

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6020092号
(P6020092)

(45) 発行日 平成28年11月2日(2016.11.2)

(24) 登録日 平成28年10月14日(2016.10.14)

(51) Int. Cl. F I
G06F 1/32 (2006.01) G06F 1/32 Z
H04M 1/73 (2006.01) H04M 1/73

請求項の数 9 (全 32 頁)

(21) 出願番号	特願2012-260627 (P2012-260627)	(73) 特許権者	000002185
(22) 出願日	平成24年11月29日(2012.11.29)		ソニー株式会社
(65) 公開番号	特開2014-106840 (P2014-106840A)		東京都港区港南1丁目7番1号
(43) 公開日	平成26年6月9日(2014.6.9)	(74) 代理人	100082131
審査請求日	平成27年1月30日(2015.1.30)		弁理士 稲本 義雄
		(74) 代理人	100121131
			弁理士 西川 孝
		(72) 発明者	須賀 佳孝
			東京都港区港南1丁目7番1号 ソニー株式会社社内
		審査官	宮下 誠

最終頁に続く

(54) 【発明の名称】 データ処理装置、データ処理方法、及び、プログラム

(57) 【特許請求の範囲】

【請求項 1】

物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、所定の基準周期の所定数倍となるタイミングで、かつ、複数のサンプリング周期それぞれのタイミングで、前記センサデータを取得する取得制御を行う取得制御部と、

前記サンプリング周期を表す情報としてインデクスの系列を採用し、複数の前記センサデータを取得したタイミングにおいて、そのタイミングが該当する前記サンプリング周期を表す周期情報を、複数の前記インデクスの系列を圧縮して生成する生成部と、

前記センサデータを、前記周期情報とともに出力する出力部と
を備えるデータ処理装置。

【請求項 2】

前記取得制御部は、所定の基準周期の、所定の整数のべき乗倍の前記サンプリング周期のタイミングで、前記センサデータを取得する取得制御を行う

請求項 1 に記載のデータ処理装置。

【請求項 3】

前記取得制御部は、所定の基準周期の、2 のべき乗倍の前記サンプリング周期のタイミングで、前記センサデータを取得する取得制御を行う

請求項 2 に記載のデータ処理装置。

【請求項 4】

前記生成部は、前記センサデータを取得したタイミングが該当する 1 つ以上の前記サン

プリング周期のうちの最大のサンプリング周期を表す情報を、前記センサデータとともに出力する前記周期情報として生成する

請求項 3 に記載のデータ処理装置。

【請求項 5】

前記サンプリング周期を表す情報は、1 ビットだけがたったビット列であり、

前記生成部は、前記センサデータを取得したタイミングが該当する 1 つ以上の前記サンプリング周期それぞれを表す前記ビット列の論理和を、前記センサデータとともに出力する前記周期情報として生成する

請求項 1 に記載のデータ処理装置。

【請求項 6】

前記取得制御部は、前記センサデータを利用するアプリケーションが要求する前記サンプリング周期のタイミングで、前記センサデータを取得する取得制御を行う

請求項 3 に記載のデータ処理装置。

【請求項 7】

前記センサデータと前記周期情報とを、前記センサデータを要求するアプリケーションを実行するプロセッサに提供する

請求項 3 に記載のデータ処理装置。

【請求項 8】

物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、所定の基準周期の所定数倍となるタイミングで、かつ、複数のサンプリング周期それぞれのタイ

ミングで、前記センサデータを取得し、前記サンプリング周期を表す情報としてインデクスの系列を採用し、複数の前記センサデータを取得したタイミングにおいて、そのタイミングが該当する前記サンプリング周期を表す周期情報を、複数の前記インデクスの系列を圧縮して生成し、

前記センサデータを、前記周期情報とともに出力する

ステップを含むデータ処理方法。

【請求項 9】

物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、所定の基準周期の所定数倍となるタイミングで、かつ、複数のサンプリング周期それぞれのタイ

ミングで、前記センサデータを取得する取得制御を行う取得制御部と、前記サンプリング周期を表す情報としてインデクスの系列を採用し、複数の前記センサデータを取得したタイミングにおいて、そのタイミングが該当する前記サンプリング周期を表す周期情報を、複数の前記インデクスの系列を圧縮して生成する生成部と、

前記センサデータを、前記周期情報とともに出力する出力部と

して、コンピュータを機能させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本技術は、データ処理装置、データ処理方法、及び、プログラムに関し、特に、例えば、負荷を軽減し、低消費電力化を図ることができるようにするデータ処理装置、データ処理方法、及び、プログラムに関する。

【背景技術】

【0002】

近年、MEMS(Micro-Electro-Mechanical Systems)技術に代表されるセンサ技術が進化し、各種の物理量をセンシングするセンサの小型化、及び、低価格化が進んでいる。かかるセンサの小型化、及び、低価格化に伴い、例えば、デジタルカメラやスマートフォン等の携帯端末には、各種のアプリケーションで利用される多数のセンサが搭載されるようになってきている。

【0003】

例えば、デジタルカメラでは、ジャイロセンサが搭載され、そのジャイロセンサを利

10

20

30

40

50

用して、撮影時の手ぶれ補正が行われる。近年のデジタルカメラでは、さらに、地磁気センサや加速度センサが搭載され、デジタルカメラの位置や姿勢、撮影の方位等を、デジタルカメラで撮影した写真（画像）のメタデータとして記録することが行われる。

【0004】

また、例えば、スマートフォンでは、各種のセンサが、スマートフォンの挙動等に応じて様々な応答を行うゲームその他のアプリケーションに利用されている。

【0005】

ここで、携帯端末に搭載されるセンサとしては、上述したセンサの他、例えば、気圧センサや、照度センサ、近接センサ等がある。

【0006】

なお、特許文献1では、センサが物理量のセンシングの結果出力する、その物理量を表すセンサデータを収集するプログラムの負荷を軽減するためのセンサデータ収集方法が提案されている。

【先行技術文献】

【特許文献】

【0007】

【特許文献1】特許第4673250号明細書

【発明の概要】

【発明が解決しようとする課題】

【0008】

携帯端末は、プログラムを実行し、各種の処理を行うCPU(Central Processing Unit)を有するが、携帯端末に搭載されるセンサは、CPUに接続され、CPUは、センサからセンサデータを取得するために、センサとの間で、頻繁に通信を行う。

【0009】

センサは、高速な通信手段を有しないため、CPUにおいて、センサから所望のセンサデータを取得するためには、センサに対して、複数回の低速なアクセスを行う必要があり、CPUは、比較的長い時間、センサデータの取得に占有される。

【0010】

近年では、携帯端末において、比較的負荷の重い画像処理や、音声処理、通信処理等を行うために、CPUは高性能化しており、かかる高性能なCPUが、センサデータの取得という単純な処理に、長時間占有されることは、効率的であるとは言い難い。

【0011】

そして、高性能なCPUは、消費電力が大きいので、センサデータの取得のために、多くの電力が消費され、結果として、携帯端末のバッテリーの持ち時間が短くなる。

【0012】

さらに、例えば、CPUが、リアルタイム性が要求される処理を行っている場合に、センサデータを取得する処理の割り込みが頻繁に行われることは、CPUの負荷が大になり、携帯端末全体のパフォーマンスを低下させることになる。

【0013】

本技術は、このような状況に鑑みてなされたものであり、センサデータの取得について、負荷を軽減し、低消費電力化を図ることができるようにするものである。

【課題を解決するための手段】

【0014】

物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、所定の基準周期の所定数倍となるタイミングで、かつ、複数のサンプリング周期それぞれのタイミングで、前記センサデータを取得する取得制御を行う取得制御部と、前記サンプリング周期を表す情報としてインデクスの系列を採用し、複数の前記センサデータを取得したタイミングにおいて、そのタイミングが該当する前記サンプリング周期を表す周期情報を、複数の前記インデクスの系列を圧縮して生成する生成部と、前記センサデータを、前記周期情報とともに出力する出力部とを備えるデータ処理装置、又は、そのようなデータ処理装

10

20

30

40

50

置として、コンピュータを機能させるためのプログラムである。

【0015】

本技術の一側面のデータ処理方法は、物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、所定の基準周期の所定数倍となるタイミングで、かつ、複数のサンプリング周期それぞれのタイミングで、前記センサデータを取得し、前記サンプリング周期を表す情報としてインデクスの系列を採用し、複数の前記センサデータを取得したタイミングにおいて、そのタイミングが該当する前記サンプリング周期を表す周期情報を、複数の前記インデクスの系列を圧縮して生成し、前記センサデータを、前記周期情報とともに出力するステップを含むデータ処理方法である。

【0016】

以上のような一側面においては、物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、所定の基準周期の所定数倍となるタイミングで、かつ、複数のサンプリング周期それぞれのタイミングで、前記センサデータが取得される。そして、前記サンプリング周期を表す情報としてインデクスの系列を採用され、複数の前記センサデータを取得したタイミングにおいて、そのタイミングが該当する前記サンプリング周期を表す周期情報が、複数の前記インデクスの系列を圧縮して生成されて、前記センサデータが、前記周期情報とともに出力される。

【0017】

なお、データ処理装置は、独立した装置であっても良いし、1つの装置を構成している内部ブロックであっても良い。

【0018】

また、プログラムは、伝送媒体を介して伝送することにより、又は、記録媒体に記録して提供することができる。

【発明の効果】

【0019】

本技術の一側面によれば、負荷を軽減し、低消費電力化を図ることができる

【図面の簡単な説明】

【0020】

【図1】メインCPUがセンサデータを取得する携帯端末の構成例を示すブロック図である。

【図2】携帯端末において、メインCPU 20が、センサからセンサデータを取得し、アプリケーションに提供する処理を説明する図である。

【図3】サブCPUがセンサデータを取得する携帯端末の構成例を示すブロック図である。

【図4】携帯端末において、サブCPU 30が、センサからセンサデータを取得し、メインCPU 20を介してアプリケーションに提供する処理を説明する図である。

【図5】サンプリング周期が、1Tないし6Tのセンサデータを取得（サンプリング）するタイミングを示す図である。

【図6】サブCPU 30が、センサ 11_nからのセンサデータをサンプリングするタイミングを示す図である。

【図7】本技術のデータ処理装置を適用した携帯端末の一実施の形態の構成例を示すブロック図である。

【図8】サンプリング周期のインデクスと、周期情報とを説明する図である。

【図9】携帯端末の処理を説明する図である。

【図10】サブCPU 40の第1のハードウェア構成例を示すブロック図である。

【図11】管理テーブルを示す図である。

【図12】CPUコア 42がセンサデータ提供プログラムを実行することによりサブCPU 40で行われるセンサデータ提供処理を説明するフローチャートである。

【図13】タイマ割り込み処理を説明するフローチャートである。

【図14】サンプリング周期のインデクスと、周期情報とを説明する図である。

【図15】サブCPU 40の第2のハードウェア構成例を示すブロック図である。

10

20

30

40

50

【発明を実施するための形態】

【0021】

<メインCPUがセンサデータを取得する携帯端末>

【0022】

図1は、メインCPUがセンサデータを取得する携帯端末の構成例を示すブロック図である。

【0023】

図1において、携帯端末は、センサ部10、及び、メインCPU20を有する。

【0024】

センサ部10は、1個以上のN個のセンサ11₁ないし11_Nを有する。センサ11_nは (n=1,2,・・・,N)、所定の物理量をセンシングし、その物理量を表すセンサデータを出力する。

10

【0025】

ここで、図1では、センサ11₁として、加速度センサが、センサ11₂として、ジャイロセンサが、センサ11₃として、地磁気センサが、それぞれ採用されている。また、センサ11_Nとして、気圧センサが採用されている。

【0026】

メインCPU20は、後述するサブCPUに比較して、処理速度が速い高性能なCPUであり、OS(Operating System)を実行し、さらに、ゲームその他の様々なアプリケーションを実行する。

20

【0027】

また、図1において、メインCPU20は、センサ部10に(直接)接続されており、必要なセンサ11_nからセンサデータを取得し、必要なアプリケーションに提供する。

【0028】

ここで、メインCPU20が実行するアプリケーションとしては、様々なアプリケーションがある。すなわち、メインCPU20が実行するアプリケーションとしては、例えば、携帯端末の姿勢を検出し、画面(の長手方向)が横又は縦のうちのいずれになっているかによって、画面の縦横表示(画像の左右方向を、画面の長手方向、又は、長手方向と直交する方向に一致させる表示)を切り替えるアプリケーション(以下、第1のアプリケーションともいう)がある。

30

【0029】

また、メインCPU20が実行するアプリケーションとしては、例えば、携帯端末の画面に表示される地図の向きを実際の向きに一致させるために、ユーザが所持している携帯端末が、いずれの方角を向いているかを計測する電子コンパスのアプリケーション(以下、第2のアプリケーションともいう)がある。

【0030】

さらに、メインCPU20が実行するアプリケーションとしては、例えば、携帯端末を所持しているユーザの歩数を計測する歩数計のアプリケーション(以下、第3のアプリケーションともいう)がある。

