APPLICATION NOTE

P82C150
Serial Linked I/O (SLIO) Device

AN94088

Philips
Semiconductors

PHILIPS

Abstract

A Serial Linked I/O device (SLIO) like the P82C150 is a port device directly interfacing to the CAN-bus. It can be used as an extension of digital and analog port functions for a remote microcontroller. The CAN-bus is used as the serial link between the microcontroller and the port extension. The 82C150 provides several digital I/O-ports, two comparators, two D/A-converters using the 'Distributed Pulse Modulation' and one A/D-converter. The port functions are programmable by the host via the serial link.

This application note gives information on the implementation of a SLIO in a CAN network and discusses different topics of interest for ease of use of the device.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

APPLICATION NOTE

P82C150
Serial Linked I/O (SLIO) Device

AN94088

Author(s):

Egon Jöhnk, Heinrich Waterholter
Product Concept & Application Laboratory Hamburg, Germany

Keywords

Automotive, Industrial
Serial Linked I/O, SLIO
CAN
P82C150
ADC
DPM
calibration

Date: 20th December, 1994

Summary

This report is intended to provide application support for designing SLIO nodes based on the P82C150 in a CAN network.

The report describes a typical application of the P82C150 with respect to interfacing to the CAN-bus. It informs about resetting and initializing the P82C150, giving some examples of reset circuits and the programming of the variable bits of the identifier. The aspect of calibration and communication in a CAN network with SLIOs are discussed. Information on the bit timing and the possible bus length are given.The structure of the I/O ports are shown and typical applications are given for the digital and analogue functions available in the P82C150.

CONTENTS

1. INTRODUCTION

A ***S***erial ***L***inked ***I/O*** device (SLIO), like the P82C150 from Philips Semiconductors, allows the design of low-cost I/O nodes in a CAN-bus system. The P82C150 includes a controller area network (CAN) protocol controller on-chip.

Fig. 1 shows the principal configuration of CAN nodes (Module 1) and different configurations with microcontrollers as “module controller”, CAN-bus controllers (integrated on the microcontroller or as separate IC) and CAN-bus transceivers for the physical interface. A SLIO node contains a CAN bus controller with I/O facilities and a transceiver.

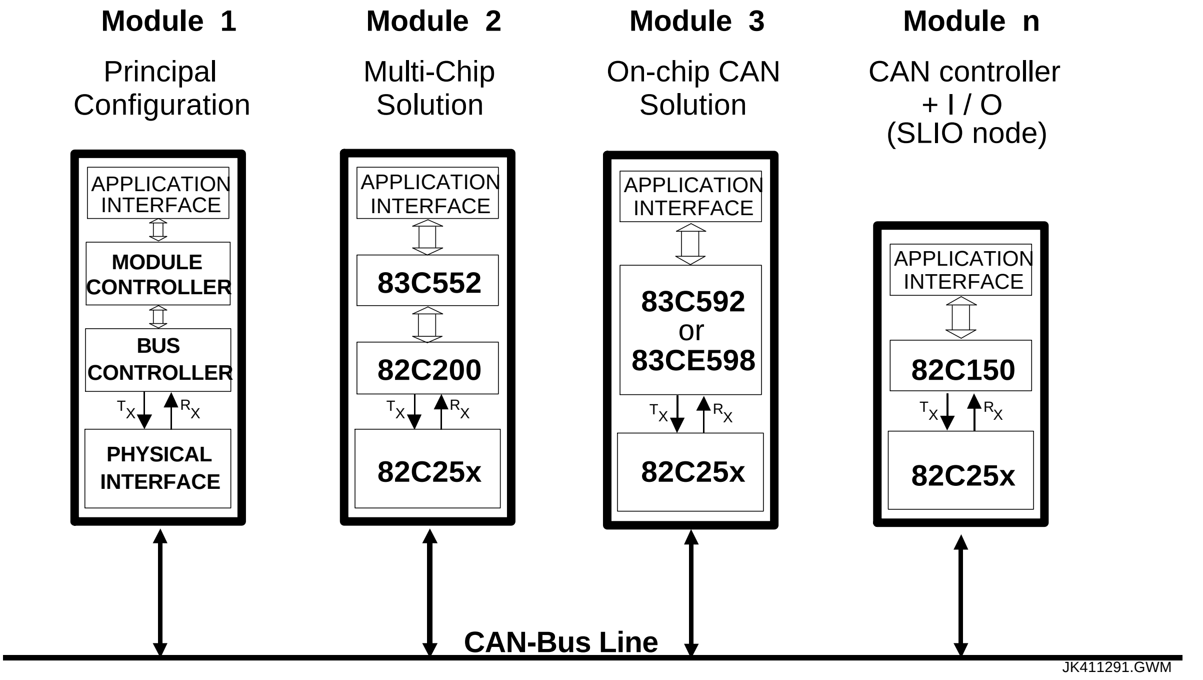


Fig. 1 CAN module setup

The principal configuration of a node is composed of

- the physical interface to the CAN-bus line
- a CAN-bus controller (takes care of the CAN protocol, Data Link Layer and Physical Layer Signalling [4], [5])
- a microcontroller, which prepares the data to be transmitted over the CAN-bus and processes data which was received from the CAN-bus,
- and an interface to the application.

The P82C150 (SLIO) takes over the tasks of the CAN-bus controller and the microcontroller. It needs a host micro controller in the CAN network in order to be able to operate (see chapters 4., 5. and 6.).

The reader of this application note is supposed to be familiar with the CAN specification ([2], [4] and [5]) and the principle functions of the P82C150. He should have the data sheet of the device (refer to [1]) available. This application note gives further information on parts of the device and hints on how to build up a network, where the SLIO is used in one or more nodes.

2. FUNCTIONS AND FEATURES OF P82C150

Fig. 2 gives an overview of all functions of the P82C150. The ‘CAN Controller‘ does all interfacing to the CAN-bus line and is responsible for the correct handling of the CAN protocol. The ‘Oscillator + Calibrator‘ unit contains the on-chip RC-oscillator and does all synchronisation and calibration for receiving messages correctly. The data received from the host CAN-controller is stored internally in a series of registers. The P82C150 organizes the functioning of all 16 I/O-pins according to the content of these registers (more information on these items may be found in the data sheet [1]).

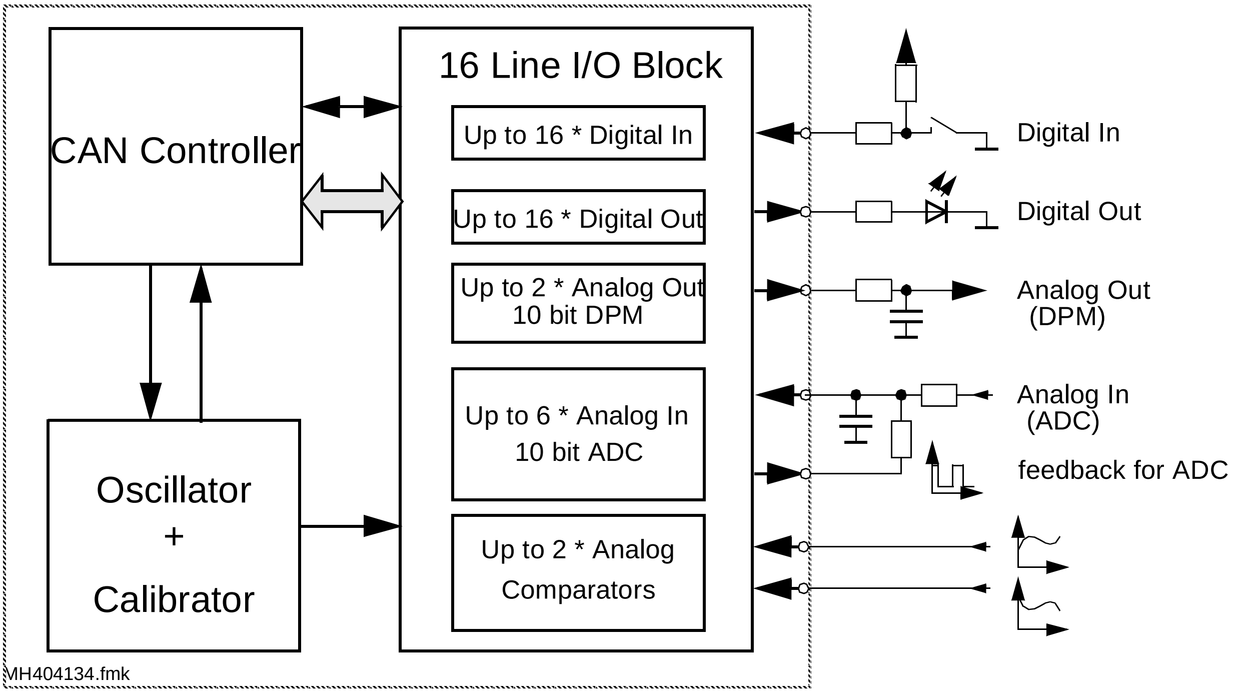


Fig. 2 Block diagram of P82C150 (SLIO)

2.1 General Features

- The P82C150
  - is a stand-alone I/O device with a CAN controller on-chip.
  - is fully compatible to the CAN specification V2.0 A and B.
  - has 16 I/O pins for various digital or analogue configurations.
  - has an on-chip oscillator (no external components).
  - supports a sleep mode with wake-up via the bus.
  - comes in a 28-pin package.
- There may be up to 16 P82C150-devices in one network.

2.2 I/O Port Features

- Each port pin‘s mode is individually configurable via the CAN-bus:
  - up to 16 pins for digital output.
  - up to 16 pins for digital input.
- The input values at the port pins are transmitted to the host
  - on request by the remote CPU via the CAN-bus (polling of input data).
  - if a signal edge is detected at the port pin (event capture mode, programmable for each pin).
- Up to 2 pins may be used for a quasi analogue output (2 DPM generators).
- 1 pin can be used for the input to a 10-bit A/D-converter.
- Up to 6 pins may be used as an analog 6:1-multiplexer. It may be used for general purpose or connected to the A/D-converter input.
- Up to 4 pins may be used for comparator functions (2 comparators are available):
  - 2 analogue signals can be compared with a reference voltage applied to other port pins.
  - the comparator outputs may be routed to digital I/Os e.g. for using the event capture possibility.

2.3 CAN Communication

- All functions are controlled by one ‘intelligent‘ communication partner (‘host‘ e.g. P8xC592, P8xCE598).
- A P82C150
  - adapts its local bit clock to the bit timing of the bus.
  - broadcasts messages to ‘intelligent‘ communication partners (not to other P82C150s).
  - supports bit rates between 20 kbit/s and 125 kbit/s.
- The communication with a host uses 2 identifiers (Standard Frame):
  - 6 identifier bits are fixed for all P82C150s.
  - 4 identifier bits are programmable by external pull-up/down resistors applied to port pins (=> max. 16 devices in one network).
  - 1 identifier bit determines the transfer direction of a message (from or to the P82C150).
- The host has to send periodically calibration messages.
  - recommended interval: 3800 bit times
  - examples of periodic calibration time intervals for different bit rates:

bit rate	time interval
20 kbit/s	190 ms
50 kbit/s	76 ms
125 kbit /s	30 ms

3. A TYPICAL APPLICATION OF P82C150

Fig. 3 shows a typical application of the P82C150. A reset circuit sets the P82C150 into a proper state after power on. Some suitable circuits are discussed in chapter 3.1. Four bits of the identifiers (used by the P82C150 for the acceptance filter and the transmission identifier) are programmed via pins (hardwired). This item is discussed in more detail in chapter 3.2 where also an example is given (Fig. 8). The connection to the bus is done via the transceiver circuit PCA82C250 (compatible with the ISO 11898 standard). See chapter 3.3 for further information. Information about configurations of the ports and how to use them are discussed under the respective functional descriptions in chapter 8 and chapter 9.

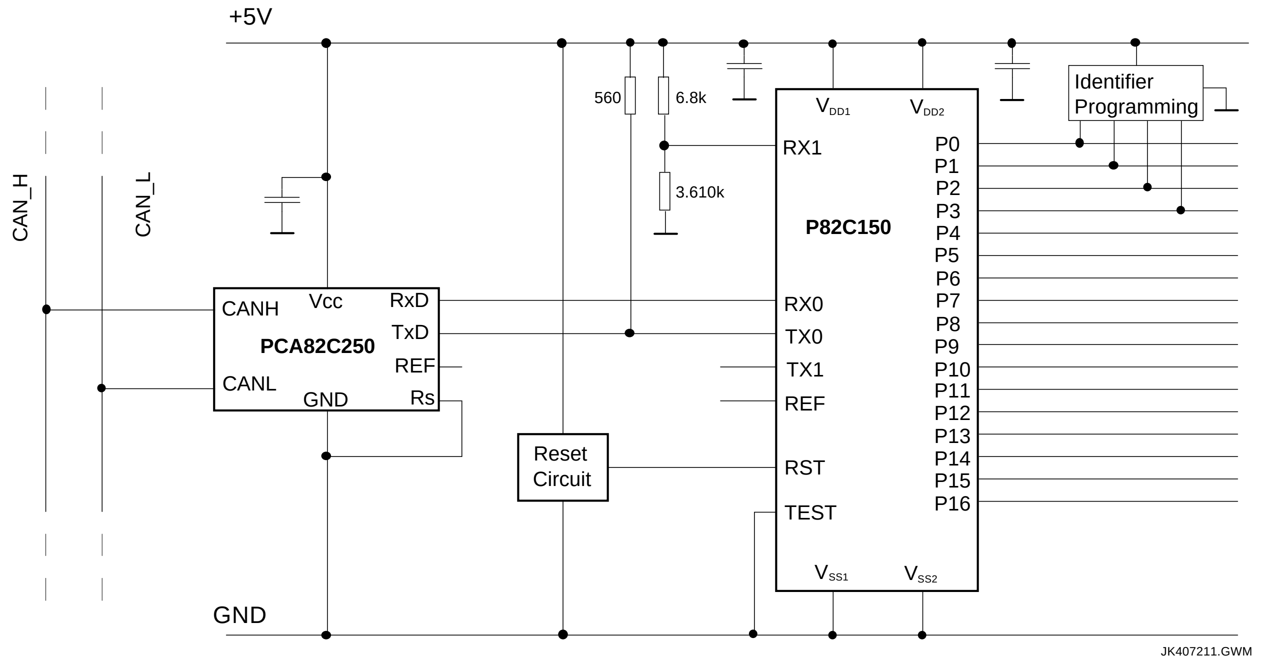


Fig. 3 Circuit diagram for a typical application of P82C150

3.1 Reset Circuit for the P82C150

The device P82C150 needs a reset signal at the input RST during ramp up of the power supply. Fig. 4 shows a possible simple reset circuit with an RC-network and the example of Fig. 8 gives a solution with a supervisory circuit.

