

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4748634号  
(P4748634)

(45) 発行日 平成23年8月17日(2011.8.17)

(24) 登録日 平成23年5月27日(2011.5.27)

(51) Int. Cl. F I  
 H O 4 L 12/40 (2006.01) H O 4 L 12/40 Z  
 G O 6 F 13/372 (2006.01) G O 6 F 13/372 C

請求項の数 13 (全 11 頁)

(21) 出願番号	特願2001-552225 (P2001-552225)	(73) 特許権者	501125231
(86) (22) 出願日	平成12年11月14日(2000.11.14)		ローベルト ボッシュ ゲゼルシャフト
(65) 公表番号	特表2003-521144 (P2003-521144A)		ミット ベシュレンクテル ハフツング
(43) 公表日	平成15年7月8日(2003.7.8)		ドイツ連邦共和国 70442 シュトゥ
(86) 国際出願番号	PCT/DE2000/004021		ットガルト ポストファッハ 30 02
(87) 国際公開番号	W02001/052074		20
(87) 国際公開日	平成13年7月19日(2001.7.19)	(74) 代理人	100095957
審査請求日	平成19年11月14日(2007.11.14)		弁理士 亀谷 美明
(31) 優先権主張番号	100 00 305.2	(74) 代理人	100096389
(32) 優先日	平成12年1月5日(2000.1.5)		弁理士 金本 哲男
(33) 優先権主張国	ドイツ(DE)	(74) 代理人	100101557
			弁理士 萩原 康司

最終頁に続く

(54) 【発明の名称】 バスシステムと結合された少なくとも2名の加入者間でデータを交換する方法と装置

(57) 【特許請求の範囲】

【請求項1】

バスシステムにより接続された少なくとも2つの加入者間でデータをメッセージで交換する方法であって、

前記データを含む前記メッセージは、前記加入者によって前記バスシステムを經由して伝送され、いずれのメッセージも、前記含まれるデータを特徴付ける識別子を含み、いずれの加入者も前記識別子によって、前記メッセージを受信するかどうかを決定する上記の方法において、

前記メッセージは、第1の加入者によって、前記第1の加入者が少なくとも1つの設定可能な時間間隔で前記バスを經由して、前記第1加入者の時間情報を含む参照メッセージを繰り返し伝送するように時間的に制御され、前記時間間隔は設定可能な長さのタイムウィンドウに区分され、前記メッセージは前記タイムウィンドウで伝送されることを特徴とする、上記の方法。

【請求項2】

前記参照メッセージは、特殊な識別子により明白に識別されることを特徴とする、請求項1に記載の方法。

【請求項3】

前記第1加入者の前記時間情報は、前記第1加入者の参照マークに対応することを特徴とする、請求項1に記載の方法。

【請求項4】

前記参照メッセージと、次の前記参照メッセージまでの後続の前記タイムウィンドウとが、設定可能な長さおよび/または設定可能な構造の1つの第1のサイクルにまとめられ、前記構造は、前記時間間隔における、前記参照メッセージの後に続く前記タイムウィンドウの長さ、個数および時間的なポジションに対応することを特徴とする、請求項1に記載の方法。

【請求項5】

前記第1のサイクルおよび後続のサイクルには番号が付され、前記参照メッセージは、現時点のサイクルの番号を含むことを特徴とする、請求項1および4に記載の方法。

【請求項6】

前記タイムウィンドウで伝送されるメッセージは、前記タイムウィンドウそれぞれに割り当てられることを特徴とする、請求項1に記載の方法。

10

【請求項7】

構造が同じである複数の第1のサイクルが1つの第2のサイクルにまとめられ、前記第2のサイクルにおいても、時間間隔が前記第1のサイクルの時間的長さよりも大きいタイムウィンドウによって、前記メッセージが繰り返し伝送されることを特徴とする、請求項1および4に記載の方法。

【請求項8】

前記タイムウィンドウでメッセージが周期的に伝送されることを特徴とする、請求項1～7のいずれか1項に記載の方法。

【請求項9】

20

前記第1のサイクルまたは第2のサイクルの少なくとも1つのタイムウィンドウにおいて、周期的なメッセージ伝送を行わず、前記タイムウィンドウにおいてアービトラージョンメッセージを伝送することを特徴とする、請求項1～8のいずれか1項に記載の方法。

【請求項10】

少なくとも1つのタイムウィンドウの一部として送信リリース区間が設けられ、メッセージは、開始部が送信リリース区間に存在する場合にのみ送信が許されることを特徴とする、請求項1～9のいずれか1項に記載の方法。

