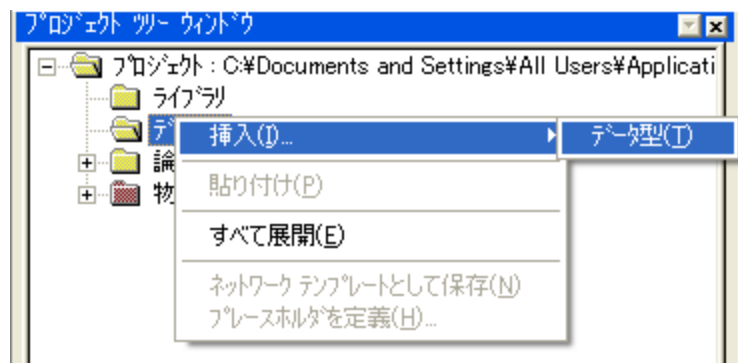


INplc タスク処理時間の計測方法

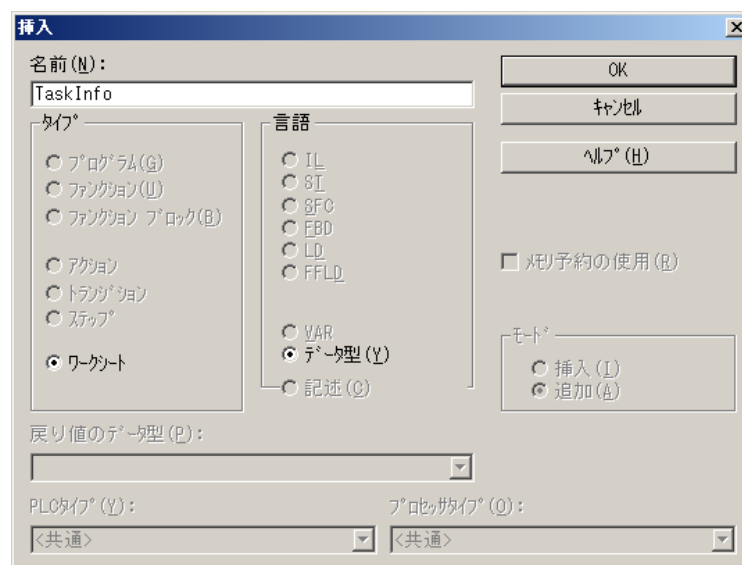
INplc では「現在のタスクの実行時間」「タスクの実行時間の最小値」「タスクの実行時間の最大値」の処理時間を計測することができます。本資料はそれらの計測方法について解説した資料です。