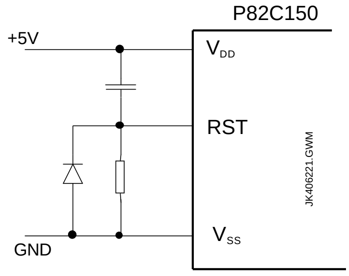


Fig. 4 Reset circuit with an RC-network

3.1.1 Reset Circuit with an RC-Network

This simple reset circuit uses an appropriate RC-network. If the power supply is rising to its nominal value the voltage at the input RST follows the supply voltage as long as the capacitor is nearly unloaded. When the supply voltage has reached its nominal voltage, the capacitor will be loaded via the resistor and the voltage at the reset input ramps down to GND.

If the supply voltage is switched off the input voltage at RST would be shifted to a value below GND as long as the capacitor still is loaded. Thus the limiting values for the DC voltage on any pin (see the data sheet [1]) would be exceeded. In order to avoid this a diode is used for discharging the capacitor quickly clamping the voltage at pin RST. Furthermore the quick discharging of the capacitor helps to generate a new reset pulse, if the supply voltage should be switched on again after a short time.

The values of the components have to be selected in such a way, that the pulse at the reset input RST meets the requirements given in the data sheet [1], as there are:

- input voltage level HIGH at RST for generating a reset of the SLIO
- min. reset pulse width after power on
- the max. allowed negative voltage at pin RST must not be exceeded

Example:

Possible values for the components of the reset circuit are: C = 10 µF, R = 27 kΩ.
The clamping can be made by a fast switching diode, like 1N4148 or BAS32L.

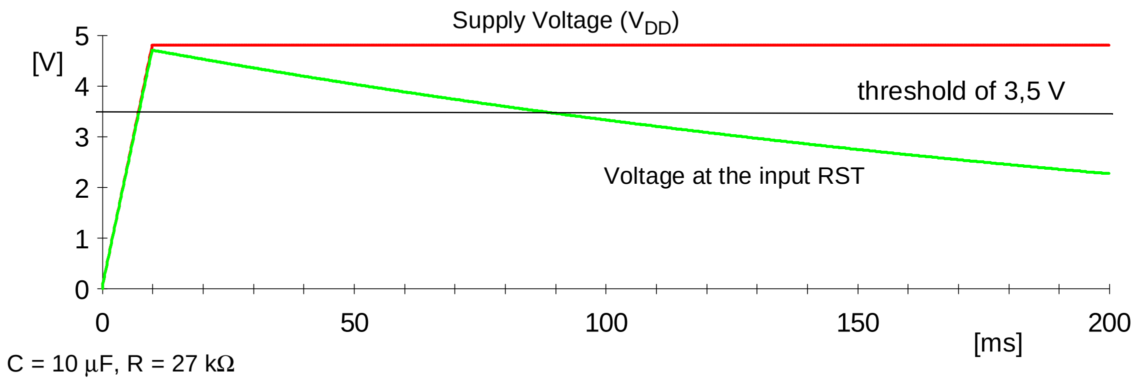


Fig. 5 Switching on the supply voltage

Fig. 5 and Fig. 6 show the supply voltage and the voltage at the RST-input during ramp-up and ramp-down (simulated results). With the selected components the voltage at the input RST stays above a threshold of 3,5 V for about 75 ms, giving enough scope for using components with greater tolerances.
These values are achieved only if the supply voltage ramps up in less than 10 ms.

This simple reset circuit has one general disadvantage: Short dips of the supply voltage (brownout condition) are not detected and don‘t result in a proper reset pulse for the P82C150. Furthermore the rise time of the supply voltage has to be shorter than a given maximum value (as indicated in the above example).

3.1.2 Reset with a Supervisory Circuit

The drawbacks of the simple reset circuit are removed by employing an integrated power supervisory circuit, either a stand-alone device or combined with a voltage regulator. The supervisory circuit has to generate a positive going reset pulse for being able to reset the P82C150. This pulse has to be held stable during a given time

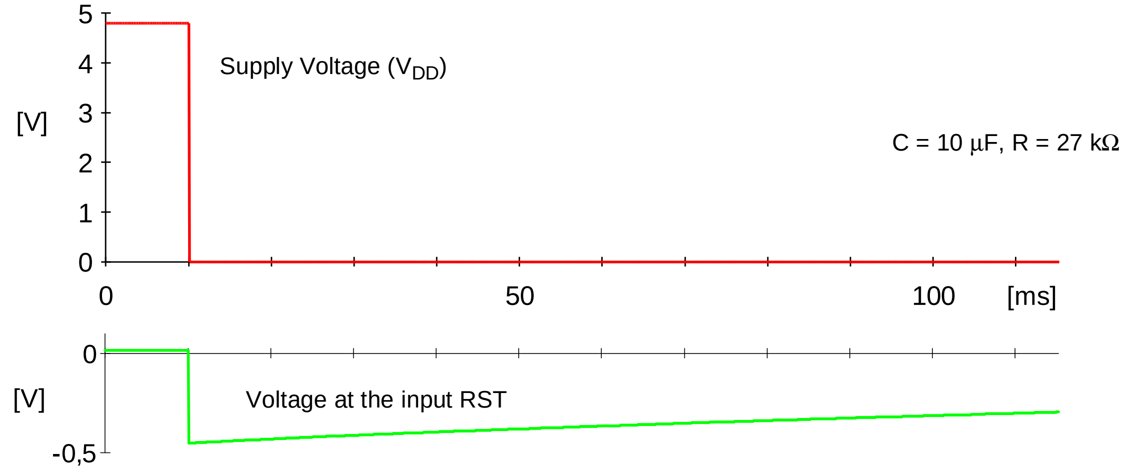


Fig. 6 Switching off the supply voltage

after power on according to [1]. Normally these supervisory circuits are designed to detect dips of the supply voltage below a certain value properly (brownout condition). The min. supply voltage for a proper operation of the P82C150 is given in the data sheet [1].

An example using the supervisory circuit PCF1252 from PHILIPS is given in chapter 3.2 (see Fig. 8).

3.2 Identifier Programming

Four bits of the identifiers used by the P82C150 are programmable via pins while 6 bits are fixed and one bit defines the direction of the message flow - to or from the SLIO (see also chapter 6 on this item). Thus the CAN-identifier distinguishes between the different SLIOs (max. 16 in a system) and their messages in a system. Each SLIO has its own identifier, which is set by applying voltage levels (V_{DD} or V_{SS}) at the pins of the I/O ports P0...P3 during the reset phase. The input levels are read into the identifier latch with the falling edge of the RST input signal. The voltage levels must be held stable for a given time (see [1]) after the falling edge of the reset (see Fig. 7).

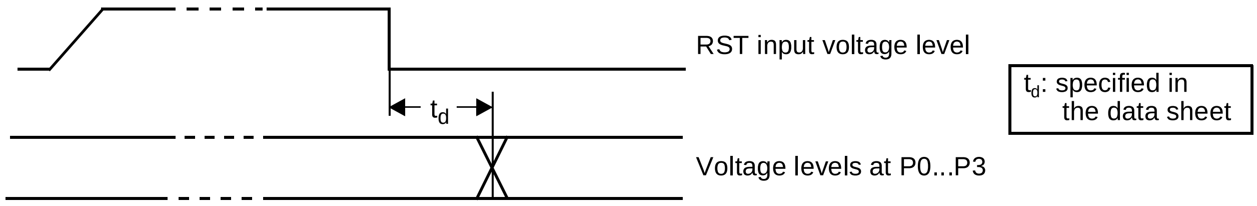


Fig. 7 Hold time definition for identifier levels at P0...P3

These four pins may be used as I/O-ports after the end of the reset. If the voltage levels for the identifier information are determined by resistors, their presence have to be taken into account when designing the circuitry attached to P0...P3.

If a pin is used as an output port, the identifier resistor (pull-up or pull-down) puts an extra load on the output. The resistor should therefore be high-ohmic (suggested range of about 100 kΩ). Care has to be taken of the reaction of the connected circuit during reset, when the port output is set to HIGH-impedance state and is used as an input reading the programmed identifier information provided by the resistor connected to the pin. The logic level at this pin must not create hazardous conditions of the connected circuit during reset. And the other way round, the connected circuit must not falsify the input port level during reset, in order to reading the programmed identifier into the internal latch correctly.

If a pin is used as an input port, the driving circuit has to override the input voltage produced by the connected resistor during normal operation. During the reset phase the connected output of the application has to be switched into a HIGH-impedance state to allow for proper voltage level generation as defined by the identifier resistor at this pin. An example is given in Fig. 8.

Example: Usage of Identifier Pins as I/O Ports after Reset

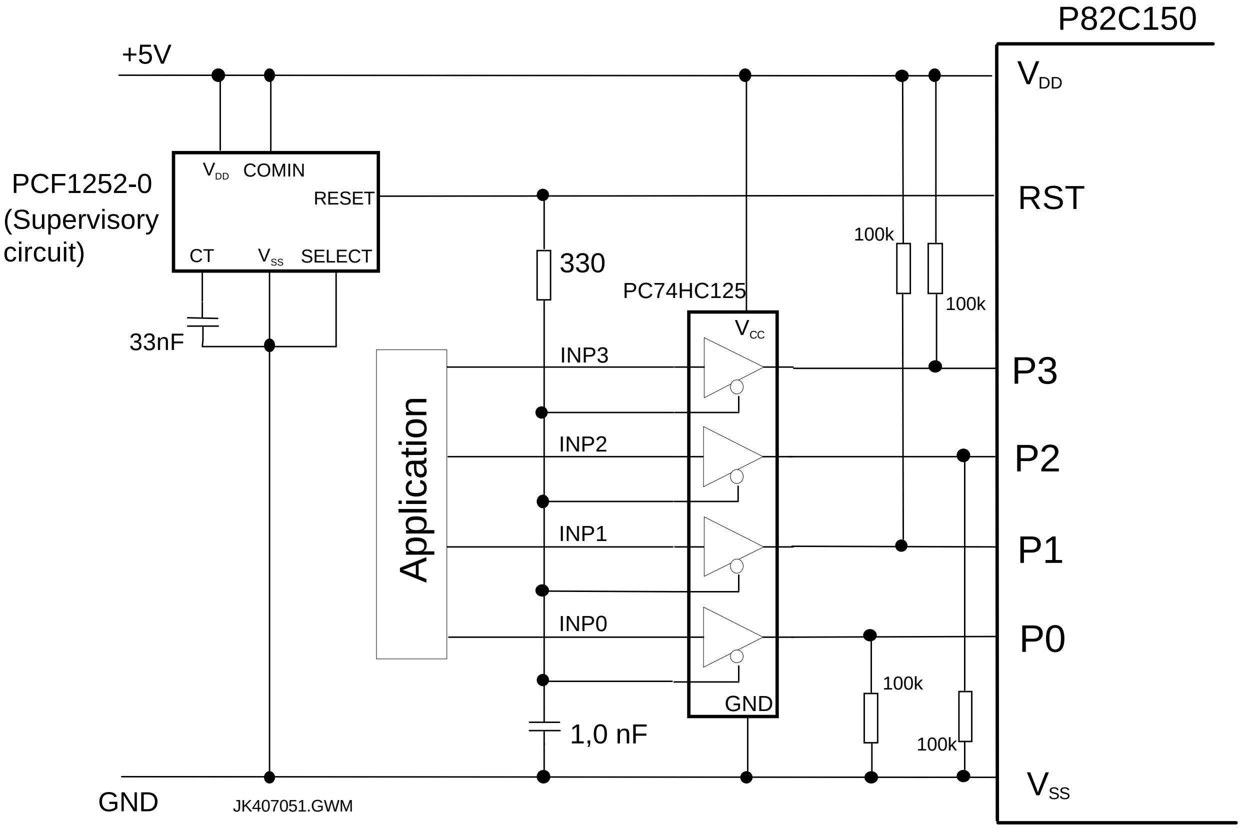


Fig. 8 Example: Using P0...P3 as inputs

The ports used for the programming of the identifier bits during reset may be used as inputs or outputs after the end of the reset pulse (see the latest data sheet [1] for the specification of the hold time). In cases, where the levels at the ports could be falsified (ports used as inputs) or the connected application would not allow certain input levels (ports used as outputs) during reset, it is necessary to implement buffers between the ports and the attached circuit. Fig. 8 shows an example where the ports are used as inputs.

