BREVET DE TECHNICIEN SUPÉRIEUR
INFORMATIQUE ET RÉSEAUX
POUR L'INDUSTRIE ET LES SERVICES TECHNIQUES

ÉTUDE D'UN SYSTÈME INFORMATISÉ

Session 2013

———

DURÉE: 6 HEURES
Coefficient 5

———

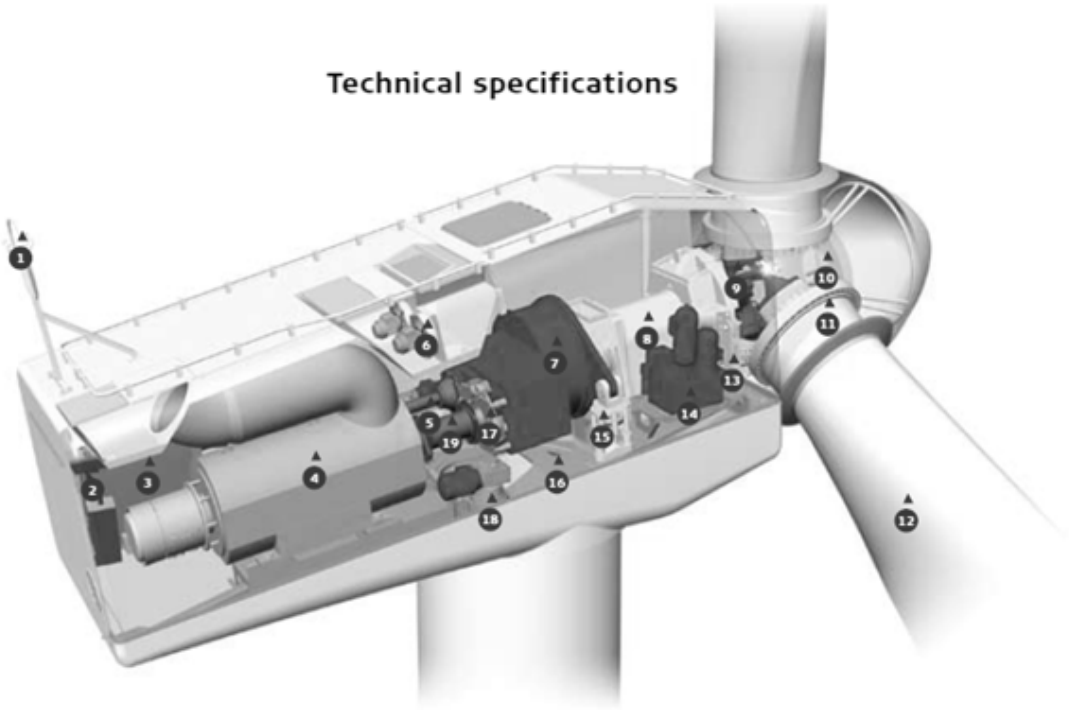
ANNEXES

(22 PAGES)

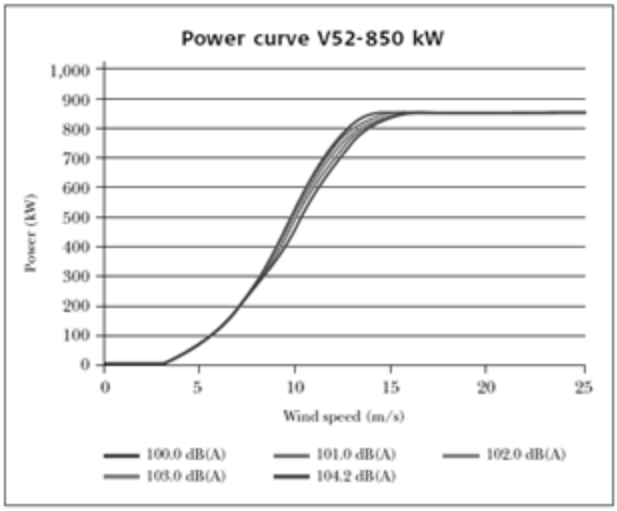
Annexe 1:	Éolienne VESTAS V52 (2 pages) ……	2
Annexe 2:	Capteur Vent (4 pages)……	4
Annexe 3:	XML, Parser TinyXML, STL (1 page)……	8
Annexe 4:	PC 104 - Carte mère et FreeRTOS (5 pages)……	9
Annexe 5:	Architecture réseau nacelle (2 pages)……	14
Annexe 6:	PC 104 – Modules de communication (1 page) ……	16
Annexe 7:	Carte Module PCM-3610 (extrait 2 pages)……	17
Annexe 8	Carte Module PCM-3680 (extrait 2 pages)……	19
Annexe 9:	Controller Can-Bus SJA 1000 (extrait 2 pages)……	21

Annexe 1 : Éolienne Vestas V52

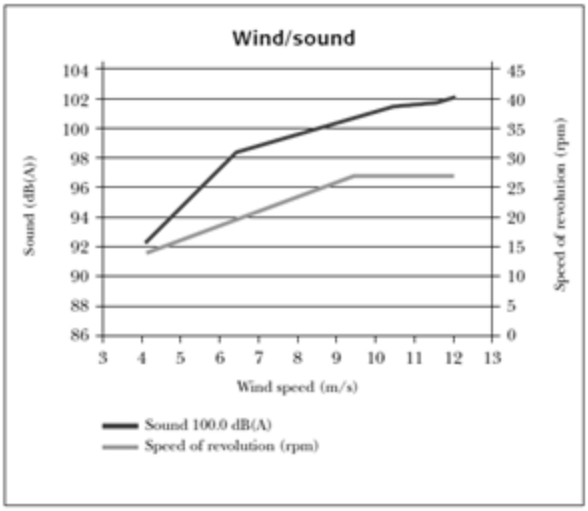
Technical specifications



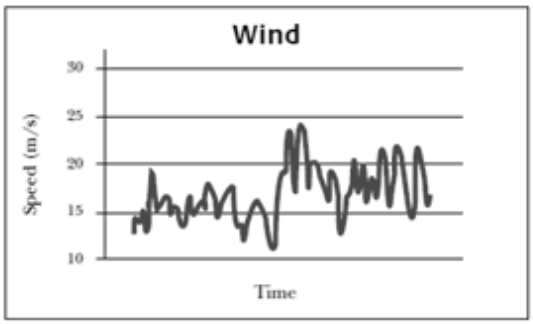
1. Ultrasonic wind sensor
2. Service crane
3. VMP-Top controller with converter
4. OptiSpeed® generator
5. Pitch cylinder
6. Oil and water coolers
7. Gearbox
8. Main shaft
9. Pitch system
10. Blade hub
11. Blade bearing
12. Blade
13. Rotor lock system
14. Hydraulic unit
15. Torque arm
16. Machine foundation
17. Mechanical disc brake
18. Yaw gear
19. Composite disc coupling

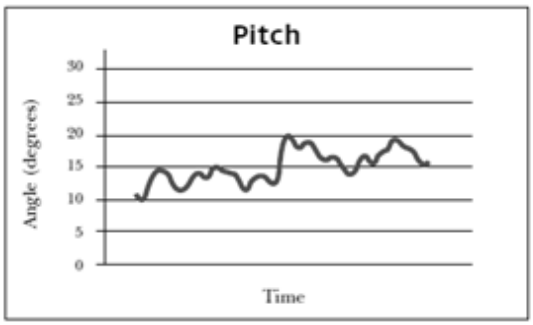


The figure above illustrates the power curves at different sound levels for the V52-850 kW turbine, which is equipped with OptiSpeed®.