【請求項11】

各ノードにおいて、特殊な相対的時点が定められており、遅くとも前記特殊な相対的時点には新しい参照メッセージが見込まれることを特徴とする、請求項1～10のいずれか1項に記載の方法。

30

【請求項12】

バスシステムにより接続された少なくとも2つの加入者間でデータをメッセージで交換する装置であって、

前記データを含む前記メッセージは、前記加入者によって前記バスシステムを經由して伝送され、いずれのメッセージも、前記含まれるデータを特徴付ける識別子を含み、いずれの加入者も前記識別子によって、前記メッセージを受信するかどうかを決定する上記の装置において、

前記メッセージは、第1の加入者によって、前記第1の加入者が少なくとも1つの設定可能な時間間隔で前記バスを經由して、前記第1加入者の時間情報を含む参照メッセージを繰り返し伝送するように時間的に制御され、前記時間間隔を設定可能な長さのタイムウィンドウに区分し前記メッセージを前記タイムウィンドウ内で伝送する手段を備えることを特徴とする、上記の装置。

40

【請求項13】

バスシステムにより請求項1から11のいずれか1項に記載の方法を実行することを特徴とする、少なくとも2つの加入者間でデータを交換するためのバスシステム。

【発明の詳細な説明】

【0001】

従来の技術

本発明は、独立請求項の上位概念に記載の、バスシステムと結合された少なくとも2名の

50

加入者間でデータを交換する方法と装置に関する。

【0002】

自動車のネットワーク化における従来の技術として、数年前からCAN プロトコルが使用されている。かかるCAN プロトコルでは、通信が事象制御により制御されている。もし様々なメッセージの送信を同時に初期設定したいときは、非常に大きな負荷を与えなければならない。CANの非破壊的なアービトレーションメカニズムは、全てのメッセージの識別子もしくはコードの優先性に従う連続的な送信を保証する。厳しいリアルタイムシステムのためには、全システムに対して実行時間とバス負荷の分析を前もって実行しなければならないが、かかる分析は、全てのメッセージ デッドラインを遵守できるように（それ自体がピーク負荷以下となるよう）、安全策を取るために実行される。

10

【0003】

例えば、TTP/Cまたはインターバス Sのように、既に時間制御されたサービスに基づく通信プロトコルが存在する。かかる場合の特色は、送信時点を指定することにより、バスアクセスを既に前もって計画することである。従って実行時間の間に、いかなる衝突も起こらない。通信バスにおけるピーク負荷も同様に避けられる。従ってこの場合、バスがフルに活用されないことが頻繁に生じる。

【0004】

従来の技術からは、いかなる点でも最適な結果を得られないことが示されている。

【0005】

発明の利点

20

本発明は、バスシステムにより接続された少なくとも2名の加入者間でメッセージに含まれたデータを交換する方法と装置に関する。この場合、データを含むメッセージは、加入者によってバスシステムを経由して伝送され、いずれのメッセージも、含まれるデータの特徴付けるコードを含む。またかかる場合に、いずれの加入者もかかるコードによって、メッセージを受信するかどうかを決定する。かかる場合に、基盤となるバスシステムもしくはバスプロトコルとして、CAN バスを使用するのが特に実用的である。しかし、本発明は、一般的にあらゆるバスシステムもしくはバスプロトコルに関するものであり、これらの場合、オブジェクト指向のメッセージ伝送もしくはデータ伝送が用いられ、従ってメッセージおよび/またはそこに含まれるデータは、コード（識別子）によって明白に認識できる。これは、加入者ではなくメッセージもしくはかかるデータがアドレス指定される全てのバスに、特にCAN バスに該当する。

30

【0006】

かかる場合にメッセージには、有利な方法としては、いずれか第1の加入者が時間的な制御を実行し、実行の際に、第1の加入者が繰り返して1つの参照メッセージを、少なくとも1つの指定可能な時間的間隔の中で、バスを経由して伝送し、時間的間隔が指定可能な長さのタイムウィンドウに区分され、かかる際にメッセージは、かかるタイムウィンドウ内で伝送されるようにする。

【0007】

従って本発明は、有利な方法として、従来の技術と比較すると、より高度のプロトコル層を本来のバス（CAN） プロトコルに含み、かかるプロトコルは、本発明の場合、時間制御された通信を変わずに維持する。有利な方法としては、時間制御された通信によって、バスをフルに活用でき、同時にいずれのメッセージに対しても待ち時間を定義された数値に維持できるようにする。