The power supply supervisory circuit, PCF1250-0, is used for generating the HIGH-level active reset signal during ramp up and short dips below 4,75 V (typ.) of the power supply. With the capacitor of 33 nF at the input CT of the circuit a duration of 33 ms (typ., slew rate of the supply voltage <25 V/ms) for the reset pulse may be achieved. For the mentioned values and for further information about the possibilities of this circuit e.g a different voltage etc., please have a look into the data sheet of the PCF1252-x [9].

The buffer circuit 74HC125 is used for disconnecting the outputs of the application from the ports during reset. The levels at the ports P3 ... P0 (determined by 100 kΩ resistors) are latched with the H-to-L edge of the reset pulse. In order to prevent the application circuit from falsifying the levels at the port inputs, the outputs of the buffer 74HC125 are switched into TRISTATE during reset. An RC-combination at the $\overline{\text{OE}}$-inputs of the buffers (330 Ω / 1,0 nF) is used to delay the reactivation of the buffer outputs to achieve a hold time >200 ns (H->L-transition of the reset pulse <100 ns).

3.3 Physical Layer Interfacing

A CAN node consists not only of the protocol controller like the P82C150 but also needs an interface to the CAN-bus. For a balanced wire implementation of the bus the basic requirements of this interface are to provide

- a differential line driver for transmitting data,
- a receiver comparator for reading the data on the bus.

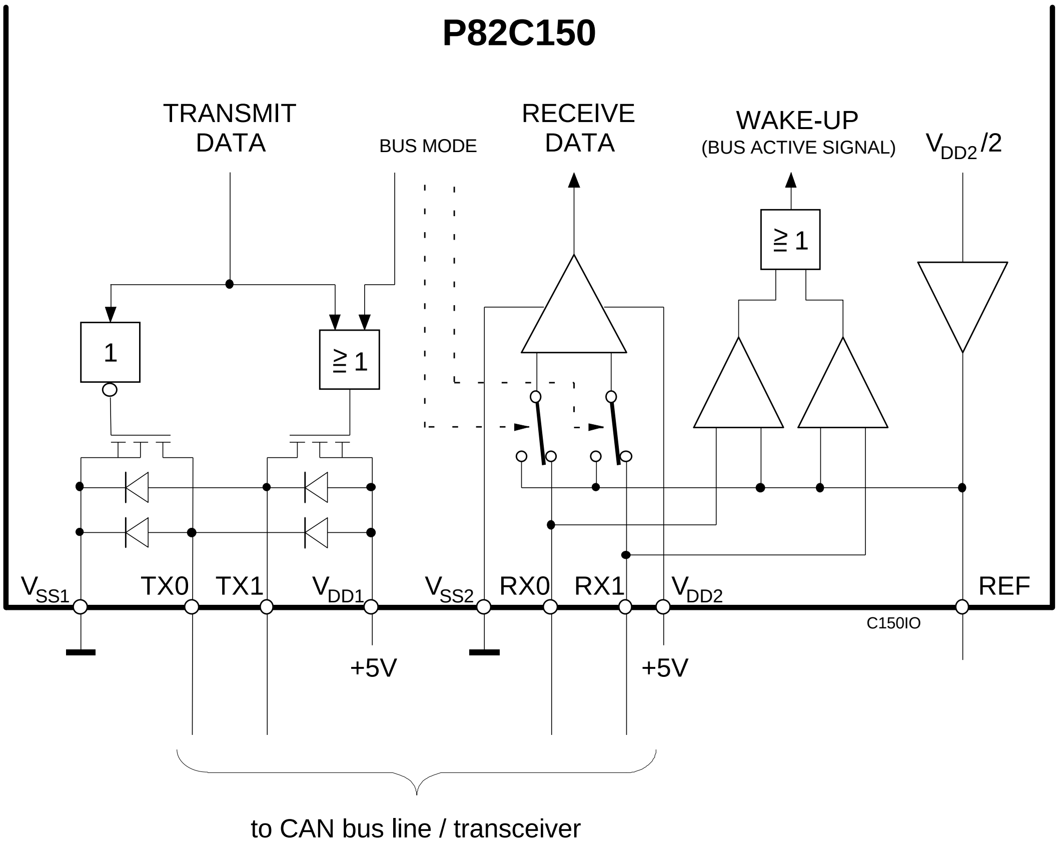


Fig. 9 Structure of the P82C150 on-chip CAN-bus interface

To some extent the controller itself comprises the main functions for such an interface which can be implemented either by external discrete components[1] or by an IC like the Philips transceiver PCA82C250 [3]. This transceiver is connected between the protocol controller and the CAN bus and represents the physical layer interface.

[1] For details on the implementation of discrete physical layer interfaces see [8]

Normally data transfer takes place on balanced bus wires in differential mode. In cases where communication in this mode fails the controller can enter one of two single wire modes in order to try to establish message transfer in this way.[2]

Table 1 CAN bus modes

		Reception Level		Transmission	
Bus mode	Status Bits BM1, BM0	Recessive	Dominant	TX1	TX0
0 Differential	00	RX0 > RX1	RX0 < RX1	enabled	enabled
1 One-wire RX1	01	RX1 < REF	RX1 > REF	enabled	enabled
2 One-wire RX0	10	RX0 > REF	RX0 < REF	disabled	enabled
3 Sleep	11	RX0 > REF and RX1 < REF	RX0 < REF or RX1 > REF	disabled	disabled

The structure of the interface that is built into the P82C150 is depicted in Fig. 9. There are two transmit outputs and two receive inputs. If the chip is connected to the bus by discrete components, TX1 and RX1 are connected to the bus wire CAN_H while TX0 and RX0 are connected to CAN_L. Of the two drivers for transmitting data one can be turned off. This is required in bus mode 2 where communication is maintained on one wire only and a short-circuit between the two wires has to be tolerated. Receiving data is done by a comparator. In bus mode 0 (normal mode) the differential signal voltage between RX0 and RX1 is evaluated. Bus modes 1 and 2 are one-wire modes. In mode 1 the input RX0 is disabled and the comparator input is connected to a reference voltage. In mode 2 input RX1 is disabled and RX0 is compared with the reference voltage. Bus mode 3 is sleep mode. Wake-up from this mode is done by two separate comparators which detect a dominant level on either wire. An overview of the bus modes is given in Table 1 (see also Fig. 12 on page 18). For more details see the data sheet of the P82C150 [1].

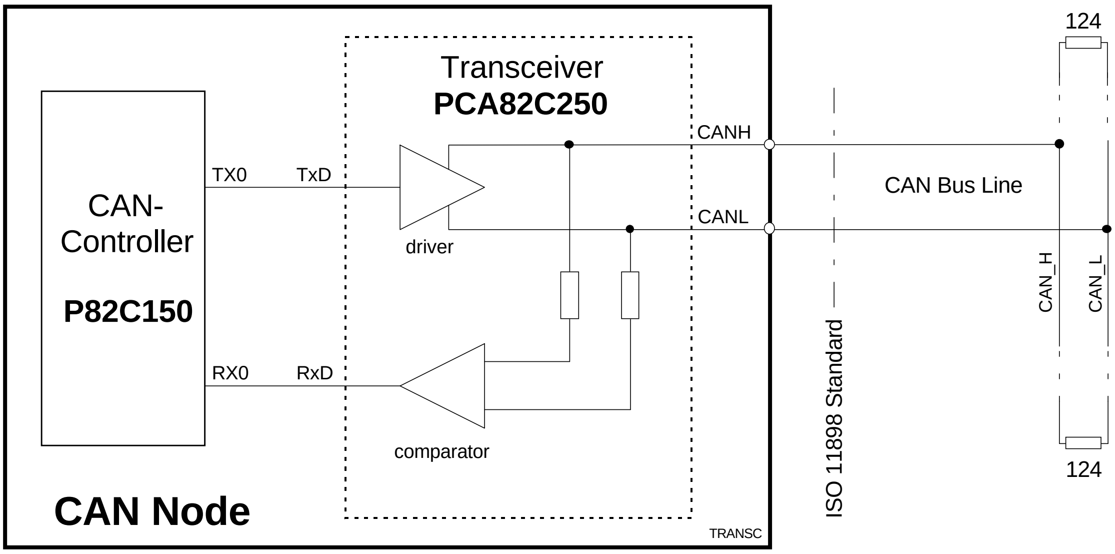


Fig. 10 Basic functions of a transceiver

Using a separate transceiver IC like the PCA82C250 not only simplifies the design and layout of a CAN node significantly but offers even more advantages. Due to its wide common mode range in receiving the differential bus signal it improves immunity against electromagnetic interference and thus provides increased reliability of the bus

[2] For details on bus failure management see ISO 11519-2 [5] and [10]

communication. Electromagnetic emission can be reduced by the ´slope control´ feature. This function decreases the slew rate of the signal transitions on the bus. In many cases this permits the extended use of twisted pair wires where otherwise a shielded cable would have to be chosen.

The basic functions of the transceiver PCA82C250 are depicted in Fig. 10. The CAN controller sends the serial data to be transmitted to the TxD input of the transceiver. Depending on the logical level the output drivers of the transceiver are activated. Due to the wired-AND characteristic of the bus only dominant levels (LOW) can be transmitted. During recessive states (HIGH) the drivers stay inactive and the bias network determines the bus voltage - unless another node is transmitting a dominant level. The receiver comparator converts the differential bus signal to a logical signal which is output at pin RxD and sent to the controller to be decoded. This receiver comparator is always active and thus monitors the bus while the node is transmitting. In this way the CAN controller can compare the received bits with the transmitted ones.

As shown in Fig. 3 on page 10 the SLIO pins Tx0 and Rx0 are used to transfer data to and from the transceiver. Tx1 is not used and Rx1 is put on a bias voltage of 1.7...1.8 V. The reason for this is to present a recessive bus level to the internal comparator, which is necessary for enabling the sleep mode (see Table 1). Furthermore a recessive bus level is seen, when the P82C150 is in bus mode 1 (a mode not used in this configuration) during the calibration procedure (see chapter 4, Fig. 12 and Fig. 13) and tries to receive data at pin Rx1 only. The transmit outputs Tx0 and Tx1 are designed to be connected directly to the bus and are therefore of open-drain type. If a transceiver is used these pins need an additional external pull-up resistor of about 560 Ω to speed up the otherwise slow LOW-to-HIGH signal transitions, because the TxD pin of the PCA82C250 is equipped with a weak internal pull-up resistor only.

The pin R_S of the transceiver is the slope control input. If it is connected to GND (V_{Rs} = 0 V, Fig. 3 on page 10), the transistors driving the bus operate at maximum speed. This gives short rise and fall times of the signals on the bus. Slope control is adjusted by connecting a resistor from pin R_S to ground. For more information about the slope control function please see the data sheet of the PCA82C250 [3].

Reducing the slew rate increases the total delay of the transceiver. This has to be taken into account when calculating the maximum achievable bus length.

The example of a bus interface as depicted in Fig. 3 on page 10 uses the same power supply ground for both the transceiver and the CAN controller. This means that the controller and its application circuitry are galvanically connected to the transceiver and the CAN bus. In cases where this is not desired, optocouplers can be placed between the transceiver and the controller. Fig. 11 shows an example with the controller and its application being isolated from the transceiver and the bus.

Using optocouplers increases the delay of a node. The signal has to pass these devices twice per node (in the transmitting and receiving path). This has to be taken into account when calculating the maximum achievable bus length (see chapter 7.4).

