

Active high-speed CAN hub

Heikki Saha, Sandvik Tamrock Oy

Research around CAN HUBs has been active, but presented solutions have not been compatible with standard high-speed physical layer. This paper introduces active HUB supporting ISO 11898-2 high-speed CAN physical layer and containing interface disable inputs for disconnection/reintegration logic introduced in recent publications.

HUBs enable use of star topology, which can better fit into system's physical structure, help avoiding uncontrolled ground currents and increase reliability. For HUB design and simulation, state-diagram based method is introduced with state-machine based behavioral CAN-bus simulation model. Timing constraints are introduced and transceiver loop-delay has found to be the most limiting parameter.

Performance test computations has been verified with a prototype. The HUB supporting ISO 11898-2 physical layer can operate at any baud rate, but in real systems the HUB can operate at baud rates up to 800kbps. In addition to the interface-specific diagnostics features, HUB-based CAN-buses can ease systems designers by enabling star topology with improved tolerance against physical layer failures and without breaking the ISO 11898-2 physical layer standard.

1 Introduction

ISO 11898 defines linear bus as a topology of CAN bus. In many systems linear structure is not optimal and may even make cabling more challenging than other topologies. In addition to the equations defining effect of unterminated drop lines /11/, literature knows solutions to implement star topology with special termination scheme /17/ and with playing with the network with long bus-shaped part and small star-shaped part at one end of the bus /4/. The main problem with the first one is a need of network-specific termination, which is different for every network not being production and maintenance friendly. The second alternative relies on relatively low baud rate (≤ 125 kbps) allowing long drop-lines or small-scale mixed-topology parts for the network. It also decreases maximum bus length about 25%. Error tolerance of CAN is also reduced when other than linear topology is used.

Previously mentioned literature presents only alternative topology solutions without reasons to go to the other than linear topology. Typical reasons for using star instead of linear bus are for example:

- A system has some subsystems located around system center point. In this case star-topology fits better.
- Power supply and bus wires are included into same cables and star shaped power supply scheme is used to minimize voltage drops.
- Star-topology increases system reliability by reducing coverage of single bus failure located at any point.

Because CAN physical layer can not generally operate in star or tree networks, active repeater /16/, HUB /13/, /14/, /15/ or switch /12/ will be needed. The biggest drawback of the repeater is double propagation delay between nodes located in separate buses isolated from backbone by repeaters. Switched CAN networks have been introduced in /12/ but CAN-switch is one additional component to be configured. In many simple systems configurable components increase too much system integration and production complexity and therefore this paper concentrates on HUBs, which operate at OSI level 1 and don't need any routing tables and other complex configuration.

First behavioral simulation of active CAN-HUB supporting ISO-11898 interfaces has been published in /12/. The HUB and its

CAN-buses have been modeled and simulated in StateCAD environment as state-diagrams. Bus propagation and HUB loop delays has been simplified into 1 clock period long to ease up the simulation as state diagrams. Typical problems with forwarding delay and bus line length reduction recognized, but many benefits have been excluded because of focus was in high-speed networks operating up to 1Mbps.

A special full-duplex physical layer utilizing point-to-point ISO high-speed interfaces has been presented in /13/. Very detailed physical layer failure analysis and a lot of online diagnostics has been presented and implemented. The main limitation is ability to connect only one node into one bus interface. The limitation is controlled feature and enables condition monitoring of every physical link and node individually. The CAN bus functionality has been implemented in FPGA with extensive set of interface oriented fault monitoring and automatic disconnect of faulty interface.

Optical fiber links have been promoted in /14/ and /15/, but in both cases supported topology is one node per each HUB interface. HUB presented in /14/ is actually a small-scale ISO-11898 CAN-bus inside the HUB. Optical interfaces in both /14/ and /15/ can be mentioned as optocouplers between CAN-controller and transceiver. The main purpose of HUBs presented in /14/ and /15/ have been EMC tolerance improvement.

Interface-specific diagnostics has been presented in both /13/ and /15/. In /13/ very detailed error-detection scheme has been presented and based on diagnostics, interfaces can automatically be disconnected and reintegrated to keep the faulty parts of the system disconnected and rest of the system operating with minimum disturbance in communication. Only optical connection existence monitoring has been implemented in /15/ with automatic disconnect and reintegration. The functionality of CAN-bus physical layer has been replaced by configurable logic, FPGA in /13/ and CPLD in /15/.