【0031】

40

第1ないし第3のアプリケーションは、いずれも、例えば、加速度センサとしてのセンサ11₁が出力するセンサデータを利用する。そのため、第1ないし第3のアプリケーションは、メインCPU20に対して、加速度センサとしてのセンサ11₁が出力するセンサデータを要求する。

【0032】

メインCPU20は、第1ないし第3のアプリケーションを実行している場合、第1ないし第3のアプリケーションからの要求に応じ、センサ11₁から、センサデータを取得して、第1ないし第3のアプリケーションそれぞれに提供(供給)する。

【0033】

ここで、画面の縦横表示を切り替える第1のアプリケーションは、例えば、5Hz程度の

50

周期（周波数）で、加速度センサとしてのセンサ 1 1₁ のセンサデータを必要とする（利用する）。また、電子コンパスの第 2 のアプリケーションは、例えば、10Hz 程度の周期で、センサ 1 1₁ のセンサデータを必要とし、歩数計の第 3 のアプリケーションは、例えば、20Hz 程度の周期で、センサ 1 1₁ のセンサデータを必要とする。

【 0 0 3 4 】

以上のように、アプリケーションが必要とするセンサデータの周期（以下、サンプリング周期ともいう）は、異なっている場合がある。

【 0 0 3 5 】

なお、以下では、説明を簡単にするために、センサ 1 1₁ ないし 1 1_N のうちの 1 個のセンサ 1 1_n からセンサデータを取得することとする。

【 0 0 3 6 】

図 2 は、図 1 の携帯端末において、メイン CPU 2 0 が、センサ 1 1_n からセンサデータを取得し、アプリケーションに提供する処理を説明する図である。

【 0 0 3 7 】

図 2 では、メイン CPU 2 0 において、外部のデバイスとの間のやりとりを行うためのデバイスドライバや、入出力の管理のためのユーティリティ（プログラム）である I/O (Input/Output) と、リソース管理その他の各種の制御を行う OS とが動作している（実行されている）。

【 0 0 3 8 】

さらに、図 2 では、メイン CPU 2 0 において、OS の管理の下、1 以上のアプリケーションとしての 3 つのアプリケーション #1、#2、及び、#3 が実行されている（動作している）。

【 0 0 3 9 】

いま、基準となるサンプリング周期（以下、基準周期ともいう）を、 T と表すとともに、アプリケーションが必要とするセンサデータのサンプリング周期が、基準周期の整数倍で表されることとする。

【 0 0 4 0 】

なお、基準周期としては、例えば、センサ 1 1_n やメイン CPU 2 0 の動作クロックの周期や、その周期の整数倍等を採用することができる。

【 0 0 4 1 】

図 2 では、アプリケーション #1 ないし #3 は、サンプリング周期が、それぞれ、 $3T$ 、 $4T$ 、及び、 $5T$ のセンサデータを必要としており、そのようなサンプリング周期のセンサデータを、OS に要求している。

【 0 0 4 2 】

OS は、アプリケーション #1 ないし #3 の要求に応じて、 $3T$ 、 $4T$ 、及び、 $5T$ のサンプリング周期のタイミングで、センサ 1 1_n からセンサデータを取得し、 $3T$ 、 $4T$ 、及び、 $5T$ のサンプリング周期のセンサデータを、アプリケーション #1 ないし #3 に、それぞれ提供する。

【 0 0 4 3 】

ここで、以下、サンプリング周期が、 $k \times T$ (k は正の整数) のセンサデータを、 $k \times T$ データともいう。

【 0 0 4 4 】

アプリケーション # i に提供されるセンサデータにジッタが生じ、センサデータの間隔（サンプリング周期）にばらつきがあると、アプリケーション # i で行われる、センサデータに対する、ローパスフィルタによるフィルタリング等の波形処理に不都合が生じることがある。

【 0 0 4 5 】

そのため、OS は、アプリケーション # i が要求するサンプリング周期のタイミングで、センサ 1 1_n から、センサデータを取得して、アプリケーション # i に提供する。

【 0 0 4 6 】

図 2 では、OS は、アプリケーション #1 ないし #3 から、それぞれ、サンプリング周期が $3T$ 、 $4T$ 、及び、 $5T$ のセンサデータの要求を受けているので、 $3T$ のタイミング（ $3T$ ごとの時刻

10

20

30

40

50

)、4Tのタイミング、及び、5Tのタイミングのそれぞれで、センサ11_nから、センサデータを取得する。

【0047】

そして、OSは、3Tのタイミングでセンサ11_nから取得したセンサデータを、アプリケーション#1に提供し、4Tのタイミングでセンサ11_nから取得したセンサデータを、アプリケーション#2に提供する。また、OSは、5Tのタイミングでセンサ11_nから取得したセンサデータを、アプリケーション#3に提供する。

【0048】

図1及び図2に示したように、メインCPU20が、センサ11_nからセンサデータを取得する場合には、センサ11_nからセンサデータを取得するタイミングは、メインCPU20で管理される。

10

【0049】

したがって、メインCPU20は、センサ11_nからセンサデータを取得したタイミングが、いずれのサンプリング周期のタイミングに該当するかを認識しているため、センサ11_nから所定のサンプリング周期のタイミングで取得したセンサデータを、そのサンプリング周期のセンサデータを要求したアプリケーションに提供することができる。

【0050】

但し、メインCPU20は、様々なアプリケーションを実行するため、上述のように、メインCPU20としては、高性能なCPUが採用される。そして、図1及び図2では、かかる高性能なメインCPU20が、センサデータの取得を行っているために、メインCPU20の負荷が大になり、その結果、携帯端末全体の消費電力が大になる。

20

【0051】

<サブCPUがセンサデータを取得する携帯端末>

【0052】

図3は、サブCPUがセンサデータを取得する携帯端末の構成例を示すブロック図である。

【0053】

なお、図中、図1の場合と対応する部分については、同一の符号を付してあり、以下では、その説明は、適宜省略する。

【0054】

図3の携帯端末は、センサ部10、及び、メインCPU20を有する点で、図1の場合と共通する。但し、図3の携帯端末は、サブCPU30が新たに設けられている点で、図1の場合と相違する。

30

【0055】

図3では、センサ部10とメインCPU20との間に、サブCPU30が設けられている。そして、センサ部10は、サブCPU30に(直接)接続されており、したがって、メインCPU20は、センサ部10とは、(直接)接続されていない。

【0056】

サブCPU30は、様々なアプリケーションを実行するメインCPU20よりも処理速度等の性能は落ちる(悪い)が、消費電力が小さいCPUであり、センサデータの取得を担当する。

40

【0057】

すなわち、図3では、サブCPU30は、メインCPU20(が実行するアプリケーション)からの要求に応じて、必要なセンサ11_nからセンサデータを取得し、メインCPU20、ひいては、センサデータを要求してきたアプリケーションに提供する。

【0058】

したがって、図3では、メインCPU20は、センサ11_nからセンサデータを取得する処理を行わずに済むので、図1及び図2の場合に比較して、メインCPU20の負荷を軽減することができる。さらに、センサデータの取得が、消費電力が大のメインCPU20ではなく、消費電力が小さいサブCPU30で行われるので、携帯端末全体の低消費電力化を図

50

ることができる。

【0059】

図4は、図3の携帯端末において、サブCPU30が、センサ11_nからセンサデータを取得し、メインCPU20を介してアプリケーションに提供する処理を説明する図である。

【0060】

メインCPU20では、図2の場合と同様に、I/O及びOSが動作しており、さらに、OSの管理の下、3つのアプリケーション#1ないし#3が実行されている。

【0061】

アプリケーション#1ないし#3は、図2の場合と同様に、サンプリング周期が、それぞれ、3T、4T、及び、5Tのセンサデータを必要とし、そのようなサンプリング周期のセンサデータを、OSに要求する。

10

【0062】

OSは、アプリケーション#1ないし#3それぞれからのセンサデータの要求に応じ、I/Oを介して、サブCPU30に、同様の要求を行う。

【0063】

サブCPU30は、OSからのセンサデータの要求、すなわち、アプリケーション#1ないし#3それぞれからのセンサデータの要求に応じて、3T、4T、及び、5Tのサンプリング周期のタイミングで、センサ11_nからセンサデータを取得し、3T、4T、及び、5Tのサンプリング周期のセンサデータを、メインCPU20の(I/O及び)OSを介して、アプリケーション#1ないし#3に、それぞれ提供する。

20

【0064】

図3及び図4では、センサデータを要求するアプリケーションを実行するメインCPU20ではなく、センサデータの取得を、いわば専属で行うサブCPU30が、センサ11_nからセンサデータを取得する。

【0065】

サブCPU30は、メインCPU20に比較して性能は劣るが消費電力が小であるため、メインCPU20がセンサデータの取得を行う場合に比較して、低消費電力化を図ることができる。

【0066】

また、メインCPU20は、通信速度の遅いセンサ11_nに(直接)アクセスせずに済むので、負荷が軽減され、画像処理や、音声処理、通信処理、その他、滑らかなユーザーフェースを実現するための高度な処理等に、いわば専念することができる。

30

【0067】

以上のように、図3及び図4の携帯端末では、アプリケーション#1ないし#3を実行するメインCPU20ではなく、サブCPU30が、アプリケーション#1ないし#3それぞれから要求されるセンサデータを取得する。そして、サブCPU30は、センサ11_nから取得したセンサデータを、メインCPU20に供給し、メインCPU20において、センサデータが、アプリケーション#1ないし#3に提供される。

【0068】

この場合、センサ11_nからセンサデータを取得するタイミングは、サブCPU30で管理され、メインCPU20は関与しない。そのため、メインCPU20は、サブCPU30から供給されるセンサデータが、いずれのサンプリング周期のタイミングで取得されたセンサデータであるかを認識することができず、サブCPU30からのセンサデータを、どのアプリケーションに提供すべきであるのかが分からないことがあり得る。

40

【0069】

図5は、サンプリング周期が、1T、2T、3T、4T、5T、及び、6Tのセンサデータを取得(サンプリング)するタイミングを示す図である。

【0070】

図5において、白抜きの矢印が、1Tデータ(サンプリング周期が1Tのセンサデータ)、2Tデータ、3Tデータ、4Tデータ、5Tデータ、及び、6Tデータのそれぞれをサンプリング(

50

取得)するタイミングを表している。

【0071】

1Tデータについては、時間(サンプリング周期)1T=Tごとのタイミングで、センサ11_nから、センサデータがサンプリングされる。同様に、k×Tデータについては、時間k×Tごとのタイミングで、センサ11_nから、センサデータがサンプリングされる。

【0072】

図6は、サブCPU30が、センサ11_nからのセンサデータをサンプリングするタイミングを示す図である。

【0073】

すなわち、図6は、図4に示したように、アプリケーション#1ないし#3から、それぞれ、3Tデータ、4Tデータ、及び、5Tデータ(サンプリング周期が、それぞれ、3T、4T、及び、5Tのセンサデータ)の要求があった場合の、サブCPU30が、センサ11_nからのセンサデータをサンプリングするタイミングを示している。

10

【0074】

サブCPU30は、例えば、3T、4T、及び、5Tのサンプリング周期それぞれのタイミング(を表す信号)を生成し、そのタイミングで、センサ11_nから、センサデータをサンプリングする。

【0075】

図6では、センサ11_nからセンサデータがサンプリングされるタイミングが、斜線を付した矢印で表されている。

20

【0076】

図6に示すように、3T、4T、及び、5Tのサンプリング周期それぞれのタイミングは、一致する場合もあれば、一致しない場合もある。

【0077】

したがって、サブCPU30がセンサ11_nからサンプリングしたセンサデータは、3Tデータ、4Tデータ、及び、5Tデータのうちのいずれか1つのデータに該当する場合の他、3Tデータ、4Tデータ、及び、5Tデータのうちのいずれか2つのデータに該当する場合、3Tデータ、4Tデータ、及び、5Tデータのすべてに該当する場合がある。

【0078】

サブCPU30は、センサ11_nからセンサデータをサンプリングすると、そのセンサデータを、メインCPU20に供給する。

30

【0079】

サブCPU30は、3T、4T、及び、5Tのサンプリング周期のうちの、少なくとも1つのサンプリング周期のタイミングに該当すれば、センサ11_nからセンサデータをサンプリングして、メインCPU20に供給する。

【0080】

そのため、サブCPU30からメインCPU20に対して、センサデータが供給されるタイミングは、図6に黒塗りの矢印で示すように、一見、ランダムなタイミングになる。

【0081】

メインCPU20では、以上のように、サブCPU30から、一見、ランダムなタイミングで供給されるセンサデータについて、そのセンサデータを提供すべき(1以上の)アプリケーションを認識する必要がある。

40

【0082】

ここで、図6では(後述する図8でも同様)、1Tないし6Tのサンプリング周期のタイミングすべてが一致する、ある時刻を、時刻 t_0 として、時刻 t_0 から、基準周期Tのm倍の時間だけ経過した時刻を、時刻 t_m と表してある。