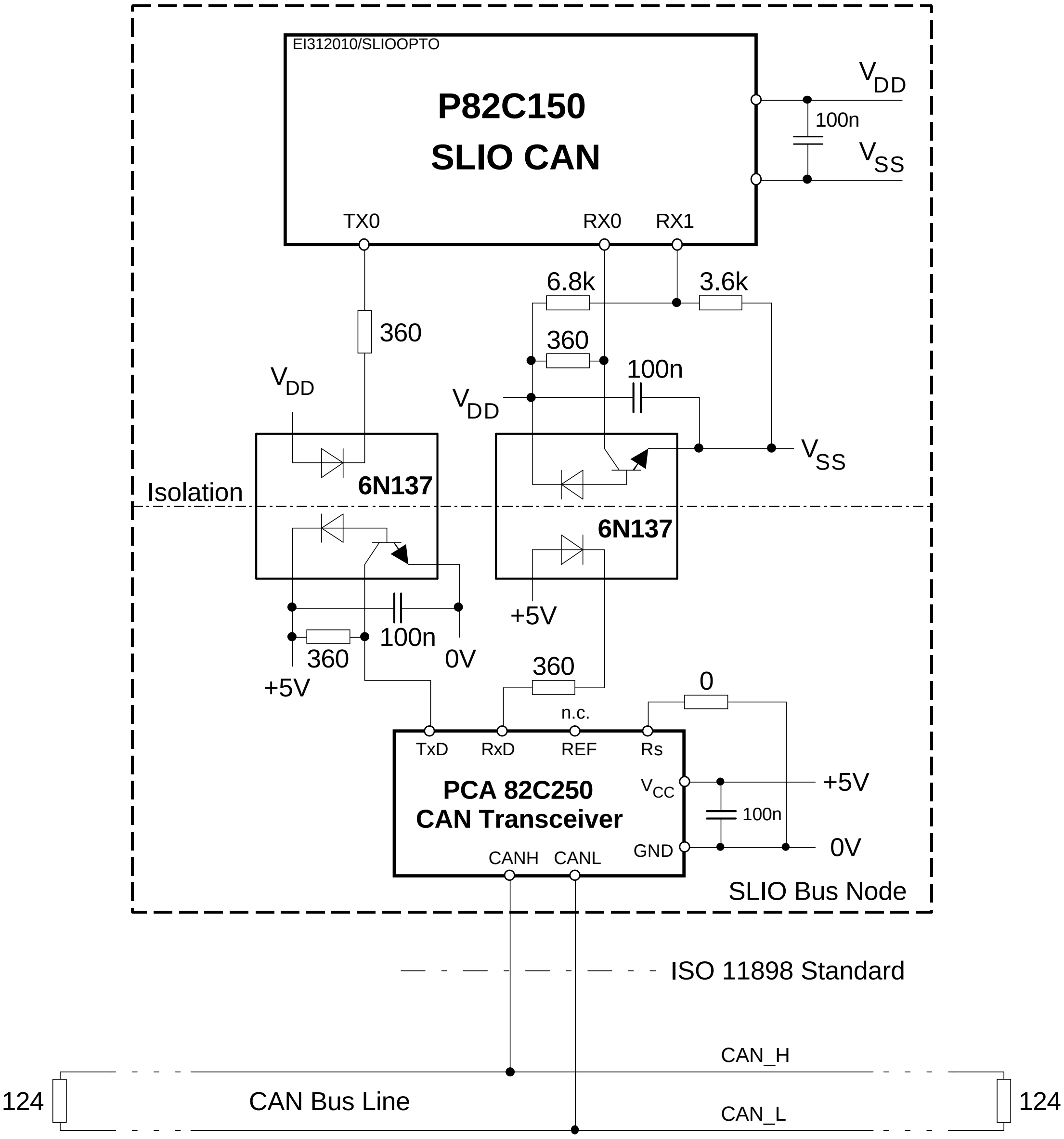
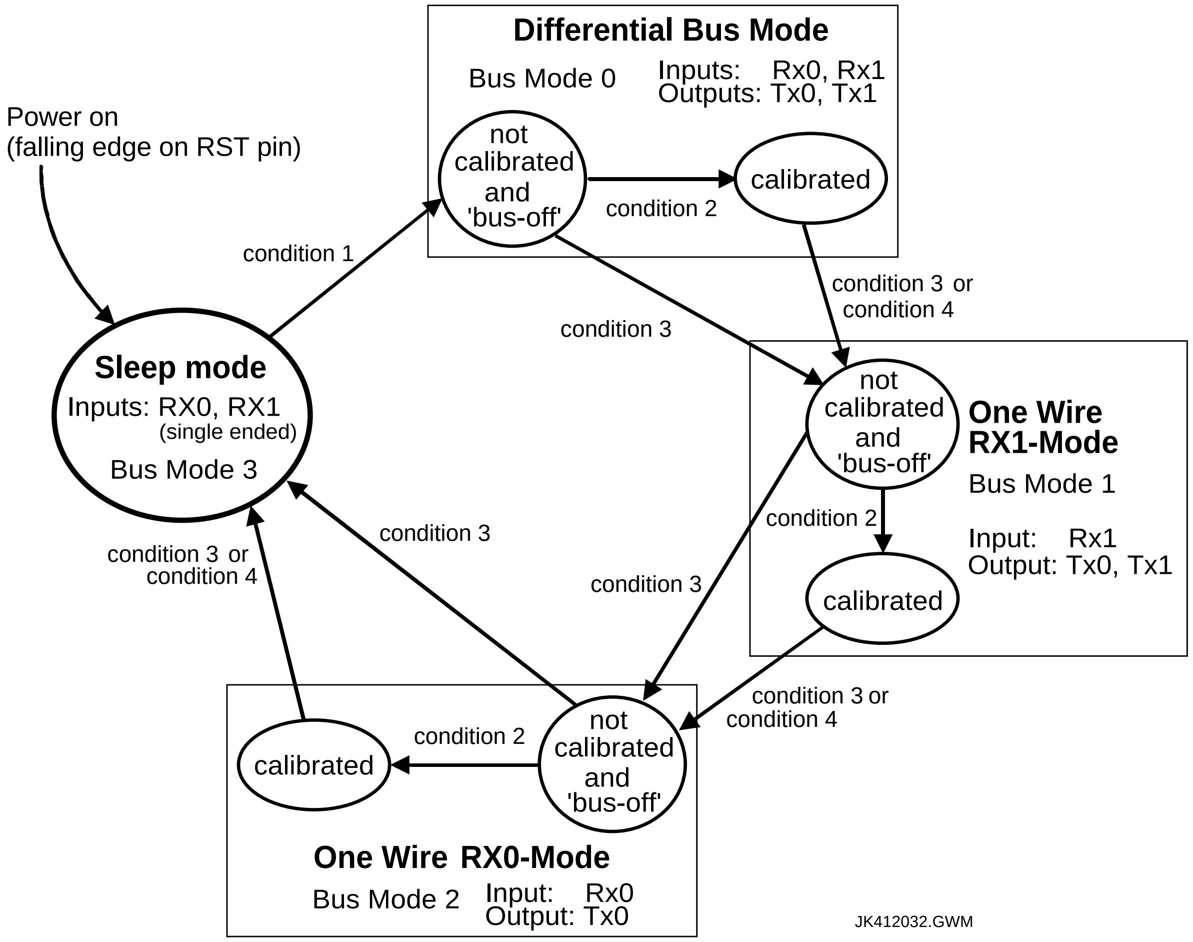


Fig. 11 Example of a CAN bus interface with galvanic isolation using optocouplers

4. INITIALIZATION OF THE P82C150

The P82C150 has been designed for the implementation of low cost nodes in a CAN network in order to reduce the overall system cost. The reduction is achieved by avoiding as many external components as possible. Therefore an on-chip oscillator without external circuitry has been designed into the P82C150 and an automatic bit rate detection has been implemented. The P82C150 calibrates its internal bit clock to the bit timing of the bus by means of a specific calibration message (see chapter 4.2).

A P82C150 in a SLIO node may be regarded as a port and function extension of a remote host microcontroller. It is programmable with respect to its I/O-ports. This implies that it has to be initialized after power-on before it may be used in a system.



Condition 1: dominant bit detected on Rx0 or on Rx1
Condition 2: successful performance of the calibration procedure according to Fig. 13
Condition 3: bit counter overflow (>8191), causes also ‘bus-off‘ state
Condition 4: Transmit Error Counter overflow (>255), causes also ‘bus-off‘ state

Fig. 12 CAN-bus modes, calibration status and switch over conditions

4.1 Reset and Bus Mode Change

After a power-on reset the P82C150 enters the sleep mode — one of 4 different bus modes, which are supported by the P82C150 (see Fig. 12 and Table 1 on page 15, further information on bus modes is found in the data sheet [1]). With the HIGH-to-LOW transition of the reset pulse the information at the port pins P3 ... P0 is latched into the identifier latch for being used as part of the acceptance filter and the transmission identifier. All internal I/O registers are cleared. In this mode the P82C150 is uncalibrated and in the 'bus-off' state ([4]) only scanning the bus line inputs (Rx0 and Rx1) for a dominant level. If a dominant level is detected, the P82C150 wakes up and starts a calibration procedure in bus mode 0 (differential mode) according to Fig. 13 and chapter 4.2. If the P82C150 has not successfully calibrated its bit clock within 8192 local bit times, it will change to bus mode 1. The bit counter is reset and a new calibration procedure is started. If the P82C150 neither can calibrate its internal bit clock in bus mode 1 nor in bus mode 2, it changes to sleep mode, again scanning the bus line inputs for a dominant level. Any dominant level will restart such a 'trying-to-calibrate' cycle until it is met with success.

If the bus transceiver PCA82C250 is connected to the P82C150 as given in the typical application of Fig. 3 on page 10, the P82C150 can never calibrate its internal bit clock in bus mode 1 as Rx1 is connected to a fixed recessive level. When changing to bus mode 1 after having been unsuccessful in bus mode 0 it will always get a bit counter overflow and change to bus mode 2 (condition 3 in Fig. 12).

4.2 Initial Calibration Procedure after Reset or Bus Mode Change

The initial calibration procedure as given in Fig. 13 is discussed in this chapter. Special aspects of the calibration of SLIO nodes in different system configurations and during normal operation are discussed in chapter 5.

After a power-on reset or a bus mode change the P82C150 is uncalibrated and 'bus-off'. The local bit time derived from the internal oscillator is set greater than 50 µs (see the data sheet [1] for the proper value after reset). It is calibrated using messages sent over the CAN-bus. A message on the bus causes the P82C150 to measure any distance between two transitions from recessive to dominant at the output of the CAN-bus input comparator. Only if one half of this time distance is shorter than the current local bit time, it is taken as a better approximation and will be used as local bit time for receiving further messages (stepwise adaptation of the internal bit clock).

In parallel to the calibration process of the local bit time length the P82C150 has started the 'bus-off' recovery sequence using the current local bit time. According to the ISO Standard [4] it has to monitor at least 128 blocks of 11 consecutive recessive bits in the bit stream before being able to leave the 'bus-off' state.

Any message on the bus may roughly calibrate the local bit time of the P82C150, but only after having received the bit sequence '1010', the P82C150 can receive messages correctly. The early occurrence of this bit sequence after reset or bus mode change shortens the time for a rough calibration of the internal bit clock. Any second message, which is received correctly according to the CAN-protocol (an acknowledge is not required but no active error flag is allowed), verifies that the local bit time is in line with the bit time on the bus. During this state of rough calibration the P82C150 may receive messages and set ports into defined states. As long as the P82C150 neither is fine calibrated nor has finished the 'bus-off' recovery sequence, the P82C150 does neither give an acknowledge nor transmit any messages on the bus.

After having verified the rough calibration of the internal bit clock, the P82C150 must receive a calibration message[3] correctly within 8192 local bit times after wake up or bus mode change. Then it has fine calibrated its internal bit clock and the local bit time corresponds to the bit time on the bus. All internal I/O registers are cleared once more, as they may contain garbled information. Before the P82C150 may take part in the communication on the CAN-bus as a member with all rights (receiving **and** transmitting), it has to finish the 'bus-off' recovery sequence. Then it is 'on-bus' as an 'error active' node and sends the sign-on message[4], which signals the controlling host, that the P82C150 now is able to transmit messages and its internal data registers are ready for a configuration (see chapter 4.3).

[3] A special message used only for calibrating the internal bit clock of the SLIOs in the CAN network. For more information see the data sheet [1] on 'Oscillator and Calibration', 'Calibration Message' and 'Bit time Calibration'.

After this initial calibration procedure the P82C150 expects to receive calibration messages at a regular basis, at least one message within 8192 local bit times (see chapter 5.3).

