리 적시 정보교환 가능성(interoperability)을 개선함과 동시에 와이어링 하네스 중량 및 복잡도를 저감함으로써 차량을 보다 신뢰성 있고, 안전하고, 연료 효율이 좋게 만드는 것이었다. CAN의 개시 이래 CAN 프로토콜은 산업 자동화 및 자동차(automotive)/트럭 응용에 폭넓은 인기를 끌었다. 노이즈 환경에서의 CAN 버스의 강성 및 고장 조건들을 검출하고 그것들로부터 복구되는 능력은 CAN을 산업 제어 장비, 의료 장비, 테스트 장비, 모바일 및 휴대용 머신, 가전제품 등의 사용에 적합하게 만들었다.

[0003] CAN은 예를 들어 두 개의 와이어를 포함하는 하나의 논리 버스를 갖는 비동기 직렬 버스 시스템이다. CAN은 동일한 버스 노드들을 갖는 개방형 선형 버스 구조를 갖는다. CAN 버스는 둘 이상의 노드들로 구성된다. 버스 상의 노드들의 수는 다른 노드들의 통신을 방해하지 않으면서 동적으로 변경될 수 있다.

[0004] CAN 논리 버스는 "와이어드-OR(wired-OR) 기구"에 대응하고, "열성(recessive)" 비트들(전부 반드시 그렇지는 않지만, 대개는 논리 레벨 "1"에 상응)은 "우성(dominant)" 비트들(대개 논리 "0")에 의해 겹쳐 써진다. 버스 노드가 우성 비트를 전송하지 않는 한, 버스 라인은 열성 상태에 있지만, 임의의 버스 노드로부터의 우성 비트는 우성 버스 라인 상태를 생성한다. 따라서 CAN 버스 라인의 경우에는, 상기 두개의 가능한 버스 상태들(우성 및 열성)을 전송할 수 있는 매체가 선택된다. 사용되는 공통적인 물리적 매체는 트위스티드 페어(twisted wire pair)이다. 그래서 버스 와이어들은 "CANH"와 "CANL"이라고 부르고, CAN 제어기 노드들에 직접적으로 또는 그 제어기를 통해 CAN 제어기 노드들에 접속될 수 있다.

[0005] CAN 버스 프로토콜에서, 어드레스되는 것은 버스 노드들이 아니라 어드레스 정보는 전송되는 메시지들에 포함된다. 이것은 메시지 콘텐츠, 예를 들어 엔진 속도, 오일 온도 등을 식별하는 식별자(각 메시지의 부분)를 통해 수행된다. 상기 식별자는 추가로 메시지의 우선순위를 나타낸다. 상기 식별자의 2진 값이 낮으면 낮을 수록 상기 메시지의 우선순위는 높아 진다(우성비트가 많아 진다).

[0006] CAN 제어기에서, 완전한 프레임들이 문자 대신 전송되는 것을 제외하고, 구조는 단일 UART와 유사한데, (전형적으로) 단일 송신 버퍼 및 이중 버퍼링된 수신 버퍼가 있다. CPU는 상기 송신 버퍼에 프레임을 입력하고, 프레임이 전송될 때 인터럽트를 발생하고, 그리고 CPU는 수신 버퍼에서 프레임을 수신하고, 인터럽트를 취하고, 버퍼를 클리어한다(후속 프레임이 수신되기 이전에). CPU는 송신 및 수신을 관리하고, 프레임들의 저장을 제어하여야 한다.