【0083】

図6において、3T、4T、及び、5Tのサンプリング周期のうちの、少なくとも1つのサンプリング周期のタイミングに該当する時刻は、時刻 $t_0, t_3, t_4, t_5, t_6, t_8, t_9, t_{10}, t_{12}, \dots$ である。

50

【 0 0 8 4 】

そして、3T、4T、及び、5Tのサンプリング周期については、例えば、時刻 t_0 は、3T、4T、及び、5Tのサンプリング周期のタイミングのすべてに該当する。

【 0 0 8 5 】

また、例えば、時刻 t_3 、 t_6 、及び、 t_9 は、3Tのサンプリング周期のタイミングに、時刻 t_4 及び t_8 は、4Tのサンプリング周期のタイミングに、時刻 t_5 及び t_{10} は、5Tのサンプリング周期のタイミングに、それぞれ該当する。

【 0 0 8 6 】

さらに、例えば、時刻 t_{12} は、3T及び4Tのサンプリング周期それぞれのタイミングに該当する。

【 0 0 8 7 】

メインCPU 2 0 は、3Tのサンプリング周期のタイミングに該当する時刻にサブCPU 3 0 から供給されたセンサデータを、3Tのサンプリング周期のセンサデータを要求したアプリケーション#1に提供しなければならない。

【 0 0 8 8 】

同様に、メインCPU 2 0 は、4Tのサンプリング周期のタイミングに該当する時刻にサブCPU 3 0 から供給されたセンサデータを、4Tのサンプリング周期のセンサデータを要求したアプリケーション#2に提供しなければならない。5Tのサンプリング周期のタイミングに該当する時刻にサブCPU 3 0 から供給されたセンサデータを、5Tのサンプリング周期のセンサデータを要求したアプリケーション#3に提供しなければならない。

【 0 0 8 9 】

しかしながら、メインCPU 2 0 では、サブCPU 3 0 から供給されるセンサデータだけでは、そのセンサデータが、いずれのサンプリング周期のタイミングでサンプリングされたセンサデータであるかを、即座に認識することが困難である。そのため、メインCPU 2 0 は、サブCPU 3 0 からのセンサデータを、どのアプリケーションに提供すべきであるのかが分からないことがある。

【 0 0 9 0 】

メインCPU 2 0 は、アプリケーション#1ないし#3が、それぞれ、3Tデータ、4Tデータ、及び、5Tデータを要求していることを認識しているので、サブCPU 3 0 からセンサデータが供給されるタイミングを、ある程度の期間に亘って観測することにより、どのタイミングで供給されるセンサデータが、3T、4T、及び、5Tのうちのどのサンプリング周期のセンサデータであるのかを推定するセンサデータ推定を行うことができる。

【 0 0 9 1 】

しかしながら、センサデータ推定を行う場合には、メインCPU 2 0 において、センサデータ推定の推定結果が得られるまでは、センサデータを要求したアプリケーションに対して、センサデータを提供することができない。したがって、アプリケーションは、起動後に、即座に、センサデータを利用することができない。

【 0 0 9 2 】

メインCPU 2 0 において、サブCPU 3 0 から供給されるセンサデータが、いずれのサンプリング周期のタイミングでサンプリングされたセンサデータであるかを、即座に認識する方法としては、例えば、サブCPU 3 0 において、基準周期Tのタイミングで、センサ 1 1_n からセンサデータをサンプリングし、メインCPU 2 0 に供給する方法がある。

【 0 0 9 3 】

この場合、サブCPU 3 0 からメインCPU 2 0 に対しては、基準周期Tのタイミングで、センサデータが供給されるので、メインCPU 2 0 は、センサ 1 1_n がメインCPU 2 0 に直接接続された図 1 及び図 2 の場合と同様に、サブCPU 3 0 からのセンサデータが、いずれのサンプリング周期のタイミングでサンプリングされたセンサデータであるかを認識することができる。

【 0 0 9 4 】

しかしながら、この場合、サブCPU 3 0 では、アプリケーションから要求されたサン

10

20

30

40

50

リング周期のセンサデータその他、アプリケーションから要求されていないサンプリング周期のセンサデータもサンプリングされる。さらに、サブCPU 30からメインCPU 20に対しては、アプリケーションから要求されたサンプリング周期のセンサデータその他、アプリケーションから要求されていないサンプリング周期のセンサデータも供給される。

【0095】

したがって、メインCPU 20及びサブCPU 30のいずれにおいても、本来行う必要がない処理が行われるため、負荷が増大し、実質的な処理の効率が低下するとともに、消費電力が増加する。

【0096】

< 本技術を適用した携帯端末の一実施の形態 >

10

【0097】

図7は、本技術のデータ処理装置を適用した携帯端末の一実施の形態の構成例を示すブロック図である。

【0098】

なお、図中、図3の場合と対応する部分については、同一の符号を付してあり、以下では、その説明は、適宜省略する。

【0099】

図7の携帯端末は、センサ部10、及び、メインCPU 20を有する点で、図3の場合と共通する。但し、図7の携帯端末は、サブCPU 30に代えて、サブCPU 40が設けられている点で、図3の場合と相違する。

20

【0100】

サブCPU 40は、図3のサブCPU 30と同様に、様々なアプリケーションを実行するメインCPU 20よりも処理速度等の性能は落ちるが、消費電力が小さいCPUであり、センサデータの取得を、少なくとも担当する。

【0101】

また、図3のサブCPU 30は、メインCPU 20（が実行するアプリケーション）からの要求に応じて、必要なセンサ11_nからセンサデータを取得し、そのセンサデータだけを、メインCPU 20に供給するが、サブCPU 40は、センサデータとともに、そのセンサデータの周期情報も、メインCPU 20に供給する。

【0102】

30

センサデータの周期情報とは、そのセンサデータをサンプリング（取得）したタイミングが該当するサンプリング周期を表す情報、すなわち、センサデータが、どのサンプリング周期のタイミングでサンプリングされたセンサデータであるのかを表す情報である。

【0103】

ここで、サンプリング周期には、そのサンプリング周期を表す識別情報としてのインデクス(index)を割り当て、周期情報は、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデクスを用いて生成することができる。

【0104】

図8は、サンプリング周期のインデクスと、周期情報とを説明する図である。

【0105】

40

すなわち、図8は、1T, 2T, 3T, 4T, 5T、及び、6Tのサンプリング周期それぞれで、センサデータをサンプリングするタイミングを示している。

【0106】

図8において、矢印が、1T, 2T, 3T, 4T, 5T、及び、6Tのサンプリング周期それぞれで、センサデータをサンプリングするタイミングを表している。

【0107】

サンプリング周期のインデクスとしては、例えば、シーケンシャルな整数を採用することができる。

【0108】

また、サンプリング周期のインデクスとしては、例えば、サンプリング周期として採用

50

し得る周期の（総）数に等しいビット列であって、そのビット列のうちの1ビットだけがたったビット列（以下、1ビットオンビット列ともいう）を採用することができる。なお、サンプリング周期として採用し得る周期は、あらかじめ決めておくことができる。

【0109】

例えば、図8に示したように、サンプリング周期として、1T, 2T, 3T, 4T, 5T、及び、6Tの6個の周期を採用し得る場合において、サンプリング周期のインデクスとして、1ビットオンビット列を採用するときには、サンプリング周期のインデクスは、1ビットだけがたった6ビットのビット列となる。

【0110】

図8では、 $k \times T$ （図8では、 $k=1, 2, 3, 4, 5, 6$ ）のサンプリング周期のインデクスとして、最上位ビットから k ビット目がたった6ビットの1ビットオンビット列が採用されている。

10

【0111】

すなわち、図8では、1Tのサンプリング周期のインデクスとして100000が、2Tのサンプリング周期のインデクスとして010000が、3Tのサンプリング周期のインデクスとして001000が、4Tのサンプリング周期のインデクスとして000100が、5Tのサンプリング周期のインデクスとして000010が、6Tのサンプリング周期のインデクスとして000001が、それぞれ採用されている。

【0112】

周期情報としては、例えば、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデクスの系列を採用することができる。

20

【0113】

ここで、例えば、図4に示したように、アプリケーション#1ないし#3から、3Tデータ、4Tデータ、及び、5Tデータ（サンプリング周期が、それぞれ、3T, 4T、及び、5Tのセンサデータ）が要求された場合、サブCPU40は、図6で説明した場合と同様にして、3T, 4T、及び、5Tのサンプリング周期それぞれのタイミング（図8において、斜線を付した矢印で示すタイミング）で、センサ11_nから、センサデータをサンプリングする。

【0114】

3T, 4T、及び、5Tのサンプリング周期については、図6で説明したように、それらのサンプリング周期のうちの、少なくとも1つのサンプリング周期のタイミングに該当する時刻は、時刻 $t_0, t_3, t_4, t_5, t_6, t_8, t_9, t_{10}, t_{12}, \dots$ であり、かかる時刻で、サブCPU40は、センサデータをサンプリングする。

30

【0115】

ここで、例えば、時刻 t_0 は、3T, 4T、及び、5Tのサンプリング周期のタイミングのすべてに該当する。したがって、時刻 t_0 のセンサデータをサンプリングしたタイミングが該当するサンプリング周期は、3T, 4T、及び、5Tである。

【0116】

このように、センサデータをサンプリングしたタイミングが該当するサンプリング周期が、3T, 4T、及び、5Tである場合において、周期情報として、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデクスの系列を採用するときには、周期情報は、例えば、3Tのサンプリング周期のインデクス001000、4Tのサンプリング周期のインデクス000100、及び、5Tのサンプリング周期のインデクス000010を並べた、 $6 \times 3 = 18$ ビットのビット列001000000100000010となる。

40

【0117】

また、例えば、時刻 t_3 は、3Tのサンプリング周期のタイミングに該当し、したがって、時刻 t_3 のセンサデータをサンプリングしたタイミングが該当するサンプリング周期は、3Tである。

【0118】

このように、センサデータをサンプリングしたタイミングが該当するサンプリング周期が、3Tである場合において、周期情報として、センサデータをサンプリングしたタイミン

50

グが該当するサンプリング周期のインデクスの系列を採用するときには、周期情報は、例えば、3Tのサンプリング周期のインデクス001000だけを並べた、 $6 \times 1 = 6$ ビットのビット列001000となる。

【0119】

以上のように、周期情報として、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデクスの系列を採用すると、センサデータをサンプリングしたタイミングが該当するサンプリング周期の数が多いほど、周期情報としてのビット列が大になる。

【0120】

そこで、サンプリング周期のインデクスとして、1ビットオンビット列を採用する場合には、周期情報としては、例えば、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデクスの論理和を採用することができる。

10

【0121】

この場合、例えば、時刻 t_0 でサンプリングされるセンサデータのように、センサデータをサンプリングしたタイミングが該当するサンプリング周期が、3T、4T、及び、5Tであるときには、周期情報は、3Tのサンプリング周期のインデクス001000、4Tのサンプリング周期のインデクス000100、及び、5Tのサンプリング周期のインデクス000010の論理和である6ビットのビット列001110となる。

【0122】

また、例えば、時刻 t_3 でサンプリングされるセンサデータのように、センサデータをサンプリングしたタイミングが該当するサンプリング周期が、3Tであるときには、周期情報は、3Tのサンプリング周期のインデクス001000に等しい6ビットのビット列001000となる。

20

【0123】

アプリケーション#1ないし#3から、3Tデータ、4Tデータ、及び、5Tデータがそれぞれ要求されている場合、サブCPU 40は、3T、4T、及び、5Tのサンプリング周期のうちの、少なくとも1つのサンプリング周期のタイミングに該当すれば、センサ11_nからセンサデータをサンプリングして、メインCPU 20に供給する。

【0124】

この場合、サブCPU 40からメインCPU 20に対して、センサデータが供給されるタイミングは、図8に黒塗りの矢印で示すように、一見、ランダムなタイミングになる。

30

【0125】

但し、サブCPU 40は、図8に示すように、メインCPU 20に対して、センサデータとともに、そのセンサデータの周期情報も供給する。

【0126】

したがって、メインCPU 20では、サブCPU 40から、一見、ランダムなタイミングで供給されるセンサデータについて、そのセンサデータを提供すべき(1以上の)アプリケーションを、周期情報に基づいて、容易に認識することができる。

【0127】

すなわち、図8において、例えば、時刻 t_{12} は、3T及び4Tのサンプリング周期のタイミングに該当するので、時刻 t_{12} において、サブCPU 40は、センサ11_nからセンサデータをサンプリングする。

40

【0128】

時刻 t_{12} において、センサデータをサンプリングしたタイミングが該当するサンプリング周期は、3T及び4Tであるので、サブCPU 40は、3Tのサンプリング周期のインデクス001000と、4Tのサンプリング周期のインデクス000100との論理和である6ビットのビット列001100を、周期情報として生成する。

【0129】

そして、サブCPU 40は、時刻 t_{12} においてサンプリングしたセンサデータとともに、周期情報001100を、メインCPU 20に供給する。

50

【 0 1 3 0 】

この場合、サブCPU 4 0 からの周期情報001100は、最上位ビットから3ビット目と4ビット目がたっているので、メインCPU 2 0 は、サブCPU 4 0 からのセンサデータが、3Tデータと4Tデータであることを、容易に認識することができる。