So called minibridge has been presented as an interconnection component between

copper backbone and nodes with fiber interface /15/. Serious problems have been met when a node with copper interface tried to be connected into HUB via fiber link. To succeed with that, the minibridge shall actually be a repeater instead of direct fiber-to-copper adapter to perform the operation properly.

Misleading terminology has been used in /15/ - passive HUB defined to equal active HUB (presented in this paper) and active HUB defined to equal switch.

2 ISO 11898-2 HUB operation

Active CAN-HUB operates in OSI-level 1, physical layer, and echoes dominant states between numerous physical CAN interfaces. Instead of direct echoing, feedback of the HUBs own transmission shall always be blocked to prevent infinite loopback of dominant state between the interfaces.

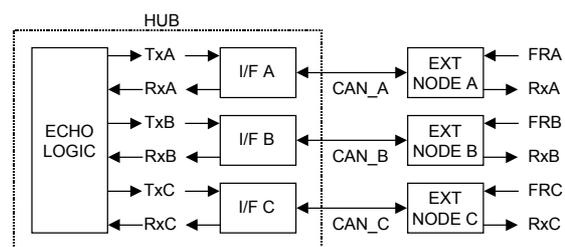


Figure 1: Signal naming convention

Signal names are introduced in Figure 1. A HUB with 3 interfaces has been used in the evaluation. To be able to properly evaluate the HUB, every interface shall have at least one external node.

The operation of active ISO-11898 CAN-HUB can be described in following three cases. In the simplest case, dominant state is received from only one interface. In that case it is trivial to echo received dominant state directly into other interfaces.

Figure 2 presents the most complicated case, where one node in bus A and another node in bus B drives partially overlapping dominant state. The challenge is to determine released dominant state in bus A and re-echo the dominant state received from bus B to the bus A as fast as possible.

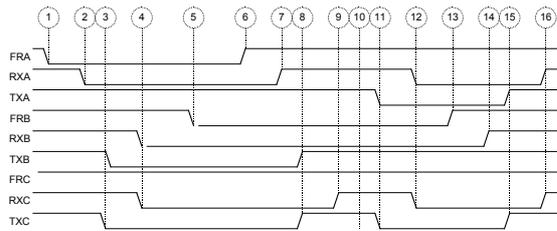


Figure 2: Overlapping dominant states

1. A node in segment A asserts dominant state
2. Dominant state of the segment A received into HUB (and also other nodes in the segment A)
3. HUB asserts dominant state into other segments
4. All nodes in the other segments receive dominant state
5. A node in segment B asserts dominant state
6. A node in the segment A negates dominant state
7. Recessive state of the segment A received into HUB (and also other nodes in the segment A)
8. HUB negates dominant state into other segments
9. All nodes in the other segments receive recessive state
10. Maximum loop-delay has elapsed and state of the bus B is determined as dominant
11. Dominant state from segment B is echoed into other segments
12. Dominant state of the segments A and C received into HUB (and also other nodes in the segments A and C)
13. A node in the segment B negates dominant state
14. Recessive state of the segment B received into HUB (and also other nodes in the segment B)
15. HUB negates dominant state into other segments

Delay between i.e. points 1 and 2 is called loop delay of a bus. It consists of following delay components:

- Transceiver TX to CAN delay

- Transceiver CAN to RX delay
- Bus propagation delay as double

Delay between i.e. 2 and 3 is HUB echoing delay and its length depends on the HUB design.

Figure 3 presents typical case during transmission of arbitration or acknowledge field of CAN-telegram, where two or more interfaces simultaneously transmit equal bus state. In this case the HUB shall identify the simultaneous reception and echo received dominant state to the interfaces which are in the recessive state.