[0007] 초기화 동안에 송신 장치 및 수신 장치는 일반적으로 소정의 CAN 주파수를 사용할 수 있도록 그것들의 CAN 모듈들을 프로그래밍한다. 그러나 많은 응용에 있어서, 상기 주파수는 공지될 수 없다. 또한, 각각의 CAN 비트의 길이는 일부 어떤 공칭 길이 근처에서 대략 20% 만큼 변할 수 있다. 더욱이, 실행 시스템(running system)에는 시스템이 동기화될 수 있는 어떠한 "공지의 초기화 전송부(known initialization transmissions)"가 없다. 환언하면, 장치가 이미 설정된 CAN 버스 시스템에 결합된 경우, 전달된 메시지들의 콘텐츠는 공지되지 않는다. 따라서 종래의 시스템에 있어서, 매우 긴 설정 시간을 가져올 수 있거나 또는 전체적으로 실패할 가능성도 있는 CAN 버스의 정확한 동작 주파수를 결정하는데 시행오차법(trial and error method)이 사용될 수 있다. 다른 직렬 버스 시스템에서, 초기 설정 단계(initialization phase)가 "5555" 또는 "AAAA"와 같은 동기화 문자 시퀀스를 이용하여 실행될 수 있다. 그러나 이러한 초기설정 루틴들은 특히 CAN 버스가 이미 동작중인 경우는 이용할

수 없다.

**발명의 내용**

**해결하려는 과제**

[0008] 따라서 CAN 시스템에 사용되는 동작 주파수를 신속하게 결정할 수 있는 CAN 모듈을 구비한 마이크로컨트롤러용 개선된 자동 속도 검출 시스템 및 방법이 필요하다.

**과제의 해결 수단**

[0009] 일 실시예에 따르면, 마이크로컨트롤러에서 CAN(Controller Area Network) 모듈을 초기화하는 방법은, CAN 신호의 복수의 하강 또는 상승 구간들 사이의 기간들을 측정하는 단계; 상기 기간들을 분류하는 단계; 상기 분류된 기간들의 인접 기간들 사이의 차이 값들을 구하는 단계; 상기 차이 값들을 분류하는 단계; 상기 분류된 차이 값들로부터 제1 차이 값을 선택하고, 상기 제1 차이 값으로부터 제1 주파수를 구하는 단계; 상기 선택된 주파수를 이용하여 상기 CAN 모듈을 초기화하는 단계; CAN 신호 프레임을 수신하는 단계; 에러가 발생했 지 여부를 판단하는 단계; 에러가 발생한 경우 다음 주파수를 선택하고, 유효한 CAN 주파수가 구해질 때까지 초기화를 반복하는 단계를 포함한다.

[0010] 또 다른 실시예에 있어서, 상기 방법은 또한 상기 기간들을 측정하기 위해 외부 CAN 라인을 상기 마이크로컨트롤러의 캡처 모듈(capture module)의 입력과 외부 결합하는 단계를 포함할 수 있다. 또 다른 실시예에 있어서, 상기 방법은 또한 상기 기간들을 측정하기 위해 CAN 신호를 상기 마이크로컨트롤러의 캡처 모듈의 입력에 내부적으로 공급하는 단계를 포함할 수 있다. 또 다른 실시예에 있어서, 상기 방법은 상기 차이 값들을 분류하는 단계 이후에 무효 주파수들을 초래하는 차이 값들을 폐기하는 단계를 더 포함할 수 있다. 또 다른 실시예에 있어서, 상기 기간들의 측정에 타이머 및 캡처 유닛이 사용될 수 있다. 또 다른 실시예에 있어서, CAN 신호의 하강 구간들 사이에 10 기간들이 측정될 수 있다. 또 다른 실시예에 있어서, 상기 방법은 유효 주파수가 발견되지 않는 경우에 반복될 수 있다. 또 다른 실시예에 있어서, 상기 차이 값으로부터 주파수를 구하는 단계는 룩업 테이블(look-up table)을 이용하여 실행될 수 있다. 또 다른 실시예에 있어서, 상기 차이 값으로부터 주파수를 구하는 단계는 주파수를 계산하기 위해 라운딩(rounding) 또는 트렁케이션(truncation)을 이용할 수 있다. 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 상기 방법을 실행하도록 프로그램될 수 있다. 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 상기 방법을 실행하는 상태 머신(state machine)을 포함할 수 있다.

