The CyCab: a New Generation of the IT-Car in Europe

Pierre Pomiers, Meftah Ghrissi, Antonella Semerano, Vincent Dupourqué
Robosoft S.A.
Technopole d'Izarbel 64210 Bidart, France
Tel.: +33 5 59 41 53 66; Fax.: +33 5 59 41 53 79
E-mail: pierre@robosoft.fr

Abstract

The CyCab is a new generation of modular electric vehicles, with advanced features such as automatic motions and safety reflexes (real-time distributed control including robotics algorithms), a user-friendly MMI (Man Machine Interface) which is Internet compatible through wireless Ethernet communications.
The main use of this new generation of vehicles is for automatic or assisted transport of goods and people in car-free areas (e.g. university campus, historical city centers, exposition-theme park, etc...). The article will describe the strategy adopted to asses the whole CyCab control.

1 INTRODUCTION

Three years ago, a new vehicle called CyCab, designed specifically for urban applications, has been presented to the public (1). Today, the CyCab is a reality: approximately ten units are used over the world in research centers.

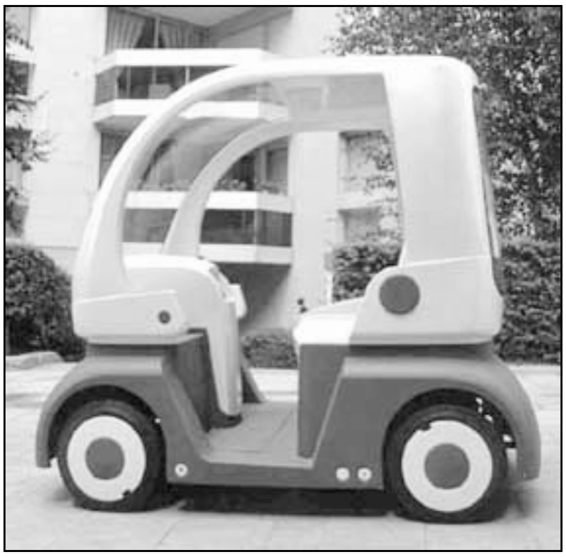


Figure 1.1: The CyCab

The CyCab application is a real-time embedded application: it must permanently interact with its environment and control it. The application reacts to input provided by number of sensors. Thus, by computations, it generates outputs handled by the multiple vehicle actuators.
In order to keep control over the environment, the CyCab has to react within bounded delays. Classical sequential architectures do not fit this aim. On the contrary, parallel architectures are convenient to satisfy real-time constraints (computation load distribution), and to take into account the distributed nature of the resources (sensor/actuator, computation, memory) for real-time applications.

To override the complexity of computing such application algorithms, the control software has been built using a dedicated software developed by INRIA, SynDEx, whose aim is to optimize the implementation of parallel and distributed code. A particular effort has been done to port the actual software distribution into the Motorola MPC555 architecture.

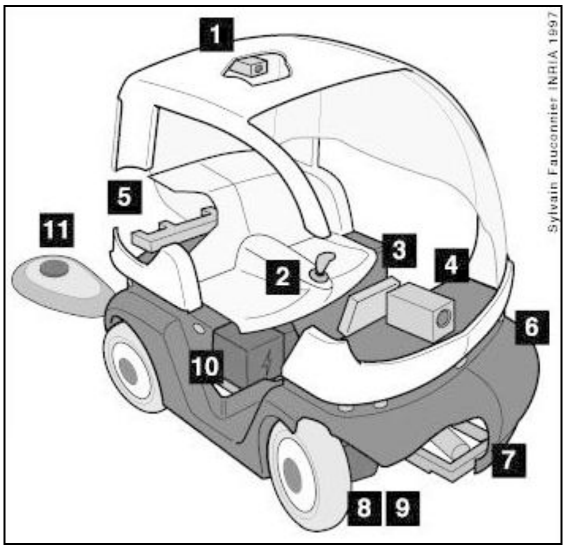


Figure 2.1: The CyCab design

2 THE CYCAB CONCEPT (2)

The CyCab is an electrical powered vehicle.
It can be seen both as a robotic platform for research purposes and as a starting block unit for fleet of automatic people movers (APMs) in several traffic reduced area (airport / station surroundings, tourist resorts, university campus…). In consideration of this latter use, the CyCab has been built around a mechanical chassis inspired by automobile design. The vehicle is 1.90 meters long, 1.20 meters large and weight approximately 300 kg (including batteries). Mechanical power is delivered by four 1 kW each DC motors per wheel: what can make the CyCab reach a theoretical velocity of 30 km/h. Each pair of rear and front wheels is steered by a DC jackservo.
With an autonomy of 2 h, this vehicle, made for two to four persons on board, and can be driven indifferently manually, using a joystick, or automatically by a supervising process. Figures 2.1 and 2.2 depict parts of the CyCab conception.