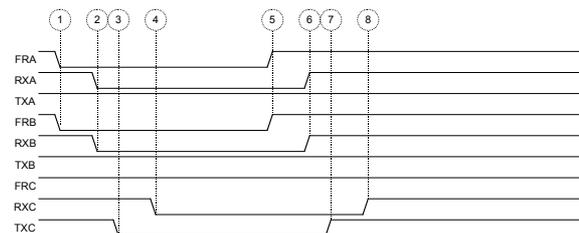


Figure 3: Simultaneously received dominant state from multiple interfaces

1. Nodes in segments A and B assert dominant state
2. Dominant state of the segments A and B received into HUB (and also other nodes in the segments A and B)
3. HUB asserts dominant state into segment C
4. All nodes in the segment C receive dominant state
5. Nodes in segments A and B negate dominant state
6. Recessive state of the segments A and B received into HUB (and also other nodes in the segments A and B)
7. HUB negates dominant state into segment C
8. All nodes in the segment C receive recessive state

3 Experimental HUB design

Experimental HUB design consists of three independent CAN-interfaces. Increasing number of interfaces increases only size of the design but not functional requirements. Main functional blocks and connections between them are presented in Figure 4. All transmission state-machines have disable feature to enable

evaluation of automatic interface disable/reintegration features later. Disable signals are not used in current work and has been left out for clarity.

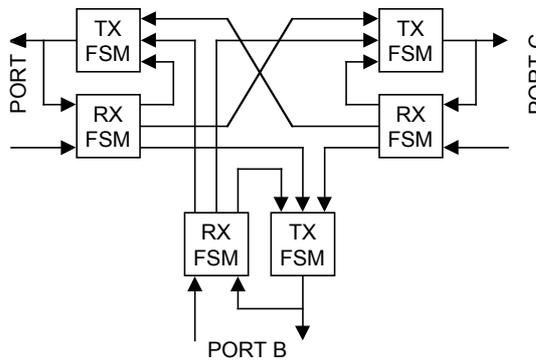


Figure 4: Block diagram of the HUB

Every reception state-machine, according to Figure 5, continuously monitors bus status and decodes dominant states driven by external CAN nodes.

In the idle state, both TX- and RX-signals are high. When external node drives dominant state, RX goes low and TX remains high. HUBs own transmissions drive TX low before RX. Received dominant state activates the ECHO state, in which received dominant state is echoed to other interfaces. Transmitted dominant state activates PASS-state, until TX is released and WAIT1-state is entered. After WAIT1-state, IDLE-state is entered when RX is released or ECHO-state when RX remains active after bus propagation time has elapsed. There is additional loop-delay timer controlled by transmission state machine.

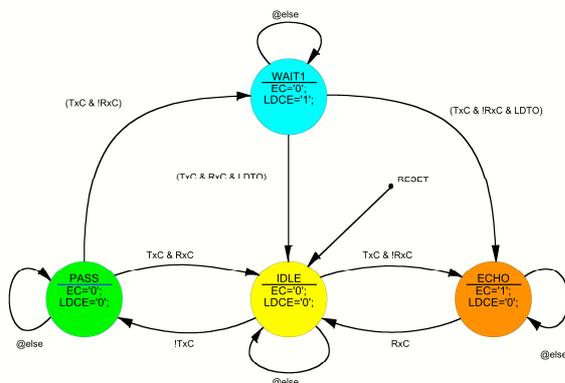


Figure 5: Reception state-machine

Transmission state-machines drive the TX-lines. They collect echoes from other interfaces and manage optional interface disable inputs, which can be driven by

interface-specific diagnostics modules. Jumpers emulate diagnostics modules in the experimental design.

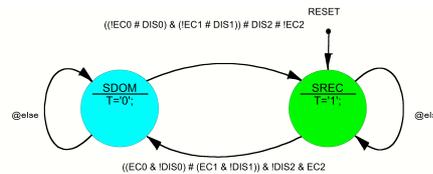


Figure 6: Transmission state-machine

HUB echoing takes two clock cycles. During the first cycle dominant state is recognized by reception state-machine and after the second cycle transmission state-machine has passed the dominant state into destination bus if it is enabled.

4 CAN bus simulation model

Behavioral simulation model has been developed for HUB simulation. The model operates at clock cycle basis and bus propagation delay can be parameterised in clock cycles. The model also has force input for emulating dominant states driven by node external to the HUB. Bus model state-machine interacts with propagation delay timer module.

Because both timers have been written in ABEL HDL, they can be used only in ABEL simulation environment. Fortunately it is easy to manually simulate functionality of both timers in state-machine simulation.