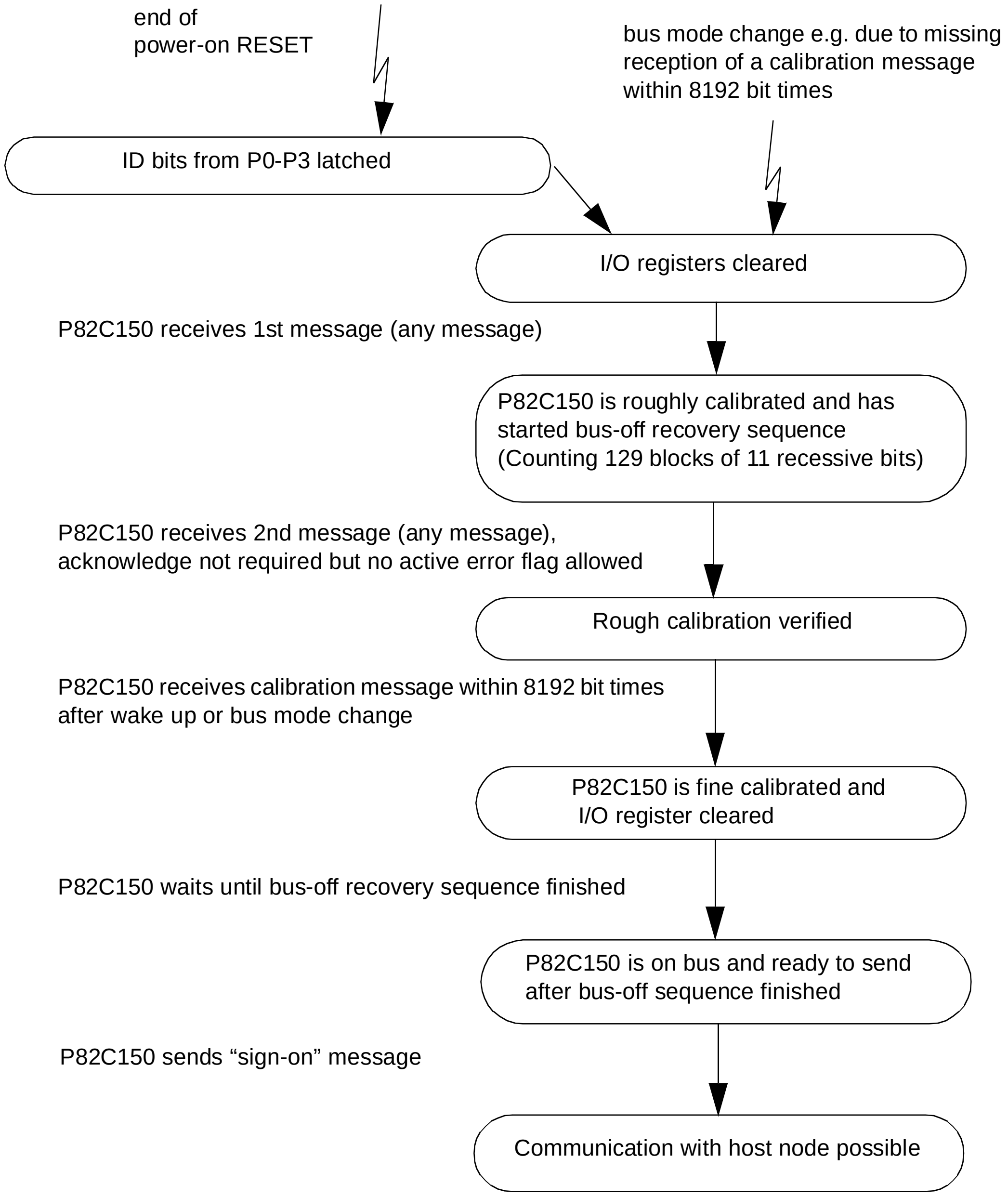

end of
power-on RESET
bus mode change e.g. due to missing
reception of a calibration message
within 8192 bit times
ID bits from P0-P3 latched
I/O registers cleared
P82C150 receives 1st message (any message)
P82C150 is roughly calibrated and has
started bus-off recovery sequence
(Counting 129 blocks of 11 recessive bits)
P82C150 receives 2nd message (any message),
acknowledge not required but no active error flag allowed
Rough calibration verified
P82C150 receives calibration message within 8192 bit times
after wake up or bus mode change
P82C150 is fine calibrated and
I/O register cleared
P82C150 waits until bus-off recovery sequence finished
P82C150 is on bus and ready to send
after bus-off sequence finished
P82C150 sends “sign-on” message
Communication with host node possible

Fig. 13 P82C150: Calibration procedure after reset and bus mode change

4. See the data sheet [1] for more information about the sign-on message.
ATTENTION: In a sign-on message the error warning bit (EW) is set, but nevertheless internally the status of the error counters are cleared. The error warning bit in a sign-on message must be ignored.

4.3 Configuration of internal Control Registers

After a successful calibration of a SLIO node, the P82C150 needs a configuration of its internal data registers. The host node has to send configuration messages, which prepare the ports for the specific needs of the connected application at the SLIO node. Table 2 gives an overview of the different port functions which may be selected.

Table 2 Alternative port functions

Port Function			Port Number																
			16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
identifier programming																X	X	X	X
digital I/O				X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
comparator input						X	X	X	X										
comparator output										X	X								
DPM output									X						X				
ADC	1 signal only	input line		X															
		AD feedback out	X																
	2 to 6 signals	input lines				X				X	X	X	X	X					
		multiplexer out			X														
		AD converter in		X															
		AD feedback out	X																
ADC comparator output												X							

JK408021.GWM

During the reset phase the output drivers of all ports have been disabled (switched into tristate) and all registers are cleared (set to ‘0‘). The host should transmit all configuration and output data to the P82C150 before switching the output drivers on. Automatic transmission of data due to edge triggering should be activated not before having finished setting switches and output data and activated the drivers of the output ports in order to be undisturbed of triggered messages during the setup of the P82C150.

The recommended sequence of configuration messages for the initialization of the P82C150 is:

1. transmit the value for the configuration of the A/D-converter inputs (SW3 .. SW1) and the ports used for monitoring the outputs of the comparators (M3 .. M1) (register address 5) without starting an A/D-conversion (ADC=‘0‘)
2. transmit the values for the DPM outputs (if used) (register addresses 6 and 7)
3. transmit the data for the digital port outputs (register address 3)
4. enable the output drivers (output enable bits in register address 4)
5. define the edge trigger mode (register address 1 and 2):
   ‘polling‘ i.e. no edge trigger enabled, trigger on a positive or negative edge, trigger on both edges.
6. An A/D conversion may now be started by setting ADC=‘1‘ in the analog configuration register (address 5) or by addressing the A/D data register (address 8).

5. CALIBRATION STRATEGY FOR THE P82C150

A network containing one or more SLIO nodes, which need calibration messages from other nodes, must have at least one node operating autonomously. Such nodes (in the following discussion they are called ***'normal node'***s in contrast with SLIO nodes) may serve as **host node**s for the communication with SLIO nodes. If normal nodes are **crystal-operated** (clock stability must be within 0,1%), they may send calibration messages to SLIO nodes. In general the host function and the calibration function are combined in one normal node, but it is allowed to distribute these functions to different normal nodes (see chapter 6.2). For the discussion in this chapter it is assumed that the host node also takes care of the calibration function.

After wake-up or bus mode change the P82C150 needs to be calibrated by an operating CAN-node sending specific calibration messages. As long as the P82C150 is uncalibrated and bus-off it is unable to acknowledge any message. Once it is calibrated and operating, it needs recalibration messages on a regular basis. Calibration messages must be sent by a crystal-operated normal node.

For the further discussion it is necessary to distinct between different sequences of calibration messages dependent on the number and the status of the normal nodes in the CAN-network:

- CAN-bus systems with only **one** normal node (host node and crystal-operated for the communication with and the calibration of SLIO nodes) and one or more **uncalibrated** SLIO nodes (see Fig. 14).
  - Sequence if the host node is error active after start-up of the SLIO nodes (see chapter 5.1.1).
  - Sequence if the host node is error passive after start-up of the SLIO nodes (see chapter 5.1.2).
- CAN-bus systems with **more** than one normal node (e.g. one crystal-operated host node for the communication with and the calibration of SLIO nodes and other normal nodes) and one or more **uncalibrated** SLIO nodes (see Fig. 17).
  - Sequence after start-up of the SLIO nodes (see chapter 5.2).
- Sequence of messages for the **re**calibration of **operating** SLIO nodes (see chapter 5.3).

For the mentioned situations different optimal sequences of calibration messages are suggested. From the discussion in the following chapters a recommended strategy for calibrating SLIO nodes may be derived (see chapter 5.4).

5.1 A Network with only One Normal Node (Host Node)

5.1.1 The Host Node is Error Active

After a wake-up or bus mode change the P82C150 is uncalibrated and bus-off. It is not allowed to acknowledge any message. Thus an error active host node, if it is the only normal node with a fixed bit rate in a CAN network (Fig. 14), will detect acknowledge errors after having sent a calibration message. It then sends an error frame destroying the 'End of Frame' field as the transmitted active error flag consists of dominant bits. Therefore the P82C150 does not receive a correct message, which is necessary for the verification of the rough calibration (see Fig. 13). After having transmitted 16 calibration messages without receiving an acknowledge, the host becomes error passive, which means that it transmits an error frame with a passive error flag (recessive bits) after having detected the acknowledge error. This will not destroy the 'End of Frame' field and the SLIO node receives the 17th calibration message correctly and can verify the rough calibration. The reception of one further calibration message (the 18th) enables the P82C150 to fine calibrate its internal bit clock. After having finished the 'bus-off' recovery sequence the P82C150 is 'on-bus' and sends its sign-on message.

Fig. 15 gives an example of a calibration procedure of an error active host node, achieving a fast calibration of all P82C150 devices in a network e.g. after power on. After having received a burst of at least 18 calibration messages the P82C150s are fine calibrated. For achieving a short 'bus-off' recovery time the host should now stop sending any further messages. This will result in a long 'bus idle' period, which may be used by the P82C150s for counting the remaining blocks of 11 recessive bits (129 blocks requested in total). If further (re)transmissions of messages are not stopped in time before the sign-on messages are sent by the SLIOs, the host node would fail to receive the first sign-on message due to its start during the passive error flag of the host.

A possible flow chart for programming a CAN controller (e.g. P8xC592 or P8xCE598) for such a bit time calibration procedure is given in the data sheet [1].

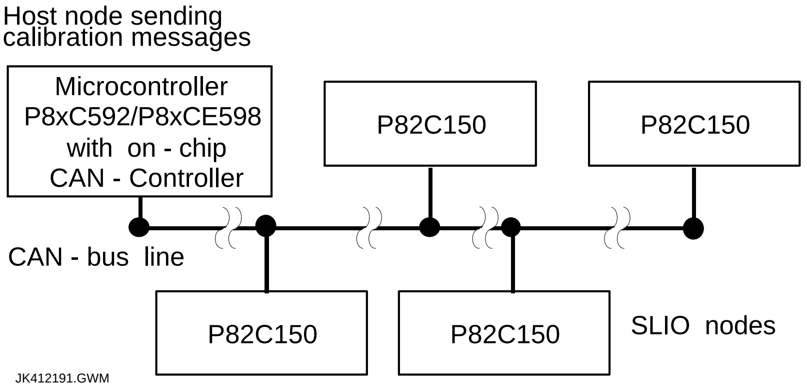

Host node sending calibration messages
Microcontroller P8xC592/P8xCE598 with on - chip CAN - Controller
P82C150
P82C150
CAN - bus line
P82C150
P82C150
SLIO nodes
JK412191.GWM

Fig. 14 CAN network with one normal node (host node)

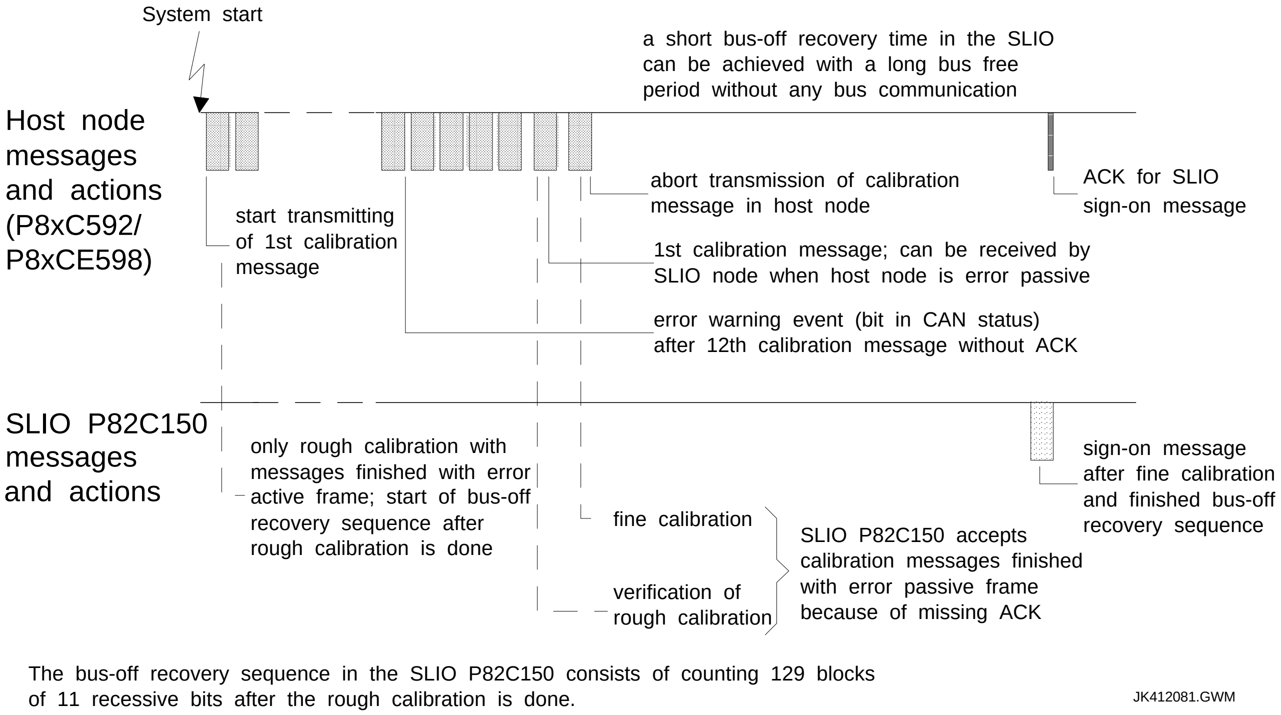

System start
a short bus-off recovery time in the SLIO can be achieved with a long bus free period without any bus communication
Host node messages and actions (P8xC592/ P8xCE598)
start transmitting of 1st calibration message
abort transmission of calibration message in host node
ACK for SLIO sign-on message
1st calibration message; can be received by SLIO node when host node is error passive
error warning event (bit in CAN status) after 12th calibration message without ACK
SLIO P82C150 messages and actions
only rough calibration with messages finished with error active frame; start of bus-off recovery sequence after rough calibration is done
fine calibration
verification of rough calibration
SLIO P82C150 accepts calibration messages finished with error passive frame because of missing ACK
sign-on message after fine calibration and finished bus-off recovery sequence
The bus-off recovery sequence in the SLIO P82C150 consists of counting 129 blocks of 11 recessive bits after the rough calibration is done.
JK412081.GWM

Fig. 15 Example of a calibration sequence if the host node is error active (e.g. after power-on)

5.1.2 The Host Node is Error Passive

In case the normal node (host node), which is sending calibration messages, is error passive already not destroying the ‘End of Frame‘ field (compare with the discussion on an error active host in chapter 5.1.1), a fast calibration sequence may be achieved with the example given in Fig. 16. The first calibration message is used for the rough calibration and starts the ‘bus-off‘ recovery sequence. The second message will be received correctly and is used for the verification of the rough calibration. Thus the third message is used by the P82C150 for the fine calibration of its internal bit clock. If the host generates a long ‘bus idle‘ period, the SLIO node can count the requested 129 blocks of 11 recessive bits for ‘bus-off‘ recovery in the shortest possible time.

