

**1 - Introduction**

2 – Conception modulaire  
A/ Électronique modulaire  
B/ Logiciel modulaire  
C/ Multi-tâches  
D/ Temps réel

3 – Application pratique  
A/ Analyse du CdC  
B/ Utilisation de PICos  
C/ Codage de l'application

4 – Pragmatec  
A/ Licence GPL  
B/ Aide au développement  
C/ Évolutions futures

5 - Conclusions

# **PICos : noyau temps réel pour PIC 16bits**

**MICROCHIP - PRAGMATEC**

**Mai 2006**

**Xavier MONTAGNE**

1 - Introduction

2 – Conception modulaire

A/ Électronique modulaire

B/ Logiciel modulaire

C/ Multi-tâches

D/ Temps réel

3 – Application pratique

A/ Analyse du CdC

B/ Utilisation de PICos

C/ Codage de l'application

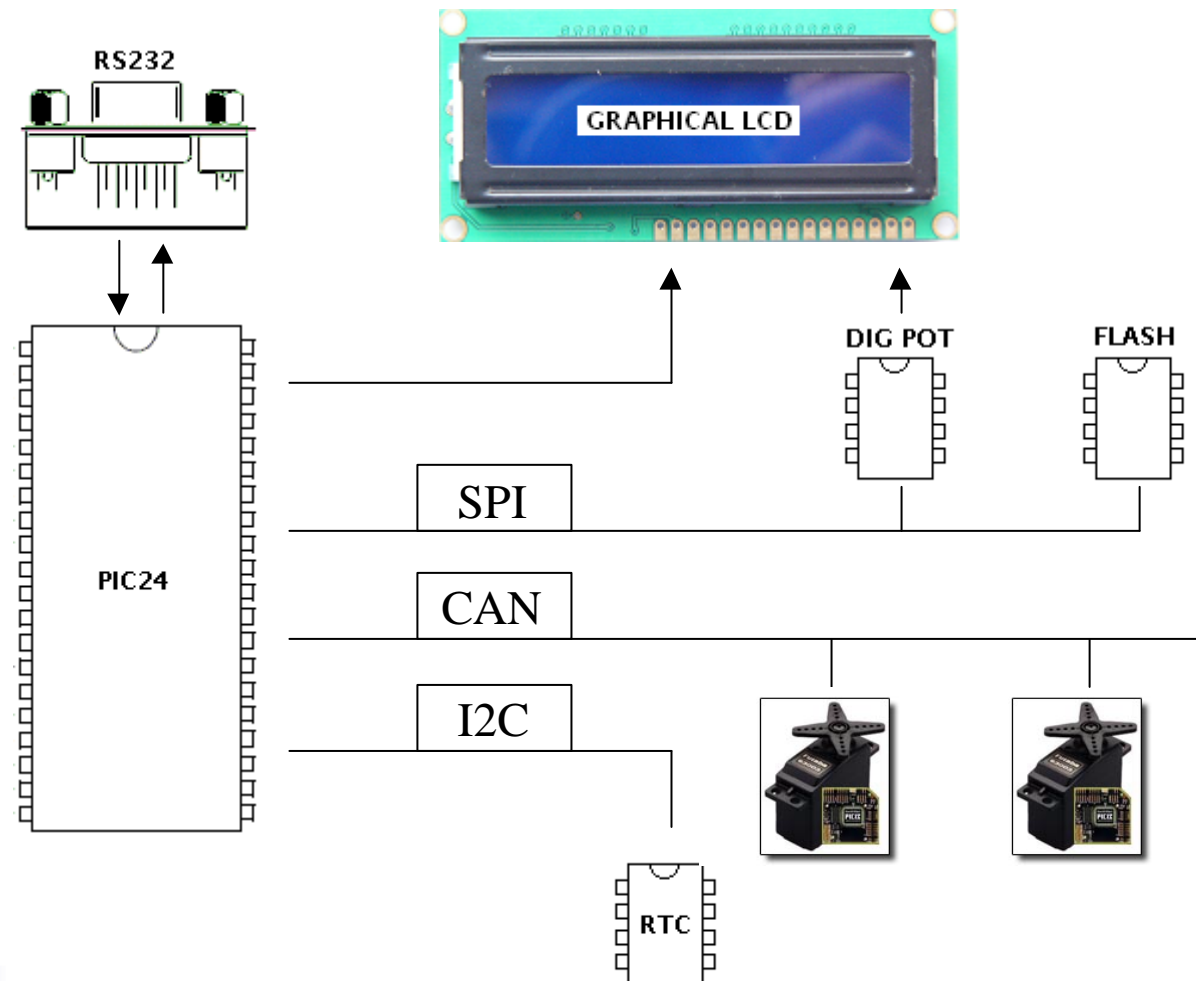
4 – Pragmatec

A/ Licence GPL

B/ Aide au développement

C/ Évolutions futures

5 - Conclusions



1 - Introduction

**2 – Conception modulaire**

A/ Électronique modulaire

**B/ Logiciel modulaire**

C/ Multi-tâches

D/ Temps réel

**3 – Application pratique**

A/ Analyse du CdC

B/ Utilisation de PICos

C/ Codage de l'application

**4 – Pragmatec**

A/ Licence GPL

B/ Aide au développement

C/ Évolutions futures

**5 - Conclusions**

Code bloquant :

- Printf()
- DelayMS()
- WaitForBusy()

Librairies

Modularité ?

Interruptions

Enchevêtrement :

- Priorité des IT
- Masquage des IT
- WaitForBusy()

Multi-tâches

Règles strictes :

