

3.25 タスク間通信

あるタスクが作成したデータを、別のタスクが加工するなど、タスクとタスクの間には、共有データをやり取りすることがあります。その際には、RTOSが提供するタスク間通信の機能を使います。

データが付随した同期制御

データのやり取りに関連するタスクの例として、データを受信するタスクと、データを解析するタスクがあったとします。データ解析タスクは、データ受信タスクがハードウェアからデータを受信するまで待ち合わせる必要があります。イベントフラグを使って制御すると、図3.25.1のようになり、タスク間通信の機能を使って制御すると図3.25.2のようになります。

このように、同期制御にデータが伴う場合、タスク間通信機能が使用できます。タスク間通信機能は、データを受信するシステムコールとデータを送信するシステムコールを持っています。データ解析タスクは、データ受信のシステムコールを呼び出すことで、データが送信されるまでウエイト状態となります。そこへ、ハードウェアからデータを取得したデータ受信タスクが、そのデータを送信すると、データ解析タスクはウエイト状態が解除され、同時にデータを受信します。データは、それを受信するタスクが待ち合わせていないときでも送信することができ、RTOSがそのデータを保持しています。つまり、データは送信できるときに送信し、受信できるときに受信すればよく、別の同期の仕組みは必要ありません。RTOSが提供するタスク間通信機能として、データの受け渡しをメモリ上にあるデータへのポインタを渡す方式をとる**メールボックス**や、データをコピーして渡す方式の**データキュー**や**メッセージバッファ**などがあります。

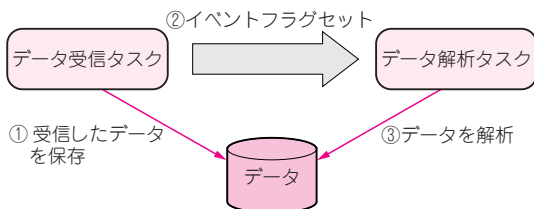


図3.25.1 イベントフラグを用いた制御



図3.25.2 タスク間通信を用いた制御

タスク間の速度差を吸収する

組込みシステムでは、受信データの発生頻度や、タスクの優先度によって、タスクの処理速度に違い

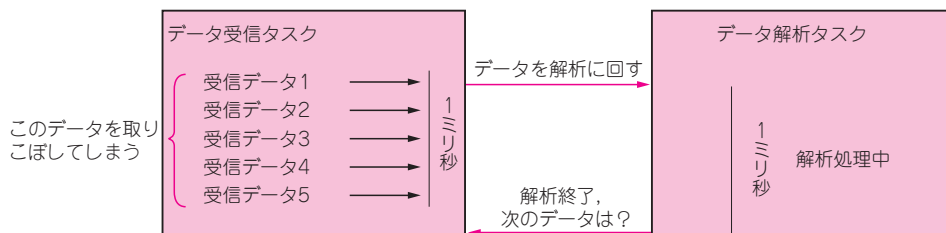


図3.25.3 タスクの速度差による問題

が出ることが多くあります。図3.25.3のように、1ミリ秒の間に、受信するデータが5回発生したとします。そして1回のデータを解析するのに、やはり1ミリ秒かかるとします。この場合、一つのデータを受信し、その解析が終わってから次のデータを受信しては間に合いません。このとき、データを受信する処理が速い処理で、データを解析する処理が遅い処理となります。

この速度差の問題を解決するには、先にデータの受信だけ行って、そのデータを保持しておき、データ解析は後から一つずつ行う方法があります。タスク間通信機能は、図3.25.4のように、このデータを保持する仕組みを提供しています。

このようにタスク間通信機能を使用することで、タスク間でのデータのやり取りに関する制御を行うことができます。

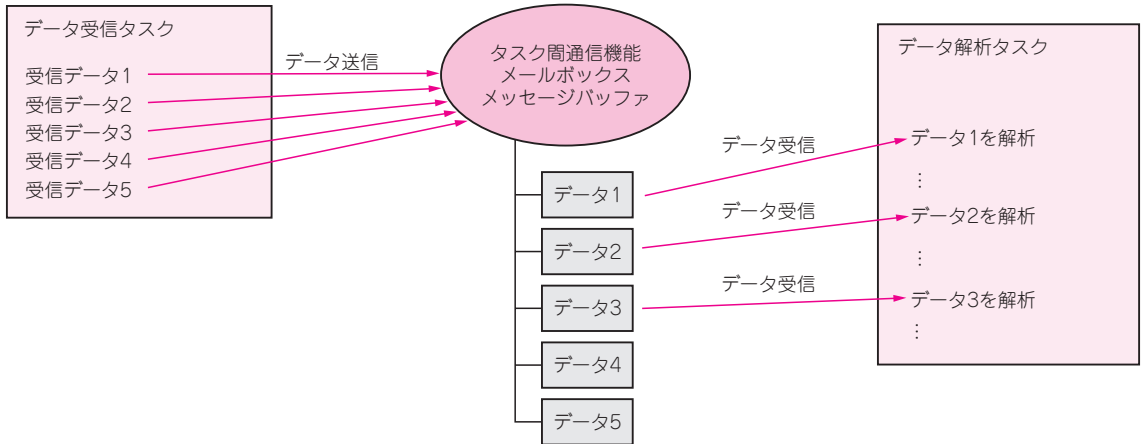


図3.25.4 タスク間通信機能を使ったタスクの速度差の吸収

タスク間通信の仕組み

タスク間通信は、データの送信と受信により、タスク間でのデータの受け渡しと、同期制御をあわせて行う機能です。ここでは、タスク間通信の仕組みとその動作について説明します。

■ メールボックス等のデータ送信システムコールを呼び出したとき

そのメールボックスでデータ受信を待ち合わせているタスクがいれば、送信データは直ちに受信され、そのタスクはウェイト状態が解除されます。

そのメールボックスでデータ受信を待ち合わせているタスクがいなければ、データはメールボックスに保持されます。このとき、データ送信のシステムコールからすぐに復帰する場合と、送信したデータが受信されるまで、送信タスクをウェイト状態にする場合があります。

データ送信には、タスクのウェイト解除をとまなうことがあるんだ。それによって、プリエンブションが起きることもあるよ。



■ メールボックス等のデータ受信システムコールを呼び出したとき

そのメールボックスに送信済みのデータが保持されていなければ、システムコールを呼び出したタスクはウェイト状態となります。このメールボックスにデータが送信されたとき、ウェイト状態が解除されます。

そのメールボックスに送信済みのデータが保持されていれば、送信データは直ちに受信され、受信タスクはウェイト状態とならずシステムコールから復帰します。



データ受信は、呼び出したタスクがウェイト状態になってしまうことがあります。そして、別のタスクにディスパッチすることがあります。