手順1. データ型の新規追加

プロジェクトツリーの「データ型」を右クリックし、[挿入]―[データ型]を選択すると挿入ダイアログが表示されます。



「名前」に任意の名前(下記の例では TaskInfo)を設定し「OK」を選択します。



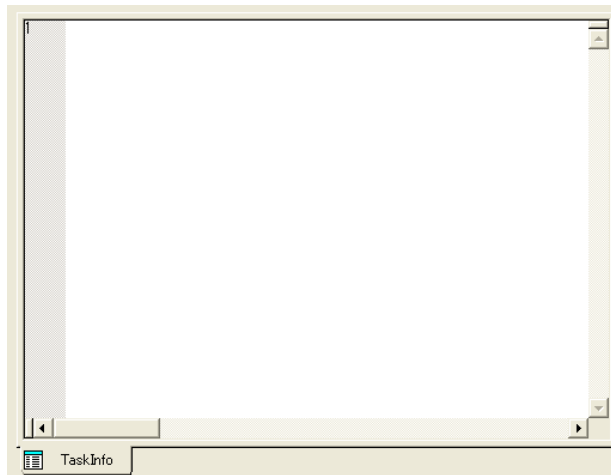
これにより新しくデータ型が作成されます。

2. 構造体の定義

新規に作成したデータ型 (TaskInfo) 項目をダブルクリックします。



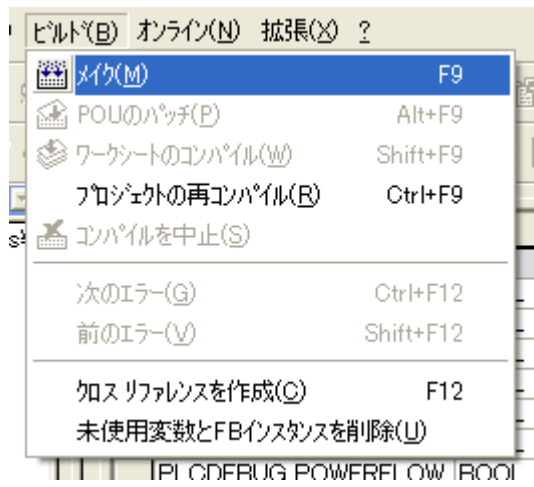
次の編集ウィンドウが表示されます。



表示された編集ウィンドウ「Task_Info」に下記の構造体を定義します。

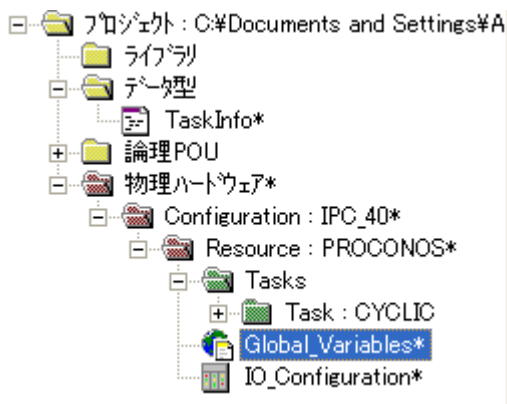
```
TYPE
(* Base アドレス %MW 1.1004 *)
Task_Name : ARRAY[1..10] OF BYTE;
Task_Info : STRUCT (* Total size = 64 Bytes *)
TASK_NAME           : Task_Name: (* Name of the task, Null terminated string *)
TASK_PRIO           : INT: (* Priority of the task *)
SystemReserve1     : INT:
TASK_PERIOD         : INT: (* Period of the task if the task type is CYCLE in milliseconds *)
TASK_STACK          : INT: (* Stack size of the task *)
SystemReserve2     : INT:
TASK_WATCHDOG       : INT: (* Watchdog time in milliseconds *)
SystemReserve3     : INT:
SystemReserve4     : INT:
SystemReserve5     : INT:
CUR_DURATION        : INT: (* Current task duration including preemption in ticks *)
MIN_DURATION        : INT: (* Minimum task duration including preemption in ticks *)
MAX_DURATION        : INT: (* Maximum task duration including preemption in ticks *)
SystemReserve6     : INT:
CUR_DELAY           : INT: (* Current task delay including preemption in ticks *)
MIN_DELAY           : INT: (* Minimum task delay including preemption in ticks *)
MAX_DELAY           : INT: (* Maximum task delay including preemption in ticks *)
(*Next task info offset calculated by 1004 + N*64 where N is the task number *)
END_STRUCT;
END_TYPE
```