- 1 tâche par action
- 1 chef d'orchestre
- gestion des priorités

1 - Introduction

**2 – Conception modulaire**

A/ Électronique modulaire

B/ Logiciel modulaire

**C/ Multi-tâches**

D/ Temps réel

**3 – Application pratique**

A/ Analyse du CdC

B/ Utilisation de PICos

C/ Codage de l'application

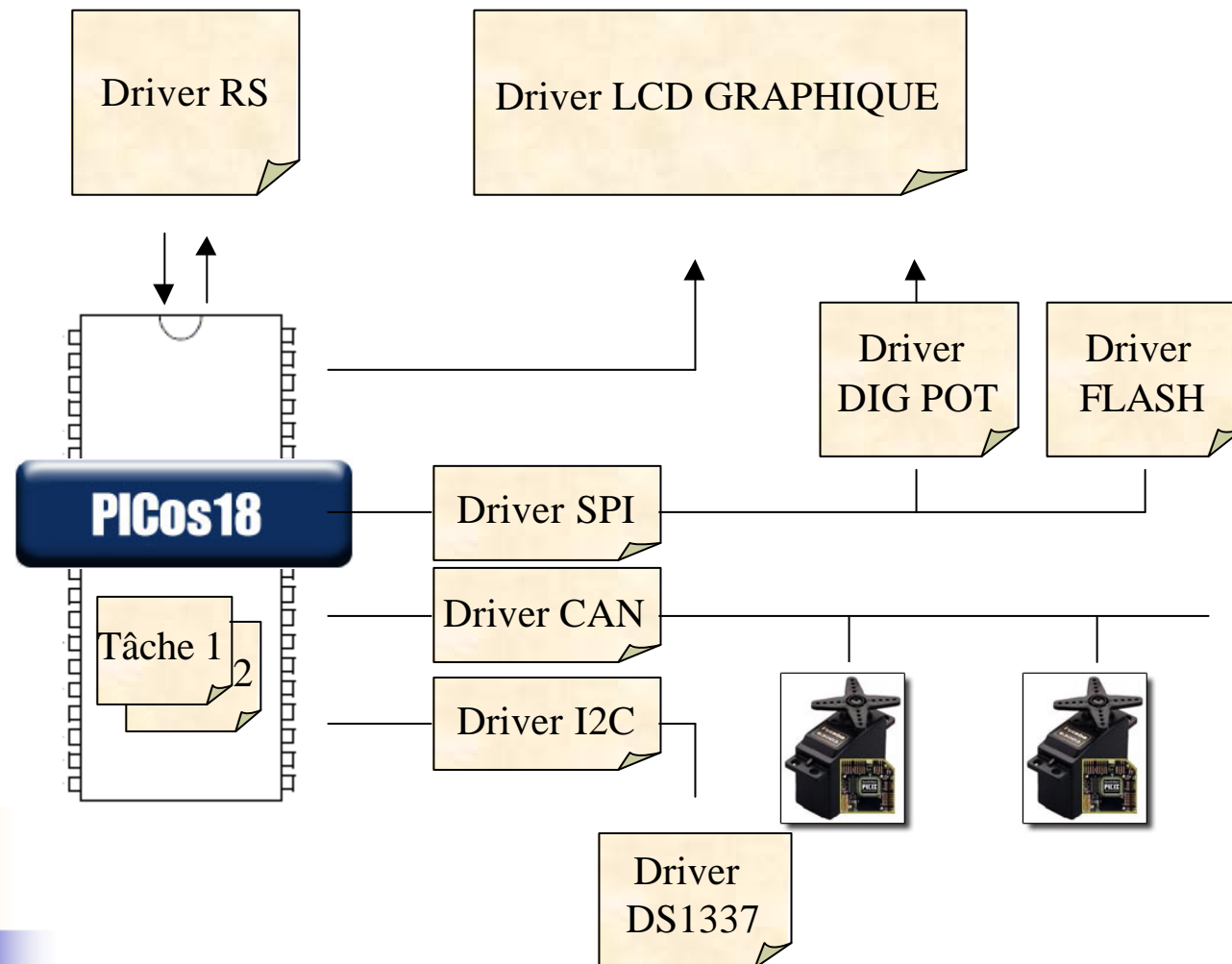
**4 – Pragmatec**

A/ Licence GPL

B/ Aide au développement

C/ Évolutions futures