The necessary three calibration messages may also be distributed equally as shown in the example given in Fig. 18.

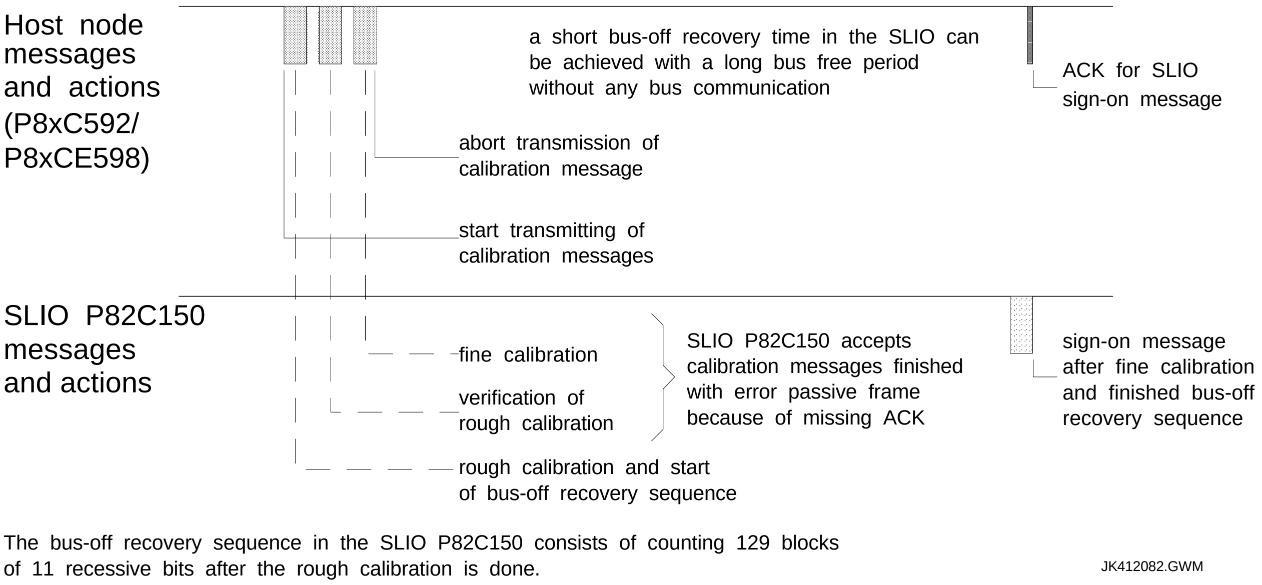

Host node messages and actions (P8xC592/ P8xCE598)
a short bus-off recovery time in the SLIO can be achieved with a long bus free period without any bus communication
ACK for SLIO sign-on message
abort transmission of calibration message
start transmitting of calibration messages
SLIO P82C150 messages and actions
fine calibration
verification of rough calibration
SLIO P82C150 accepts calibration messages finished with error passive frame because of missing ACK
sign-on message after fine calibration and finished bus-off recovery sequence
rough calibration and start of bus-off recovery sequence
The bus-off recovery sequence in the SLIO P82C150 consists of counting 129 blocks of 11 recessive bits after the rough calibration is done.
JK412082.GWM

Fig. 16 Example of a calibration sequence if the host node is error passive

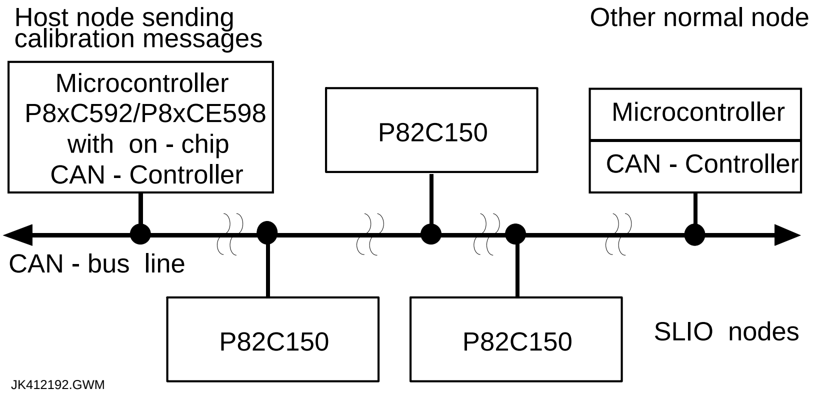


Fig. 17 CAN network with more than one normal node

5.2 A Network with More than One Normal Node

If there are more than one normal node in a CAN system (see Fig. 17), each message always gets an acknowledge from the other normal node(s), i.e. the messages are not destroyed by error frames, which corresponds to the case discussed in chapter 5.1.2 for an error passive host. Each calibration message may be used by the SLIO node for the calibration of its internal bit clock achieving a fast fine calibration after having received three messages correctly. Both examples given in Fig. 16 and Fig. 18 apply for this situation.

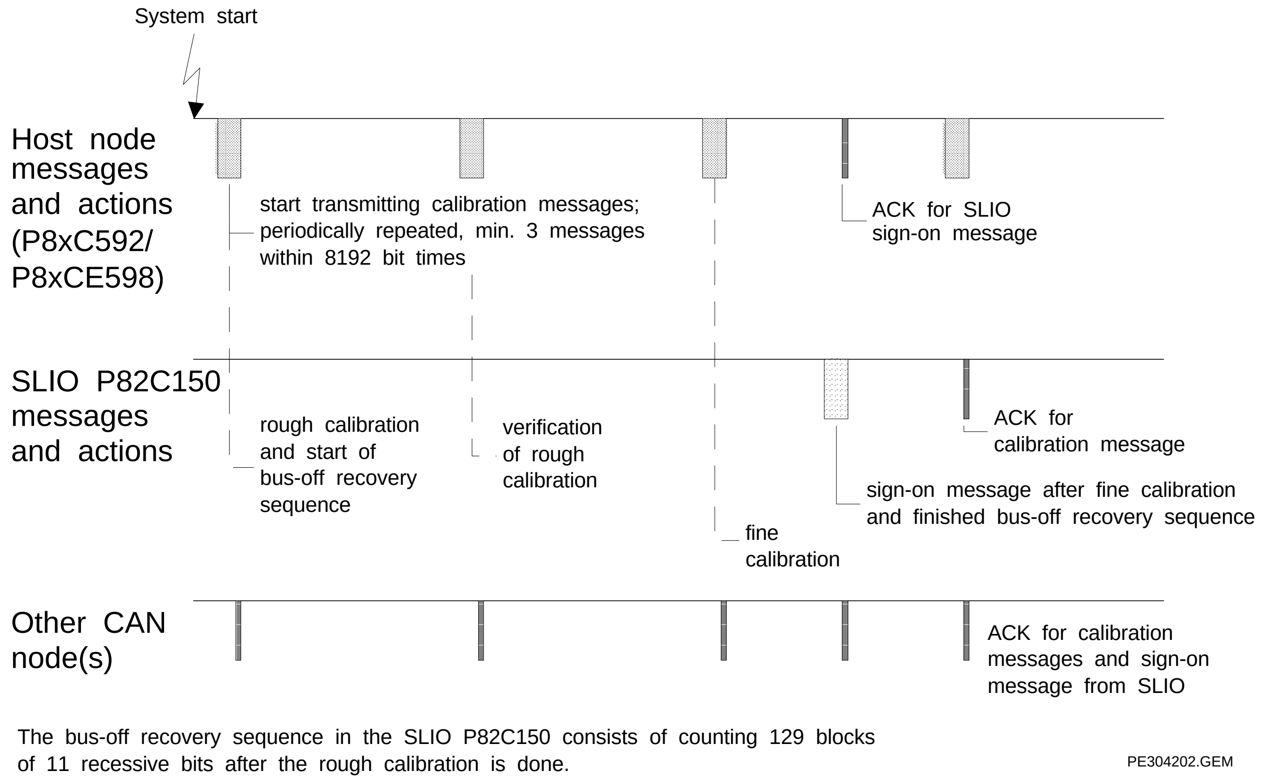

System start
Host node messages and actions (P8xC592/ P8xCE598)
start transmitting calibration messages; periodically repeated, min. 3 messages within 8192 bit times
ACK for SLIO sign-on message
SLIO P82C150 messages and actions
rough calibration and start of bus-off recovery sequence
verification of rough calibration
ACK for calibration message
sign-on message after fine calibration and finished bus-off recovery sequence
fine calibration
Other CAN node(s)
ACK for calibration messages and sign-on message from SLIO
The bus-off recovery sequence in the SLIO P82C150 consists of counting 129 blocks of 11 recessive bits after the rough calibration is done.
PE304202.GEM

Fig. 18 Example of a calibration sequence with 2 normal nodes

5.3 Recalibration of an Operating P82C150

Following the initial calibration the P82C150 needs a regular recalibration, as the frequency of the internal oscillator varies with temperature and supply voltage changes. It is expected that the temperature of the environment and the supply voltage does not change significantly between the reception of two calibration messages (short time stability of temperature and supply voltage). If the P82C150 does not receive a calibration message within 8192 local bit times, it switches into the ‘bus-off‘ state and tries to calibrate its local bit time in a different bus mode (see data sheet [1], Fig. 12 and Fig. 13).

Fig. 12 on page 18 gives an overview of the different conditions under which the P82C150 switches between the ‘calibrated‘ and ‘not calibrated‘ state. After power-on the P82C150 is not calibrated (sleep mode). It enters the calibrated state after having received calibration messages as discussed in chapter 4.2, chapter 5.1 and chapter 5.2. After a successful execution of the ‘bus-off‘ recovery sequence it is fully operating on the CAN-bus. Two conditions may bring the P82C150 back into the ‘not calibrated‘ state at any time:

- The bit counter overflows, because a calibration message has not been received within 8192 bit times.
- The Transmit Error Counter overflows (>255) and the node is switched into the ‘bus-off‘ state ([2], [4] or [5]).

In order to keep a network with SLIO nodes alive at least one calibration message within 8192 local bit times is requested for a recalibration of the P82C150s. Due to the inaccuracy of the calibration process and the allowed oscillator tolerance in a CAN-bus system a repetition period lower than 8000 bit times (at the host node) is recommended [1].

5.4 Recommended Calibration Strategy for the P82C150

In the previous chapters four possible sequences of calibration messages were discussed, which adapts to different situations of the network:

1. One burst of 18 calibration messages (see Fig. 15).
2. Blocks of three calibration messages every 8000 bit times (host node) (see Fig. 16).
3. Equally distributed calibration messages every 3800 bit times (host node) (see Fig. 18).
4. One calibration message every 8000 bit times (host node).

In Table 3 characteristics of the discussed sequences are given.

Table 3 Characteristics of Calibration Sequences

Number	start-up (after power-on) or restart (after bus mode change) of SLIOs	recalibration	bus load (recalibration)*
1.	fast	not applicable	not applicable
2.	slow	yes	high (*~2,5%*)
3.	slow	yes	medium (*~1,75%*)
4.	not applicable	yes	very low (*~0,8%*)

* Calculated with a calibration message of 67 bits and an error free transmission.

The sequences 2. and 3. may be used in all situations but are not very fast for being used during start-up or restart of P82C150 devices. Therefore it is recommended to use one of the following combinations of sequences.

Table 4 Recommended Sequences of Calibration Messages

	Description	Properties
1 + 2 or 1 + 3	Sequence 1 (a burst of 18 calibration messages) is used after the power-on of the host node only (host is always error active). Afterwards sequence 2 or 3 is used continuously (applies for restart due to bus mode change and for recalibration).	fast start-up after power-on slow restart after bus mode change easy to implement, as no complex decisions have to be taken

In the demonstration software of the SLIO Evaluation Board (OM4272) the sequences 1., 2. and 3. are used for the calibration process. This solution may be used as an example for a more complex calibration, where each situation of the system is met with an optimized sequence of calibration messages[5].

Sequence 4. is optimized for the recalibration process with a very low bus load but is not applicable during the start-up or restart of an uncalibrated P82C150 (e.g. after power-on of a SLIO node). It should only be used in case a very low bus load is required, as it lacks the possibility to calibrate P82C150 devices which have been disconnected (e.g. hot-plugging of SLIOs, see chapter 6.2.2).

[5] for further information refer to the software flow given in the User Manual [6]

6. GENERAL COMMUNICATION ASPECTS OF SYSTEMS WITH SLIO NODES

The aspects of power-on reset, calibration after power-on, initialization and recalibration of a SLIO node equipped with the P82C150 have already been discussed in the previous paragraphs (see chapter 3, chapter 4 and chapter 5). This chapter discusses some other general aspects of CAN networks where one or more nodes are SLIO nodes.