構造体を追加したら「メイク」を実行します。

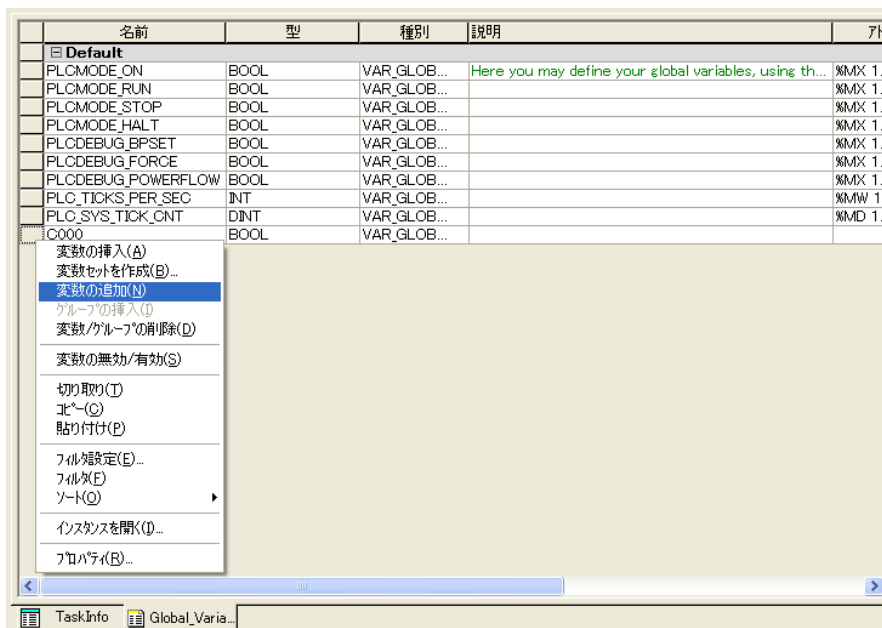


3. グローバル変数の定義

グローバル変数定義に変数を追加するために、[Global_Variables]をダブルクリックします。



[Global_Variables]一覧に表示されている何れかの変数名の上にマウスカーソルを移動し、右クリックすると以下のメニューが表示されます。メニュー中の「変数の追加」を選択します。



追加に成功すると、以下の[NewVar1]変数が追加されます。

CO00	BOOL	VAR_GLOB...	
NewVar1	BOOL	VAR_GLOB...	

追加された NewVar1 変数の「名前」「型」「アドレス」を下記の通りに変更します。

	初期値	変更後
名前	NewVar1	—任意の名前—
型	BOOL	Task_Info
アドレス	—空欄—	%MW1.1004

下記の図は変更後の例です。

CO00	BOOL	VAR_GLOB...		
TaskData	Task_Info	VAR_GLOB...		%MW1.1004

※補足事項

タスクが複数存在し、それらのタスクの処理時間を計測する場合のアドレス割付は以下の通りです。

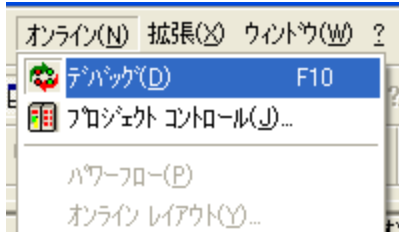
なお、タスクが1つの場合のアドレスは「%MW1.1004」です。

タスク	オフセット	アドレス記入例
1つ目のタスク	0	%MW1.1004
2つ目のタスク	64	%MW1.1068
3つ目のタスク	128	%MW1.1132

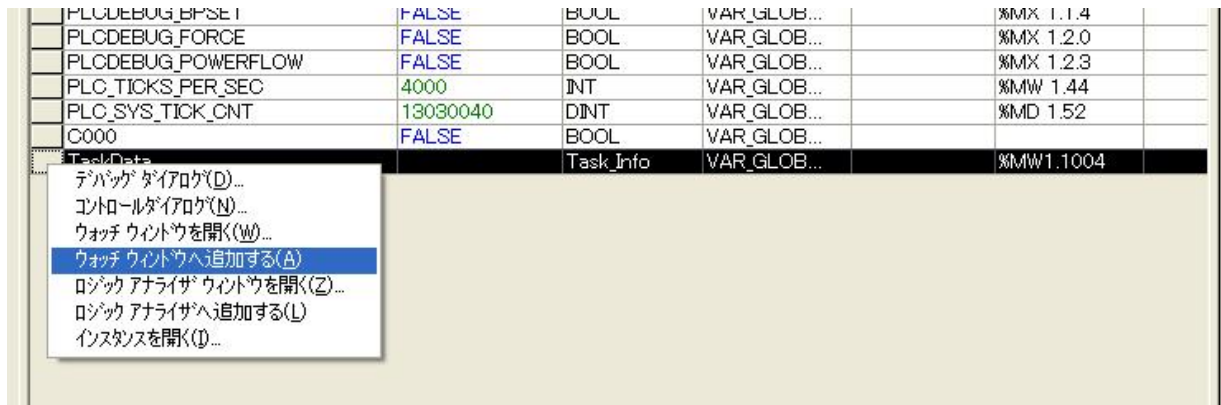
4つ目移行のタスクについても64ずつ加算したアドレス値になります。

4. 処理時間の確認

- 1) 作成したラダープログラムをダウンロードし実行します。
- 2) ラダープログラムが実行したら、下記[デバック]メニューを選択しデバッグモードに移行します。