**5 - Conclusions**



1 - Introduction

## 2 – Conception modulaire

A/ Électronique modulaire

B/ Logiciel modulaire

C/ Multi-tâches

**D/ Temps réel**

3 – Application pratique

A/ Analyse du CdC

B/ Utilisation de PICos

C/ Codage de l'application

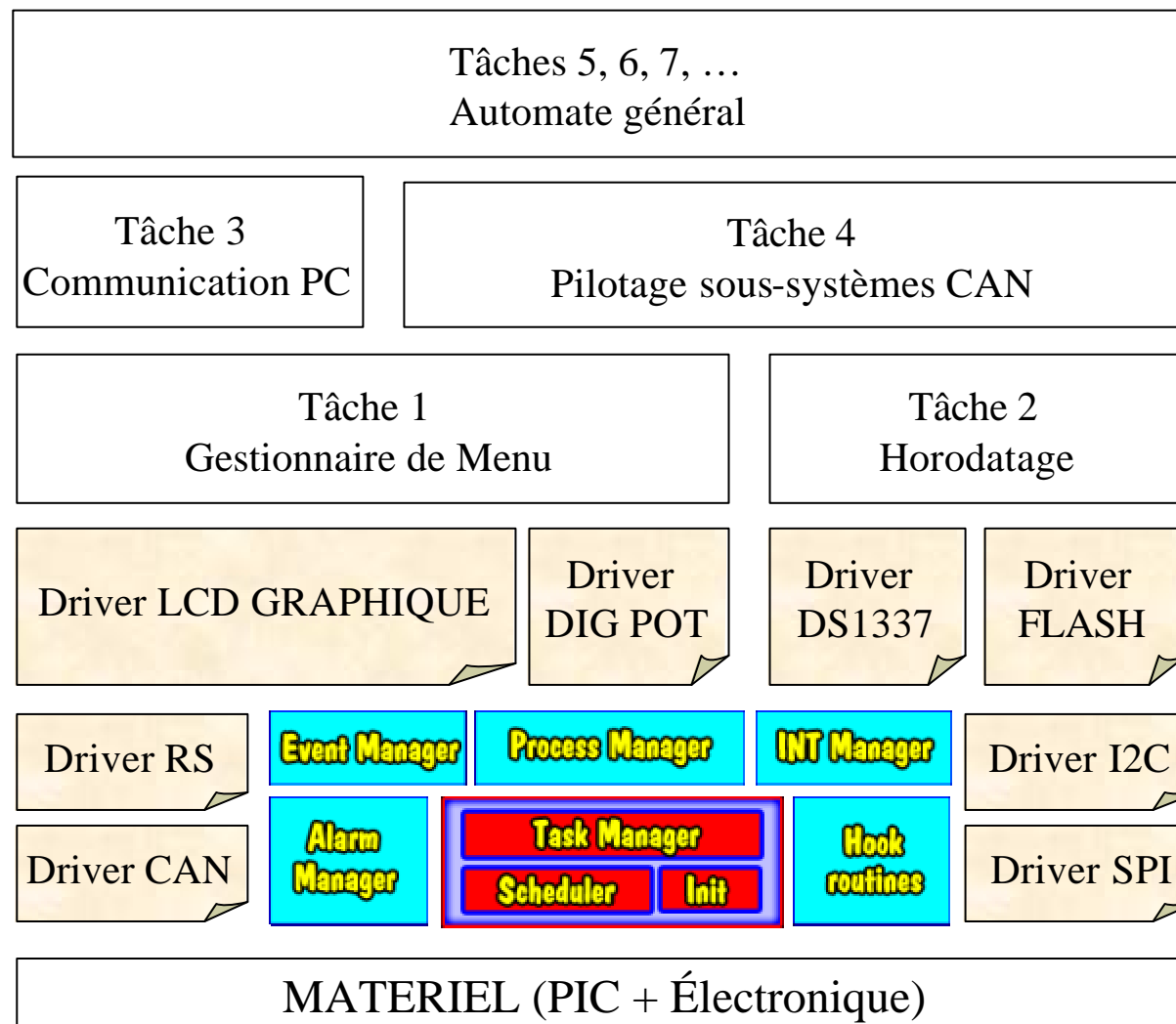
4 – Pragmatec

A/ Licence GPL

B/ Aide au développement

C/ Évolutions futures

5 - Conclusions



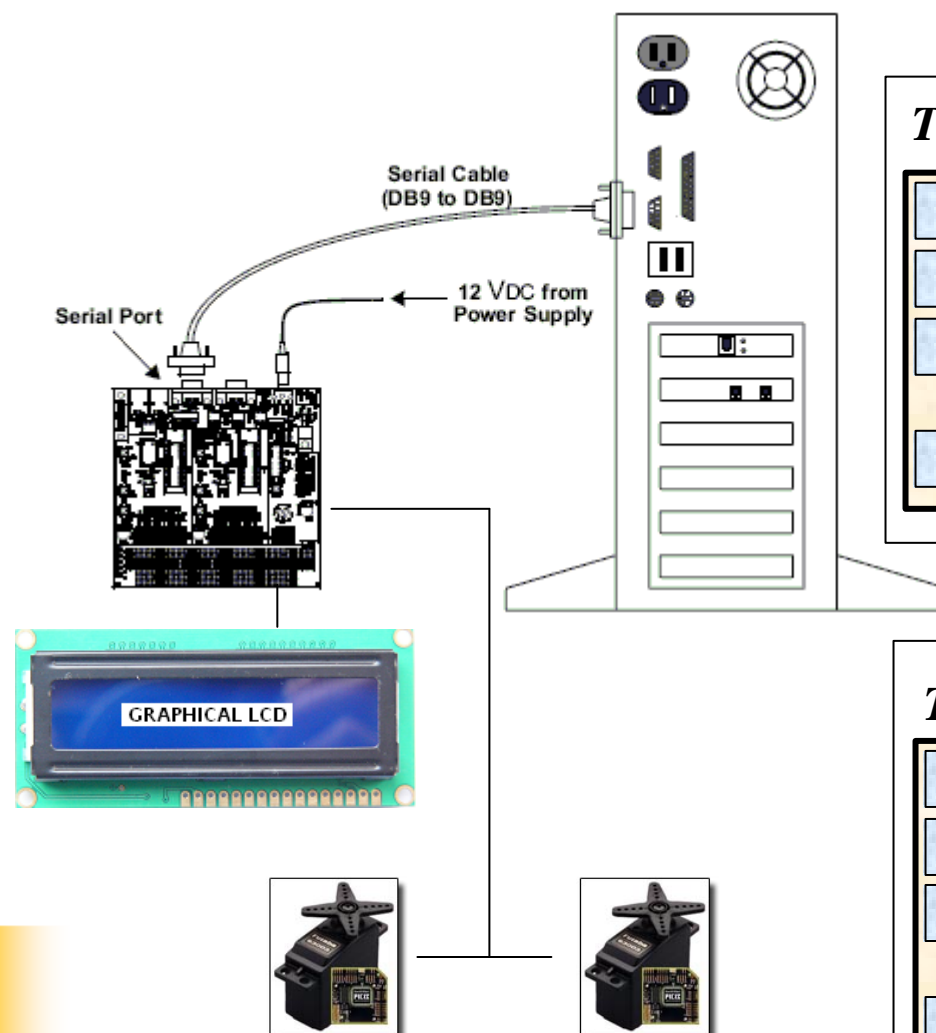
1 - Introduction