40

【0008】

従って本発明は、周期的に経過するバス（CAN） メッセージ伝送を含む。これにより決定論的であって多重化可能な通信システムが生まれる。かかるシステムは、本発明の場合、TT CANとも呼ばれる。同様に、またCAN バスを出発点とするものでもあって、上記に述べたように、この構想は一般に全てのバスシステム、もしくはオブジェクト指向のメッセージ伝送を伴うバスプロトコルに該当するものである。

【0009】

50

実用的な方法として、参照メッセージと、次の参照メッセージまで後続のタイムウィンドウとを、指定可能な長さおよび/または指定可能な構造の第1のサイクルにまとめるものとする。この場合のかかる構造は、時間的間隔の中で参照メッセージに続くタイムウィンドウの長さ、個数、時間的ポジションに対応する。

【0010】

他の有利な方法として、同じ構造を持つ複数の第1のサイクルを1つの第2のサイクルにまとめ、その際に、第2のサイクルでもメッセージをタイムウィンドウで繰り返し伝送し、時間的間隔が第1のサイクルの時間的間隔より大きいものとする。

【0011】

実用的な方法としては、第1または第2のサイクルのタイムウィンドウの少なくとも1つにおいて、周期的なメッセージ伝送を実行しないものとする。かかる最初の空白のタイムウィンドウにおいて、アービトレーションメッセージを伝送する。すなわちこれは、周期的に伝送する必要はなく、例えば特定のプロセスが終了した場合に提供されるメッセージである。

【0012】

他の長所と有利な実施形態を、下記の説明と請求項の特徴に挙げた。

【0013】

添付の図面を用いて、下記に本発明を説明する。

【0014】

実施例の説明

TTCANは、主に時間制御された周期的通信に基づいており、かかる通信には、タイマー(ノード、加入者)が時間参照メッセージまたは短い参照メッセージRNを用いて、クロックパルスを付与している。次の参照メッセージRNまでの周期は、基本サイクルと呼ばれ、n個のタイムウィンドウに区分される(図2を参照)。いずれのタイムウィンドウも、異なる長さの周期的メッセージの排他的送信を許容する。かかる周期的メッセージは、TTCANコントローラにおいてタイミングマークを使用することにより送信され、かかるタイミングマークは、論理的な相対時間の経過と結合されている。TTCANは、空白のタイムウィンドウを考慮に入れることも許容する。かかるタイムウィンドウは、いわゆる自発的メッセージに利用でき、その場合に、タイムウィンドウ内におけるバスへのアクセスを、CANのアービトレーションスキーマを介して利用する(アービトレーションメッセージ)。タイマー時計(大域時間gZ)と個々のノード内部のローカル時間LZ1~LZ4との同期が考慮され、能率的に変換される。

【0015】

図1は、複数のバス加入者101~105を含むバスシステム100を示す。この場合いずれの加入者101~105も、自らの時間基準106~110を持ち、かかる時間基準を、時計、カウンター、パルス発生器等の内部の手段、あるいは外部から、各加入者に伝送できる。各々のローカルな時間基準LZ1~LZ4は、特にカウンター、例えば順方向カウントして16ビットのものであって、かかるカウンターに影響を加えてよいのはHWリセットによるものだけである。このローカルな時間基準は、いずれのノードもしくは加入者102~105でも実行される。この場合に加入者、タイマー101は、外部と接触するポジションを持つ。加入者の時間基準は、大域時間106であって、かかる大域時間106を大域時間gZと呼ぶ。この時間基準は、タイマー101で実行されているか、あるいは外部からこのタイマーに伝送される。大域時間gZは、原則的にいずれのノードでもローカル時間基準107~110もしくはローカル時間LZ(LZ1~LZ4)と、オフセットOS1~OS4から形成される。タイマー101におけるかかるオフセットOSgは、通常ゼロに等しい(OSg=0)。他の全てのノードは、ローカル時間LZ(LZ1~LZ4)とローカルオフセットOS1~OS4およびOSg(OSg=0のとき)から得た大域時間gZ上に、その視点を構築する。OSgがゼロでない場合は、例えば大域時間gZが外部からタイマー101に伝送され、かかるタイマーがさらに自分の時間基準106をも持っている場合に生じる。かかる場合は、タイマーも、大域時間gZに時刻

10

20

30

40

50

を合わせるが、 $gZ$ と時間基準106が一致しないことがある。ローカルオフセットは、参照メッセージの送信時点(SOF, Start of Frame)のローカル時間と、タイマーがかかる参照メッセージで伝送する大域時間との差である。