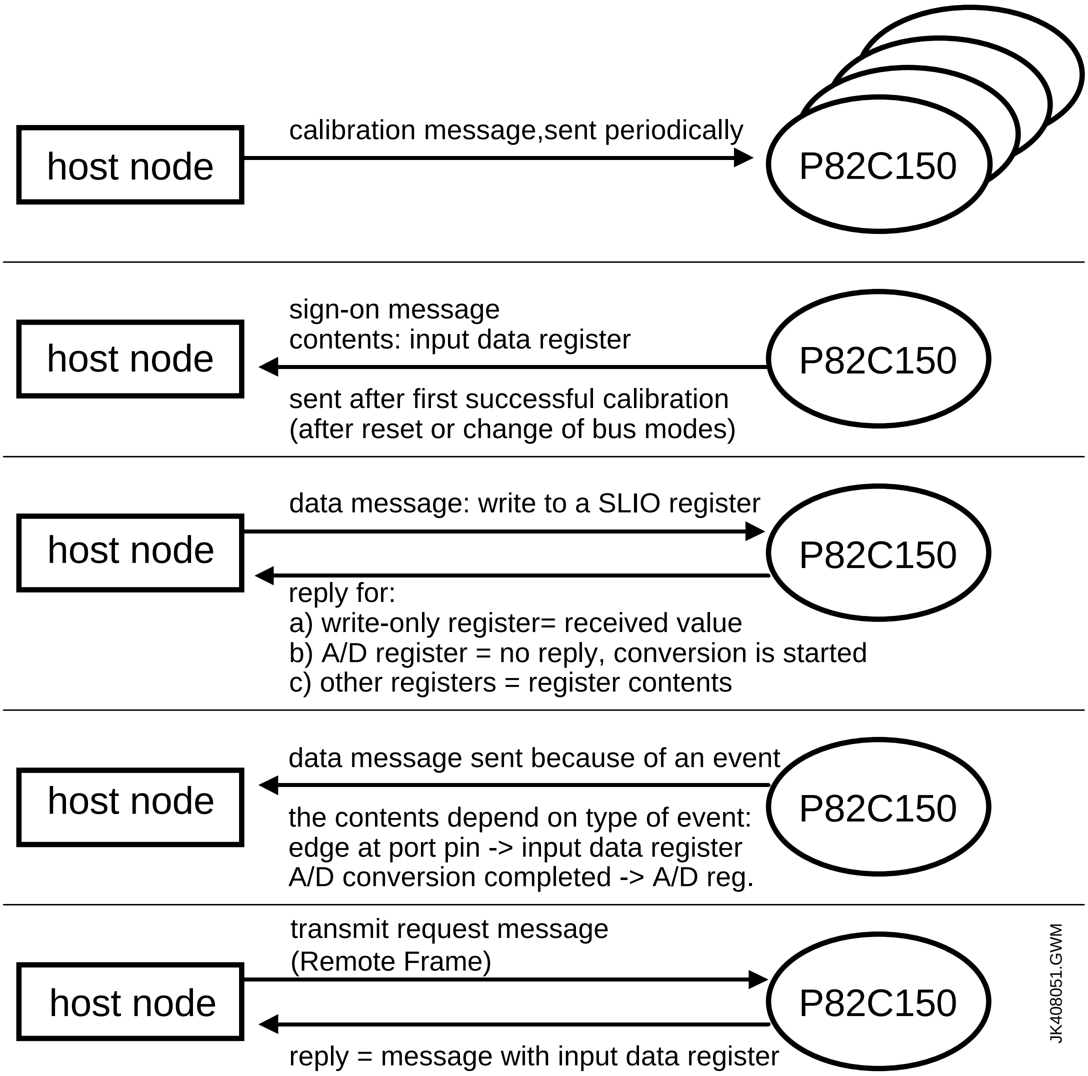

host node
calibration message,sent periodically
P82C150
host node
sign-on message
contents: input data register
sent after first successful calibration
(after reset or change of bus modes)
P82C150
host node
data message: write to a SLIO register
reply for:
a) write-only register= received value
b) A/D register = no reply, conversion is started
c) other registers = register contents
P82C150
host node
data message sent because of an event
the contents depend on type of event:
edge at port pin -> input data register
A/D conversion completed -> A/D reg.
P82C150
host node
transmit request message
(Remote Frame)
reply = message with input data register
P82C150
JK408051.GWM

Fig. 19 Exchange of messages between a host CAN controller and a P82C150

6.1 Message Structure

A CAN network cannot be built up only of SLIO nodes because they need at least one normal node (crystal-operated, host node), which takes care of the calibration and initialization of the SLIO nodes (see Fig. 14 on page 23). The messages, which are exchanged between a host and the SLIOs it is controlling, are in general calibration messages, the sign-on message, data messages and remote frames [1]. Fig. 19 gives an overview of all mentioned messages indicating the direction of the transmission and the reaction of the P82C150 on received messages.

The acceptance filter of the P82C150 has a fixed value for receiving calibration messages (identifier: “000 1010 1010”) and a partly fixed value for receiving data and remote frames. The variable part is defined by the voltage levels at P3 ... P0 during reset (see chapter 3.2, chapter 4.2 and [1]) and by the type of frame (identifier bit ‘DIR‘ [1]). The composition of the different messages are shown in Fig. 20 (general message format as defined by the CAN standard) and Table 5 (defining bits and bytes for the different messages of the P82C150).

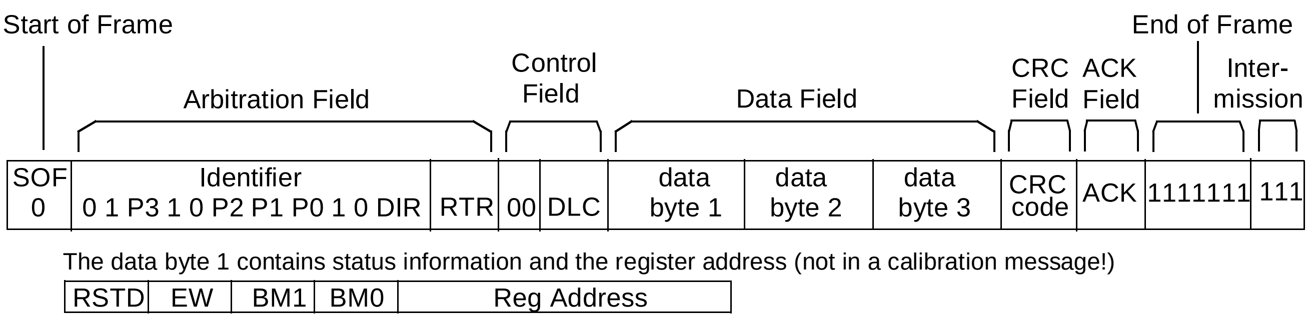


Fig. 20 General message format for the P82C150

Table 5 Specification of Bits and Data Bytes in the SLIO messages

				Data Byte 1				
message type	**DIR**	**RTR**	**DLC**	**RSTD**	**EW**	**BM1**	**BM0**	**Reg.Address**
data message (host to SLIO)	0	0	0011	X	X	X	X	current
data message (SLIO to host)	1	0	0011	0	EW	BM1	BM0	current
sign-on message	1	0	0011	1	1*	BM1	BM0	0000
Remote Frame	1	1	0011					

* In the sign-on message the error warning bit (EW) is set to “1”. Nevertheless the error counters are cleared. The error warning bit of the sign-on message must be ignored.

Because the priority of a SLIO message can not be changed (fixed identifier), the P82C150 delays the transmission of a further pending message for three bit times after each successful transmission (special ‘Suspend Transmission‘). This enables other CAN nodes with lower priority to transmit before a SLIO node starts a new transmission.

The fixed acceptance filter for each P82C150 in the system defines a bundle of messages (sign-on message, data message to SLIO, data message from SLIO and remote frames) for that specific SLIO. Not more than 16 such ‘bundle messages‘ (corresponds to the number of SLIO nodes) can exist in a CAN network and each bundle represents all messages for one SLIO node. It is not possible to define messages for functions e.g. with separate identifiers for ‘digital port functions‘ and ‘analog functions‘, which are sent to more than one SLIO node.

Only one host may send messages to that SLIO node it is controlling (except calibration messages). Messages from different hosts, if they are transmitted simultaneously, could be superimposed as they differ first after the arbitration field - in the data fields - and cause errors. Calibration messages have one special identifier, which is accepted by all P82C150s in a network. It does not matter, if the calibration messages are sent by one host to all SLIOs or if each host sends the calibration messages to its own SLIOs only. As long as the calibration messages contain the same data, a superimposition does not lead to errors.

6.2 Network Management

6.2.1 Calibration Server

The initial calibration of all P82C150s and holding them alive with a regular recalibration is one of the important tasks in a network. If the same bit stream for the calibration is used throughout the network for all nodes equipped with a P82C150, each host may calibrate its own SLIO nodes (Fig. 21 case 3), as errors are not produced, even in case the hosts would transmit simultaneously. If one host has been set into the ‘bus-off‘ state or is not more available for one reason or other, the second host, sending calibration messages to its own SLIO nodes, holds also the SLIO nodes of the first one alive.

In order to reduce the bus load it is recommended to assign the calibration task to one normal node (host node, crystal-operated) in the network. This node is then acting as a **calibration server** (Fig. 21 case 1). This solution has the disadvantage, that if the calibration server runs into severe problems and perhaps is switched into the ‘bus-off‘ state, all SLIO nodes will be taken out of the communication process due to the missing calibration messages. As every CAN node may listen to any message on the bus and use the information for its own purpose, another normal node (crystal-operated) may listen to the calibration messages and take over the task, if the calibration messages fail to be transmitted after a certain time (Fig. 21, case 2). With such a feature implemented in the network management the redundancy for keeping the SLIO nodes alive has been recovered and the bus load is reduced.

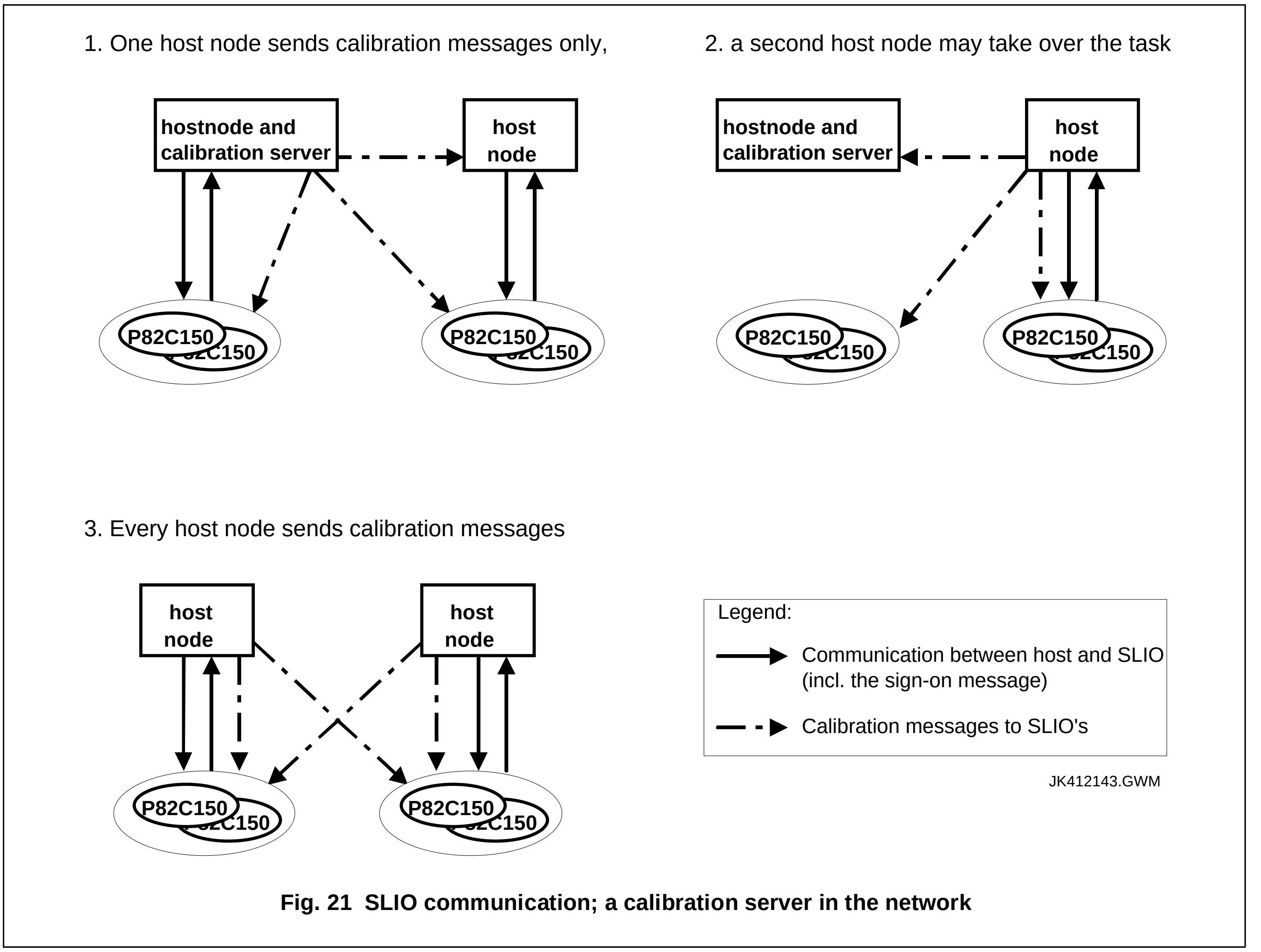

1. One host node sends calibration messages only,
2. a second host node may take over the task
hostnode and calibration server
host node
P82C150
P82C150
3. Every host node sends calibration messages
Legend:
Communication between host and SLIO (incl. the sign-on message)
Calibration messages to SLIO's
JK412143.GWM

Fig. 21 SLIO communication; a calibration server in the network

6.2.2 Hot-Plugging of SLIO Nodes

In some applications it is necessary to allow for ‘hot-plugging‘ of CAN nodes, which means that nodes can be switched on and off or taken physically out of the network without disturbing the communication in the network. The calibration procedure, which ends with the transmission of the sign-on message, is an instrument which may be used in a network management for this purpose.