2 – Conception modulaire  
A/ Électronique modulaire  
B/ Logiciel modulaire  
C/ Multi-tâches  
D/ Temps réel

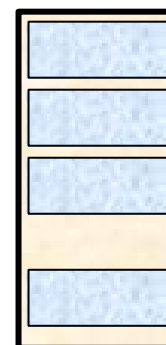
3 – Application pratique  
A/ Analyse du CdC  
B/ Utilisation de PICos  
C/ Codage de l'application

4 – Pragmatec  
A/ Licence GPL  
B/ Aide au développement  
C/ Évolutions futures

5 - Conclusions

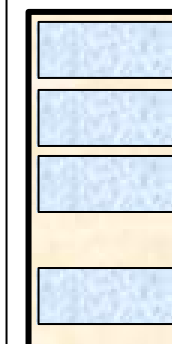


## Tâche RS232



- attente d'ordres
- synchronisation avec d'autres tâches
- acquittement des ordres

## Tâche CAN



- attente de trames
- envoi de trames
- traitement des données
- synchronisation avec d'autres tâches

## 1 - Introduction

## 2 – Conception modulaire

- A/ Électronique modulaire
- B/ Logiciel modulaire
- C/ Multi-tâches
- D/ Temps réel

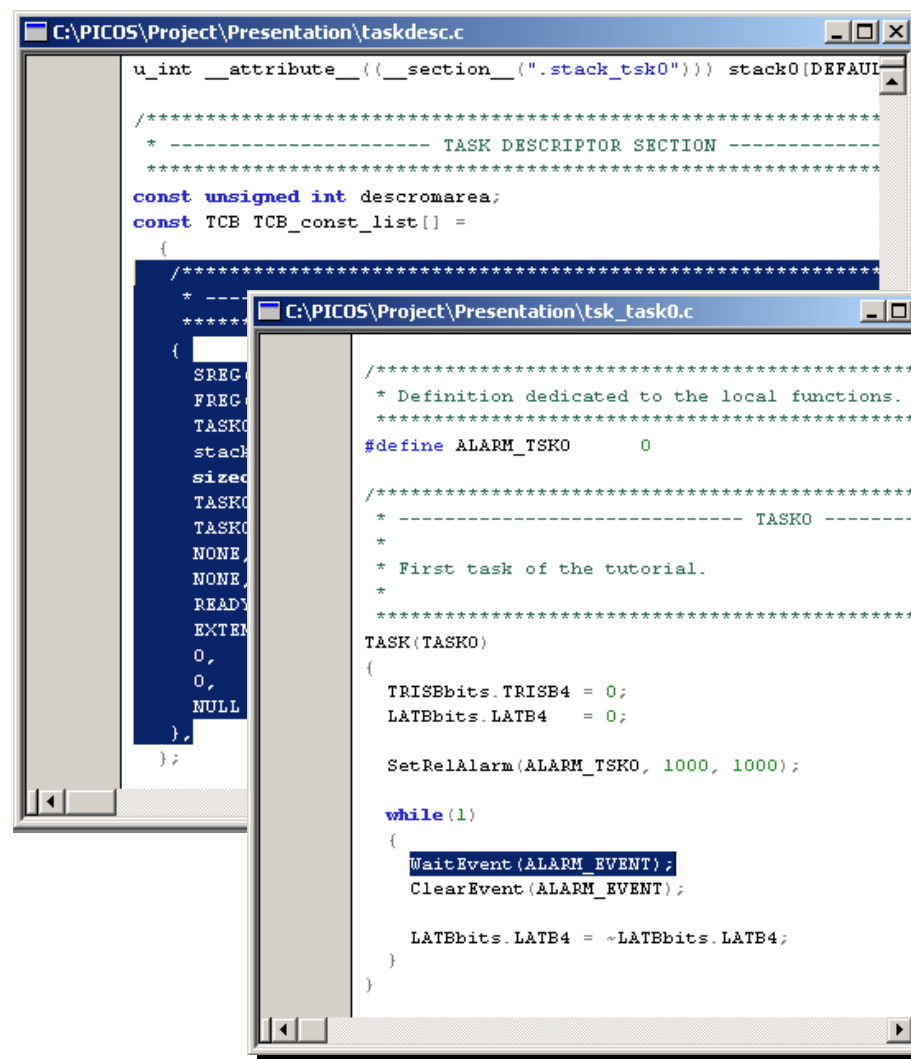
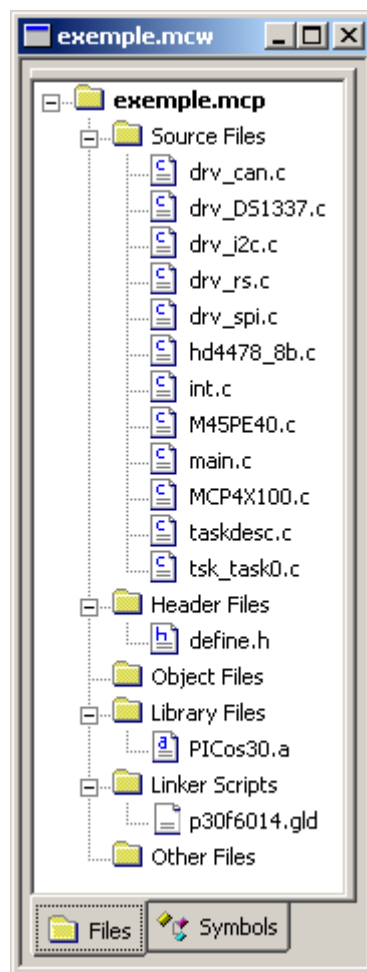
## 3 – Application pratique