- 3) [Global_Variables]一覧を表示し、今回追加したデータ変数(下記では TaskData)上でマウスを右クリックします。メニューの[ウォッチウィンドウへ追加する]を選択します。



- 4) ウォッチウィンドウに追加されると、下記のように追加した構造体情報が表示されます。

変数	値	デフォルト値	タイプ	インスタンス
TaskData			Task_Info	Configuration.Resource.Global_Variables.TaskData
TASK_NAME			Task_Name	Configuration.Resource.Global_Variables.TaskData.TA...
TASK_PRIO	0		INT	Configuration.Resource.Global_Variables.TaskData.TA...
SystemReserve1	2		INT	Configuration.Resource.Global_Variables.TaskData.Sys...
TASK_PERIOD	1		INT	Configuration.Resource.Global_Variables.TaskData.TA...
TASK_STACK	1000		INT	Configuration.Resource.Global_Variables.TaskData.TA...
SystemReserve2	2		INT	Configuration.Resource.Global_Variables.TaskData.Sys...
TASK_WATCHDOG	1		INT	Configuration.Resource.Global_Variables.TaskData.TA...
SystemReserve3	0		INT	Configuration.Resource.Global_Variables.TaskData.Sys...
SystemReserve4	0		INT	Configuration.Resource.Global_Variables.TaskData.Sys...
SystemReserve5	0		INT	Configuration.Resource.Global_Variables.TaskData.Sys...
CUR_DURATION	0		INT	Configuration.Resource.Global_Variables.TaskData.CU...
MIN_DURATION	0		INT	Configuration.Resource.Global_Variables.TaskData.MIN...
MAX_DURATION	0		INT	Configuration.Resource.Global_Variables.TaskData.MA...
SystemReserve6	0		INT	Configuration.Resource.Global_Variables.TaskData.Sys...
CUR_DELAY	0		INT	Configuration.Resource.Global_Variables.TaskData.CU...
MIN_DELAY	0		INT	Configuration.Resource.Global_Variables.TaskData.MIN...
MAX_DELAY	1		INT	Configuration.Resource.Global_Variables.TaskData.MA...

ウォッチ ウィンドウ: ウォッチ 1 / ウォッチ 2 / ウォッチ 3 / ウォッチ 4 /

5) 各項目の内容は次の表の通りです。

項目	内容	単位
CUR_DURATION	現在のタスクの実行時間	カーネルティック値
MIN_DURATION	タスクの実行時間の最小値	カーネルティック値
MAX_DURATION	タスクの実行時間の最大値	カーネルティック値

カーネルティック値とは・・・

カーネルティックとは、INplc のベース OS である INtime が時間経過を把握する目的で使用する基準クロックのことで、PC アーキテクチャが標準的に搭載しているインターバルタイマコントローラが使用されます。インターバルタイマコントローラは非常に短い時間周期でハードウェア割り込み(インターバル割り込み)を繰り返し発行するハードウェアです。

例えば INtime のカーネルティックを 500μ 秒に設定した場合、インターバルタイマは 500μ 秒ごとにハードウェア割り込みを生じるよう初期化されます。このハードウェア割り込みを受け取る INtime カーネルは、この個数を計上することにより時間経過を判断できます。

カーネルティックが 500μ 秒の場合の計算例・・・

値 = 2

タスクの処理時間は 2~3 ティック間となり、1m 秒 (500μ 秒 \times 2) ~ 1.5m 秒 (500μ 秒 \times 3) となります。

値 = 4

タスクの処理時間は 4~5 ティック間となり、2m 秒 (500μ 秒 \times 4) ~ 2.5m 秒 (500μ 秒 \times 5) となります。

以上