【 0 1 3 1 】

その結果、メインCPU 2 0 は、サブCPU 4 0 からの、3Tデータ及び4Tデータとしてのセンサデータを、3Tデータを要求したアプリケーション#1、及び、4Tデータを要求したアプリケーション#2にそれぞれ提供することができる。

【 0 1 3 2 】

以上のように、サブCPU 4 0 では、センサ 1 1_n からセンサデータをサンプリングし、そのセンサデータを、周期情報とともに、メインCPU 2 0 に供給するので、メインCPU 2 0 は、センサデータがサンプリングされたタイミングを管理せずに、サブCPU 4 0 からの周期情報を参照するだけで、センサデータをサンプリングしたタイミングが該当するサンプリング周期を認識することができる。

10

【 0 1 3 3 】

さらに、上述した、サブCPU 4 0 において、基準周期Tのタイミングで、センサ 1 1_n からセンサデータをサンプリングし、メインCPU 2 0 に供給する場合のように、サブCPU 4 0 において、アプリケーションから要求されたサンプリング周期のセンサデータの他に、アプリケーションから要求されていないサンプリング周期のセンサデータがサンプリングされることはなく、サブCPU 4 0 からメインCPU 2 0 に対して、アプリケーションから要求されたサンプリング周期のセンサデータの他に、アプリケーションから要求されていないサンプリング周期のセンサデータが供給されることもない。

20

【 0 1 3 4 】

したがって、サブCPU 4 0 において、基準周期Tのタイミングで、センサ 1 1_n からセンサデータをサンプリングし、メインCPU 2 0 に供給する場合に比較して、センサデータの取得（サンプリング及び供給）について、負荷を軽減し、低消費電力化を図ることができる。

【 0 1 3 5 】

また、サブCPU 4 0 は、メインCPU 2 0（が実行するアプリケーション）からの要求に応じて、センサ 1 1_n からセンサデータをサンプリングし、そのセンサデータを、周期情報とともに、メインCPU 2 0 に供給するので、メインCPU 2 0 が要求するセンサデータの変更に対して、ロバストに対応することができる。

30

【 0 1 3 6 】

すなわち、例えば、メインCPU 2 0 が、所定のサンプリング周期（既に提供が行われているセンサデータのサンプリング周期とは異なるサンプリング周期）のセンサデータの提供（供給）を新たに要求した場合には、サブCPU 4 0 は、その所定のサンプリング周期のタイミングでも、センサデータのサンプリングを開始する。そして、サブCPU 4 0 は、既に提供されているセンサデータのサンプリング周期のインデクスと、所定のサンプリング周期とから、新たな周期情報を生成して、その新たな周期情報と、サンプリングによって得られるセンサデータとを、メインCPU 2 0 に提供することを、即座に開始することができる。

40

【 0 1 3 7 】

また、例えば、メインCPU 2 0 が、既に提供されているサンプリング周期のセンサデータのうちの所定のサンプリング周期のセンサデータの提供の解除を要求した場合には、サブCPU 4 0 は、その所定のサンプリング周期のタイミングでのセンサデータのサンプリングを停止し、まだ提供の解除が要求されていないセンサデータのサンプリング周期のインデクスと、所定のサンプリング周期とから、新たな周期情報を生成して、その新たな周期情報と、まだ提供の解除が要求されていないサンプリング周期のセンサデータとを、メインCPU 2 0 に提供することを、即座に開始することができる。

【 0 1 3 8 】

50

さらに、サンプリング周期のインデクスとして、1ビットオンビット列を採用するとともに、周期情報として、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデクスの論理和を採用することにより、サブCPU 4 0 からメインCPU 2 0 に提供されるセンサデータのオーバーヘッドとしての周期情報を、サンプリング周期のインデクスである1ビットオンビット列と同一のデータ量に抑えることができる。その結果、センサデータの他に、周期情報をやりとりすることによる、メインCPU 2 0 及びサブCPU 4 0 の負荷及び消費電力の増大を抑制することができる。

【 0 1 3 9 】

図 9 は、図 7 の携帯端末の処理を説明する図である。

【 0 1 4 0 】

図 9 では、メインCPU 2 0 において、アプリケーション#1ないし#3が実行されている。そして、ステップ S 1 1 において、アプリケーション#1が、3Tデータの提供を、OSに要求し、ステップ S 2 1 において、アプリケーション#2が、4Tデータの提供を、OSに要求している。さらに、ステップ S 3 1 において、アプリケーション#3が、5Tデータの提供を、OSに要求している。

【 0 1 4 1 】

OSは、アプリケーション#1からの3Tデータの提供の要求を受信し、ステップ S 4 1 において、アプリケーション#1からの3Tデータの提供の要求に応じて、3Tデータの提供を、サブCPU 4 0 に要求する。

【 0 1 4 2 】

また、OSは、アプリケーション#2からの4Tデータの提供の要求、及び、アプリケーション#3からの5Tデータの提供の要求を受信する。

【 0 1 4 3 】

そして、OSは、ステップ S 4 2 において、アプリケーション#2からの4Tデータの提供の要求に応じて、4Tデータの提供を、サブCPU 4 0 に要求する。さらに、OSは、ステップ S 4 3 において、アプリケーション#3からの5Tデータの提供の要求に応じて、5Tデータの提供を、サブCPU 4 0 に要求する。

【 0 1 4 4 】

サブCPU 4 0 は、OS (メインCPU 2 0) からの3Tデータ、4Tデータ、及び、5Tデータの要求を受信し、その要求に応じて、3T、4T、及び、5Tのサンプリング周期それぞれのタイミング、すなわち、図 8 において、斜線を付した矢印で示すタイミングでの、センサ 1 1 n からのセンサデータのサンプリングを開始する。

【 0 1 4 5 】

サブCPU 4 0 は、センサデータをサンプリングすると、そのセンサデータの周期情報を生成する。そして、サブCPU 4 0 は、ステップ S 5 1 において、センサデータと周期情報とを、OS (メインCPU 2 0) に供給 (送信) する。

【 0 1 4 6 】

OSは、サブCPU 4 0 からのセンサデータと周期情報とを受信し、ステップ S 4 4 において、その周期情報に基づいて、サブCPU 4 0 からのセンサデータがサンプリングされたタイミングが該当するサンプリング周期を認識する。

【 0 1 4 7 】

そして、OSは、ステップ S 4 5 において、サブCPU 4 0 からのセンサデータを、センサデータを要求してきたアプリケーション#1ないし#3のうちの、サブCPU 4 0 からの周期情報から認識されたサンプリング周期のセンサデータを要求してきたアプリケーションに供給する。

【 0 1 4 8 】

サブCPU 4 0 によるステップ S 5 1、並びに、メインCPU 2 0 によるステップ S 4 4 及び S 4 5 の処理は、サブCPU 4 0 において、センサデータがサンプリングされ、そのセンサデータの周期情報が生成されるたびに行われる。

【 0 1 4 9 】

10

20

30

40

50

<サブCPU 4 0 の第 1 のハードウェア構成例>

【 0 1 5 0 】

図 1 0 は、図 7 のサブCPU 4 0 の第 1 のハードウェア構成例を示すブロック図である。

【 0 1 5 1 】

図 1 0 において、サブCPU 4 0 は、タイマ 4 1、CPUコア 4 2、RAM(Random Access Memory) 4 3、ROM(Read Only Memory) 4 4、センサI/O 4 5、及び、コミュニケーションI/O 4 6を有する。そして、タイマ 4 1ないしコミュニケーションI/O 4 6は、バスに接続されている。

【 0 1 5 2 】

タイマ 4 1 は、時間を計時 (カウント) する。

10

【 0 1 5 3 】

CPUコア 4 2 は、ROM 4 4 に記憶されたプログラムを実行することにより、主として、センサデータを取得する処理、すなわち、センサ 1 1_n からセンサデータをサンプリングし、メインCPU 2 0 に供給する処理を行う。

【 0 1 5 4 】

RAM 4 3 は、CPUコア 4 2 の動作上必要なデータ等を一時記憶する。RAM 4 3 が記憶するデータには、後述する管理テーブルが含まれる。

【 0 1 5 5 】

ROM 4 4 は、OSや所定のアプリケーション (プログラム) 等を記憶している。ROM 4 4 に記憶されているアプリケーションには、少なくとも、センサデータを取得して提供するためのセンサデータ提供プログラム (サブCPU 4 0 を、センサデータを取得して提供するデータ処理装置として機能させるためのプログラム) が含まれる。

20

【 0 1 5 6 】

センサI/O 4 5 は、センサ 1 1_n にアクセスするためのインターフェースであり、センサ 1 1_n からセンサデータを受信 (サンプリング) する。

【 0 1 5 7 】

コミュニケーションI/O 4 6 は、メインCPU 2 0 との間で通信を行うためのインターフェースである。

【 0 1 5 8 】

ここで、メインCPU 2 0 は、サブCPU 4 0 と同様に構成される。

30

【 0 1 5 9 】

但し、メインCPU 2 0 については、タイマ 4 1 及びセンサI/O 4 5 は、必須ではない。また、メインCPU 2 0 では、少なくとも、CPUコア 4 2 については、サブCPU 4 0 よりも高性能な製品が用いられている。さらに、メインCPU 2 0 では、RAM 4 3 やROM 4 4 については、必要に応じて、サブCPU 4 0 よりも高性能な製品が用いられている。

【 0 1 6 0 】

また、サブCPU 4 0 では、ROM 4 4 に代えて、フラッシュメモリ等の書き換え可能なメモリを採用することができる。ROM 4 4 に代えて採用する書き換え可能なメモリに記憶されるプログラム (OSを含む) については、例えば、インターネット等から、コミュニケーションI/O 4 6 を介してダウンロードしてインストールや更新を行うことができる。

40

【 0 1 6 1 】

また、メインCPU 2 0 及びサブCPU 4 0 を有する携帯端末では、その携帯端末が内蔵する図示せぬハードディスク等のストレージに、インターネットやリムーバブル記録媒体から、プログラムをインストールして、メインCPU 2 0 やサブCPU 4 0 では、そのようにしてインストールされたプログラムを実行することができる。

【 0 1 6 2 】

図 1 0 に示したように構成されるサブCPU 4 0 では、CPUコア 4 2 が、ROM 4 4 に記憶されたセンサデータ提供プログラムを実行することにより、センサ 1 1_n からセンサデータをサンプリングし、そのサンプリングデータを、メインCPU 2 0 に提供 (供給) するセンサデータ提供処理が行われる。

50

【0163】

センサデータ提供処理では、サブCPU40において、メインCPU20からの要求に応じて、センサ11_nからセンサデータがサンプリングされる。さらに、サブCPU40では、センサ11_nからサンプリングされたセンサデータの周期情報が生成され、センサデータとともに、メインCPU20に提供される。

【0164】

センサデータ提供処理において、センサ11_nからのセンサデータのサンプリングや、サンプリングデータの、メインCPU20への提供（供給）は、管理テーブルによって管理される。

【0165】

管理テーブルは、CPUコア42がセンサデータ提供プログラムを実行することにより、RAM43に展開（記憶）される。

【0166】

図11は、管理テーブルを示す図である。

【0167】

管理テーブルには、サブCPU40がサンプリングすることができるセンサデータのサンプリング周期が登録される。図11では、サンプリング周期として、1T, 2T, 3T, 4T, 5T、及び、6Tの6個が登録されている。

【0168】

管理テーブルには、さらに、サンプリング周期に対して、インデクス(Index)、要求フラグ、初期値、及び、残カウント値が対応付けられて登録される。

【0169】

管理テーブルにおいて、サンプリング周期に対応付けられているインデクスは、そのサンプリング周期のインデクス（サンプリング周期を表すインデクス）である。図11では、サンプリング周期に対して、図8で説明したのと同様の1ビットオンビット列が、サンプリング周期のインデクスとして登録されている。

【0170】

サンプリング周期に対応付けられている要求フラグは、そのサンプリング周期のセンサデータの提供が要求されているかどうかを表す、例えば、1ビットのフラグである。所定のサンプリング周期のセンサデータの提供が要求されている場合、そのサンプリング周期に対応付けられている要求フラグはオン(on)（0及び1のうちの、例えば、1）にされる。また、所定のサンプリング周期のセンサデータの提供が要求されていない場合（センサデータの提供が解除されている場合）、そのサンプリング周期に対応付けられている要求フラグはオフ(off)（0及び1のうちの、例えば、0）にされる。

【0171】

サンプリング周期に対応付けられている初期値は、そのサンプリング周期に対応付けられている残カウント値に設定される初期値である。すなわち、所定のサンプリング周期に対応付けられている残カウント値が0になると、その残カウント値は、所定のサンプリング周期に対応付けられている初期値に初期化される。

【0172】

サンプリング周期に対応付けられている残カウント値は、基準周期Tごとに、1だけデクリメントされる。この残カウント値のデクリメントにより、管理テーブルに登録されている各サンプリング周期がカウントされる。