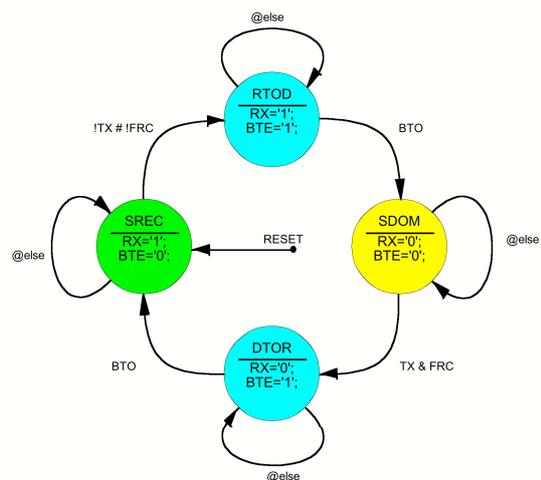


Figure 7: CAN bus state machine

Though state-diagrams have been used as a primary design entry format in this case, the use of standard HDL as module integration format must be emphasized. There are many times IP-blocks or

modules, which are easier to write manually to be easily integrated into same design with state-machines. Standard HDL files also provide excellent backup of state-diagrams for manual or automatic translation to next generation HDLs. Use of standard HDL offers total tool and chip vendor independence after design entry. /18/

5 HUB timing analysis

The transceiver loop-delay $T_{LOOP}=105\text{ns}$ has been found out from transceiver specifications /1/, /2/, /3/. Transceiver loop-delay has been measured to guarantee correctness of the computations.

Bus propagation delay is assumed to be $\lambda=5\text{ns/m}$ according to /8/, /9/, /10/ and /11/.

Both /8/ and /10/ give equal bit time configuration with 8 time quanta/bit time with 87.5% sample point position:

- 1 time quantum for synchronization segment
- 5 time quanta for propagation segment. For simplicity 4 TQ propagation segment has been used in computations.
- 1 time quantum for TSEG1 (SJW < 1 TQ /10/)
- 1 time quanta for TSEG2

Table 1 presents most common bit timing parameters used in the following analysis:

Baud rate	Bit time	Time quantum
1Mbps	1 μs	125ns
800kbps	1.25 μs	156.25ns
500kbps	2 μs	250ns
250kbps	4 μs	500ns

Table 1: Bit timing parameters according to different baud rates

8 time quanta per bit time don't meet i.e. CiA DS 301 at all baud rates, but it is used for simplicity. Default HUB system clock rate of $F_{HUBCLK}=40\text{MHz}$ is used.

ECU internal delay is defined in /8/:

$$T_{ECU} = T_{TX} + T_{RX} + T_{LOGIC} + T_{CHOKE} \quad (1)$$

When choke and extra logic do not exist, equation 1 can be written as:

$$T_{ECU} = T_{TX} + T_{RX} = T_{LOOP} \quad (2)$$

Effective propagation delay is defined in /8/ as:

$$T_{PROP} = 2 \cdot (T_{ECU} + \lambda \cdot L_{BUS}) \quad (3)$$

In case of HUB device, HUB forwarding delay of 2 HUB system clock cycles shall be added to equation 3:

$$T_{PROP} = 2 \cdot (T_{ECU} + \lambda \cdot L_{BUS}) + 2 \cdot T_{HUB} \quad (4)$$

Effective propagation delay shall fit into propagation segment:

$$T_{PROP} = 4 \cdot TQ \quad (5)$$

Combining previous two equations and taking dual propagation feature of the HUB into account, we get:

$$4 \cdot TQ = 2 \cdot 2 \cdot (T_{LOOP} + \lambda \cdot L_{BUS} + \frac{2 \cdot T_{HUB}}{2}) \quad (6)$$

L_{BUS} can be solved from Equation 6:

$$L_{BUS} = \frac{1}{\lambda} \left(\frac{1}{8 \cdot BR} - T_{LOOP} - \frac{1}{F_{HUBCLK}} \right) \quad (7)$$

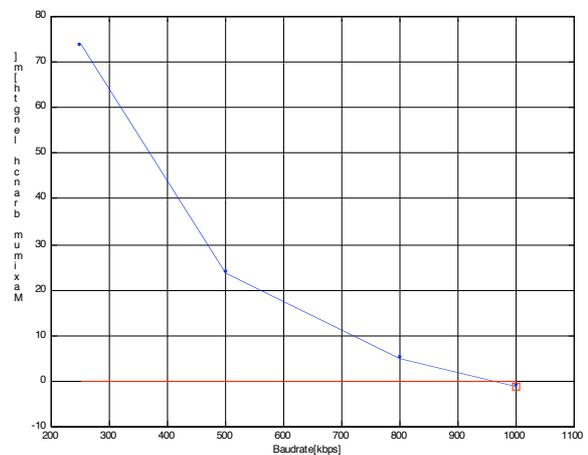


Figure 8: Maximum achievable branch length vs. typical baud rates

Figure 8 depicts maximum branch length versus few most typical baud rates at 40MHz HUB system clock rate. One can see that 800kbps is the highest achievable standard baud rate.