The sound output level can be adjusted by varying the revolution speed of the turbine as illustrated in the figure above. It clearly shows the sound level advantages of lower speeds of revolution because the sound level is approximately 7 dB(A) lower at 4 m/s than at 8 m/s. For other sound levels, the benefit can be as much as 10 dB(A). Please note that a decrease of 3 dB(A) represents a halving of the sound level.





Rotor

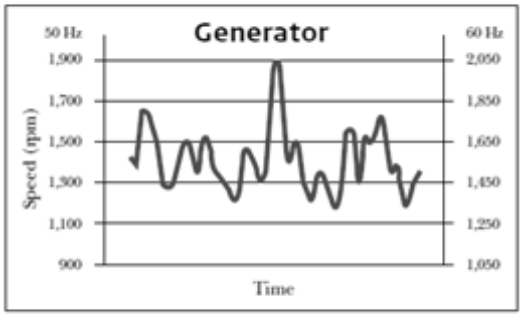
Diameter:	52 m
Area swept:	2,124 m²
Nominal revolutions:	26 rpm
Operational interval:	14.0-31.4 rpm
Number of blades:	3
Power regulation:	Pitch/OptiSpeed®
Air brake:	Full blade pitch

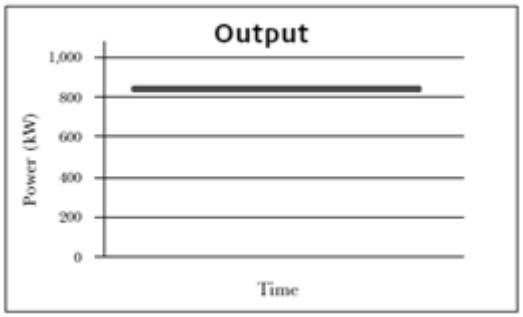
Tower

Hub height:	40 m, 44 m, 49 m, 55 m, 60 m, 65 m, 74 m, 86 m

Operational data

Cut-in wind speed:	4 m/s
Nominal wind speed:	16 m/s
Cut-out wind speed:	25 m/s





Generator

Type:	Asynchronous with OptiSpeed®
Nominal output:	850 kW
Operational data:	50 Hz/60 Hz 690 V

Gearbox

Type:	1 planet step/2-step parallel axle gears

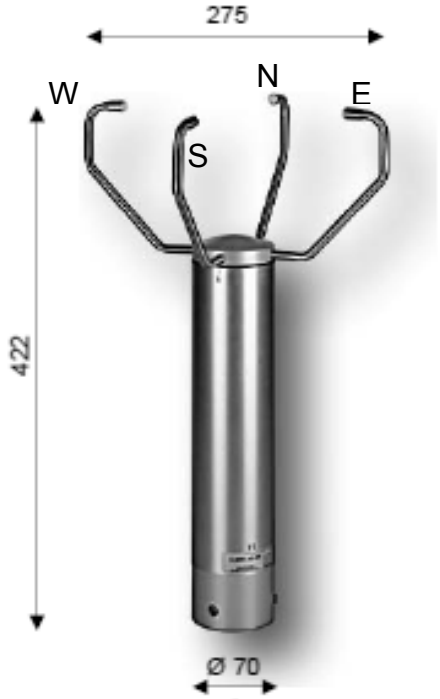
Control

Type:	Microprocessor-based monitoring of all turbine functions as well as OptiSpeed® output regulation and OptiTip® pitch regulation of the blades.

Weight

Nacelle:	22 t
Rotor:	10 t

Annexe 2 : Capteur Vent



Les capteurs adaptés au givrage utilisent un anémomètre ultrasonique pouvant se réchauffer facilement. Ces équipements sont principalement destinés à être installés en montagne.

Les capteurs réalisent des mesures de vitesse et de direction du vent 2 fois par secondes et effectuent les calculs nécessaires à l'exploitation du vent conformément aux règles de l'OMM* et de l'OACI**.

Principaux calculs effectués :
Vitesse maximum du vent et sa direction associée, calculées sur une et 10 minutes
Variation calculée sur 10 minutes de la vitesse et de la direction du vent
Vitesse moyenne du vent et l'argument du vecteur vent moyen sur 2 et 10 minutes
Vent passé sur une minute
Écart type de la vitesse et de la direction du vent sur 10 minutes
Indicateur de grain.

Les capteurs intelligents gèrent le protocole Cibus et transmettent les données à un afficheur pour visualiser les valeurs instantanées du vent

Câble : longueur 10m

Cette chaîne de mesure du vent est utilisée dans le réseau national de **Météo-France**

Caractéristiques du capteur :

Vitesse :
Étendue de mesure : 0-65m/s
Précision : ± 0.1m/s jusqu'à 5m/s
± 2% au-delà de 5m/s

Direction :
Étendue de mesure : 0-360°
Précision : ± 1°

Domaine d'utilisation : -40°C ....70°C
Réchauffage : 24V - 70W

Fixation sur tube $1^{1/2}$" (40/49 en mm)



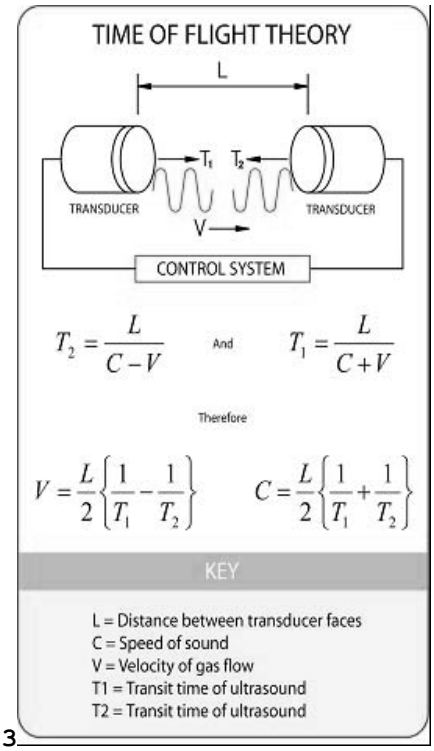
*OMM Organisation Météorologique Mondiale
**OACI Organisation de l'Aviation Civile Internationale

1



Figure 1: Le capteur vent : vitesse et direction.

2



3

Figure 2 : Calcul de la vitesse et direction du vent.

4***Extrait du manuel d'utilisation du capteur.***

Communication

***1*Protocole CIBus :**

Le protocole CIBus est décrit dans le

5**Cahier des Spécifications Techniques**

YB8 1400 B 7400

Protocole d'échange réduit CIBUS entre capteurs interactifs et systèmes d'acquisition

1Requête du centralisateur

6<ENQ> N <SOH> 0 00 A <EOT><CRC>

Le caractère N prend successivement les valeurs .0. .1, …...9. .A. B, .M. .0….

14.4.1.2.Réponse transmise par le capteur, CIBus version 3

7<SOH>Raa A<STX>Cx VSS DDFF1 DDFF2 DDFF3 ddDD fffFFF DDFF4 VVVV<EOT><CRC>

Raa Adresse physique du capteur dans le réseau CIBUS
A Identification de message de données minutes
Cx Identification logique de la mesure de vent
VSS Version du message du capteur vent/status public/status privé (V= 3)
DDFF1 Vent moyen 10 min
DDFF2 Vent instantané : maximum dans les 10 min précédentes
DDFF3 Vent moyen 2 min
ddDD Vent moyen 3 s : domaine de variation des directions dans les 10 min précédentes
fffFFF Vent moyen 3 s : domaine de variation des vitesses dans les 10 min précédentes
DDFF4 Vent instantané : maximum dans la minute précédente ;
VVVV Vent passé dans la minute précédente.