- A/ Analyse du CdC
- B/ Utilisation de PICos**
- C/ Codage de l'application

## 4 – Pragmatec

- A/ Licence GPL
- B/ Aide au développement
- C/ Évolutions futures

## 5 - Conclusions





## 1 - Introduction

## 2 – Conception modulaire

- A/ Électronique modulaire
- B/ Logiciel modulaire
- C/ Multi-tâches
- D/ Temps réel

## 3 – Application pratique

- A/ Analyse du CdC
- B/ Utilisation de PICos
- C/ Codage de l'application**

## 4 – Pragmatec

- A/ Licence GPL
- B/ Aide au développement
- C/ Évolutions futures

## 5 - Conclusions

```

drv_can.txt - Bloc-notes
Fichier Edition Format ?

And declare a CAN message (type CAN_message_t):
CAN_message_t CAN_msg;

See hereafter a basic example to wait for a message with a specific
CAN ID value :

#include "define.h"
#include "drv_can.h"

/* Replace by the correct value of your alarm ID */
#define CAN_TASK_ALARM 0

/* *****
 * variables shared with the rest of application.
 * *****
CAN_message_t RCV_message;
unsigned char data1, data2, length;

/* *****
 * ----- Internal test TASK -----
 *
 * The task wait for the CAN message with 64 as SIDH CAN ID.
 * *****
TASK(CAN_reciever)
{
    unsigned char i;
    RCV_message.CANID = 0x7AA;
    //RCV_message.CANID = 0x1A55AA55; // In 29 bits mode only
    CAN_RCV_Register(&RCV_message);

    while(1)
    {
        WaitEvent(CAN_QUEUE_FULL);
        ClearEvent(CAN_QUEUE_FULL);

        length = RCV_message.length;
        data1 = RCV_message.data[length-2];
        data2 = RCV_message.data[length-1];
    }
}

```

```

C:\PICOS\Project\Presentation\tsk_task0.c
*****
CAN_message_t INJ_Base_rapide;

/*****
 * ----- TASK0 -----
 *
 * First task of the tutorial.
 *
 *****/
TASK(TASK0)
{
    unsigned int EngineRPM;

    INJ_Base_rapide.CANID = 0x0FA;
    CAN_RCV_Register(&INJ_Base_rapide);

    while(1)
    {
        WaitEvent(CAN_QUEUE_FULL);
        ClearEvent(CAN_QUEUE_FULL);

        EngineRPM = INJ_Base_rapide.data[1] << 8;
        EngineRPM += INJ_Base_rapide.data[0];
    }
}