[0011] 또 하나의 실시예에 따르면, 마이크로컨트롤러는, 중앙 처리 장치(CPU), CAN(Controller Area Network) 모듈, 타이머, 상기 타이머와 결합된 캡처 모듈을 포함할 수 있으며, 상기 마이크로컨트롤러는, CAN 신호의 복수의 상승 또는 하강 구간들 사이의 기간들을 측정하고; 상기 기간들을 분류하며; 상기 분류된 기간들의 인접 기간들 사이의 차이 값들을 구하고; 제1 차이 값을 선택하고, 상기 제1 차이 값으로부터 제1 주파수를 구하며; 상기 선택된 주파수를 이용하여 상기 CAN 모듈을 초기화하고; CAN 신호 프레임을 수신하고; 에러가 발생했는지 여부를 판단하며; 에러가 발생한 경우 다음 주파수를 선택하고, 유효한 CAN 주파수가 구해질 때까지 상기 CAN 모듈의 초기화를 반복하여 CAN 주파수를 자동으로 검출하도록 구성된다.

[0012] 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 또한 무효 주파수들을 초래하는 차이 값들을 폐기하도록 구성될 수 있다. 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 캡처 모듈은 상기 타이머와 캡처 레지스터를 결합하는 드라이버를 제어하는 예지 검출기를 포함할 수 있다. 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 상기 CAN 버스 모듈의 입력과 상기 캡처 모듈의 상기 입력 사이에 결합된 제어가능한 드라이버를 포함할 수 있다. 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 룩업 테이블을 이용하여 상기 차이 값으로부터 주파수를 구하도록 구성될 수 있다. 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 라운딩 또는 트렁케이션을 이용하여 상기 차이 값으로부터 주파수를 구하도록 구성될 수 있다. 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 상기 CAN 주파수를 구하기 위해 소프트웨어에 의해 프로그램될 수 있다. 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 상기 CAN 주파수를 구하기 위한 상태 머신을 포함할 수 있다. 마이크로컨트롤러의 또 다른 실시예에 있어서, 상기 마이크로컨트롤러는 상기 상태 머신을 제어하는 특수 기능 레지스터(special function register)를 포함할 수 있다.

**발명의 효과**

[0013] 본 발명에 의하면 CAN 시스템에 사용되는 동작 주파수를 신속하게 결정할 수 있는 CAN 모듈을 구비한 마이크로컨트롤러용 개선된 자동 속도 검출 시스템 및 방법을 얻을 수 있다.

**도면의 간단한 설명**

[0014] 도 1은 집적 CAN 모듈을 구비한 마이크로컨트롤러의 일 실시예의 블록도이다.

도 2는 예시의 CAN 신호의 타이밍도이다.

도 3은 CAN 동작 주파수를 구하기 위한 일 실시예에 따른 원리를 도시한다.

도 4는 일 실시예에 따른 흐름도를 도시한다.

**발명을 실시하기 위한 구체적인 내용**

[0015] 도 1을 참조하면, 마이크로컨트롤러는 파선으로 도시된다. 마이크로컨트롤러(100)는 중앙 처리 장치(CPU)(110) 및 CPU(110)와 결합되는 CAN 버스 모듈 또는 주변 장치를 포함한다. 외부 핀들(130 및 135)이 제공되고, 이 외부핀들을 통해 CAN 버스 모듈(120)이 그 별개의 송신 및 수신 라인들(CANTX, CANRX)을 상기 외부 CAN 버스(180)의 트위스티드 페어(CANH 및 CANL)에 접속할 수 있다. 이는 일반적으로 상기 별개의 내부 신호들(CANRX 및 CANTX)을 각각의 CAN 버스 신호들(CANH 및 CANL)로 변환하는 외부 인터페이스 드라이버(105)를 통해 수행된다. 그러나 다른 실시예들은 CAN 버스의 직접적인 접속을 가능케 하는 상기 인터페이스 장치를 내부적으로 제공할 수 있다.