14.4.1.3. Réponse transmise par le capteur, CIBus version 4

2<SOH>Raa A<STX>Cx VSS DDFF1 DDFF2 DDFF3 ddDD fffFFF DDFF4 G sdff sdd<EOT><CRC>

Raa Adresse physique du capteur dans le réseau CIBUS
A Identification de message de données minutes
Cx Identification logique de la mesure de vent

VSS Version du message du capteur vent/status public/status privé (V=4)
DDFF1 Vent moyen 10 min
DDFF2 Vent instantané : maximum dans les 10 min précédentes
DDFF3 Vent moyen 2 min
ddDD Vent moyen 3 s : domaine de variation des directions dans les 10 min précédentes
fffFFF Vent moyen 3 s : domaine de variation des vitesses dans les 10 min précédentes
DDFF4 Vent instantané : maximum dans la minute précédente ;
G Indicateur de grain, codé 1 si présent, 0 sinon
sdff Ecart type de la force du vent exprimé en 1/100ème de m/s
sdd Ecart type de la direction en °

Codage status public

Codage	Désignation
B1B0 = 00	Capteur HS
B1B0 = 01	DD HS
B1B0 = 10	FF HS
B1B0 = 11	Capteur OK
B2 = 0	EEPROM_CI non programmée ou HS.
B3 = 1	Forcé à 1
B4 = 0	Alimentation locale < 6V
B5 = 1	Forcé à 1
B6 = 1	Forcé à 1
B7 = 1	Non utilisé, forcé à 1

1Codage status privé

Codage	Désignation
B0 = 0	Courant consommé > 6mA
B1 = 1	Forcé à 1
B2 = 1	Forcé à 1
B3 = 1	Forcé à 1
B4 = 1	Forcé à 1
B5 = 1	Forcé à 1
B6 = 1	Forcé à 1
B7 = 0	Alerte grain

1Diffusion de données sur le 2ème canal

Ce 2ème canal permet de connecter des afficheurs autonomes.
L'unité centrale émet spontanément toutes les 500 ms une trame dont le format est le suivant :

<SOH>RaaTCiDATACxDATA<EOT>CRC

R numéro de réseau.
aa adresse dans le réseau.
T Identificateur de trame.
Ci,Cx identificateur de données vents (Ci : données instantanées, Cx : autres données).
DATA données.

La direction DD ou DDi est exprimée en dizaine de degrés (2 caractères ASCII),
une direction invalide ou non calculée est codée : //.,
une direction non significative ou dite de vent calme est codée.00
La force FFF ou FFi est exprimée en dixième de m/s (3 caractères ASCII),

une force invalide ou non calculée est codée : ///
Le vent passé VVVV est exprimé en mètres (caractère ASCII),
un vent passé non calculé ou invalide est codé : ////
L'émission de données s'effectue par l'émission de 8 trames consécutives.

Trame n° 1 : trame d'identification avec les arguments suivants :
2<SOH>Raa I CiDDFFi CI QFU<EOT>CRC

DD direction instantanée.
FFi force instantanée.
QFU nom du site exprimé sur 4 caractères ASCII.

Trame 2 : trame de status avec les arguments suivants :
3<SOH>Raa S CiDDFFi CS VSS<EOT>CRC

DD direction instantanée.
FFi force instantanée.
V version logicielle du capteur (3 ou 4)
SS status public et status privé.

Trame n° 3 à 8 : trame des mesures avec les arguments suivants :
4<SOH>Raa M CiDDFFi Cx DATA<EOT>CRC

DD direction instantanée.
FFi force instantanée.
X et **DATA** suivant le tableau suivant.

X	DATA
A	DDFFF (vent moyen 2'.)
B	DDFFF vent moyen 10')
C	ddDD (secteur de variabilité direction moyenne 3 » sur 10')
D	fffFFF (secteur de variabilité force et moyenne 3 » sur 10')
E	DDFFF (max vent instantané. 1')
F	VVVV (vent passé sur la minute)

Les paramètres force et direction instantanés transmis dans chaque trame ne sont pas affectés par la condition dite de vent calme.

CRC = CRC sur 16 bits avec polynôme générateur du type

$$X^{16} + X^{12} + X^{5} + 1.$$

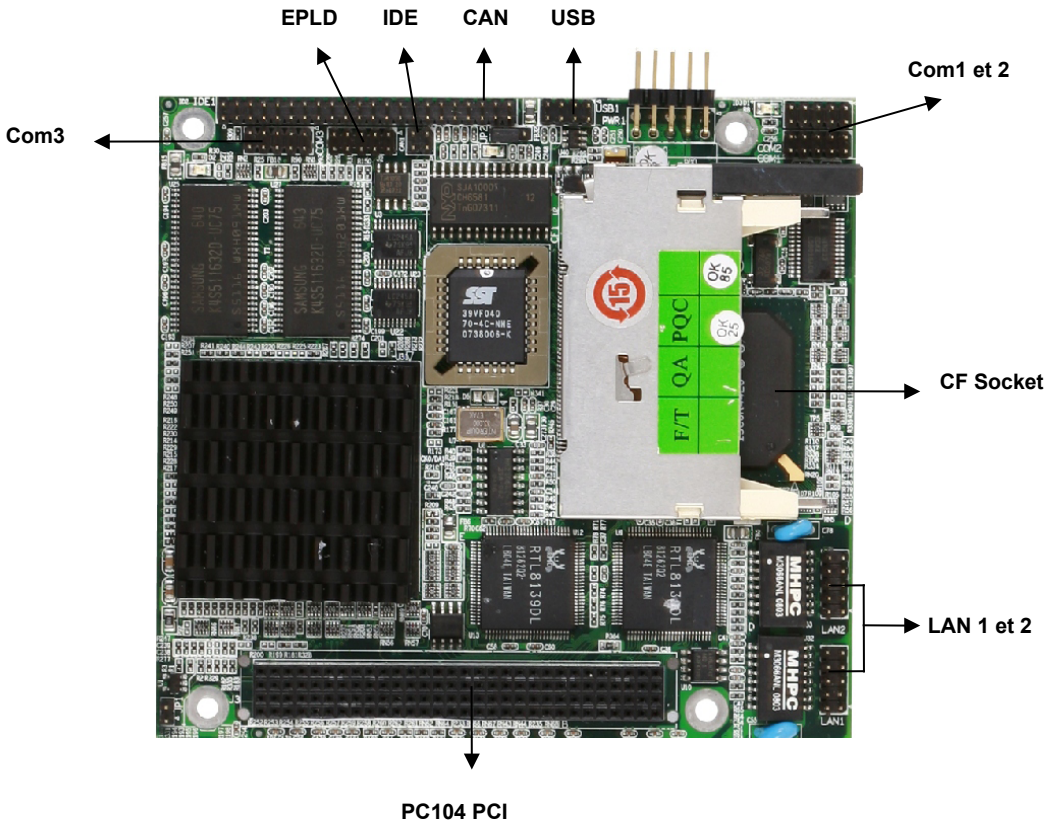
Le CRC est calculé du caractère SOH au caractère EOT inclus.
1
2Caractéristiques:
Modulation FSK-V23 centrée à 1700 Hz avec déplacement de fréquence de ±400 Hz. Au repos, la fréquence de 1300 Hz est transmise en permanence. Un boitier supplémentaire permet d'adapter cette modulation à une liaison série RS232. Les paramètres de transmission sont : 1200 bauds, 8 bits, sans parité, 1 stop.