```

```

C:\PICOS\Project\Presentation\tsk_task0.c
* First task of the tutorial.
*
*****
TASK(TASK0)
{
    I2Ctime.year = 05; // 2005
    I2Ctime.month = 8; // August
    I2Ctime.date = 5; // Friday
    I2Ctime.day = 12;
    I2Ctime.hour = 16;
    I2Ctime.minutes = 44;
    I2Ctime.seconds = 25;
    DS1337_Write(&I2Cmsg, &I2Ctime);

    /* Do not forget to declare an alarm in tascdesc.c
    SetRelAlarm(0, 60000, 60000); /* every minute */

    while (1)
    {
        WaitEvent(ALARM_EVENT);
        ClearEvent(ALARM_EVENT);

        DS1337_Read(&I2Cmsg, &I2Ctime);
    }
}

```



## 1 - Introduction

## 2 – Conception modulaire

- A/ Électronique modulaire
- B/ Logiciel modulaire
- C/ Multi-tâches
- D/ Temps réel

## 3 – Application pratique

- A/ Analyse du CdC
- B/ Utilisation de PICos
- C/ Codage de l'application

## 4 – Pragmatec

- A/ Licence GPL
- B/ Aide au développement
- C/ Évolutions futures

## 5 - Conclusions

```

C:\PICOS\Project\Presentation\tsk_task0.c
/*****
/*
/* File name: tsk_task0.c
/*
/*
/* Since: 2005-April-18
/*
/*
/* Version: PICos v1.03 - RTOS for PIC24/30/33 families.
/* Copyright (C) 2005-2006 Pragmatec.
/*
/*
/* Author: Designed by Pragmatec S.A.R.L. www.pragmatec.net
/* MONTAGNE Xavier [XM] xavier.montagne@pragmatec.net
/*
/* Purpose: First task of the tutorial.
/*
/*
/* Distribution: This file is part of PICos.
/* PICos is free software; you can redistribute it
/* and/or modify it under the terms of the GNU General
/* Public License as published by the Free Software
/* Foundation; either version 2, or (at your option)
/* any later version.
/*
/*
/* PICos is distributed in the hope that it will be
/* useful, but WITHOUT ANY WARRANTY; without even the
/* implied warranty of MERCHANTABILITY or FITNESS FOR A
/* PARTICULAR PURPOSE. See the GNU General Public
/* License for more details.
/*
/*
/* You should have received a copy of the GNU General
/* Public License along with gpsim; see the file
/* COPYING.txt. If not, write to the Free Software
/* Foundation, 59 Temple Place - Suite 330,
/* Boston, MA 02111-1307, USA.
/*
/*
/* > A special exception to the GPL can be applied should
/* you wish to distribute a combined work that includes
/* PICos, without being obliged to provide the source
/* code for any proprietary components.
/*
/*
/* History:
/* 2005/04/18 [XM] Create this file.
/*
/*
*****/

```

Descriptif  
du fichier

Licence  
GPL

Avenant  
commercial

## 1 - Introduction

## 2 – Conception modulaire

- A/ Électronique modulaire
- B/ Logiciel modulaire
- C/ Multi-tâches
- D/ Temps réel