【0016】

ローカル時間基準と大域時間

ローカル時間基準：ローカル時間基準とは1つのカウンター、例えば順方向カウントして16ビットのものであって、影響を与えうるのは、HWリセットによる場合のみである。

ローカル時間基準は、いずれのノードでも実行されている。

【0017】

バッファレジスタの参照マーク：SOFを受け入れるときは、いつも、バッファレジスタにローカル時間基準がロードされる。

【0018】

リファレンスマーカー：現時点のメッセージが参照メッセージとして認識されると、その値は、バッファレジスタからリファレンスマーカーに転送される(参照マークとして)。リファレンスマーカーは、例えば16ビットレジスタとして設計される。

【0019】

タイマー参照マーク：参照メッセージにおいてタイマーの計時装置が受信した参照マークである。

【0020】

大域時間に対するローカルオフセット：大域時間に対するローカルオフセットとは、バッファレジスタにおける参照マークと、参照メッセージ中で受信された大域タイミングマークとの差である。かかるローカルオフセットは、ローカル時間から大域時間を計算するために使用される。タイマー自体のオフセットは、一定のままである。参照メッセージ中で、タイマーは、ローカル参照マークにローカルオフセットを加えて送信する。

【0021】

従ってタイマー101は、時間参照メッセージ111、より略して言うところの参照メッセージRNを送信するノードもしくは加入者でもある。矢印112は、参照メッセージRN111がそのほかの加入者102~105に、特に同時間に、送信されることを示す。

【0022】

参照メッセージRNは、TTCANを時間制御により周期的に運用するための基盤である。参照メッセージは、特殊な識別子、特殊なコードによって明白に表示されており、全てのノードに、ここでは102~105に、クロックとして受信される。これは、タイマー101から原則として周期的に送信される。かかる参照メッセージは、次のデータを含む：現時点の基本サイクル $BZ_n$ の数、大域時間におけるタイマーの参照マーク。

【0023】

参照マークは、タイマーの参照メッセージを受信する場合、「Start of Frame」ビット(SOF)の時点における内部カウンタステータスを静止化することによって発生する。従って、参照マークは、参照メッセージの受信時点におけるローカル時間基準のスナップショットである。加入者の欄に挙げた相対時間 $RZ_1 \sim RZ_4$ および $RZ_g$ は、ローカル時間基準と最後の参照マークの差である。使用されたタイミングマークに関する全ての定義は、個々の加入者の相対時間に関連する。参照マークは、例えば、持続的に信号として存在することができる(例えば、ゲート回路を経由して2つのレジスタ値を結ぶことによって)。参照マークは、TTCANバスにおける全てのノードの相対時間を決定する。

【0024】

同様に図示されているウォッチドッグ $W_g$ および $W_1 \sim W_4$ は、特殊な相対時点である。いずれのノードでも、遅くとも新たな参照メッセージと参照マークも期待されるような相対時間(ウォッチドッグ)が定義される。従って、かかるウォッチドッグは、特殊なタイミングマークである。ウォッチドッグは、特に初期設定と再初期設定の際に、通信が可能な状態にあるかどうかを監視するのに使用される。ウォッチドッグは、かかる場合、各参

10

20

30

40

50

照メッセージ間の時間的間隔よりも常に大きくするべきである。

【0025】

この場合にタイミングマークは、相対時間と、当初のバス(CAN)コントローラーにおけるアクションとの関連を生成する相対時点である。タイミングマークがレジスターとして示され、その際コントローラーが複数のタイミングマークを管理できる。1つのメッセージに複数のタイミングマークを割り当てることができる(例えば、図4参照:送信グループAが、タイムウィンドウZF1aだけでなく、タイムウィンドウZF4aにも現れている)。

【0026】

アプリケーションの場合は、特にアプリケーションウォッチドッグが用いられる。TTCANコントローラーに正しい操作信号を出すため、アプリケーションの際には、かかるウォッチドッグを定期的に使用しなければならない。かかるウォッチドッグが用いられる場合のみ、CANコントローラーは、メッセージを送信する。