【0173】

ここで、本実施の形態では、センサデータのサンプリング周期は、基準周期Tの整数倍kであるk×Tで表されるが、k×Tのサンプリング周期に対応付けられる初期値には、kが採用される。

【0174】

k×Tのサンプリング周期に対応付けられている残カウント値は、初期値kに初期化され、その後、基準周期Tごとに、1だけデクリメントされていく。

10

20

30

40

50

【0175】

その結果、 $k \times T$ のサンプリング周期に対応付けられている残カウント値が0になるタイミングは、 $k \times T$ のサンプリング周期のタイミングとなる。

【0176】

図12は、図10のCPUコア42がセンサデータ提供プログラムを実行することにより、サブCPU40で行われるセンサデータ提供処理を説明するフローチャートである。

【0177】

CPUコア42は、サブCPU40の起動後、センサデータ提供プログラムを実行し、これにより、サブCPU40では、センサデータ提供処理が行われる。

【0178】

センサデータ提供処理では、管理テーブル(図11)がRAM43に記憶される。なお、センサデータ提供処理の開始直後では、例えば、管理テーブルの要求フラグは、すべてオフにされ、管理テーブルの残カウント値は、すべて初期値に初期化される。

【0179】

センサデータ提供処理では、ステップS71において、CPUコア42が、タイマ41を起動し、これにより、タイマ41が、基準周期Tを繰り返し計時することを開始する。

【0180】

その後、処理は、ステップS71からステップS72に進み、CPUコア42は、(タイマ41の起動、又は、タイマ41による前回の基準周期Tの計時から)基準周期Tが経過したかどうかを判定する。

【0181】

ステップS72において、基準周期Tが経過していないと判定された場合、処理は、ステップS73をスキップして、ステップS74に進む。

【0182】

また、ステップS72において、基準周期Tが経過したと判定された場合、処理は、ステップS73に進み、CPUコア42は、タイマ割り込みを発生し、処理は、ステップS74に進む。ここで、タイマ割り込みの発生によって、後述するタイマ割り込み処理が行われる。

【0183】

ステップS74では、CPUコア42は、メインCPU20から、センサデータの提供の要求(提供要求(コマンド))、又は、センサデータの提供の解除の要求(解除要求(コマンド))があったかどうかを判定する。

【0184】

ステップS74において、提供要求、及び、解除要求のいずれもないと判定された場合、処理は、ステップS75をスキップして、ステップS72に戻り、以下、同様の処理が繰り返される。

【0185】

また、ステップS74において、提供要求、又は、解除要求があったと判定された場合、処理は、ステップS75に進み、CPUコア42は、提供要求、又は、解除要求に応じて、管理テーブルを更新して、処理は、ステップS72に戻る。

【0186】

ここで、ステップS75の管理テーブルの更新では、提供要求があった場合には、管理テーブルにおいて、提供要求によって提供が要求されたセンサデータのサンプリング周期に対応付けられている要求フラグが、センサデータの提供を表すオンに設定される。さらに、提供要求によって提供が要求されたセンサデータのサンプリング周期に対応付けられている残カウント値が、初期値に初期化される。

【0187】

また、解除要求があった場合には、管理テーブルにおいて、解除要求によって提供の解除が要求されたセンサデータのサンプリング周期に対応付けられている要求フラグが、センサデータの提供の解除(キャンセル)を表すオフに設定される。

10

20

30

40

50

【 0 1 8 8 】

図 1 3 は、図 1 2 のステップ S 7 3 のタイマ割り込みの発生によって行われるタイマ割り込み処理を説明するフローチャートである。

【 0 1 8 9 】

タイマ割り込み処理では、ステップ S 8 1 において、CPUコア 4 2 が、管理テーブル（図 1 1）のすべての残カウント値を、1だけデクリメントして、処理は、ステップ S 8 2 に進む。

【 0 1 9 0 】

ステップ S 8 2 では、CPUコア 4 2 は、管理テーブルを参照することにより、（現在のタイミングが、）メインCPU 2 0 で実行されている、いずれかのアプリケーションからセンサデータが要求されているタイミングであるかどうかを判定する。

10

【 0 1 9 1 】

ステップ S 8 2 において、いずれかのアプリケーションからセンサデータが要求されているタイミングでないと判定された場合、タイマ割り込み処理を終了してリターンする。

【 0 1 9 2 】

また、ステップ S 8 2 において、いずれかのアプリケーションからセンサデータが要求されているタイミングであると判定された場合、すなわち、管理テーブル（図 1 1）において、要求フラグがオンで、残カウント値が0になっているサンプリング周期（以下、該当サンプリング周期ともいう）が存在する場合、処理は、ステップ S 8 3 に進み、CPUコア 4 2 は、センサI/Oを制御して、センサ 1 1_nからのセンサデータを取得させて、処理は、ステップ S 8 4 に進む。

20

【 0 1 9 3 】

ステップ S 8 4 では、CPUコア 4 2 は、管理テーブルの該当サンプリング周期に対応付けられている残カウント値を、その該当サンプリング周期に対応付けられている初期値に初期化して、処理は、ステップ S 8 5 に進む。

【 0 1 9 4 】

ここで、該当サンプリング周期になっているサンプリング周期は、1つのサンプリング周期であるとは限らず、複数のサンプリング周期が該当サンプリング周期になっていることがある。複数のサンプリング周期が該当サンプリング周期になっている場合、ステップ S 8 4 では、その、該当サンプリング周期になっている複数のサンプリング周期それぞれについて、残カウント値が初期化される。

30

【 0 1 9 5 】

ステップ S 8 5 では、CPUコア 4 2 は、管理テーブルを参照することにより、該当サンプリング周期（になっているサンプリング周期）のインデクスを認識し、そのインデクスの論理和を、ステップ S 8 3 で取得されたセンサデータの周期情報として求め、処理は、ステップ S 8 6 に進む。

【 0 1 9 6 】

ステップ S 8 6 では、CPUコア 4 2 は、例えば、ステップ S 8 3 で取得されたセンサデータに、ステップ S 8 5 で生成された周期情報を付加することにより、1個のデータにまとめ、コミュニケーションI/O 4 6 を制御することにより、メインCPU 2 0 に送信させて、タイマ割り込み処理を終了してリターンする。

40

【 0 1 9 7 】

< インデクス、及び、周期情報の他の例 >

【 0 1 9 8 】

図 1 4 は、サンプリング周期のインデクスと、周期情報とを説明する図である。

【 0 1 9 9 】

図 8 では、サンプリング周波数として、基準周期Tの、1ずつ増加する整数倍kの周期である1T, 2T, 3T, 4T, 5T、及び、6Tを採用したが、サンプリング周期としては、その他、例えば、基準周期Tの、2のべき乗倍 2^k の周期である、例えば、 $2^0 \times T$, $2^1 \times T$, $2^2 \times T$, $2^3 \times T$, $2^4 \times T$ 、及び、 $2^5 \times T$ を採用することができる。

50

【 0 2 0 0 】

図 1 4 は、基準周期 T の、2のべき乗倍 2^k の周期である $2^0 \times T$, $2^1 \times T$, $2^2 \times T$, $2^3 \times T$, $2^4 \times T$ 、及び、 $2^5 \times T$ それぞれをサンプリング周期とする場合の、センサデータをサンプリングするタイミング（図中、矢印で示す部分）を示している。

【 0 2 0 1 】

図 1 4 では、 $2^0 \times T$ ないし $2^5 \times T$ のサンプリング周期のタイミングすべてが一致する、ある時刻を、時刻 t_0 として、時刻 t_0 から、基準周期 T の m 倍の時間だけ経過した時刻を、時刻 t_m と表してある。

【 0 2 0 2 】

図 1 4 のように、基準周期 T の、2のべき乗倍 2^k の周期を、サンプリング周期に採用する場合には、サンプリング周期を、例えば、 $1T, 2T, 3T, 4T, 5T, 6T$ のように、基準周期 T 単位で指定することは困難になるが、図 8 のように、基準周期 T の、1ずつ増加する整数倍 k の周期を、サンプリング周期に採用する場合に比較して、例えば、 $2^0 \times T$ ないし $2^5 \times T$ のような、広範囲な周期の中から、サンプリング周期を指定することが可能となる。

10

【 0 2 0 3 】

図 1 4 では、サンプリング周期のインデックスとして、例えば、シーケンシャルな整数が採用されている。

【 0 2 0 4 】

すなわち、図 1 4 では、例えば、 $2^0 \times T$ のサンプリング周期のインデックスとして1が、 $2^1 \times T$ のサンプリング周期のインデックスとして2が、 $2^2 \times T$ のサンプリング周期のインデックスとして3が、 $2^3 \times T$ のサンプリング周期のインデックスとして4が、 $2^4 \times T$ のサンプリング周期のインデックスとして5が、 $2^5 \times T$ のサンプリング周期のインデックスとして6が、それぞれ採用されている。

20

【 0 2 0 5 】

サンプリング周期のインデックスとして、シーケンシャルな整数を採用する場合、周期情報としては、例えば、図 8 で説明した、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデックスの系列を採用することができる。

【 0 2 0 6 】

この場合、図 1 4 において、例えば、時刻 t_8 は、 $2^0 \times T$, $2^1 \times T$, $2^2 \times T$ 、及び、 $2^3 \times T$ のサンプリング周期それぞれのタイミングに該当し、したがって、時刻 t_8 に、センサデータがサンプリングされたときには、そのセンサデータの周期情報は、 $2^0 \times T$ のサンプリング周期のインデックス1、 $2^1 \times T$ のサンプリング周期のインデックス2、 $2^2 \times T$ のサンプリング周期のインデックス3、及び、 $2^3 \times T$ のサンプリング周期のインデックス4を並べた1234となる。

30

【 0 2 0 7 】

ところで、図 1 4 に示したように、基準周期 T の、2のべき乗倍 2^k の周期である $2^0 \times T$ ないし $2^5 \times T$ を、サンプリング周期に採用する場合には、基準周期 T の、ある2のべき乗倍 2^A のサンプリング周期 $2^A \times T$ のタイミングは、必ず、基準周期 T の、 2^A 未満の2のべき乗倍 2^B のサンプリング周期 $2^B \times T$ のタイミングにもなっている ($A > B$)。

【 0 2 0 8 】

すなわち、図 1 4 において、例えば、時刻 t_8 は、サンプリング周期 $2^3 \times T$ のタイミングに該当するが、サンプリング周期 $2^3 \times T$ 未満のサンプリング周期 $2^0 \times T$, $2^1 \times T$, $2^2 \times T$ それぞれのタイミングにも、必ず該当する。

40

【 0 2 0 9 】

また、例えば、時刻 t_4 は（時刻 t_{12} 及び t_{20} も同様）は、サンプリング周期 $2^2 \times T$ のタイミングに該当するが、サンプリング周期 $2^2 \times T$ 未満のサンプリング周期 $2^0 \times T$, $2^1 \times T$ それぞれのタイミングにも、必ず該当する。

【 0 2 1 0 】

以上のように、基準周期 T の、2のべき乗倍 2^k の周期である $2^0 \times T$ ないし $2^5 \times T$ を、サンプリング周期に採用する場合には、あるサンプリング周期 $2^A \times T$ のタイミングは、必ず、そのサンプリング周期 $2^A \times T$ 未満のサンプリング周期 $2^B \times T$ のタイミングにもなっている。し

50

たがって、周期情報として、センサデータをサンプリングしたタイミングが該当するサンプリング周期のインデックスの系列を採用するときには、その周期情報としてのサンプリング周期のインデックスの系列は、最大のサンプリング周期のインデックスに圧縮することができる。

【0211】

すなわち、例えば、上述したように、時刻 t_g は、 $2^0 \times T$ 、 $2^1 \times T$ 、 $2^2 \times T$ 、及び、 $2^3 \times T$ のサンプリング周期それぞれのタイミングに該当するので、時刻 t_g に、センサデータがサンプリングされたときには、そのセンサデータの周期情報は、 $2^0 \times T$ のサンプリング周期のインデックス1、 $2^1 \times T$ のサンプリング周期のインデックス2、 $2^2 \times T$ のサンプリング周期のインデックス3、及び、 $2^3 \times T$ のサンプリング周期のインデックス4を並べた1234となる。

10

【0212】

この周期情報1234は、時刻 t_g が該当するサンプリング周期 $2^0 \times T$ 、 $2^1 \times T$ 、 $2^2 \times T$ 、及び、 $2^3 \times T$ 、すなわち、時刻 t_g にセンサデータがサンプリングされた場合に、そのセンサデータがサンプリングされたタイミングが該当するサンプリング周期 $2^0 \times T$ 、 $2^1 \times T$ 、 $2^2 \times T$ 、及び、 $2^3 \times T$ のうちの最大のサンプリング周期 $2^3 \times T$ のインデックス4に圧縮することができる。

【0213】

メインCPU 20が、センサデータとともに周期情報4を受信した場合、その周期情報4から、センサデータがサンプリングされたタイミングが、インデックスが4のサンプリング周期 $2^3 \times T$ のタイミングであること、すなわち、センサデータが、インデックスが4のサンプリング周期 $2^3 \times T$ のタイミングでサンプリングされたセンサデータであることを認識することができる。