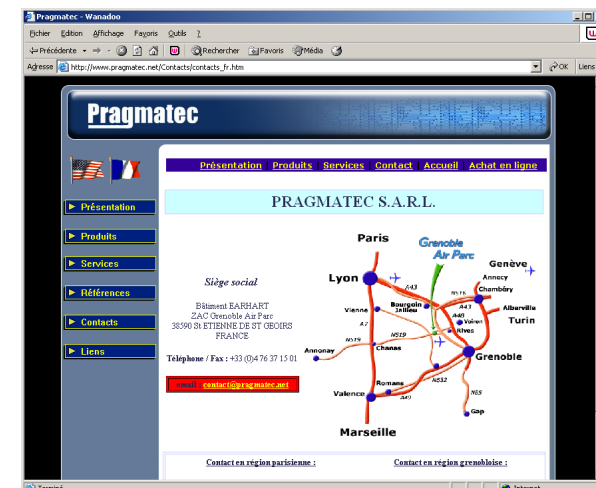
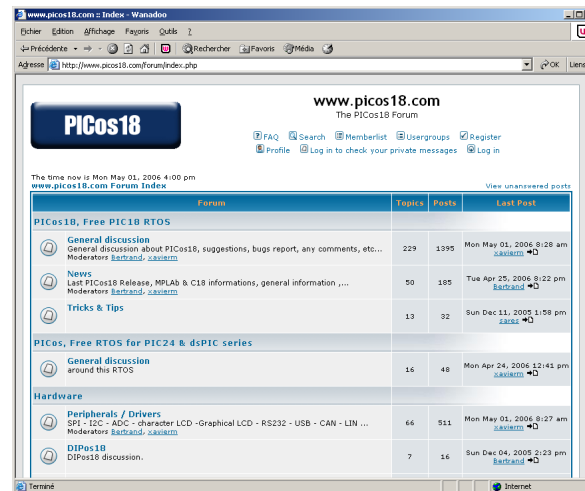
## 3 – Application pratique

- A/ Analyse du CdC
- B/ Utilisation de PICos
- C/ Codage de l'application

## 4 – Pragmatec

- A/ Licence GPL
- B/ Aide au développement
- C/ Évolutions futures

## 5 - Conclusions



## 1 - Introduction

## 2 – Conception modulaire

- A/ Électronique modulaire
- B/ Logiciel modulaire
- C/ Multi-tâches
- D/ Temps réel

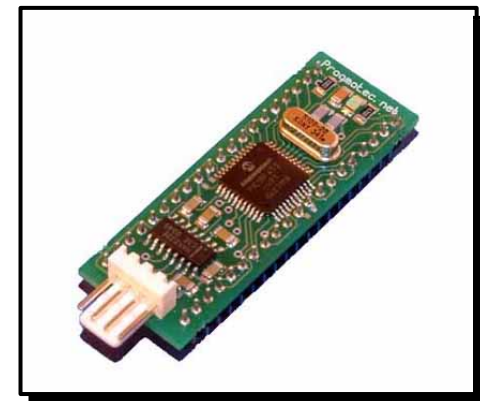
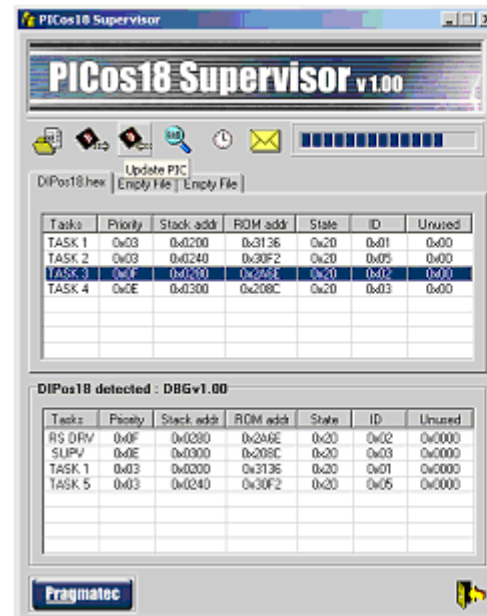
## 3 – Application pratique

- A/ Analyse du CdC
- B/ Utilisation de PICos
- C/ Codage de l'application

## 4 – Pragmatec

- A/ Licence GPL
- B/ Aide au développement
- C/ Évolutions futures**

## 5 - Conclusions



1 - Introduction

2 – Conception modulaire  
A/ Électronique modulaire  
B/ Logiciel modulaire  
C/ Multi-tâches  
D/ Temps réel

3 – Application pratique  
A/ Analyse du CdC  
B/ Utilisation de PICos  
C/ Codage de l'application

4 – Pragmatec  
A/ Licence GPL  
B/ Aide au développement  
C/ Évolutions futures

**5 - Conclusions**

# L'intérêt de PICos

- Solution entièrement gratuite
- Aucune royaltie
- Code éprouvé (nombreux utilisateurs)
- Intégration facilitée par le multi-tâches
- Rigueur du temps réel