Annexe 3 : Fichier XML, Parser TinyXML, Librairie STL

Extrait du fichier XML: evt_arret.xml

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<EVT_ARRET>
        <EOLIENNE NUM_EOLIENNE="1">
                <DATE>14:57:36</DATE>
                <VITESSE_VENT>4.8</VITESSE_VENT>
                <ORIENTATION>31:54</ORIENTATION>
                <VITESSE_ORIENTATION>0.3</VITESSE_ORIENTATION>
        </EOLIENNE>
        <EOLIENNE NUM_EOLIENNE="9">
                <DATE>14:59:37</DATE>
                <VITESSE_VENT>4.2</VITESSE_VENT>
                <ORIENTATION>28:60</ORIENTATION>
                <VITESSE_ORIENTATION>0.5</VITESSE_ORIENTATION>
        </EOLIENNE>
        <EOLIENNE NUM_EOLIENNE="6">
                <DATE>15:29:13</DATE>
                <VITESSE_VENT>5.6</VITESSE_VENT>
                <ORIENTATION>34:12</ORIENTATION>
                <VITESSE_ORIENTATION>1.3</VITESSE_ORIENTATION>
        </EOLIENNE>
        <EOLIENNE NUM_EOLIENNE="2">
                <DATE>16:14:18</DATE>
                <VITESSE_VENT>5.3</VITESSE_VENT>
                <ORIENTATION>29:35</ORIENTATION>
                <VITESSE_ORIENTATION>0.6</VITESSE_ORIENTATION>
        </EOLIENNE>
        <EOLIENNE NUM_EOLIENNE="7">
                <DATE>16:35:54</DATE>
                <VITESSE_VENT>5.8</VITESSE_VENT>
                <ORIENTATION>18:36</ORIENTATION>
                <VITESSE_ORIENTATION>0.1</VITESSE_ORIENTATION>
        </EOLIENNE>
        <EOLIENNE NUM_EOLIENNE="1">
                <DATE>17:18:04</DATE>
                <VITESSE_VENT>3.8</VITESSE_VENT>
                <ORIENTATION>25:21</ORIENTATION>
                <VITESSE_ORIENTATION>1.7</VITESSE_ORIENTATION>
        </EOLIENNE>
        <EOLIENNE NUM_EOLIENNE="9">
                <DATE>18:03:45</DATE>
                <VITESSE_VENT>4.5</VITESSE_VENT>
                <ORIENTATION>13:37</ORIENTATION>
                <VITESSE_ORIENTATION>0.8</VITESSE_ORIENTATION>
        </EOLIENNE>
</EVT_ARRET>
```

Annexe 4 : PC104 – Carte Mère 104-3511CD2N et FreeRTOS



Features

- Onboard MOTOROLA MPC8245 350MHz processor **- RISC design**
- Onboard 256MB system memory, 64MB Flash ROM
- Two 10/100Mbps Ethernet controllers
- One PCI-104 connector, one CAN bus connector
- Three serial ports, two USB1.0 ports
- +5V, +12V power supply

Operating system

Embedded OS Vxworks/Linux patch FreeRTOS

Exemple d'applications FreeRTOS:

Une application FreeRTOS aura généralement la forme suivante :
Initialisation sommaire du matériel (assez pour les deux étapes suivantes)
Copie de la section .data et initialisation de la section .bss
Exécution du programme principal (à partir de l'étape suivante)
Fin de l'initialisation du matériel (horloges supplémentaires, wait states, etc.)
Initialisation des périphériques
Création des structures FreeRTOS (tâches et structures de contrôle)
Démarrage de l'ordonnanceur

Ce premier exemple listing 1 est une application simple mettant en œuvre deux tâches indépendantes **fonctionnant concurremment. Le but est de montrer comment à l'installation d'un environnement multitâche, on définit et créer les tâches, lancer un ordonnanceur de préemption, aussi bien que** montrer quelques fonctions de contrôle de temps de tâche.

Dans le premier exemple il y a deux tâches indépendantes La première appelée LEDS fait juste clignoter des leds à une fréquence fixe. La seconde appelée 'la TEMPERATURE' demande périodiquement au capteur externe de température une mesure, pour l'afficher sur un PC via une liaison série. Les deux tâches fonctionnent avec la même priorité et l'ordonnanceur peut anticiper les tâches pour faire exécuter .l'autre

La tâche **vLEDTask** configure d'abord le port de sortie matériel et initialise une variable PRINCIPALE elle entre alors dans la boucle infinie où elle active les LEDS selon la variable d'état, incrémente cette variable, demande ensuite à l' ordonnanceur de bloquer la tâche pour un temps donné.

La tâche **vTempTask** initialise d'abord la communication UART avec de capteur de température, Le capteur température demande un échantillonnage continu et entre dans la boucle infinie. Dans la boucle, la tâche lit la température du capteur, l'envoie sur le port série puis attend un temps donné.

La fonction principale du programme doit ajouter les tâches à l'ordonnanceur et le commencer. Ceci est fait utilisant les fonctions de vTaskStartScheduler et le xTaskCreate. Voici le code complet du fichier de main.c.

```
/**
 * \file main.c
 */
#include "Projet.h"

/* Hardware initialization */
static void prvSetupHardware( void );
static void vLEDTask(void* pvParameters);
static void vTempTask(void* pvParameters);

/**
 * Putchar function required by stdio.h module to be able to use printf()
 */
int putchar(int c)
{
   return uart0_putchar(c);
}

/* The main function. */
int main( void )
{
   /* Setup the hardware. */
   prvSetupHardware();

   /* Add the two tasks to the scheduler */
```

```
    xTaskCreate(vLEDTask, "LED", configMINIMAL_STACK_SIZE, NULL, 1, NULL );
    xTaskCreate(vTempTask, "Temperature", configMINIMAL_STACK_SIZE, NULL, 1, NULL );


    /* Start the scheduler. */
    vTaskStartScheduler();

    /* As the scheduler has been started we should never get here! */
    return 0;
}

/* Initialize the main hardware parameters. */
static void prvSetupHardware( void )
{
…..

/* Rôle de la tâche*/
…….

}

/* The LED task function.It increments a variable and displays its value on the 3 LEDs, then waits for a given delay and start over again.*/
 * \param pvParameter NULL is passed as parameter.
 */
static void vLEDTask(void* pvParameters)
{
…..

/* Rôle de la tâche*/
…….
}

/* The temperature measurement task function. It reads the temperature from the sensor and print it on the serial port.*/
 * \param pvParameters NULL is passed, unused here.
 */
static void vTempTask(void* pvParameters)
{
…..

/* Rôle de la tâche*/
…….
}
```

Listing 1

Task synchronization

Ce deuxième exemple est une application simple montrant une tâche qui se synchronise par l'intermédiaire d'un sémaphore avec une routine de service s'interrompant..

La solution utilisée est un sémaphore binaire.

Sémaphores binaires sont utilisées à la fois pour l'exclusion mutuelle et à des fins de synchronisation.

Sémaphores binaires et les mutex sont très similaires, mais ont quelques différences subtiles: Mutex inclue un mécanisme d'héritage de priorité, les sémaphores binaires n'en ont pas. Cela rend les sémaphores binaires le meilleur choix pour la mise en œuvre de la synchronisation (entre les tâches ou entre les tâches et une interruption) et des mutex le meilleur choix pour la mise en œuvre simple, l'exclusion mutuelle. La description de la façon dont un mutex peut être utilisé comme un mécanisme d'exclusion mutuelle détient également des sémaphores binaires

Les Fonctions API de sémaphores permettent un temps de blocage à préciser. Le temps de blocage indique le nombre maximal de «tiques» que la tâche doit attendre dans l'état bloqué lorsque l'on tente de «prendre» un sémaphore, le sémaphore ne doit ne pas être immédiatement disponible. Si une tâche sur le même sémaphore que la tâche en cours tente d'accéder à la même zone alors la tâche avec la plus haute priorité sera la tâche qui sera servie la prochaine fois que le sémaphore devient disponible.

Les macros pdTrue indique que le sémaphore est occupé et pdFalse que le sémaphore est libre
Le paramètre xBlockTime de la première macro est le nombre maximal de ticks pour attendre que le sémaphore soit disponible. Il correspond au même paramètre avec les fonctions de file d'attente.

La fonction xSemaphoreGiveFromISR libère le sémaphore en présence d'une interruption.
.

Voici le code complet.