[0016] 또한, 마이크로컨트롤러는 캡처/비교/펄스 폭 변조(CCP) 모듈(140), 타이머 모듈(170), 및 클록 시스템(170)을 포함하고, 이들 모두는 CPU(110)에 결합된다. CCP 유닛(140)은 외부 신호들을 평가하는 또 하나의 외부 핀(150)에 결합된다. 이 목적을 위해, 일 실시예에 따르면, CCP 모듈(140) 내부의 제어가능한 드라이버(146)는 외부 핀(150)에 접속될 수 있다. 그러나 도 1에 도시한 예시의 장치는 또한 파선들로 도시한 내부 신호들을 평가/처리할 수 있다. 예를 들어, 일 실시예에서, 내부 CAN 신호(CANRX)는 제어가능한 드라이버(185)를 통해 CCP 모듈(140)로 공급될 수 있다. 이 실시예는 특히 내부 CANRX 신호들의 평가할 수 있도록 내부 CAN 버스 인터페이스(105)가 제공되는 경우에 고려될 수 있다. 그러나 본 분야의 당업자라면 알 수 있는 바와 같이, 다른 실시예들은 CAN 버스 라인들을 직접적으로 평가할 수 다. 이 경우에, 내부에 또는 외부에 배치되는 차동 드라이버/컨버터 유닛이 CAN 버스 상의 신호들을 단일 디지털 신호로 변환하고, 이들을 CCP 모듈(140)에 공급하는데 사용될 수 있다.

[0017] CCP 모듈(140)은 드라이버들(145 및 185)의 출력을 수신하는 프리스케일러(142) 및 내, 외부 입력 신호를 선택하는데 이용될 수 있는 임의의 다른 제어가능한 드라이버를 포함할 수 있다. 상기 프리스케일러(142)의 출력은 수신 신호의 상승 구간 또는 하강 구간 중 어느 하나를 프로그램가능하게 검출할 수 있는 예지 검출기(148)와 결합된다. 상기 예지 검출기의 출력 신호는 타이머 모듈(170)의 타이머 레지스터(175)를 CCP 모듈(140)의 캡처 레지스터(144)에 결합하는 또 하나의 드라이버(145)를 제어한다. 예지 검출기(160)는 클록 시스템(160)으로부터의 클록 신호를 수신할 수 있다.

[0018] 자동 고속 검출 기능을 제공하기 위해 상기 CCP 모듈의 입력 또는 CCP 모듈이 다수 입력을 포함하는 경우 그 입력들 중 하나는 핀(130) 상에서 이용가능한 CANRX 라인과 결합된다. 도 1에 도시한 바와 같이, 이는 CANRA 핀(130)과 외부 핀(150)을 접속하거나 또는 제어가능한 드라이버(185)를 통한 내부 결합을 선택함으로써 이루어질 수 있다. 이 설정은 자동 고속 검출 단계 동안에만 가능하다. 일단 속도가 결정되면, CCP 입력은 다른 신호들을 처리하는데 사용될 수 있다.

[0019] 도 2는 외부 CAN 버스(180) 상의 예시의 신호를 도시한다. CCP 모듈(140)은 CAN 신호(200)의 하강 구간들을 분석하도록 프로그램된다. 그러나 다른 실시예들에 있어서, 다른 신호 특성들이 구간을 결정하기 위하여 분석될 수 있다. 도 2에 따르면, 하강 구간이 발생할 때 마다, CCP 모듈(140)은 타이머 모듈로부터 타이머 값을 입수한다. 입수한 예시의 타이머 값들을 도시한다. 값들은 방법의 원리의 이해를 단지 용이하게 하기 위하여 가장 가까운 100 자리값으로 어림 계산된다(round). 도 2에 도시한 예에서, 10 하강 구간들이 조사된다. 그러나 더 많거나 적은 구간들이 처리될 수 있다. 입수되는 구간들이 많을 수록 자동 고속 검출이 더 정밀하게 된다. 타이머는 100ns의 해상도로 초기화될 수 있다. 이 목적을 위해 타이머는 10MHz 또는 임의의 다른 적합한 클록 주파수로 클럭화될 수 있다. 타이머 값들은 예를 들어 도 1에 도시한 바와 같이 16 비트 값들일 수 있다. 따라

서 클록 소스에 따라 각각 다른 해상도들을 얻을 수 있다. 타이머 해상도가 높아지면 더 정확한 추정이 가능하다. 가장 빠른 가능한 타이머 소스를 이용하는 것은 타이머의 해상도를 증가시키므로 바람직한 방법이 된다. 예를 들어, 40MHz 클록이 타이머 해상도를 25ns까지 개선하는데 사용될 수 있다. 그리고 이들 타이머 값들은 어레이에 저장될 수 있다.