20

【0214】

さらに、センサデータがサンプリングされたタイミングが、サンプリング周期 $2^3 \times T$ のタイミングであることから、そのセンサデータがサンプリングされたタイミングが、サンプリング周期 $2^3 \times T$ 未満のサンプリング周期 $2^0 \times T$ 、 $2^1 \times T$ 、及び、 $2^2 \times T$ それぞれのタイミングでもあることを認識することができる。

【0215】

以上のように、基準周期 T の、2のべき乗倍 2^k の周期を、サンプリング周期に採用し、周期情報を、センサデータがサンプリングされたタイミングが該当する1つ以上のサンプリング周期のうちの最大のサンプリング周期のインデックスに圧縮する場合には、各時刻で、センサデータがサンプリングされたときの周期情報は、図14に示すようになる。

30

【0216】

すなわち、サンプリング周期 $2^0 \times T$ のタイミングにのみ該当する時刻 t_1 等でサンプリングされるセンサデータの周期情報は、サンプリング周期 $2^0 \times T$ (のうちの最大のサンプリング周期 $2^0 \times T$)のインデックス1となる。

【0217】

また、サンプリング周期 $2^0 \times T$ 及び $2^1 \times T$ のタイミングにのみ該当する時刻 t_2 等でサンプリングされるセンサデータの周期情報は、サンプリング周期 $2^0 \times T$ 及び $2^1 \times T$ のうちの最大のサンプリング周期 $2^1 \times T$ のインデックス2となる。

40

【0218】

さらに、サンプリング周期 $2^0 \times T$ 、 $2^1 \times T$ 、及び、 $2^2 \times T$ のタイミングにのみ該当する時刻 t_4 等でサンプリングされるセンサデータの周期情報は、サンプリング周期 $2^0 \times T$ ないし $2^2 \times T$ のうちの最大のサンプリング周期 $2^2 \times T$ のインデックス3となる。

【0219】

また、サンプリング周期 $2^0 \times T$ 、 $2^1 \times T$ 、 $2^2 \times T$ 、及び、 $2^3 \times T$ のタイミングにのみ該当する時刻 t_g 等でサンプリングされるセンサデータの周期情報は、サンプリング周期 $2^0 \times T$ ないし $2^3 \times T$ のうちの最大のサンプリング周期 $2^3 \times T$ のインデックス4となる。

【0220】

さらに、サンプリング周期 $2^0 \times T$ 、 $2^1 \times T$ 、 $2^2 \times T$ 、 $2^3 \times T$ 、及び、 $2^4 \times T$ のタイミングに

50

のみ該当する時刻 t_{16} 等でサンプリングされるセンサデータの周期情報は、サンプリング周期 $2^0 \times T$ ないし $2^4 \times T$ のうちの最大のサンプリング周期 $2^4 \times T$ のインデクス5となる。

【0221】

そして、サンプリング周期 $2^0 \times T$ 、 $2^1 \times T$ 、 $2^2 \times T$ 、 $2^3 \times T$ 、 $2^4 \times T$ 、 $2^5 \times T$ のタイミングのすべてに該当する時刻 t_0 等でサンプリングされるセンサデータの周期情報は、サンプリング周期 $2^0 \times T$ ないし $2^5 \times T$ のうちの最大のサンプリング周期 $2^5 \times T$ のインデクス6となる。

【0222】

なお、図14では、センサデータを取得(サンプリング)するサンプリング周期として、基準周期 T の、2のべき乗倍 2^k の周期を採用したが、サンプリング周期としては、その他、例えば、2以外の3や5等の所定の整数のべき乗倍の周期を採用することができる。

10

【0223】

また、図14では、サンプリング周期のインデクスとして、シーケンシャルな整数を採用したが、図14でも、図8の場合と同様に、1ビットオンビット列を、サンプリング周期のインデクスとして採用することができる。

【0224】

<サブCPU40の第2のハードウェア構成例>

【0225】

図15は、図7のサブCPU40の第2のハードウェア構成例を示すブロック図である。

【0226】

なお、図中、図10の場合と対応する部分については、同一の符号を付してあり、以下では、その説明は、適宜省略する。

20

【0227】

図15において、サブCPU40は、CPUコア42、RAM43、ROM44、センサI/O45、及び、コミュニケーションI/O46を有する点で、図10の場合と共通する。

【0228】

但し、図15では、サブCPU40は、タイマ41を有しておらず、センサデータ取得制御部50、周期情報生成部56、及び、周期情報付加部57を、新たに有する点で、図10の場合と相違する。

【0229】

ここで、図10のサブCPU40では、ソフトウェアにより、すなわち、CPUコア42がセンサデータ提供プログラムを実行することにより、アプリケーションからセンサデータが要求されているタイミングの判定(図13のステップS82)、すなわち、センサデータのサンプリングのタイミングの生成が行われるが、センサデータのサンプリングのタイミングの生成は、ハードウェアにより行うことができる。

30

【0230】

図15では、センサデータのサンプリングのタイミングの生成が、ハードウェアであるセンサデータ取得制御部50(を構成する後述するタイミング生成部51、及び、カウンタ制御部52)により行われる。

【0231】

また、図10のサブCPU40では、ソフトウェアにより、センサI/O45によるセンサデータの取得が制御されるが(図13のステップS83)、センサI/O45によるセンサデータの取得の制御は、ハードウェアにより行うことができる。

40

【0232】

図15では、センサI/O45によるセンサデータの取得の制御が、ハードウェアであるセンサデータ取得制御部50(を構成する後述するセンサI/O制御部53)により行われる。

【0233】

さらに、図10のサブCPU40では、ソフトウェアにより、センサデータに、そのセンサデータの周期情報が付加されるが、センサデータへの、周期情報の付加は、ハードウェアにより行うことができる。

50

【 0 2 3 4 】

図 1 5 では、ハードウェアである周期情報付加部 5 7 により、センサデータへの、周期情報の付加が行われ、周期情報が付加されたセンサデータが、RAM 4 3 に記憶される。

【 0 2 3 5 】

以上のように、図 1 0 において、ソフトウェアで行われている処理の一部を、ハードウェアで行うことにより、その分、CPUコア 4 2 の負荷を軽減し、低消費電力化を図ることができる。

【 0 2 3 6 】

図 1 5 において、センサデータ取得制御部 5 0 は、メインCPU 2 0 (アプリケーション) から要求されたサンプリング周期のタイミングで、センサ 1 1_n から、センサデータを取得するように、センサI/O 4 5 を制御する。

10

【 0 2 3 7 】

すなわち、センサデータ取得制御部 5 0 は、タイミング生成部 5 1、カウンタ制御部 5 2、及び、センサI/O制御部 5 3 を有する。

【 0 2 3 8 】

タイミング制御部 5 1 は、基準周期Tのクロックを生成し、カウンタ制御部 5 2 に供給する。

【 0 2 3 9 】

カウンタ制御部 5 2 は、バスに接続されており、カウンタ制御部 5 2 には、コミュニケーションI/O 4 6 及びバスを介して、メインCPU 2 0 からの、センサデータの提供の要求 (提供要求) や、センサデータの提供の解除の要求 (解除要求) が供給される。

20

【 0 2 4 0 】

カウンタ制御部 5 2 は、図示せぬカウンタを内蔵しており、そのカウンタによって、タイミング制御部 5 1 からのクロックをカウントする。

【 0 2 4 1 】

そして、カウンタ制御部 5 2 は、カウンタのカウント値に基づいて、メインCPU 2 0 から提供要求があり、かつ、まだ解除要求がないセンサデータのサンプリング周期のタイミングを表すタイミング信号を生成し、センサI/O制御部 5 3 に供給する。

【 0 2 4 2 】

さらに、カウンタ制御部 5 2 は、タイミング信号が表すタイミングが該当するサンプリング周期のインデクスを、周期情報生成部 5 6 に供給する。

30

【 0 2 4 3 】

センサI/O制御部 5 3 は、カウンタ制御部 5 2 から供給されるタイミング信号に従って、センサI/O 4 5 を制御することにより、センサ 1 1_n から、センサデータを取得させる。

【 0 2 4 4 】

ここで、図 1 5 では、センサI/O 4 5 は、センサI/O制御部 5 3 の制御に従って、センサデータを取得 (サンプリング) するが、そのセンサデータは、センサI/O 4 5 から、周期情報付加部 5 7 に供給される。

【 0 2 4 5 】

周期情報生成部 5 6 は、カウンタ制御部 5 2 からのサンプリング周期のインデクスから、周期情報を生成し、周期情報付加部 5 7 に供給する。

40

【 0 2 4 6 】

ここで、周期情報としては、上述したように、サンプリング周期のインデクスの系列や、インデクスの論理和を採用することができる。また、周期情報として、サンプリング周期のインデクスの系列を採用する場合には、その周期情報は、上述したように圧縮可能であれば、圧縮することができる。

【 0 2 4 7 】

周期情報付加部 5 7 は、センサI/O 4 5 から供給されるセンサデータを、周期情報生成部 5 6 から供給される周期情報とともに出力する出力部として機能する。

50

【 0 2 4 8 】

すなわち、周期情報付加部 5 7 は、センサ I/O 4 5 から供給されるセンサデータに、周期情報生成部 5 6 から供給される周期情報を付加する。そして、周期情報付加部 5 7 は、周期情報が付加されたセンサデータ（以下、周期情報付きセンサデータともいう）を、バスを介して、RAM 4 3 に供給して記憶させる。

【 0 2 4 9 】

ここで、RAM 4 3 に記憶された周期情報付きセンサデータは、バスを介して、コミュニケーション I/O 4 6 から、メイン CPU 2 0 に送信される。

【 0 2 5 0 】

以上のように構成されるサブ CPU 4 0 では、タイミング制御部 5 1 は、基準周期 T のクロックを生成し、カウンタ制御部 5 2 に供給する。

【 0 2 5 1 】

カウンタ制御部 5 2 は、内蔵する図示せぬカウンタによって、タイミング制御部 5 1 からのクロックのカウントを開始する。

【 0 2 5 2 】

そして、カウンタ制御部 5 2 は、バスを介して、メイン CPU 2 0 から、センサデータの提供の要求（提供要求）があると、カウンタのカウント値に基づいて、メイン CPU 2 0 から提供要求があったセンサデータのサンプリング周期のタイミングを表すタイミング信号を生成し、センサ I/O 制御部 5 3 に供給する。

【 0 2 5 3 】

さらに、カウンタ制御部 5 2 は、タイミング信号が表すタイミングが該当するサンプリング周期のインデクスを、周期情報生成部 5 6 に供給する。

【 0 2 5 4 】

センサ I/O 制御部 5 3 は、カウンタ制御部 5 2 から供給されるタイミング信号に従って、センサ I/O 4 5 を制御することにより、センサ 1 1_n から、センサデータを取得させる。

【 0 2 5 5 】

センサ I/O 4 5 で取得されたセンサデータは、センサ I/O 4 5 から、周期情報付加部 5 7 に供給される。

【 0 2 5 6 】

一方、周期情報生成部 5 6 は、カウンタ制御部 5 2 からのサンプリング周期のインデクスから、周期情報を生成し、周期情報付加部 5 7 に供給する。

【 0 2 5 7 】

周期情報付加部 5 7 は、センサ I/O 4 5 からのセンサデータに、周期情報生成部 5 6 からの周期情報を付加し、その結果得られる周期情報付きセンサデータを、バスを介して、RAM 4 3 に転送して記憶させる。

【 0 2 5 8 】

周期情報付加部 5 7 は、周期情報付きセンサデータの、RAM 4 3 への転送を完了すると、その旨を表す転送完了通知を、CPU コア 4 2 に送信する。

【 0 2 5 9 】

CPU コア 4 2 は、周期情報付加部 5 7 からの転送完了通知を受信すると、コミュニケーション I/O 4 6 を制御することにより、RAM 4 3 に記憶された周期情報付きセンサデータを、メイン CPU 2 0 に送信させる。

【 0 2 6 0 】

ここで、本明細書において、コンピュータ（プロセッサ）（CPU）がプログラムに従って行う処理は、必ずしもフローチャートとして記載された順序に沿って時系列に行われる必要はない。すなわち、コンピュータがプログラムに従って行う処理は、並列的あるいは個別に実行される処理（例えば、並列処理あるいはオブジェクトによる処理）も含む。

【 0 2 6 1 】

また、プログラムは、1 のコンピュータにより処理されるものであっても良いし、複数

10

20

30

40

50

のコンピュータによって分散処理されるものであっても良い。さらに、プログラムは、遠方のコンピュータに転送されて実行されるものであっても良い。

【0262】

なお、本技術の実施の形態は、上述した実施の形態に限定されるものではなく、本技術の要旨を逸脱しない範囲において種々の変更が可能である。

【0263】

例えば、本実施の形態では、サンプリング周期と、そのサンプリング周期のインデクスとの対応関係が、固定であり（変動せず）、かつ、その固定の対応関係をメインCPU 20が認識していることを、暗黙の前提としているが、サンプリング周期とインデクスとの対応関係は、変更可能とすることができる。