3 THE CYCAB HARDWARE ARCHITECTURE

3.1 General features

In order to control all the vehicle elements, a distributed hardware architecture has been developed. It includes 2 or more (up to 4) intelligent Motorola MPC555 based controllers, able to drive both motors and steering jackservo, and an Intel Pentium 233 MHz based PC computer. Details about this hardware architecture are given on figure 3.1.

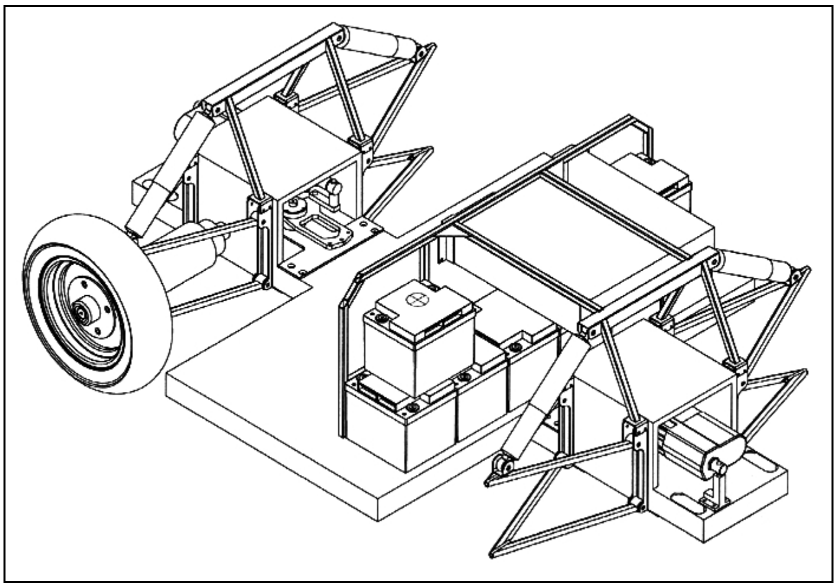


Figure 2.2: The CyCab mechanical platform

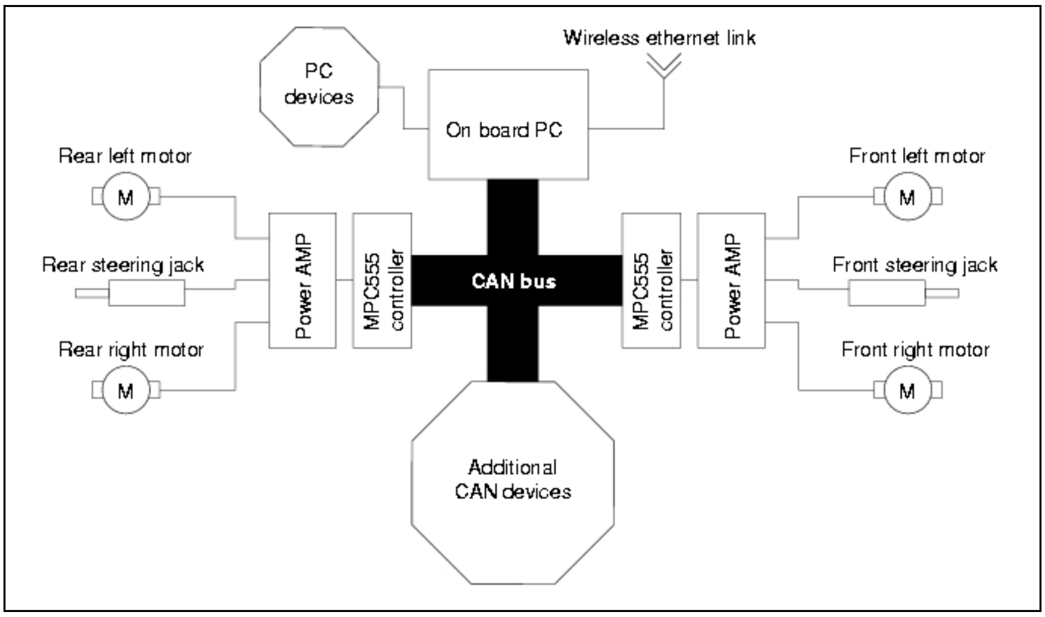


Figure 3.1: The CyCab hardware architecture

This type of architecture allows to split applications in different levels. Typically, the MPC555 controllers set handles, in priority, low level vehicle control under real time constraints. As for the PC computer, it handles high level applications as MMI or other highly time consuming task such as, image processing, path planning, etc... All these hardware components are connected to a same CAN (Controller Area Network) bus.
The main advantage of such a solution, compared with classical "centralized" architectures, is to get low level computation power close to actuators: this allows to reduce, considerably, wiring inside the vehicle and by this way to reduce risks of damages. Thus, in this architecture, only "high level" data is routed in the CAN bus, which is one of the most secured and reliable bus (it is commonly used in automobile industry).

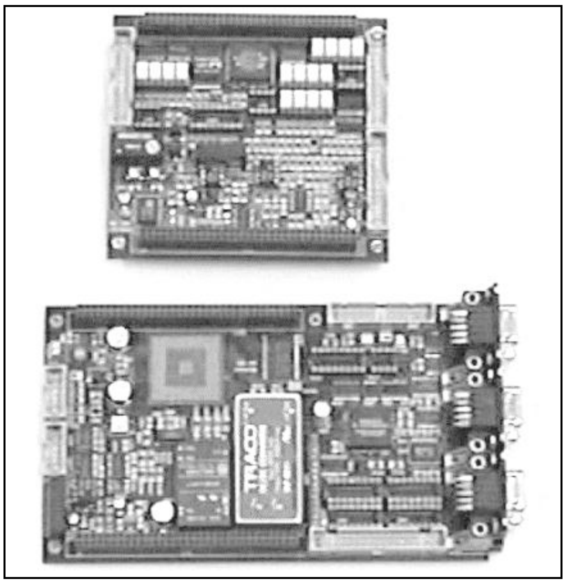


Figure 3.2: The MPC555 based controller

3.2 The MPC555 based controller

The controller unit developed for the CyCab embeds a Motorola MPC555, which is a new micro controller based on a 32 bits PowerPC core, including a floating point unit, running at 40 MHz. Due to its RISC internal architecture, the MPC555 provides high performance, what suits well real time motion control computation requirements. Moreover, this micro controller includes, on the same chip, numbers of functionalities. The controller board has been designed to provide access to all of these MPC555 resources. The main features of the controller board are listed below:

- Memory
  - 26Kbytes of internal FLASH
  - 448Kbytes of internal FLASH EEPROM with 5V programming
  - 128Kbytes of external SRAM

- Internal computing unit:
  - 6Kbytes of TPU Microcode RAM
  - 1 TPU (Time Processor Unit). This unit is a programmable controller that can be used for several tasks, such as: PWM (Pulse Width Modulation), fast quadrature calculation
  - measurements, computation and signal processing

- Input / output:
  - 4 incremental optical encoder interface channels
  - 4 PWM channels
  - 16 analog input channels
  - 4 12 bits analog output channels
  - 16 logical input channels
  - 16 logical output channels

- Communication ports:
  - 2 CAN (2.0A or 2.0B) ports
  - 2 RS232 serial ports
  - 1 SPI synchronized serial port

- Miscellaneous:
  - 1 LCD display (2 lines of 16 characters)
  - 1 hardware watchdog (refreshed each 2 ms)
  - 1 "wire cut" detection

4 THE CYCAB SOFTWARE ARCHITECTURE

4.1 Real-Time embedded applications (9)

Real-time applications are mostly found in the domains of signal and image processing and process control under real-time constraints. They must permanently interact with the environment that they control. They react to input stimuli received from the environment, by triggering computations to generate output reactions and/or change their internal state. In order to keep control over their environment, they have to react within bounded delays.
When the complexity of the application algorithm and the ratio between its computation volume and the response time bound are high, classical sequential architectures are inadequate, and parallel multicomponent hardware architectures are required. Programming such architectures is an order of magnitude harder than with mono-component sequential ones, and even more when hardware resources must be minimized to match cost, power and volume constraints required for embedded applications (3). The SynDEx CAD (6) software, that support AAA methodology (4), has been developed by INRIA, to help designers in optimizing the implementation of such applications.