```
#include "Projet.h"

/* Function Prototypes */
static void prvSetupHardware( void );
static void vLEDTask(void* pvParameters);
static void rx_char_cb(uint8_t c);

int putchar(int c)
{
   return uart0_putchar(c);
}

/* Global Variables */
xSemaphoreHandle xSemaphore;

/* The main function. */
int main( void )
{
   /* Setup the hardware. */
   prvSetupHardware();

   /* Create the Semaphore for synchronization between UART and LED task */
   vSemaphoreCreateBinary( xSemaphore)

   /* Add the only task to the scheduler */
   xTaskCreate(vLEDTask, "LED", configMINIMAL_STACK_SIZE, NULL, 1, NULL );

   /* Start the scheduler. */
```

```
    vTaskStartScheduler();

    /* As the scheduler has been started we should never get here! */
    return 0;
}
/ * Initialize the main hardware parameters. */
static void prvSetupHardware( void )
{
    /* Stop the watchdog timer. */
    WDTCTL = WDTPW + WDTHOLD;

    /* Setup MCLK 8MHz and SMCLK 1MHz */
    set_mcu_speed_xt2_mclk_8MHz_smclk_1MHz();

    /* Configure the UART module for serial communication */
    uart0_init(UART0_CONFIG_1MHZ_115200);
    uart0_register_callback(rx_char_cb);
    printf("type any char to update the LEDs\r\n");

    /* Enable Interrupts */
    eint();
}

/* The LEDs task function.It waits the semaphore is given, then takes it and update the LEDs
 * \param pvParameters NULL is passed, unused here.
 */
static void vLEDTask(void* pvParameters)
{
    uint16_t leds_state = 0;

    /* Initialize the LEDs */
    LEDS_INIT();

    /* Infinite loop */
    while(1)
    {
        /* Increment the LED state */
        leds_state ++;
        /* Block until the semaphore is given */
        xSemaphoreTake(xSemaphore, portMAX_DELAY);
        /* update the LEDs and loop */
        LEDS_SET(leds_state);
    }
}