According to the Equation 7, signal propagation delay λ and node loop delay T_{LOOP} can not be modified, so the branch length can be increased by decreasing the

baud rate or by increasing HUB system clock rate F_{HUBCLK} .

Unfortunately effective loop-delay is dominating with 210ns, which means that only minimum effect can be reached by increasing HUB system clock rate. Figure 9 depicts the maximum branch length at 1Mbps baud rate versus some usable HUB system clock rates.

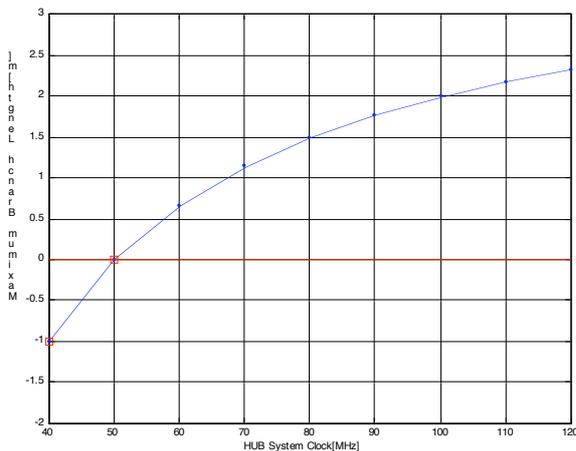


Figure 9: Maximum branch length vs. HUB clock frequency

One can notice that even with the higher clock rates the branch length at 1Mbps remains very short. Maximum achievable branch length can be approximated:

$$L_{BUS_{Max}} = \lim_{F_{HUBCLK} \rightarrow \infty} \frac{1}{\lambda} \left(\frac{1}{8 \cdot BR} - T_{LOOP} - \frac{1}{F_{HUBCLK}} \right) \rightarrow 4m \quad (8)$$

According to the Equation 8 the maximum branch length of 4 meters can be reached with infinitely high HUB clock rate. With baud rates up to 800kbps and 1Mbps branches of 8.5m and 2.3m respectively can be reached with 120MHz HUB system clock.

6 HUB prototype evaluation

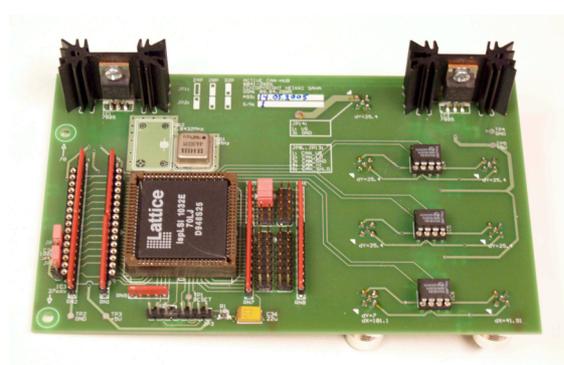


Figure 10: The HUB prototype

A prototype HUB has been manufactured to enable verification of computations. The prototype consists of a CPLD and three CAN-interfaces.

The prototype was tested with three standard baud rates - 500kbps, 800kbps and 1Mbps. CAN-telegrams were sent into every interface with some different identifiers and at as high load as possible. According to Figures 11 and 12, there are no problems at baud rates up to 800kbps. At 1Mbps every bus immediately entered bus-off. The results proved correctness of the computations.