4.2 Porting SynDEx into the Motorola MPC555 architecture

At the begin of the CyCab software development, a big effort was made to port the SynDEx macro-executive structure into this new Motorola MPC555 microcontroller architecture. Indeed, SynDEx was able to generate executive for various targets (such as C/Unix Workstation, i80386 processor, M68332 microcontroller, 87C196KC microcontroller, TMS320C40 DSP, SHARC ADSP2106x DSP and T800 Transputer) and various communication media (such as serial link, Unix TCP/IP and CAN bus), but did not include MPC555 support.

In order to generate executives, SynDEx relies on a set of target language dependent software primitives (called macros). Typically, they implement everything concerning:

- downloading procedure
- bootloader
- hardware initialization
- data type definitions
- memory allocations
- memory transfers
- control structure (like if-then-else and infinite or iterative loops)
- communication transfers
- communication receptions
- communication synchronizers
- interrupt handler
- chronometrical functions
- standard logic/arithmetic operations
- standard onchip input / output functionalities

In other words, this set of macros provide a very low level hardware definition of the system. On higher level, a specific macro language, interface these basic macros with SynDEx macro executive generator. Then each macroprocessed file is compiled and linked, producing an MPC555 compatible machine code.
These indirected inclusions, through the intermediate SynDEx macro language, also provide a very flexible and powerful mechanism needed to support efficiently heterogeneous architectures, with heterogeneous languages and compilation chains. This property is extremely useful as it allows MPC555 controllers to interact with any other SynDEx supported architectures.

4.3 Using SynDEx to describe the CyCab application (7)

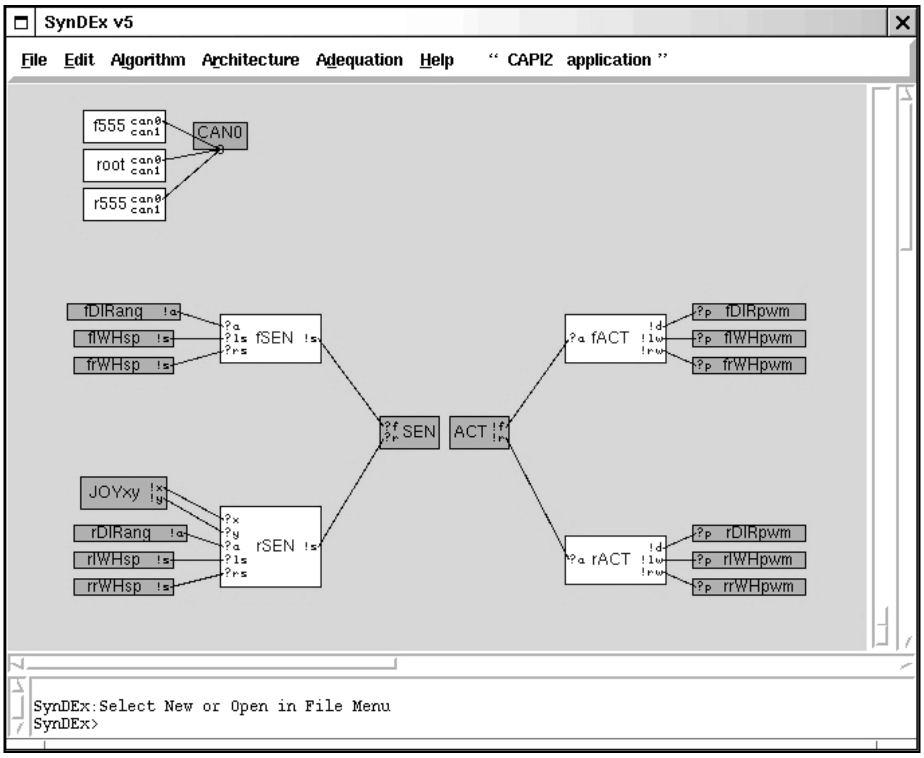
As it was described before, the CyCab platform is able to handle many different applications. Nevertheless, one underlying mode can be seen as a common basis: a low level software, performing vehicle actuators control and sensors acquisition (running each 1 ms), supervised by an high level application running on a PC, making use of sensors data and refreshing actuators commands (running each 10 ms). For instance, by high level applications we mean time consuming software that implement path planning or a vision tracking procedures. In the next section we focus on the benefits of using SynDEx CAD for such an application implementation.