【0027】

図2は、時間制御された周期的なメッセージ伝送もしくはデータ伝送の原理を経時的に示す。このメッセージ伝送には、タイマーが参照メッセージを用いてクロックパルスを与える。この場合に期間 $t_0 \sim t_6$ は、基本サイクルBZと呼ばれ、 $k$ 個のタイムウィンドウ( $k \leq N$ )に区分される。従って、かかる場合にタイムウィンドウZF $1 \sim N$ における $t_0 \sim t_1$ ,  $t_6 \sim t_7$ ,  $t_{12} \sim t_{13}$ および $t_{18} \sim t_{19}$ によって、各基本サイクルBZ $0 \sim BZ_3$ の参照メッセージRNが伝送される。参照メッセージRNに続くタイムウィンドウZF $1 \sim ZF_5$ 、すなわちタイムウィンドウの長さ( $t_s = t_{sb} - t_{sa}$ を含むセグメントSにおいて)、タイムウィンドウの個数およびタイムウィンドウの時間上のポジションは、指定可能である。これにより同じ構造の複数の基本サイクルから1つの全体サイクルGZ $1$ を形成することができ、かかるサイクルは、 $t_0$ で開始し、 $t_{24}$ で終了して改めてバスが実行される。これらのタイムウィンドウは、例えば、各々が32ビット時間である2~5個のセグメントを含む。タイムウィンドウの個数は、例えば2~16個であるが、この場合タイムウィンドウをただ1個、あるいは16個以上とすることも可能である。1つの全体サイクル中の基本サイクルの個数は例えば $2^m$ で、特には $m \geq 4$ とする。

【0028】

例えば、 $t_{zf1}$ および $t_{zf2}$ により、2つの送信リリース区間もしくはタイムウィンドウリリース区間を表示する。かかる区間は、例えば、16または32ビット時間続き、その範囲内であれば基本サイクルに関するメッセージ送信を開始してもよい時間フレームを表す。

【0029】

いずれのタイムウィンドウも、長さが異なる周期的メッセージの排他的送信を許容する。例えば、図3は、長さが異なる2つのメッセージと、タイムウィンドウにおける割り当てを示す。ブロック300のメッセージ1(N $1$ )は、例えば130ビットを、ブロック301のメッセージ2(N $2$ )は、例えば47ビットを含む。

【0030】

既に述べたように、最大および最小のタイムウィンドウは、メッセージの長さに依存して指定でき、ここに示す例では、例えば1つのタイムウィンドウにつき、2~5セグメントである。従って最大のタイムウィンドウZF $max$ は、それぞれ32ビット時間の5セグメント(S $1 \sim S_5$ )を含むブロック302として、最小のタイムウィンドウZF $min$ は、それぞれ32ビット時間の2セグメントを含むブロック303として指定される。かかるタイムウィンドウでメッセージN $1$ およびN $2$ が伝送され、従ってこの場合メッセージは、タイムウィンドウを完全に満たす必要はなく、むしろメッセージの長さに応じてタイムウィンドウの大きさが指定される。従ってZF $max$ は、あり得る最長のメッセージ、例えば130ビットもしくはビット時間のメッセージに、十分な時間もしくはスペースを提供しなければならず、ZF $min$ は、起こり得る最短の、例えば47ビットのメッセ

10

20

30

40

50

ージに適合させることができる。

【0031】

一般的には、ある特定のメッセージに提供される時間フレームのタイムウィンドウがある。(図3参照)。送信リリースが実行されると、メッセージのタイムウィンドウが開放され、かかるウィンドウの開始部は、定義されたタイミングマークと原則的に一致する。タイムウィンドウの長さは、例えば32ビット時間(ブロック304a参照)の*i*個のセグメントから決定される。この場合、セグメント化を特に32ビット時間ずつとすることにより、HWに適した大きさとなる。タイムウィンドウは、タイムウィンドウに現れる最長のメッセージより短くしてはならない。ビット時間は、特に定格のCANビット時間とする。

10

【0032】

送信リリース区間またはタイムウィンドウリリース区間は、その範囲内でメッセージの送信を開始してよい時間フレームである。送信リリース区間は、タイムウィンドウの一部である。従ってリリースは、タイミングマークとタイミングマーク+デルタの間に行われる。デルタの数値は、タイムウィンドウの長さより明らかに小さい(例えばZFF1またはZFF2の場合16または32)。その開始部が送信リリース区間中にあるメッセージは、送信してはならない。

【0033】

図4は、全体サイクル(送信マトリックス)GZ2を示す。

全体サイクル(送信マトリックス):全ての加入者の全てのメッセージ(RN, A~Fおよびアービトレーション)は、1つの送信マトリックスの成分として編成される(図4参照)。かかる送信マトリックスは、個々の基本サイクルBZ0a~BZ7aから構成される。全体サイクルGZ2の全ての基本サイクルは、同じ構造を持つ。これらの基本サイクルは、排他的(A~F)成分およびアービトレーション成分のいずれかを選択して構成できる。行の総数(つまり基本サイクルBZ0a~BZ7a)は、この場合 $2^m = 8$ でありことより、 $m = 3$ である。