	CAN 1	CAN 2	CAN 3
Busload [%]	75.84	75.88	75.05
Peakload [%]	95.05	95.05	94.78
Std. Data [fr/s]	3000	3000	2991
Std. Data [total]	8376389	8376390	8376389
Ext. Data [fr/s]	0	0	0
Ext. Data [total]	0	0	0
Std. Remote [fr/s]	0	0	0
Std. Remote [total]	0	0	0
Ext. Remote [fr/s]	0	0	0
Ext. Remote [total]	0	0	0
Errorframe [fr/s]	0	0	0
Errorframes [total]	0	0	0
Chip state	Active	Active	Active

Figure 11: 500kbps test results

	CAN 1	CAN 2	CAN 3
Busload [%]	15.74	15.68	15.26
Peakload [%]	57.81	57.77	59.11
Std. Data [fr/s]	993	991	1004
Std. Data [total]	4951760	4951760	4951760
Ext. Data [fr/s]	0	0	0
Ext. Data [total]	0	0	0
Std. Remote [fr/s]	0	0	0
Std. Remote [total]	0	0	0
Ext. Remote [fr/s]	0	0	0
Ext. Remote [total]	0	0	0
Errorframe [fr/s]	0	0	0
Errorframes [total]	0	0	0
Chip state	Active	Active	Active

Figure 12: 800kbps test results

7 Conclusions

Detailed analysis of active CAN high-speed HUB has shown that the concept works. Star-topology high-speed CAN network with active HUB can decently support baud rates up to 500kbps in real systems. In some small-scale systems baud rates up to 800kbps and 1Mbps can be used with branches of approximately 8.5m and 2.3m respectively.

The biggest benefits of the ISO high-speed physical layer are 100% compatibility with current node offering and ability to handle multiple nodes connected into each interface while the other HUB implementations need dedicated nodes or modified physical layer

and support only one node per interface. But meeting ISO high-speed physical layer drastically drops the maximum branch length.

The HUB-based star-shaped networks can be used with any baud rate, but if large coverage is needed together with one of the highest baud rates, switched CAN-network provides better solution. As long as 500kbps is fast enough, there are enough bus length available for most mobile machinery and automotive applications.

State-diagram based design entry and behavioral simulation was found very successful for this kind of designs. State-diagram design entry tool significantly improved design and functional verification of the state machines with complex transition conditions. Possibility to use "else" statement in the state transitions has found very productive and time saving feature.

References

- [1] PCA82C251 CAN transceiver for 24V systems, Product specification, Philips Semiconductors, 2000
- [2] TJA1050 High speed CAN transceiver, Preliminary specification, Philips Semiconductors, 2000
- [3] TJA1050 CAN-Transceiver, Application Note AN00020, Philips Semiconductors, 2000
- [4] Peter Dierauer: Limits of CAN high-speed multi-drop networks, Proceedings of 7:th International CAN Conference, Amsterdam, Netherlands, 2000, pp. 05-2 - 05-9
- [5] CAN Specification Version 2.0, Robert Bosch GmbH, 1991
- [6] ISO 11898-1:2002
- [7] ISO 11898-2:1999
- [8] SAE J1939-11/October 1999
- [9] SAE J1939-15/November 2003
- [10] CiA DS-301 v. 4.02, 13.02.2002
- [11] CiA DRP 303-1 v 1.3, 30.12.2004
- [12] Heikki Saha, Comparison of system level networking solutions with high-speed CAN networks, Proceedings of 9:th International CAN Conference, Munich, Germany, 2003, pp. 09-1 - 09-8
- [13] M. Barranco, J. Proenza, G. Rodriguez-Navas, L. Almeida, A CAN hub with improved error detection and isolation, Proceedings of 10:th International CAN conference, Rome, Italy, 2005, pp. 04-16 - 04-23
- [14] J.J. Serrano, J.C. Campelo, J.V. Capella, R. Ors, A. Bonastre, Commutations system based on CAN and extended with optical fiber for the implementation of a mobile robot immune to the sabotage provoked by electromagnetic noise, Proceedings of 10:th International CAN conference, Rome, Italy, 2005, pp. 04-9 - 04-15
- [15] S. Rohr, H. Kabza, High Speed Optical Controller Area Networks (CAN), Proceedings of 10:th International CAN conference, Rome, Italy, 2005, pp. 04-1 - 04-8
- [16] M. Hietikko, J. Alanen, R. Tiusanen, Työkoneiden ja automaation CAN vävyläsovellusten turvallisuus, VTT Information Service, Finland, 1996, ISBN 951-38-4900-7 (In Finnish)
- [17] PCA82C250/251 CAN Transceiver, Application Note AN96116, Philips Semiconductors, 23.10.1996
- [18] Heikki Saha, Life Cycle Control of Low-Volume Embedded Processing Platforms, Proceedings of International Conference of Machine Automation, Tampere, Finland, 2002, pp. 77-83