[0020] 도 3은 입수된 타이머 값들이 추가로 처리되는 방법을 도시한다. 첫 번째 단계에서, 10개의 입수된 타이머 값들을 갖는 어레이가 최 좌측 컬럼에 보여진 바와 같이 가장 낮은 값으로부터 가장 높은 값으로 분류된다. 그러면 각각의 두 개의 인접하는 타이머 값들 사이의 차이가 제2 컬럼에 표시된 바와 같이 계산된다. 다음에 다른 차이값들은 제3 컬럼에 도시한 바와 같이 다시 분류된다. 또 하나의 단계에서, 무효 차이값들은 폐기될 수 있다. 예를 들어, 명확히 1 $\mu$ s 이하의 값들 또는 임의의 소정의 최대 주기 또는 소정의 최대 주파수 이하의 값들은 도 3의 제3 컬럼에 도시한 바와 같이 폐기될 수 있다. 도 3의 제4 컬럼에 보여진 바의 나머지 값들의 경우에, 나머지 주기 값들에 대한 결과적인 주파수가 결정된다. 그리고 제1 값은 CAN 모듈 주파수의 설정에 사용된다. 이 목적을 위해 일 실시예에 따라 윈도우가 CAN 시스템에서 성립하는 가장 가까운 수로 주파수를 어림계산하는데(round) 사용될 수 있다. 예를 들어, 250kbps 또는 275kbps가 적용되고 263kbps는 적용되지 않는다. 일단 이 주파수가 결정되면, 예를 들어, 도 3에 보여진 바와 같이, 500kbps가 결정되면, CAN 모듈(120)은 이 주파수를 이용하여 초기화되고, "리슨 온리(listen only)" 모드로 설정되어 정상의 버스 트래픽은 방해되지 않는다. 그리고 나서 시스템은 대기하고, 수신된 메시지가 유효한지 또는 에러 메시지가 CAN 모듈에 의해 생성되는지를 결정한다. 에러 메시지의 경우에, 자동 고속 검출 방법은 도 3에 보여진 바와 같은 다음의 이용가능한 주파수 예를 들어 450kbps를 선택한다. 이 단계들은 정확한 주파수가 설정될 때까지 반복된다. 따라서 이 방법은 도 3의 최종 컬럼에서 결정된 바와 같이 각각 다른 주파수들을 이용하여 CAN 모듈(120)의 초기화를 반복한다. 이들 결정된 값들의 어느 것도 정확한 설정을 생성하지 않는 경우, 방법은 다시 시작하여 CAN 신호의 또 하나의 10 개의 구간들을 평가한다.

[0021] 도 3의 제4 컬럼에 보여진 타이머 차이 값들은 실제 주파수 값들을 포함하는 테이블을 직접적으로 액세스하는데 사용될 수 있다. 이러한 방법에 의해 유용한 주파수 값들을 얻기 위해 트렁케이션 또는 라운딩을 용이하게 실행할 수 있다.