20

【0034】

基本サイクル(送信マトリックスの行)は、参照メッセージRN中の参照マークと共に開始し、定義された長さの(RNのための最初のタイムウィンドウZF0もしくはZF RN)、順次続く複数(*i*)個のタイムウィンドウからなる。基本サイクル内のメッセージの配置は、自由に規定することができる。1つのタイムウィンドウは、排他的成分の場合、CANメッセージオブジェクトと結合される。タイムウィンドウは空けたままにしておくことも可能であり(409, 421, 441, 417, 445)、あるいはアービトレーション成分のために利用することも可能である(403, 427)。

30

【0035】

1つの送信グループ(送信マトリックスの列, A~F)は、常に同じタイムウィンドウであるが、異なる基本サイクルで送信されるメッセージを形成する(図4参照)。従って1つの周期を構成することができる。例えば、ZF1aおよびZF4aにおける送信マトリックスA: 401, 407, 413, 419, 425, 431, 437, 443および404, 410, 416, 422, 428, 434, 440, 446。1つの送信グループの中で、(1つのタイムウィンドウの)1つのメッセージオブジェクトを何回も送信できる。1つの送信グループ内における1つのメッセージの周期は、 $2^1$ でなければならず、この場合 $1 - m$ である。

40

【0036】

メッセージオブジェクトもしくはメッセージは、バスのメッセージオブジェクト、特にCANにおけるバスのメッセージオブジェクトに相当し、識別子もしくはコード、およびデータ自体を含む。TT CANにおいてはメッセージオブジェクトが、少なくとも次に挙げるもののうちの1つを、好ましくは3つ全てを、送信マトリックスの中で補完される: タイムウィンドウ、基本マーク、繰返し率。

【0037】

50

タイムウィンドウは、基本サイクル（ $BZ_n$ ，送信マトリックスの行）におけるポジション（ $ZF_0$ ， $ZF_{1a} \sim ZF_{5a}$ ）である。タイムウィンドウの開始部は、特定のタイミングマークに達することにより定義される。

【0038】

基本マークは、全体サイクル中のいずれの基本サイクル（ $BZ_{0a} \sim BZ_{7a}$ ）で、メッセージが最初に送信されるかを表示する。

【0039】

繰り返し率は、基本サイクルがいくつ経過した後に、この伝送が繰り返されるかを定義する。

【0040】

CAN コントローラーに対するメッセージオブジェクトの有効性を表示するために、オブジェクトの持続的なリリースを意味する「持続的送信リクエスト」と（排他的成分については下記参照）、オブジェクトが1回だけ有効であることを意味する「個別的送信リクエスト」とがある（アービトレーション成分については下記参照）。

【0041】

CANからの自動的再伝送は、TTCANにおけるメッセージに対しては遮断するのが実用的である。

【0042】

他にここでは、もう一度以下のメッセージ伝送について、すなわち基本サイクルもしくは全体サイクルにおける周期的メッセージおよび自発的メッセージについて、特にアプリケーションに関して説明する。この場合も、排他的メッセージすなわち周期的メッセージと、アービトレーション的すなわち自発的メッセージとを区別する。

【0043】

排他的メッセージオブジェクト（周期的メッセージ）

排他的メッセージが送信されるのは、アプリケーションウォッチドッグが設定され、CAN コントローラーに対するアプリケーションの「持続的な送信要求」が設定され、付随するタイムウィンドウの送信リリース区間が開かれている場合である。この場合、メッセージオブジェクトに対するタイミングマークは、相対時間と一致する。持続的送信要求は、アプリケーション自体によって取り消されるまで、設定されたままとなる。

【0044】

アービトレーションのメッセージオブジェクト（自発的メッセージ）

アービトレーションのメッセージオブジェクトが送信されるのは、アプリケーションウォッチドッグが設定され、CAN コントローラーに対するアプリケーションの「持続的な送信要求」が設定され、そのために特定された次のタイムウィンドウの送信リリース区間が開かれている場合である。この場合に、かかるタイムウィンドウに対するタイミングマークは相対時間に等しい。送信要求は、CAN コントローラーによる送信が成功した後に取り消される。さまざまな自発的メッセージの同時アクセスは、CANのビット・アービトレーションを介して調整される。ある自発的メッセージがある1つのタイムウィンドウでほかの自発的メッセージに敗れた場合、敗れたメッセージは、結果的に特定された次のタイムウィンドウで初めて、再びバスアクセスを巡って争うことができる。