4.3.1 Implementation principles

Using the SynDEx Graphical User Interface (GUI), the first step consists in designing the application as a data flow graph (see figure 4.1), in order to highlight the application potential parallelism. At hardware level, the user describes the architecture design as an hypergraph where each vertex represent a machine and each edge between vertices, physical communication buses.
Computing the adequation (based on AAA methodology) SynDEx will propose both a distributed version of the application and the corresponding scheduling (see figure 4.2). At that time, it suffices to verify if the proposed application meets the real-time constraints. If not, there are two ways to improve this result. First of all, it is possible to modify the actual hardware architecture. But it is the most costly solution. On a second instance, it can be

attempted to redesign the application algorithm using a smaller granularity, in order to allow a better distribution.
When these preliminary steps are satisfactory, SynDEx may generate a dead lock free executive for the real-time execution of the application algorithm on the multicomponent architecture. On relevance, is also the fact that SynDEx generates communication media dependent parts of code used for loading executives, through the designed architecture, into destination processors.

4.3.2 The CyCab application design
Figure 4.1 shows a snapshot of the implemented CyCab application using SynDEx. Represented at the top of the picture, the hardware architecture graph is composed of three unit: two MPC555 controller (f555 (resp. r555) designating front (resp. rear) MPC555 controller) and a PC (named root), all linked through a CAN bus.



```
SynDEx:Select New or Open in File Menu
SynDEx>
```

Figure 4.1: The CyCab software architecture graph

At the bottom of the picture, appears the data flow graph of the application algorithm. The vertices represent the operations to be executed on data and the edges, the data dependencies between operations. On this graph, three different types of operations are visible. Let us define them and list their contents:

- input operations: that only produce data
  - the two vertices flWHsp and frWHsp (running on f555) and the two vertices rlWHsp and rrWHsp (running on r555) respectively produce as output front left, front right, rear left and rear right wheels speed information
  - the two vertices fDIRang (running on f555) and rDIRang (running on r555) respectively provide front and rear steering angle

    - the vertex JOYxy (running on r555) gives x-axis and y-axis positions get from an analog joystick
    - the vertex ACT (running on PC) provides actuators commands computed by the high level software
- output operations: that only consume data
    - the two vertices flWHpwm and frWHpwm (running on f555) and the two vertices rlWHpwm and rrWHpwm (running on r555) respectively consume as input front left, front right, rear left and rear right motor PWM commands
    - the two vertices fDIRpwm (running on f555) and rDIRpwm (running on r555) respectively consume front and rear PWM steering commands
    - the vertex SEN (running on PC) gets sensors data for high level software computations
- processing operations: that consume and produce data
    - the two vertices fSEN (running on f555) and rSEN (running on r555) take as input sensors data, proceed filtering if needed, and produce as output tables containing computation results
    - the two vertices fACT (running on f555) and rACT (running on r555) take as input data tables from high level software and produce as output PWM commands for motors and steering jackservo

Finally, figure 4.2 presents the predicted execution time given by SynDEx after the adequation was completed. It shows only one execution of the graph. Each column represents the sequence of operations assigned to each processor, showing the time progress from top to bottom. The edges represent processor communications.