/* The callback function called when a char is received. It gives the semaphore.
 * \param c the received char
 */
static void rx_char_cb(uint8_t c)
{
    static portBASE_TYPE xHigherPriorityTaskWoken;
    /* Give the semaphore */
    xSemaphoreGiveFromISR(xSemaphore, &xHigherPriorityTaskWoken);
}
```

Listing 2

Annexe 5 : Architecture réseau Nacelle.

Definition of Terms

CT316	The processor dedicated to VCMS
CT317	Connection box for the RS485 network.
CT5200	High sensitive accelerometer for VCMS
CT5210	Microphone for VCMS
CT5221	Tower accelerometer
CT5230	Low sensitive accelerometer for VCMS
VCMP	See CT316
VCMS	Vestas Condition Monitoring System
VCPM	Vestas Converter Processor Module. Variable speed processor.
VMP	Vestas Multi Processor controller

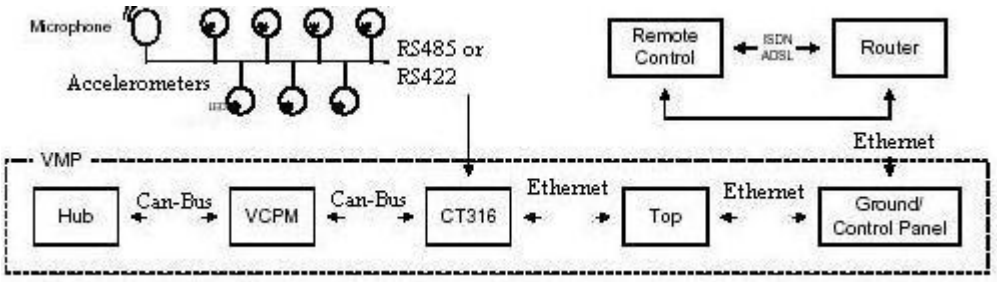
Processor unit:

The hardware consists of 2 CT316 (processor unit), 3 CT317, 6 high sensitive accelerometers (CT5200), two low sensitive accelerometers (CT5230) and a microphone (CT5210).

The CT316 is similar to the other processors in the turbine placed on the ARCnet or Ethernet, which it uses to communicate possible warnings and alarms and receive operational data from the central turbine monitor (VMP).

The CT316 handles all calculations in connection with VCMS and this has the advantage that the significantly large number of calculations does not burden the rest of the processors in the VMP. The processor contains 16 MB RAM and 32 MB flash disc.

Therefore, a second CT316 is placed in the hub in order to collect and process data from these sensors. This is only a slave processor which only is used to sample data from the nodes and transfer them to the master CT316 in the top control panel.



RS485 or RS422 network:

The network between the accelerometers, the microphone and the CT316 has been chosen especially to improve signal reliability in the noisy environment of the turbine.
In order to ensure that the network is free of reflections, the sensors are connected in series and terminated in both ends. However the network will still be functional if e.g. the power is disconnected from one of the sensors. The reason is that there is a direct physical connection through the accelerometer and the transceiver circuit is connected parallel to the network.
By measuring the voltage before every network command broken cables can be found. The CT317 is a connection box, in which it is possible to disable the connection to one or more sensors.

Figure 4: RS485 network principle.

SPECIFICATIONS		RS232	RS423	RS422	RS485
Mode of Operation		SINGLE -ENDED	SINGLE -ENDED	DIFFERENTIAL	DIFFERENTIAL
Total Number of Drivers and Receivers on One Line (One driver active at a time for RS485 networks)		1 DRIVER 1 RECVR	1 DRIVER 10 RECVR	1 DRIVER 10 RECVR	32 DRIVER 32 RECVR
Maximum Cable Length		50 FT.	4000 FT.	4000 FT.	4000 FT.
Maximum Data Rate (40ft. - 4000ft. for RS422/RS485)		20kb/s	100kb/s	10Mb/s-100Kb/s	10Mb/s-100Kb/s
Maximum Driver Output Voltage		+/-25V	+/-6V	-0.25V to +6V	-7V to +12V
Driver Output Signal Level (Loaded Min.)	**Loaded**	+/-5V to +/-15V	+/-3.6V	+/-2.0V	+/-1.5V
Driver Output Signal Level (Unloaded Max)	**Unloaded**	+/-25V	+/-6V	+/-6V	+/-6V
Driver Load Impedance (Ohms)		3k to 7k	>=450	100	54
Max. Driver Current in High Z State	**Power On**	N/A	N/A	N/A	+/-100uA
Max. Driver Current in High Z State	**Power Off**	+/-6mA @ +/-2v	+/-100uA	+/-100uA	+/-100uA
Slew Rate (Max.)		30V/uS	Adjustable	N/A	N/A
Receiver Input Voltage Range		+/-15V	+/-12V	-10V to +10V	-7V to +12V
Receiver Input Sensitivity		+/-3V	+/-200mV	+/-200mV	+/-200mV
Receiver Input Resistance (Ohms), (1 Standard Load for RS485)		3k to 7k	4k min.	4k min.	>=12k

PCM-3610
PCM-3612
PCM-3614

RS-232/422/485 Module with Isolation Protection

2-port RS-422/485 Module

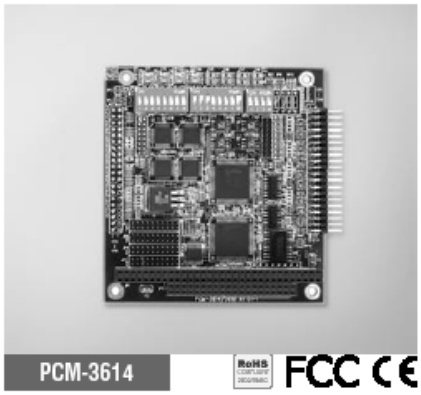
4-port RS-422/485 High-speed Module



PCM-3610



PCM-3612



PCM-3614

Features

- High speed transmission rate
- Automatic RS-485 data flow control
- Jumper selectable interrupt level
- Supports Windows® 98/2000/XP
- Supports WinCE 3.0, 4.2, 5.0
- Powerful and easy-to-use utility (ICOM Tools)

Specifications

General

- Card Interface: PC/104
- Certifications: CE
- Connectors: 2 x DB9-M
- Ports: 2
- Power Consumption: +5 V @ 400 mA typical, ±12 V @ 950 mA max

Communications

- Channel 1: RS-232, 422, or 485
- Channel 2: RS-422, or RS-485
- Character Length: 5, 6, 7, or 8 bits
- IRQ: 3, 4, 5, 6, 7, 9
- Parity: Even, odd, or none
- Speed: 50 bps ~ 115.2 kbps
- Stop Bit: 1, 1.5, or 2

Protection

- Isolation Protection: 1,000 V_{DC}

Environment

- Humidity (Operating): 0 ~ 90 % RH
- Operating Temperature: 0 ~ 65° C (32 ~ 149° F)
- Storage Temperature: -40 ~ 85° C (-40 ~ 185° F)

Ordering Information

- PCM-3610-B: Isolated RS-232/422/485 Module

Features

- Long distance communication
- Automatic RS-485 data flow control
- Jumper selectable interrupt level
- Supports Windows® 98/2000/XP
- Supports WinCE 3.0, 4.2, 5.0
- Powerful and easy-to-use utility (ICOM Tools)

Specifications

General

- Card Interface: PC/104
- Certifications: CE
- Connectors: 2 x DB9-M
- Indicators: Red LED for TX, Green LED for RX
- Ports: 2
- Power Consumption: +5 V @ 400 mA typical

Communications

- Channel 1 and 2: RS-422, or RS-485
- Character Length: 5,6,7, or 8 bits
- IRQ: 3, 4, 5, 6, 7, 9, 10, 11, 12 or 15
- Parity: Even, odd, or none
- Speed: 50 bps ~ 115.2 kbps
- Stop Bit: 1, 1.5, or 2

Environment

- Humidity (Operating): 0 ~ 90 % RH
- Operating Temperature: 0 ~ 65° C (32 ~ 149° F)
- Storage Temperature: -40 ~ 85° C (-40 ~ 185° F)

Ordering Information

- PCM-3612-A: Dual Port RS-422/485 Module

Features

- Automatic RS-485 data flow control
- Shared IRQ settings for each ports