【0045】

送信マトリックス全体もしくは全体サイクルが実行されると、周期的に時間制御されたメッセージ伝送が得られる。時間制御されるとは、いずれのアクションも、ある特定の時点に到達してから開始するということの意味する（タイミングマークおよび相対時間の項を参照）。全体サイクルが完全に実行された場合、すなわち全ての基本サイクルがひとたび処理された場合、再び送信マトリックスの最初の基本サイクルを開始できる。その過渡状態において、時間的空隙を生じることはない。タイマーによるこのような時間制御された通信システムの概要を、発明の詳細な説明と図面により、以上の通り説明した。

【図面の簡単な説明】

【図1】 複数の加入者を含むバスシステムを図式的に示す。

10

20

30

40

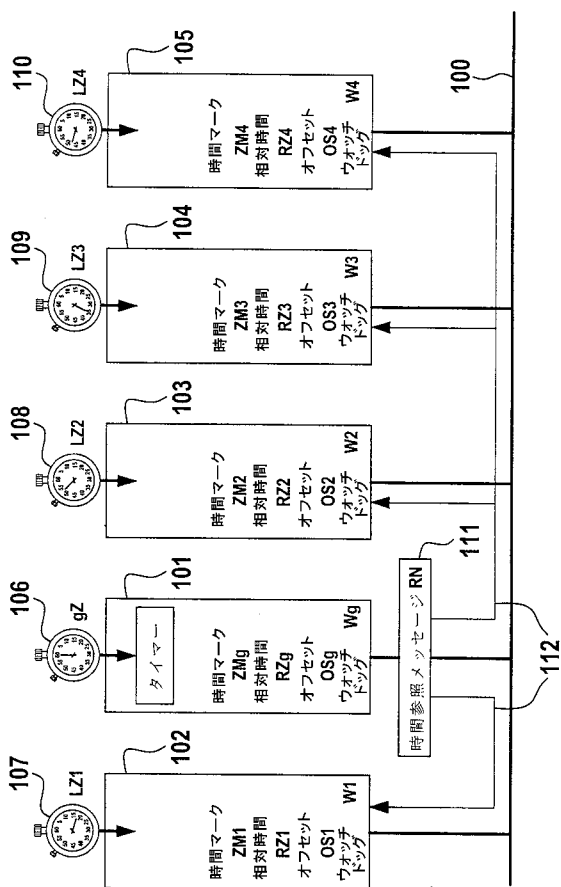
50

【図2】 第1のサイクルまたは基本サイクル，および第2のサイクル，全サイクルの原理的な経過を経時的に示す。

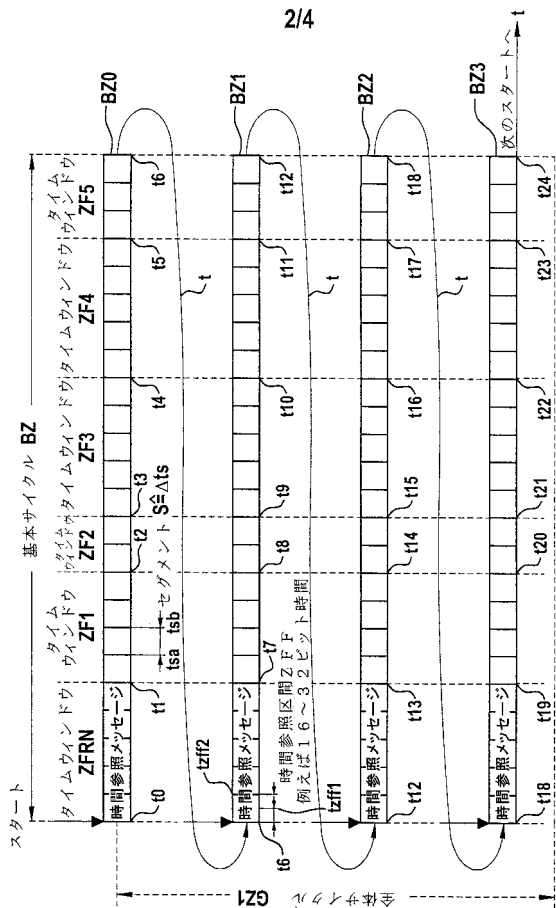
【図3】 タイムウィンドウの構造とメッセージレイアウトの詳細を示す。

【図4】 7つの基本サイクルと，メッセージおよびアービトレーションメッセージのさまざまな送信グループとを含む全体サイクルを示す。

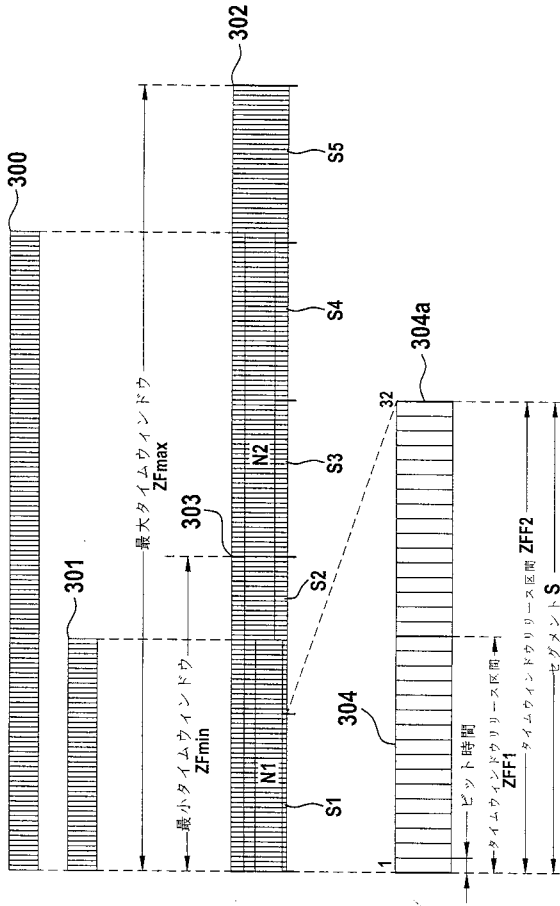
【図1】



【図2】



【図3】



【図4】

基本サイクル BZ (マトリックス中の行)

タイムウィンドウ (マトリックス中の列)	ZF1a	ZF2a	ZF3a	ZF4a	ZF5a
時間参照メッセージ 400	A	402 E	403	404	A
時間参照メッセージ 406	A	408 E	409	410	A
時間参照メッセージ 412	A	414 E	415	416	A
時間参照メッセージ 418	A	420 E	421	422	A
時間参照メッセージ 424	A	426 E	427	428	A
時間参照メッセージ 430	A	432 E	433	434	F