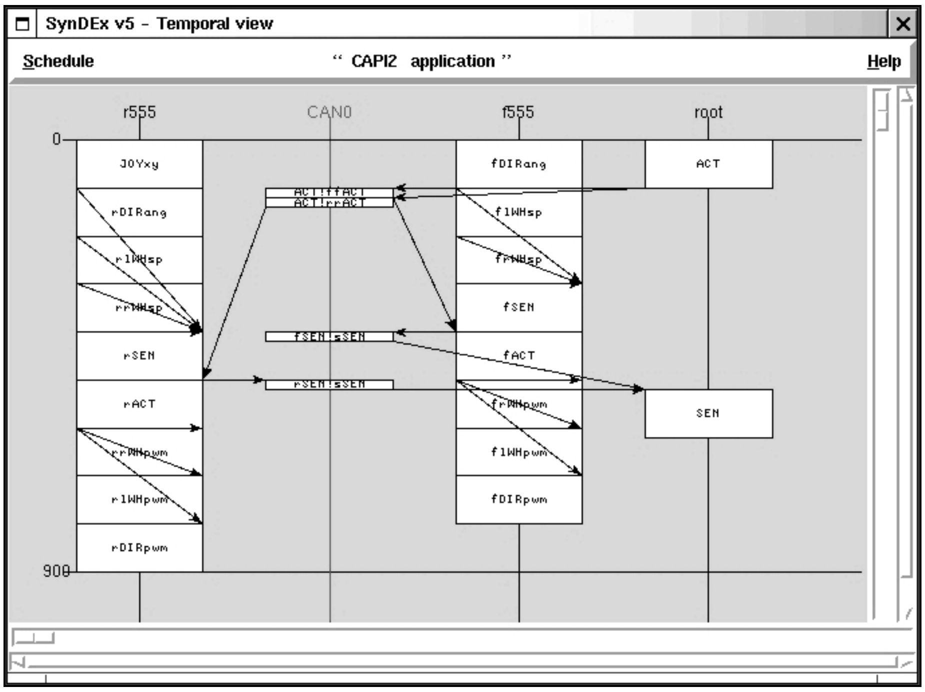


Figure 4.2: The CyCab application time schedule

CONCLUSIONS

The CyCab is a new concept of modular and intelligent electrical vehicle whose most promising application is as fleet unit of APMs in car-free areas (e.g. university campus, leisure parks, large company sites, etc...). In consideration of this application, the CyCab robotic platform has been inspired by automobile mechanical chassis design. As the CyCab was intended to react with the environment within bounded delays, a new type of hardware and software implementation was required. Using a parallel distributed and synchronized architecture was a real guarantee in term of security and computation load absorption. Development of such an application was made easier using SynDEX CAD tools. Unlike other classical tools, making use of the AAA methodology provides facilities for implementing software on heterogeneous parallel distributed robotic hardware architecture. In other words, the CyCab design shows that this new global design approach allows rapid optimized prototyping of robotic system under severe industrial constraints.

REFERENCES

(1) Michel Parent, E. Benejam-Francois, and N. Hafez. Praxitèle, "A New Public Transport with Self-service Electric Cars", In ISATA Congress, Florence, Italy, June 1996.

(2) Gérard Baille, Philippe Garnier, Hervé Mathieu, Roger Pissard-Gibollet, "Le Cycab de l'INRIA Rhône-Alpes", in INRIA Technical report N° 0229, April 1999.

(3) Behrooz A. Shirazi, A. Hurson, and Krishna M. Kavi, "Scheduling and load balancing in parallel and distributed system", IEEE Computer Society Press, 1995.

(4) Christophe Lavarenne, Yves Sorel, "A Unified Model for Software-Hardware Co-design", in CODES'99 7th International Workshop on Hardware/Software Co-Design, Rome, May 1999.

(5) Christophe Lavarenne and Yves Sorel, "Performance optimization of multiprocessor real-time applications by graphs transformations", In Proc. of the PARCO93 conference, France, 1993.

(6) Thierry Grandpierre, Christophe Lavarenne, Yves Sorel, "Modèle d'exécutif distribué temps réel pour SynDEx", in INRIA Technical report N° 3476, August 1998.

(7) Rémi Kocik, Yves Sorel, "A methodology to design and prototype optimized embedded robotic systems", 2nd IMACS International Multiconference CESA'98, Hammamet, Tunisia, April 1998.

(8) N. Halbwachs, "Synchronous programming of reactive systems",Kluwer Academic Publishers, Dordrecht Boston, 1993.

(9) SynDEx webpage, http://www-rocq.inria.fr/syndex.

(10)Motorola, "MPC555 RISC PowerPC MicrocontrollersUser's Manual", 15 July1999.

(11)Motorola, "RCPU Risc Central Process Unit Reference Manual revision 1", 1 February 1999.

(12)Motorola, "Time Processor Unit Reference Manual (including TPU2)", 1996.

(13)Motorola, "Fast Quadrature Decode TPU Function (FQD)", 1997.

(14)Philps, "SJA1000 Stand-alone CAN controller", 4 January 2000.