- LED indicators: TX, RX
- Standard PC ports: COM1, COM2, COM3, COM4 compatible
- Supports Windows® 98/2000/XP
- Supports WinCE 4.2, 5.0
- Powerful and easy-to-use utility (ICOM Tools)

Specifications

General

- Card Interface: PC/104
- Certifications: CE
- Connectors: 4 x DB9-M
- Ports: 4
- Power Consumption: +5 V @ 450 mA

Communications

- Data Bits: 5, 6, 7, 8
- I/O Address Range: 0 x 000 ~ 0 x 3F8
- IRQ: 3, 4, 5, 6, 7, 9, 10, 11, 12, or 15
- Parity: Even, odd, or none
- Data Signals: RS-422: TxD+, TxD-, RxD+, RxD-, CTS+, CTS-, RTS+ and RTS-; RS-485: DATA+, DATA-, CTS+, CTS-
- Speed: 50 bps ~ 921.6 kbps
- Stop Bits: 1, 1.5, 2

Protection

- EFT Protection: 1000 V_{DC}
- Termination Resistor: 120 W

Environment

- Humidity (Operating): 0 ~ 90 % RH
- Operating Temperature: 0 ~ 65° C (32 ~ 149° F)
- Storage Temperature: -40 ~ 85° C (-40 ~ 185° F)

Ordering Information

- PCM-3614-A: 4-port RS-422/485 High-speed Module

PCM-3610
PC/104 Isolated Dual-port RS-232 and RS-485/422 Module

Initial inspection

We carefully inspected the PCM-3610 both mechanically and electrically before we shipped it. It should be free of marks and scratches and in perfect electrical order on receipt.

Handle the board only by its edges. The static charge on your body may damage its integrated circuits. Keep the card in its anti-static package whenever it is not installed. You can use this package to return the card if it should need repair.

Switches and jumpers

The following chart shows the switches and jumpers corresponding to each serial interface channel:

Ch. 1	Ch. 2	Function
SW1	SW2	I/O base address
JP2	JP3	Interrupt level
JP4	JP5	RS-485 or RS-422
JP10	N/A	RS-232 or RS-485/422

Switch locations appear in the figure below.

RS-232/422/485 selection (JP4, 5, 10)

Channel 1

RS-232 JP10 up to 232

RS-422 JP10 down to 485/422
JP4 right to RS-422

RS-485 JP10 down to 485/422
JP4 left to RS-485

Note: *You must connect channel 1's external signal cable to JP7 for RS-232 or JP8 for RS-422/485.*

Channel 2

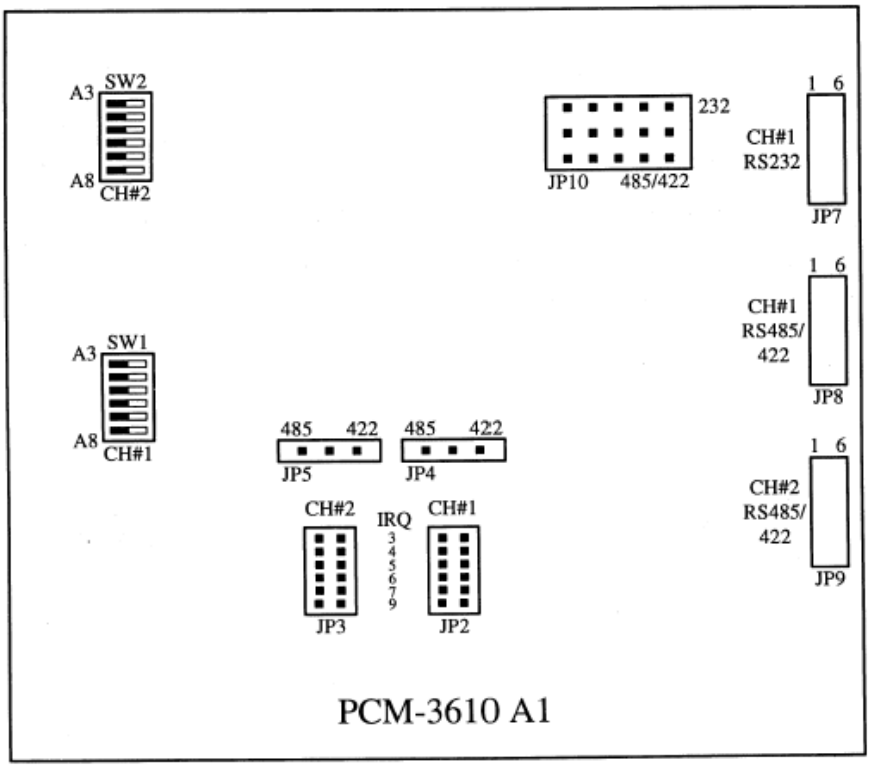
RS-422 JP5 right to RS-422

RS-485 JP5 left to RS-485

Base address (SW1 and SW2)

Two 6-position DIP switches select the I/O port base address for each channel. SW1 controls Ch. 1, and SW2 controls Ch. 2. See the figure below for switch locations.

You can set the base address anywhere from hex 200 to 3F8. The default settings are 3F8 (COM1) for Channel 1 and 2F8 (COM2) for Channel 2. The following table shows switch settings for various base addresses:



PCM-3610 A1

Module I/O addresses (SW1 and SW2)

Range (hex)	Switch position					
	1	2	3	4	5	6
200 - 207	O	O	O	O	O	O
208 - 20F	●	O	O	O	O	O
⋮						
2E8 - 2EF	●	O	●	●	●	O
* 2F8 - 2FF (COM2)	●	●	●	●	●	O
3E8 - 3EF	●	O	●	●	●	●
* 3F8 - 3FF (COM1)	●	●	●	●	●	●

O = On ● = Off * = defaults

NOTE: *Switches 1-6 control the PC bus address lines as follows:*

Switch	1	2	3	4	5	6
Line	A3	A4	A5	A6	A7	A8

Interrupt level-IRQ (JP2 and JP3)

You can set each port for any interrupt level from 3 to 9, except 8. Jumper JP2 controls Ch. 1, and JP3 controls Ch. 2. Simply short the pins on the jumper corresponding to the interrupt level (see figure below).

JP2 (Ch 1, COM1) IRQ JP3 (Ch, 2, COM2) IRQ



If you use the ports as standard COM1 and COM2, you will need to set Channel 1 to IRQ 4 and Channel 2 to IRQ 3.

Baud rate	Divisor
50	2304
75	1536
110	1047
133.5	857
150	768
300	384
600	192
1200	96
1800	64
2000	58
2400	48
3600	32
4800	24
7200	16
9600	12
19200	6
38400	3
56000	2
115200	1

LCR : line control register

LCR : line control register				
Bit	**Value**			**Description**
	Bit 1		**Bit 0**	Data word length
	0		0	5 bits
	0		1	6 bits
	1		0	7 bits
0,1	1		1	8 bits
		0		1 stop bit
2		1		1.5 stop bits (5 bits word) 2 stop bits (6, 7 or 8 bits word)
	Bit 5	**Bit 4**	**Bit 3**	
	x	x	0	No parity
	0	0	1	Odd parity
	0	1	1	Even parity
	1	0	1	High parity (stick)
3,4,5	1	1	1	Low parity (stick)
		0		Break signal disabled
6		1		Break signal enabled
		0		DLAB : RBR, THR and IER accessible
7		1		DLAB : DLL and DLM accessible

PCM-3680 PC/104 Dual Port CAN Interface Module

PCM-3680 PC/104
Dual Port CAN Interface Module

Jumper Setting

The PCM-3680 is a special purpose communication card that brings the Control Area Network to your PC. With the built-in CAN controller, the PCM-3680 provides bus arbitration and error detection with automatic transmission repeat function. This drastically avoids data loss and ensures system reliability. The on-board CAN controllers are located at different positions in the memory. You can run both CAN controllers at the same time, independently. The PCM-3680 operates at baud rates up to 1 Mbps and can be installed directly into the expansion slot of your PC.

Control Area Network

The CAN (Control Area Network) is a serial bus system especially suited for networking "intelligent" I/O devices as well as sensors and actuators within a machine or plant. Characterized by its multi-master protocol, real-time capability, error correction, high noise immunity, and the existence of many different silicon components, the CAN serial bus system, originally developed by Bosch for use in automobiles, is increasingly being used in industrial automation.

Direct Memory Mapping

The PCM-3680 is assigned with memory address, which allows direct access to the CAN controller. This is the simplest and fastest way of programming any board in a PC because the board is regarded as standard RAM.

Optical Isolation Protection

On-board optical isolators protect your PC and equipment against damage from ground loops, increasing system reliability in harsh environments.

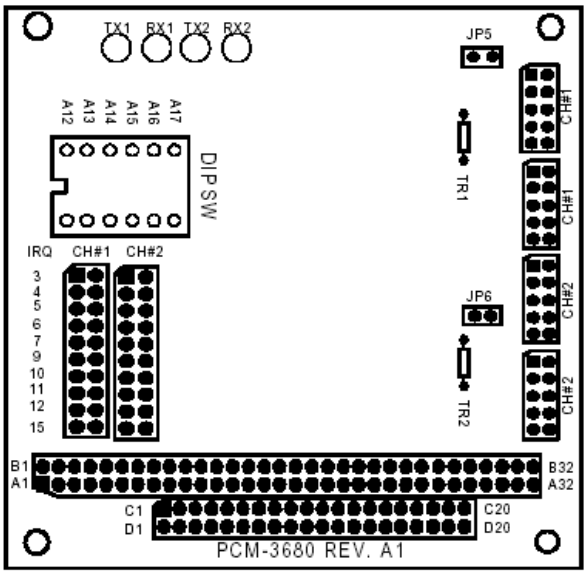
Specifications

- **Ports**: 2
- **CAN controller**: SJA-1000