時間参照メッセージ 436	A	438 E	439	440	A
時間参照メッセージ 442	A	444 E	445	446	A

送信グループ (マトリックス中の列)

送信グループ	ZF1a	ZF2a	ZF3a	ZF4a	ZF5a
BZ0a	401	402 E	403	404	405
BZ1a	407	408 E	409	410	411
BZ2a	413	414 E	415	416	417
BZ3a	419	420 E	421	422	423
BZ4a	425	426 E	427	428	429
BZ5a	431	432 E	433	434	435
BZ6a	437	438 E	439	440	441
BZ7a	443	444 E	445	446	447

A ~ F: 送信グループ  
アービトラリオン: アービトラリオンメッセージのため確保された

## フロントページの続き

- (72)発明者 ヴァイグル, アンドレーアス  
ドイツ連邦共和国 7 6 3 5 1 リンケンハイム - ホークステッテン ワーナー - フォン - シーメ  
ンス - シュトラーセ 1 0
- (72)発明者 フェーラー, トーマス  
ドイツ連邦共和国 7 0 8 3 9 ゲルリンゲン パッベルヴェーク 6
- (72)発明者 ムエラー, ベルント  
ドイツ連邦共和国 7 1 2 2 9 リオンバーグ ユーゲン - ヒーゲル - ヴェーク 1 9
- (72)発明者 ハルトウィッヒ, フローリアン  
ドイツ連邦共和国 7 2 7 6 2 ロイトリンゲン レルヒェンシュトラーセ 1 7 / 1
- (72)発明者 ヒューゲル, ローベルト  
ドイツ連邦共和国 7 6 1 9 9 カールスルーエ ジョセフ - フォン - アイヒェンドルフ - シュト  
ラーセ 9

審査官 岩田 玲彦

(56)参考文献 国際公開第9 8 / 0 5 4 8 7 2 (WO, A 1)

(58)調査した分野(Int.Cl., D B名)

H04L 12/40

G06F 13/372