[0022] 도 4는 전술한 방법의 일례의 흐름도를 도시한다. 단계 400에서 상승 또는 하강 구간들을 검출하고, 관련 타이머 값들 t(1) 내지 t(n)을 저장함으로써 CCP 모듈을 이용하여 CAN 신호가 평가된다. 이후 이들 값들은 단계 410에서 값들 ts(1) - ts(n)로 분류된다. 단계 420에서 x=1...n-1에 대한 d(x)=ts(x+1)-ts(x)를 계산함으로써  $\Delta$ 값들 d(1) - d(n-1)가 구해진다. 상기  $\Delta$ 값들은 단계 430에서 ds(1)...ds(n-1)로 다시 분류된다. 이어서 선택적으로 단계 440에서 유효 주파수들을 형성하지 않는 값들 ds()는 폐기된다. 나머지 값들은 단계 450에서 주파수 값들 f(x) = 1/ds(x)를 결정하는데 사용된다. 단계 460에서, 제1 값 f(1)이 CAN 모듈(120)을 초기화하는데 사용된다. 이어서, 단계 470에서 프레임이 수신된다. 단계 480에서는 CAN 모듈이 임의의 에러 메시지들을 생성했는지 여부를 검사하여 정확한 타이밍이 결정된다. 에러 메시지가 생성되지 않으면, 루틴은 종료한다. 에러가 생성된 경우에는 루틴은 다음 값 f(i)을 획득함으로써 예를 들어, 단계 490에서 록업 테이블을 이용함으로써 진행된다. 단계 500에서는 이용할 수 있는 더 이상의 값들이 있는지 없는지 여부를 결정된다. 이용할 수 있는 더 이상의 값들이 없으면 루틴은 단계 400으로 점프하고, 활용할 수 있으면 루틴은 단계 460으로 복귀한다.

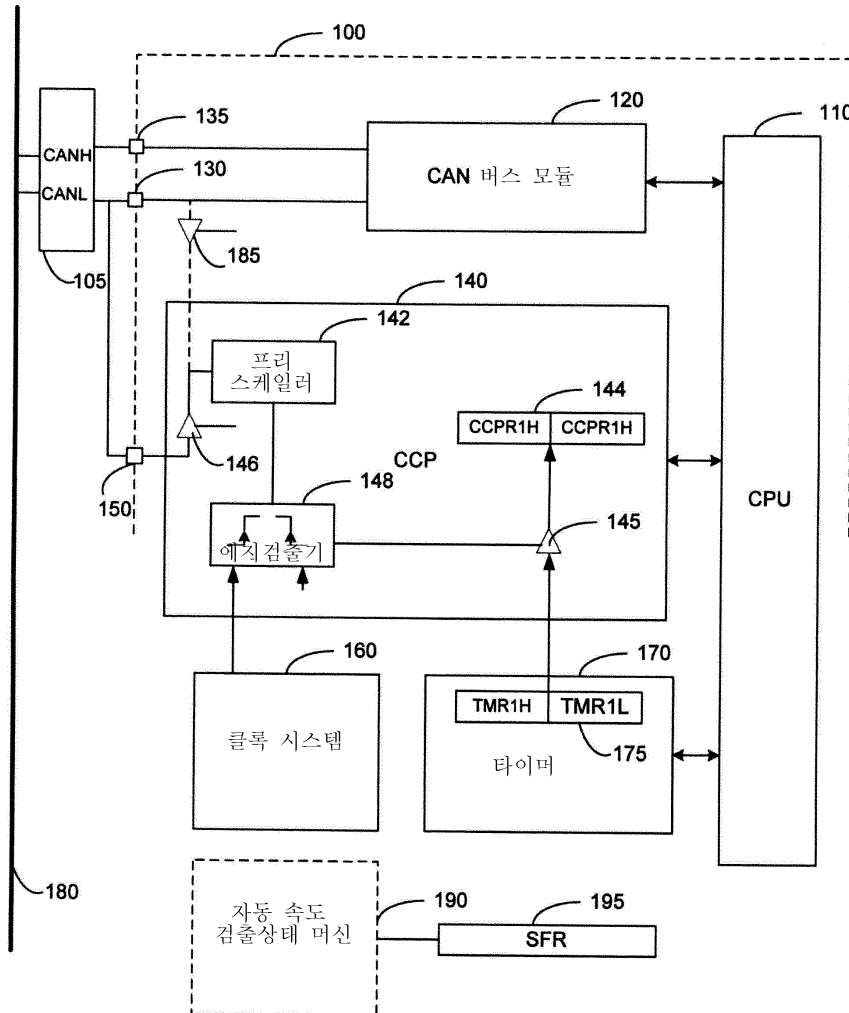
[0023] 일 실시예에 따르면, 전술한 시퀀스는 소프트웨어에 의해 실행될 수 있다. 그러나 또 하나의 실시예에 따르면, 상태 머신(190)은 전술한 특정 시퀀스들을 제어하는데 사용될 수 있다. 도 1은 상태 머신(190)을 도시한다. 이 상태 머신은 예를 들어 도 1에 도시한 제어가능한 드라이버(185)를 통해 CAN 버스 모듈(120)에의 입력 신호를 내부적으로 선택할 수 있는 마이크로컨트롤러(100)에서 특히 바람직하다. 상기 상태 머신은 특수 기능 레지스터(special function register; 195)에서 비트 설정을 통해 시동될 수 있다. 일단 시동되면, 상태 머신(190)은 드라이버(185)를 통해 내부 CAN 버스 신호를 CCP 모듈(140)에 결합하고, 전술한 바와 같이 CAN 모듈(120)의 초기화를 실행하고, CAN 모듈(140)에서 CAN 신호를 분리한다. 그리고 난 후에 유지는 다른 신호들의 평가를 위해 이 모듈을 사용할 수 있다.