10

【0264】

すなわち、例えば、サブCPU 40が、1T, 2T, . . . , 1000Tの1000個のサンプリング周期を採用し得る場合（1000個のサンプリング周期のセンサデータの提供が可能である場合）において、いままで、メインCPU 20から、2Tデータの提供を要求されていたが、その2Tデータの提供が解除され、その後、1000Tデータの提供を要求されたときには、サブCPU 40において、サンプリング周期とインデクスとの対応関係を変更し、いままで2Tデータのインデクスとして用いられていた値Vを、1000Tデータのインデクスとして用いることができる。

【0265】

サンプリング周期とインデクスとの対応関係を、変更可能とした場合、その対応関係の変更が行われたときには、その変更後の対応関係を、サブCPU 40からメインCPU 20に知らせる処理が必要になる。但し、インデクスについては、そのインデクスとして用いる値のビット数を少なくすることができる。

20

【0266】

すなわち、例えば、サンプリング周期のインデクスとして、1ビットオンビット列を採用する場合、サンプリング周期とインデクスとの対応関係を固定にすると、サブCPU 40が、1Tないし1000Tの1000個のサンプリング周期を採用し得るときには、インデクスとしての1ビットオンビット列は、1000ビットという膨大なビット数になる。

【0267】

サンプリング周期とインデクスとの対応関係を、変更可能とすることにより、時間をずらして、異なるサンプリング周期に対して、同一の値のインデクスを対応付けることが可能となり、インデクスとしての1ビットオンビット列が膨大なビット数になることを防止することができる。

30

【0268】

すなわち、例えば、サブCPU 40が、1Tないし1000Tの1000個のサンプリング周期を採用し得るが、同時に提供することができるサンプリングデータのサンプリング周期がP (< 1000) 個である場合には、サンプリング周期とインデクスとの対応関係を、変更可能とすることにより、インデクスとしての1ビットオンビット列のビット数を、Pビットにすることができる。

【0269】

また、本実施の形態では、センサ11_nからセンサデータを取得して提供するセンサデータ取得ブロックとしてのサブCPU 40と、センサデータの提供を受けるセンサデータ受信ブロックとしてのメインCPU 20とが、携帯端末という1個の筐体内に存在する場合を例に、本技術を説明したが、本技術は、その他、例えば、センサデータ取得ブロックと、センサデータ受信ブロックとが異なる装置に存在し、センサデータ取得ブロックが存在する装置から、センサデータ受信ブロックが存在する装置に、ネットワークを介して、センサデータを提供する場合にも適用することができる。

40

【0270】

さらに、本技術は、メインCPU 20とサブCPU 40とが、別個のIC(Integrated Circuit)チップとして構成される場合にも、複数のCPUが1個のICチップに含められるマルチCPU構

50

成のICチップとして構成される場合にも、適用することができる。メインCPU 20とサブCPU 40とが、マルチCPU構成の1個のICチップとして構成される場合、その1個のICチップにおいて、メインCPU 20とサブCPU 40との間の通信路は、そのメインCPU 20とサブCPU 40とで共有されるメモリ（共有メモリ）という形で実装することができる。

【0271】

また、本実施の形態では、センサ11_nからセンサデータを取得して提供するセンサデータ取得ブロックとして、ソフトウェアにより各種の処理の実行が可能なCPUであるサブCPU 40を採用したが、センサデータ取得ブロックとしては、その他、例えば、センサ11_nからセンサデータを取得して提供する用途に特化してハードウェア化されたICを採用することができる。

10

【0272】

さらに、本実施の形態では、説明を簡単にするために、センサ11₁ないし11_N（図3）のうちの1個のセンサ11_nからセンサデータを取得することとしたが、本技術は、複数のセンサから同時にセンサデータを取得する場合にも適用することができる。複数のセンサから同時にセンサデータを取得して提供するのに、管理テーブル（図11）を用いる場合、管理テーブルは、センサごとに用意され、センサからのセンサデータの取得、及び、提供は、そのセンサの管理テーブルを用いて行われる。

【0273】

なお、本技術は、以下のような構成をとることができる。

【0274】

20

< 1 >

物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、1つ以上のサンプリング周期それぞれのタイミングで、前記センサデータを取得する取得制御を行う取得制御部と、

前記センサデータを、そのセンサデータを取得したタイミングが該当する前記サンプリング周期を表す周期情報とともに出力する出力部と
を備えるデータ処理装置。

< 2 >

前記取得制御部は、所定の基準周期の所定数倍の、1つ以上の前記サンプリング周期のタイミングで、前記センサデータを取得する取得制御を行う

30

< 1 >に記載のデータ処理装置。

< 3 >

前記取得制御部は、所定の基準周期の、所定の整数のべき乗倍の前記サンプリング周期のタイミングで、前記センサデータを取得する取得制御を行う

< 2 >に記載のデータ処理装置。

< 4 >

前記取得制御部は、所定の基準周期の、2のべき乗倍の前記サンプリング周期のタイミングで、前記センサデータを取得する取得制御を行う

< 3 >に記載のデータ処理装置。

< 5 >

40

前記センサデータを取得したタイミングが該当する1つ以上の前記サンプリング周期のうち最大のサンプリング周期を表す情報を、前記センサデータとともに出力する前記周期情報として生成する生成部をさらに備える

< 4 >に記載のデータ処理装置。

< 6 >

前記サンプリング周期を表す情報は、1ビットだけがたったビット列であり、

前記センサデータを取得したタイミングが該当する1つ以上の前記サンプリング周期それぞれを表す前記ビット列の論理和を、前記センサデータとともに出力する前記周期情報として生成する生成部をさらに備える

< 2 >ないし< 4 >のいずれかに記載のデータ処理装置。

50

< 7 >

前記取得制御部は、前記センサデータを利用するアプリケーションが要求する前記サンプリング周期のタイミングで、前記センサデータを取得する取得制御を行う

< 1 > ないし < 6 > のいずれかに記載のデータ処理装置。

< 8 >

前記センサデータと前記周期情報とを、前記センサデータを要求するアプリケーションを実行するプロセッサに提供する

< 1 > ないし < 7 > のいずれかに記載のデータ処理装置。

< 9 >

物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、1つ以上のサンプリング周期それぞれのタイミングで、前記センサデータを取得し、

前記センサデータを、そのセンサデータを取得したタイミングが該当する前記サンプリング周期を表す周期情報とともに出力する

ステップを含むデータ処理方法。

< 10 >

物理量をセンシングし、その物理量を表すセンサデータを出力するセンサから、1つ以上のサンプリング周期それぞれのタイミングで、前記センサデータを取得する取得制御を行う取得制御部と、

前記センサデータを、そのセンサデータを取得したタイミングが該当する前記サンプリング周期を表す周期情報とともに出力する出力部と

して、コンピュータを機能させるためのプログラム。

【符号の説明】

【0275】

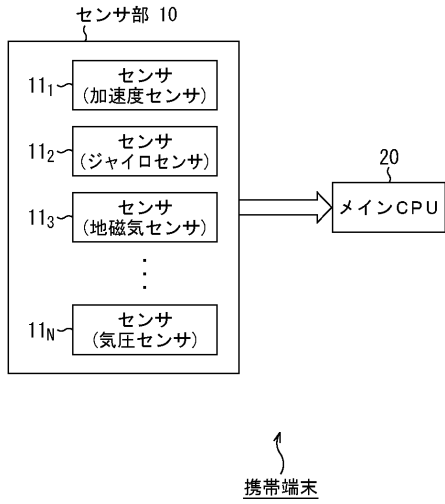
10 センサ部, 11₁ ないし 11_N センサ, 20 メインCPU, 30, 40 サブCPU, 41 タイマ, 42 CPUコア, 43 RAM, 44 ROM, 45 センサI/O, 46 コミュニケーションI/O, 50 センサデータ取得制御部, 51 タイミング生成部, 52 カウンタ制御部, 53 センサI/O制御部, 56 周期情報生成部, 57 周期情報付加部