- **CAN transceiver**: 82C250
- **Signal support**: CAN-L, CAN-H
- **Memory address**: From C800H to EF00H
- **IRQ**: 3, 4, 5, 6, 7, 9, 10, 11, 12, 15
- **Isolation voltage**: 1000 V_{DC}
- **Power consumption**: +5 V @ 400 mA typical, 950 mA max.
- **Connectors**: Dual DB-9 male connectors
- **Operating temperature**: 32 to 122˚F (0 to 50˚C)
- **PC/104 form factor**: 3.6" x 3.8" (90 mm x 96 mm)
- **Shipping weight**: 0.9 lb (0.4 kg)

Features

- Operates 2 separate CAN networks at the same time
- High speed transmission up to 500 kbps
- 16 MHz CAN controller frequency
- Takes a 4 KB address space, 40 base address adjustable in steps from C800H up to EF00H
- Optical isolation protection of 1000 V_{DC} ensures system reliability
- Wide IRQ selection for each port includes: IRQ 3, 4, 5, 6,7, 9, 10, 11, 12, 15
- LED indicates Transmit/Receive status on each port
- Direct memory mapping enables speedy access to the CAN controllers
- C library and examples included

Jumper & Switch Locations



Wiring

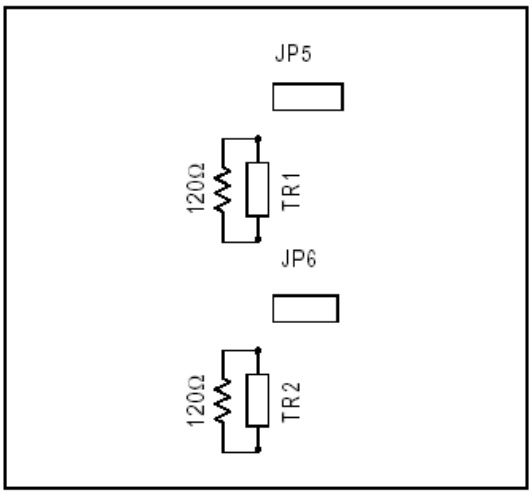
Pin Assignments

The following figure shows the pin assignments for the card's DB-9 connectors.

Termination Resistor Setup

Termination restistors are factory-installed to allow for impedance matching. These resistors can be enabled by using jumpers number 5 and 6 (shown below). Jumper 5 enables the terminal resistor for port 1, while jumper 6 enables the terminal resistor for port 2. The value of the resistor equals the characteristic impedence of the signal wires (approximately 120 Ω)

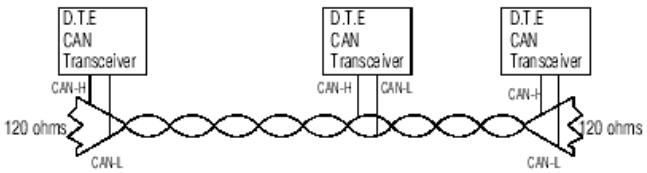
The following figure shows resistor placements. Port TR1 enables jumper JP5, and TR2 enables jumper JP6



CAN termination resistor installation

CAN signal wiring

The CAN standard supports half-duplex communication. This means that just two wires are used to transmit and receive data.



Wiring topograhpy

Wiring connections are as follows:

PCM-3860	DTE (male DB-9)	Terminal DTE
Pin	Signal	Signal
7	CAN-H	CAN-H
3	GND	GND
2	CAN-L	CAN-L

Register Structure

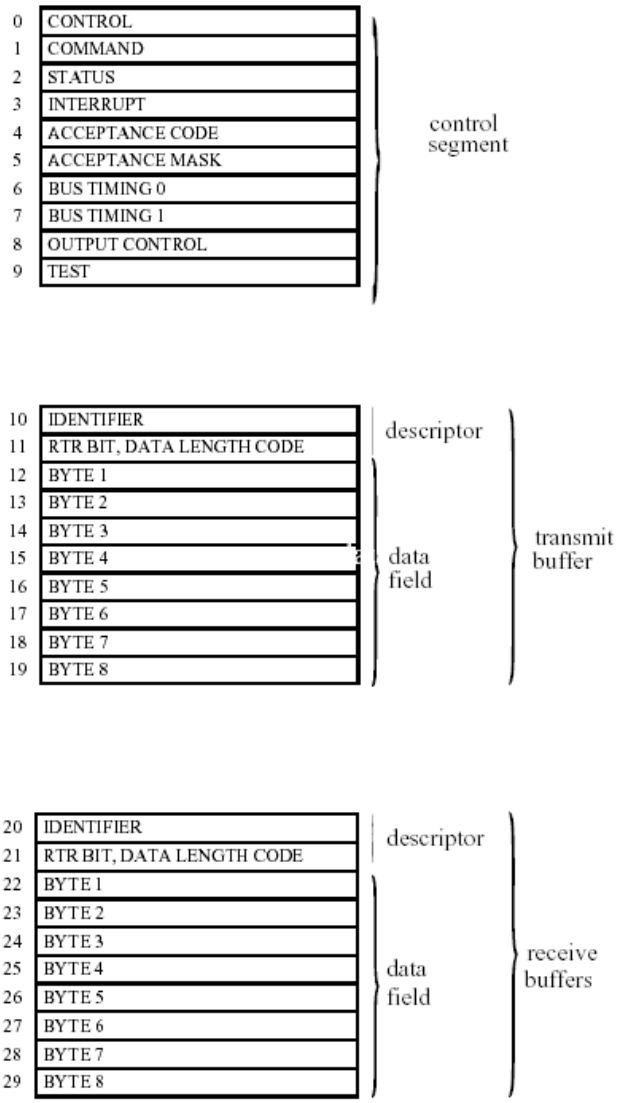
This appendix gives a brief description of the CAN controller registers. For more detailed information, please refer to the *Stand-alone CAN-controller Data Book* from Philips Semiconductors Microcontroller Products. (You ma also find the information on the enclosed disk under the "Manual" directory, in the Word 6.0 file: REGISTER.DOC.)

CAN Controller Address Allocation

Philips PCX82C200 CAN Controller

ADDRESS

Register address map



6.5 Common registers

6.5.1 BUS TIMING REGISTER 0 (BTR0)

The contents of the bus timing register 0 defines the values of the Baud Rate Prescaler (BRP) and the Synchronization Jump Width (SJW). This register can be accessed (read/write) if the reset mode is active.

In operating mode this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a 'FFH' is reflected.

Table 44 Bit interpretation of bus timing register 0 (BTR0); CAN address 6

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

6.5.1.1 Baud Rate Prescaler (BRP)

The period of the CAN system clock t_{scl} is programmable and determines the individual bit timing. The CAN system clock is calculated using the following equation:

$$t_{scl} = 2 \times t_{CLK} \times (32 \times BRP.5 + 16 \times BRP.4 + 8 \times BRP.3 + 4 \times BRP.2 + 2 \times BRP.1 + BRP.0 + 1)$$

where t_{CLK} = time period of the XTAL frequency = $\frac{1}{f_{XTAL}}$

6.5.1.2 Synchronization Jump Width (SJW)

To compensate for phase shifts between clock oscillators of different bus controllers, any bus controller must re-synchronize on any relevant signal edge of the current transmission. The synchronization jump width defines the maximum number of clock cycles a bit period may be shortened or lengthened by one re-synchronization:

$$t_{SJW} = t_{scl} \times (2 \times SJW.1 + SJW.0 + 1)$$

6.5.2 BUS TIMING REGISTER 1 (BTR1)

The contents of bus timing register 1 defines the length of the bit period, the location of the sample point and the number of samples to be taken at each sample point. This register can be accessed (read/write) if the reset mode is active.

In operating mode, this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a 'FFH' is reflected.

Table 45 Bit interpretation of bus timing register 1 (BTR1); CAN address 7

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

6.5.2.1 Sampling (SAM)

BIT	VALUE	FUNCTION
SAM	1	triple; the bus is sampled three times; recommended for low/medium speed buses (class A and B) where filtering spikes on the bus line is beneficial
	0	single; the bus is sampled once; recommended for high speed buses (SAE class C)

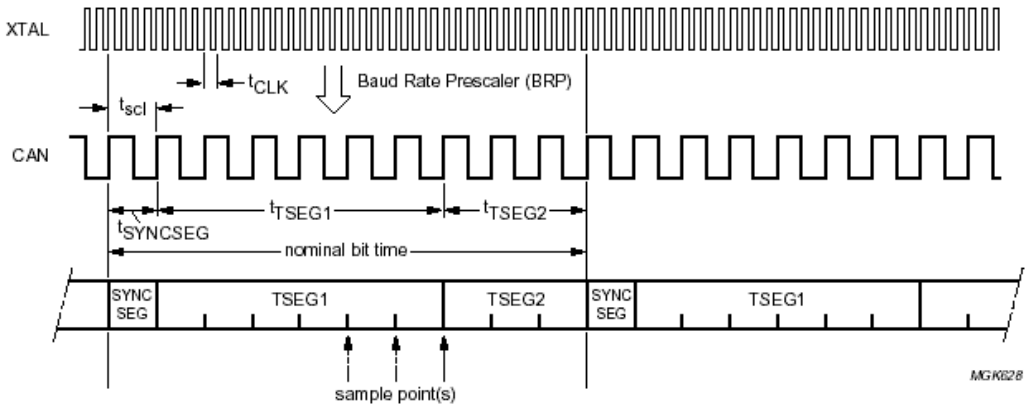
6.5.2.2 Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2)

TSEG1 and TSEG2 determine the number of clock cycles per bit period and the location of the sample point, where:

$t_{SYNCSEG} = 1 \times t_{scl}$

$t_{TSEG1} = t_{scl} \times (8 \times TSEG1.3 + 4 \times TSEG1.2 + 2 \times TSEG1.1 + TSEG1.0 + 1)$

$t_{TSEG2} = t_{scl} \times (4 \times TSEG2.2 + 2 \times TSEG2.1 + TSEG2.0 + 1)$



Possible values are BRP = 000001, TSEG1 = 0101 and TSEG2 = 010.

Fig.13 General structure of a bit period.

6.5.3 OUTPUT CONTROL REGISTER (OCR)

The output control register allows the set-up of different output driver configurations under software control.

This register may be accessed (read/write) if the reset mode is active. In operating mode, this register is read only, if the PeliCAN mode is selected. In BasicCAN mode a 'FFH' is reflected.

Table 46 Bit interpretation of the output control register (OCR); CAN address 8

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	OCPOL0	OCMODE1	OCMODE0