[0024] 따라서 본 발명은 전술한 목표 및 목적들(the ends)을 잘 수행하고, 뿐만 아니라 내부의 기타 목적들을 실행하도록 구성되어 있다. 본 발명은 본 발명의 바람직한 실시예들을 참조로 하여 서술 및 기술 그리고 정의되었지만, 이러한 참조는 본 발명을 제한하는 것이 아니고 그러한 제한을 암시하는 것도 아니다. 본 발명은 본 분야의 당업자라면 형태 및 기능에 있어서 변형, 및 변경 그리고 등가물을 고려할 수 있을 것이다. 서술되고 기술된 본 발명의 바람직한 실시예들은 단지 예시의 목적이고, 본 발명의 영역을 한정하는 것이 아니다. 결론적으로

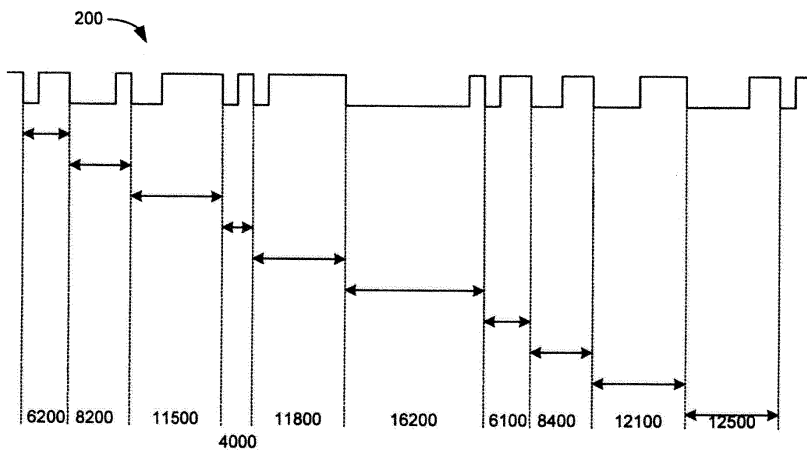
로, 본 발명은 첨부 청구범위들의 사상 및 영역에 의해서만 한정되는 것을 지향하고, 모든 점에서 등가물의 완전한 인지를 제공한다.

도면

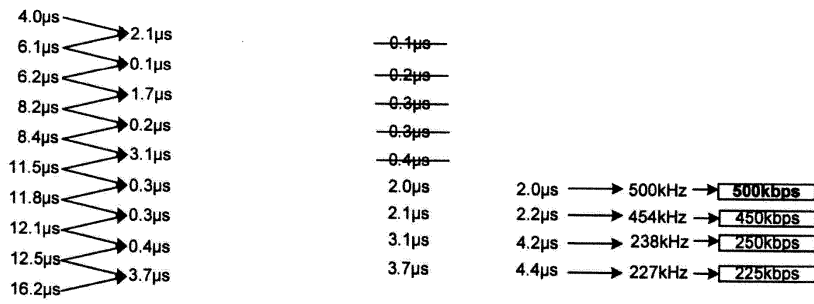
도면1



도면2



도면3



도면4