10

20

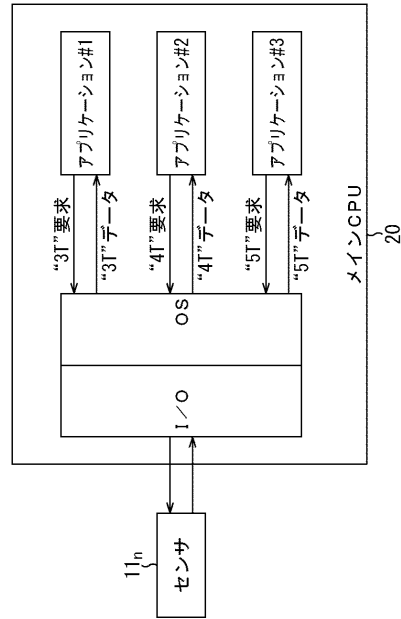
【図1】

図1



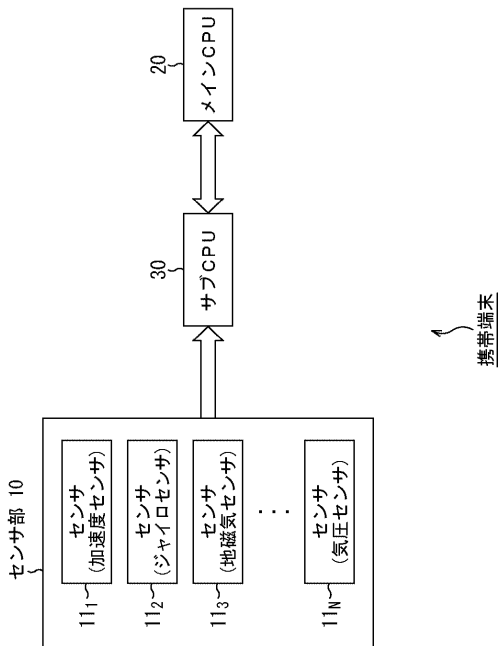
【図2】

図2



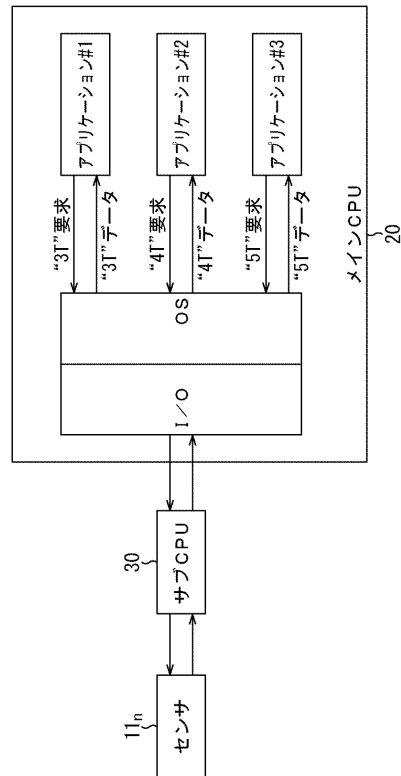
【図3】

図3

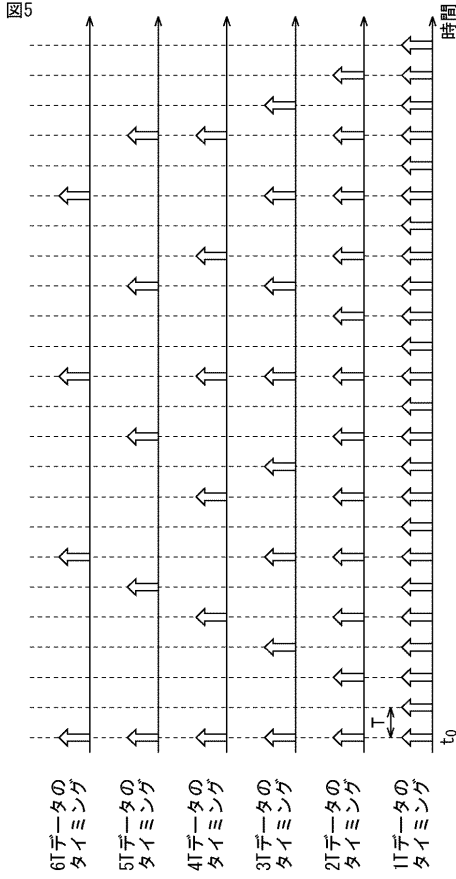


【図4】

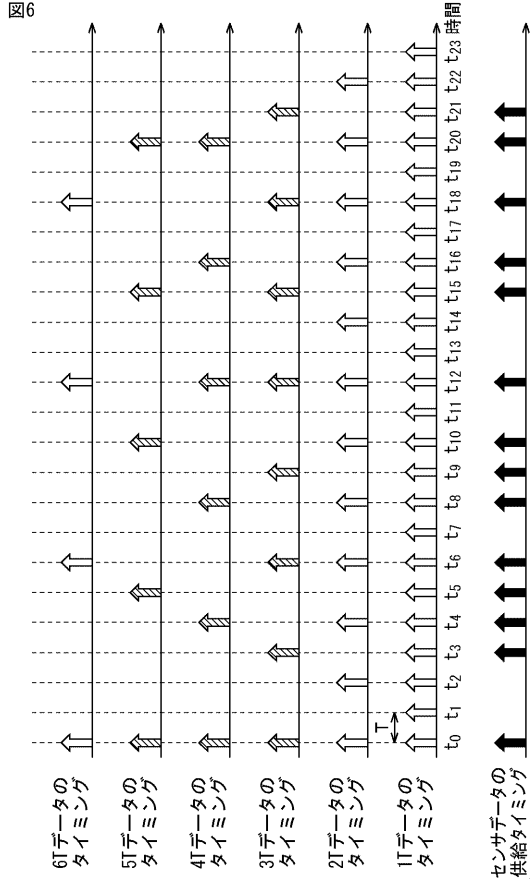
図4



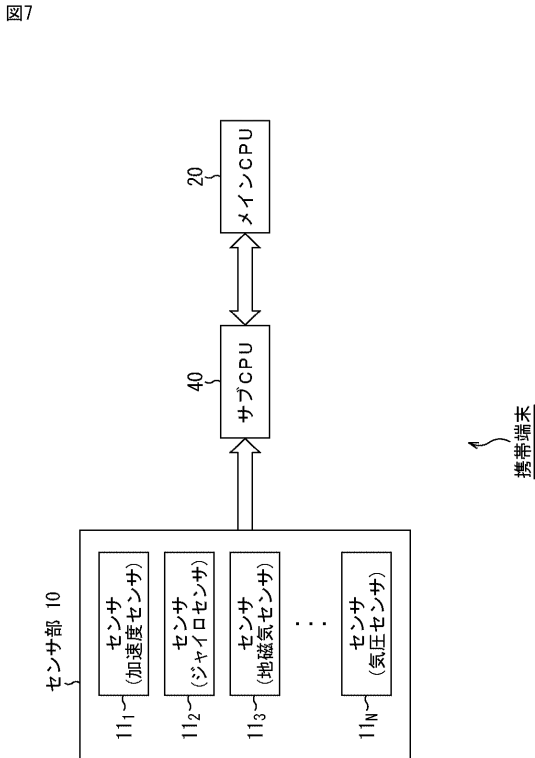
【図5】



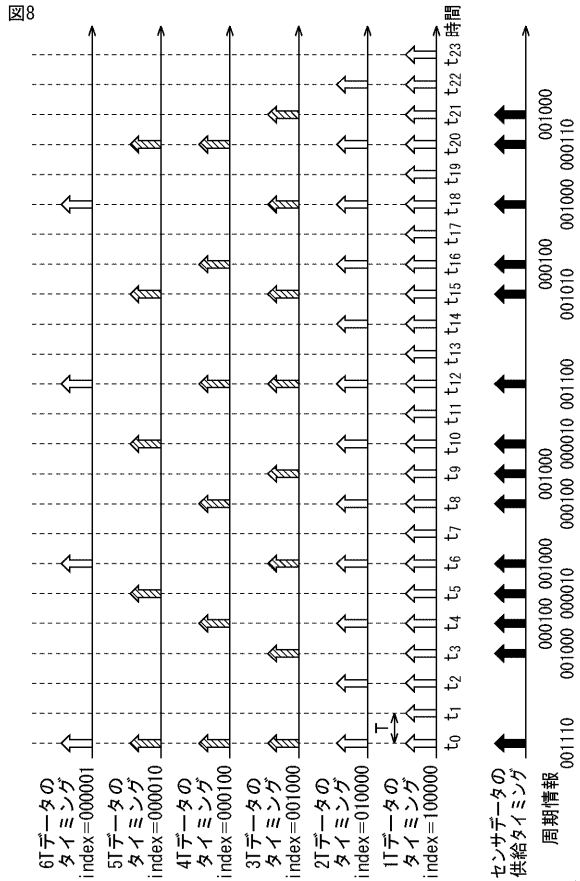
【図6】



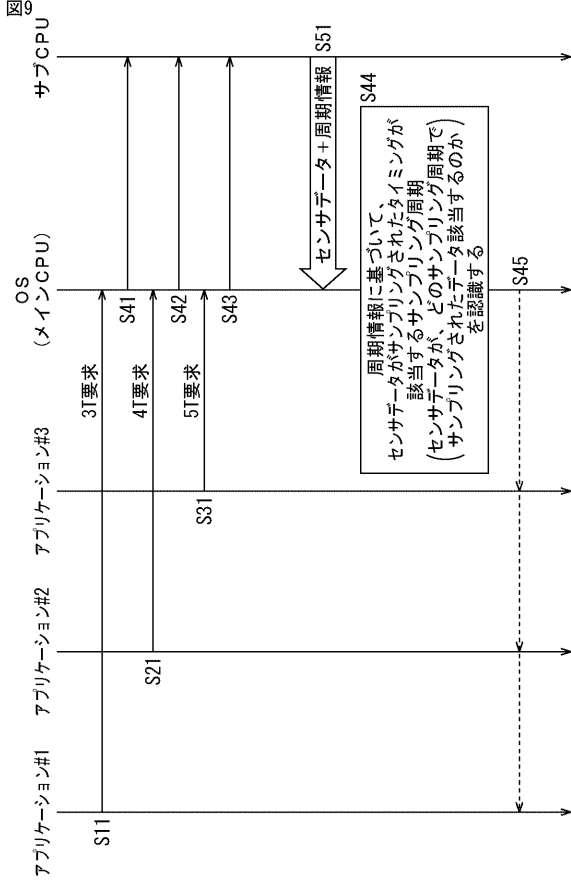
【図7】



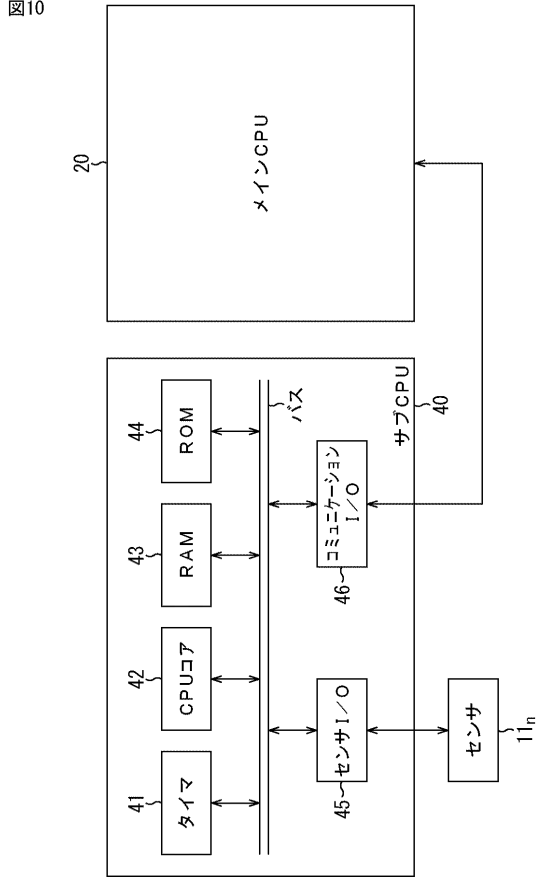
【図8】



【図9】



【図10】



【図11】

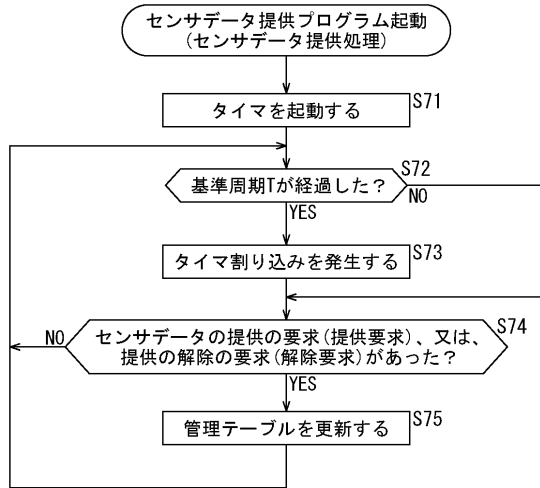
図11

サンプリング周期	Index	要求フラグ	初期値 (固定値)	残カウント値
1T	"100000"	off	1	1
2T	"010000"	off	2	2
3T	"001000"	on	3	1
4T	"000100"	on	4	2
5T	"000010"	on	5	3
6T	"000001"	off	6	5

センサデータの提供を管理する管理テーブル

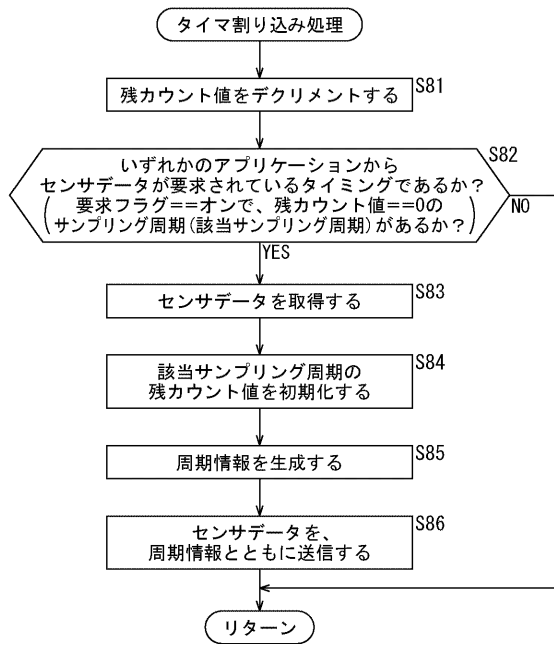
【図12】

図12



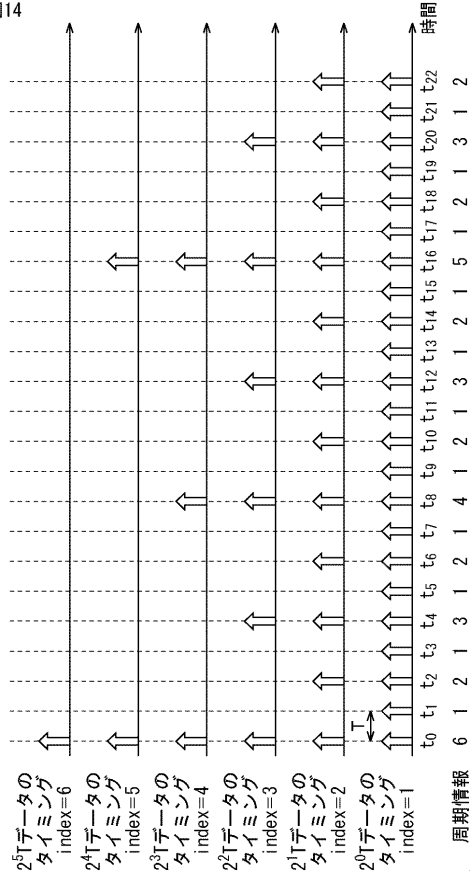
【図13】

図13



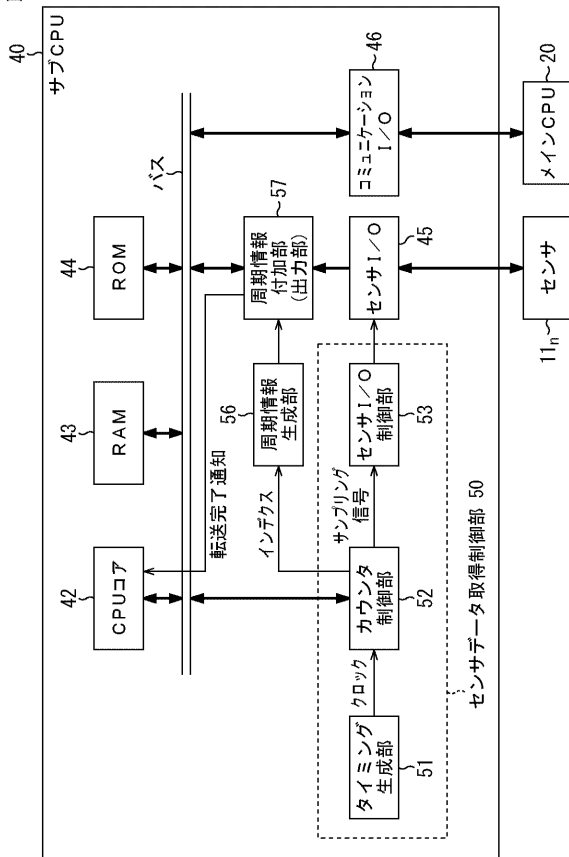
【図14】

図14



【図15】

図15



フロントページの続き

- (56)参考文献 特開2011-158990(JP,A)
特開2000-209225(JP,A)
特開2011-135141(JP,A)
米国特許出願公開第2012/0254878(US,A1)

(58)調査した分野(Int.Cl., DB名)

G06F 1/26 - 1/32
H04M 1/73
G06F 15/16 - 15/177
G06F 13/10 - 13/14