If a calibration message does not reach a certain SLIO node, the host must not necessarily be informed (as long as the messages are acknowledged by other nodes). Thus SLIO nodes may be physically taken out, switched off or disconnected from the communication on the bus (e.g. due to a malfunctioning of bus lines). In the latter case the host is even sure that the P82C150 does not disturb the communication, as it will have entered the sleep mode after having tried - without success - to calibrate its bit time in the other bus modes. All its output drivers will be switched off.

The host will learn at the reception of a sign-on message from a SLIO node, that this node has been disconnected and can react in an appropriate way again. The sign-on message, which is sent by the P82C150 only after a successful calibration of its local bit timing after reset or bus mode change, is set up as a normal data message to the host transmitting the content of the input data register, but with the status bit RSTD set to ‘1‘ (see Fig. 20 and Table 5). Thus it is easily distinguished from all other data messages from the P82C150. The sign-on message indicates to the host, that a SLIO node had lost calibration due to some reason or other or had been ‘bus-off‘ since the last message, which had been sent to that node. This ‘lost&found‘ node needs an initialization of its internal data registers for further proper operation.

In order to be able to calibrate ‘lost&found‘ nodes again the strategy recommended in chapter 5.4 should be implemented in the network management.

6.3 Summary of Communication Conditions

- Who may communicate with whom in a network containing SLIO nodes?
  - A P82C150 should receive data messages (data frames) from one single host only.
  - A host node may transmit data messages and remote frames to several P82C150s.
  - A crystal-operated node may transmit calibration messages to all P82C150s in the network.
  - It is allowed, that several crystal-operated nodes transmit (the same) calibration messages.
  - Every normal node may transmit remote frames to each P82C150 in the network.
  - Every normal node may receive data messages and sign-on messages sent by all P82C150s in the network and use the received information for its own purpose.
  - Every normal node may receive the calibration message and use it for its own purpose.
  - A direct message exchange between P82C150s is **not** possible.
- Calibration of the local bit timing of the P82C150
  - An initial calibration of the bit timing is necessary.
  - A regular recalibration of the bit timing is necessary (e.g. one calibration message every 3800 bit times).
- The bit rate in a network with P82C150 nodes is restricted, as described in the data sheet [1].
- The location of the sample point and the size of the resynchronization jump width of the other CAN nodes have to match that of the P82C150, as described in the data sheet [1]. See also chapter 7.3.
- The maximum distance between the outermost bus nodes is reduced, compared to networks with ‘conventional‘ CAN controllers (see chapter 7.4).
- If CAN controllers according to the CAN specification V1.1 (e.g. 82C200 V0) are used together with SLIOs, a more stable power supply voltage for the SLIOs is required (refer to the data sheet [1]).

7. BIT TIMING AND BUS LENGTH

The SLIO P82C150 represents an easy and cost-effective way to implement a CAN node. The main reason for this is that the CAN controller does not need any on-chip or on-board software. It also does not need an own crystal oscillator, as it has a built-in clock generator and its local bit clock is automatically calibrated by the received bit stream. Any bit rate between 20 kbit/s and 125 kbit/s can be used. The calibration procedure is described in chapter 4.2 on page 19 of this application note.

7.1 Influences on the Clock Accuracy

This automatic bit clock calibration gives great flexibility considering the bit rate being used. A SLIO node can be hooked onto the bus without having its bit timing registers set - the only adjustments possible apply to the identifier bits.

However this calibration mechanism gives only limited accuracy. The main reasons for this are the following:

- The clock stability of the node sending the calibration messages must be within 0.1%. In calculating the SLIO node’s absolute clock accuracy this must always be included.
- The calibration mechanism does not continually monitor the bit stream for adjusting the clock frequency, but relies on particular messages which are required to be sent by crystal-operated host nodes only. During a calibration message the P82C150 determines the time between two recessive-to-dominant transitions which have to be 32 bit periods apart. The tolerance of this calibration process also affects the overall accuracy.
- Between two calibrations the stability of the SLIO´s clock frequency can basically be influenced by temperature and supply voltage changes. The effect of temperature changes however usually can be neglected because of the short time between two calibrations: even at the lowest bit rate of 20 kbit/s and a calibration occurring every 8000 bits the interval is as short as 400 ms.
- The main source for clock frequency changes therefore are supply voltage variations. In order to keep the overall clock accuracy within the required limits of 1.58%[6] the voltage must be kept constant[7]. This refers to short-term changes (from one calibration to the next one), the nominal voltage is not that critical in this case.

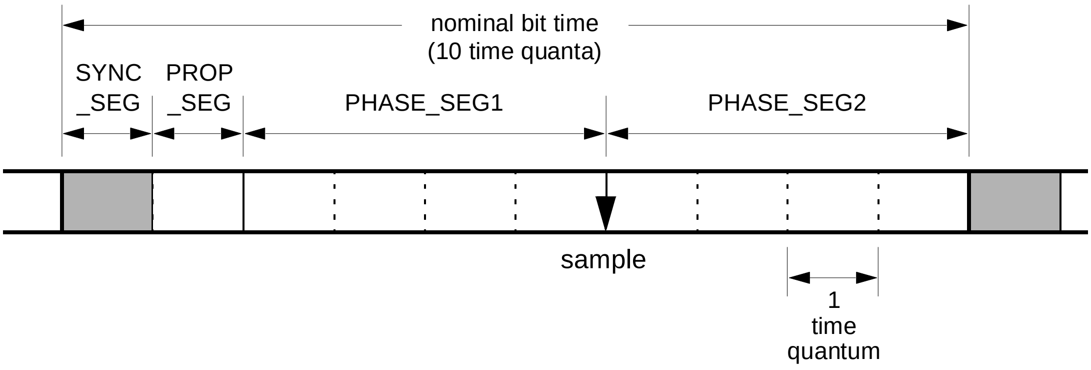


Fig. 22 Bit timing optimized for oscillator tolerance for P82C150

7.2 Bit Time Optimized for Tolerance

Since each node runs on its own clock, certain clock frequency differences are inevitable (even between crystal-operated nodes). Apart from the P82C150 and its calibration mechanism, a node generally is unable to detect a frequency offset, but can only notice the integral over time of this offset: a phase difference between its own

[6] More on oscillator tolerance including a derivation of this value can be found in the CAN specification [2].

[7] For exact data on the required stability of the supply voltage refer to the data sheet of the P82C150 [1].

internal bit clock and the bit rate on the bus. In order to realign its clock phase to the bus the CAN specification offers the process of resynchronization. A node can shorten or lengthen a bit time by up to four time quanta to get its synchronization segment (the first segment of a bit) back in phase with the transition on the bus. With a bit stuffing length of 5 a recessive-to-dominant transition on the bus can be expected at least every 10 bits during normal data transmission. Between crystal-operated nodes the amount of phase difference that can accumulate during these 10 bits is so small that a resynchronization jump width (SJW) setting of 1 is sufficient for these nodes.

If a SLIO is used in the network, the accumulated phase difference between two transitions can be much larger. A SLIO has set its SJW to 4. Moreover the bit time is divided into 10 time quanta only, so SJW amounts to 40% of the bit time. This results in a bit timing with the highest possible oscillator tolerance for CAN. Fig. 22 shows the bit timing as it is implemented in the P82C150.

7.3 Bit Timing of different Nodes in a Network

According to the CAN specification [2] nodes in a CAN network may run on clocks having up to 1.58% of frequency deviation. This permits devices like the P82C150 which run a non-crystal-operated clock to take part in the communication in the network.

If the SLIO P82C150 is used all other nodes in the network should have the same bit timing as the P82C150 in order to maintain a safe link. When programming a node the difference between the protocol definition used so far and the implementation definition of the bit timing should be noticed (see Fig. 23). The length of a bit (NBT, Nominal Bit Time) is determined by programming TSEG1 and TSEG2 (see Fig. 23 for definitions):

NBT = TSEG1 + TSEG2 + 1

The time when a node samples the bus is at the beginning of TSEG2. Therefore the length of TSEG2 determines the position of the sample point.

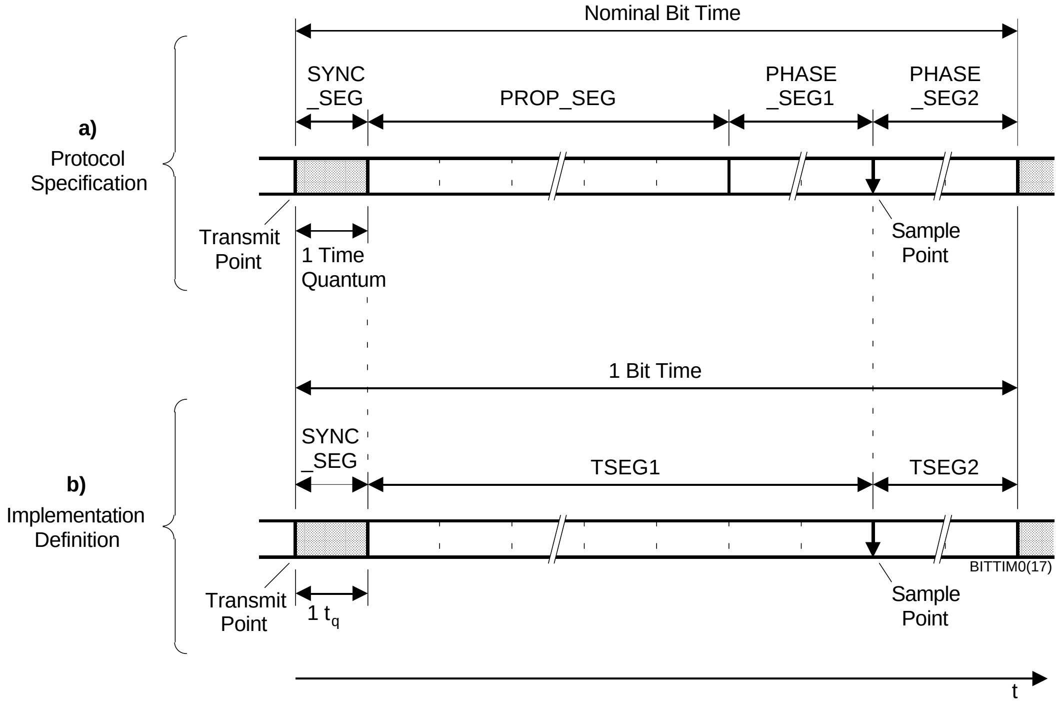


Fig. 23 Structure of a CAN bit time

In programming a node for the communication with a SLIO the bit time parameters should be set according to Table 6:

Table 6 Bit time parameters for communication with SLIOs P82C150

Parameter	**value**
bit rate	between 20 kbit/s and 125 kbit/s, set baud rate prescaler BRP appropriately
synchronization jump width (SJW)	4 time quanta
sampling	0, the bus is sampled once
time segment 1 (TSEG1)	5 time quanta
time segment 2 (TSEG2)	4 time quanta

	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Bus Timing Register 0	SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0
for SLIO communication:	*1*	*1*	*	*	*	*	*	*
Bus Timing Register 1	SAM	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0
for SLIO communication:	*0*	*0*	*1*	*1*	*0*	*1*	*0*	*0*

* *... value depends on oscillator frequency and desired bit rate*

BRP	Baud Rate Prescaler	t_q = 2 t_{CLK} (32 BRP.5 + 16 BRP.4 + 8 BRP.3 + 4 BRP.2 + 2 BRP.1 + BRP.0 + 1) (${}_{tCLK}$ = time period of the oscillator of the CAN controller)
SJW	Synchronization Jump Width	t_{SJW} = t_q (2 SJW.1 + SJW.0 + 1)
SAM	Sampling	SAM = 1: 3 samples are taken SAM = 0: the bus is sampled once
TSEG1	Time Segment 1	t_{TSEG1} = t_q (8 TSEG1.3 + 4 TSEG1.2 + 2 TSEG1.1 + TSEG1.0 + 1)
TSEG2	Time Segment 2	t_{TSEG2} = t_q (4 TSEG2.2 + 2 TSEG2.1 + TSEG2.0 + 1)

Fig. 24 Bus timing register layout of Philips CAN controllers

Fig. 24 shows the layout of the bus timing registers 0 and 1 of the Philips CAN controllers and gives the rules of how to program the individual bits.

Generally, nodes can have different bit timings. If all nodes are crystal-operated the only requirement for establishing a communication is, that the bit rate is the same throughout the network. Differences in TSEG2 and thus the position of the sample point only influence the maximum achievable bus length. If however a P82C150 is present in the network the synchronization jump width must be set to 40% of the bit time (4 out of 10 time quanta). Since the maximum value for SJW is 4 this requirement can only be met if the total number of time quanta in a bit is 10 (see Fig. 25).

Fig. 25 also shows that the requirement for compensating a maximum of oscillator tolerance is contrary to the requirement of achieving a maximum of bus length. Moreover the relative time left for propagation delay (bus length) increases with the number of time quanta per bit and with the sample point being positioned closer to the end of the bit time.

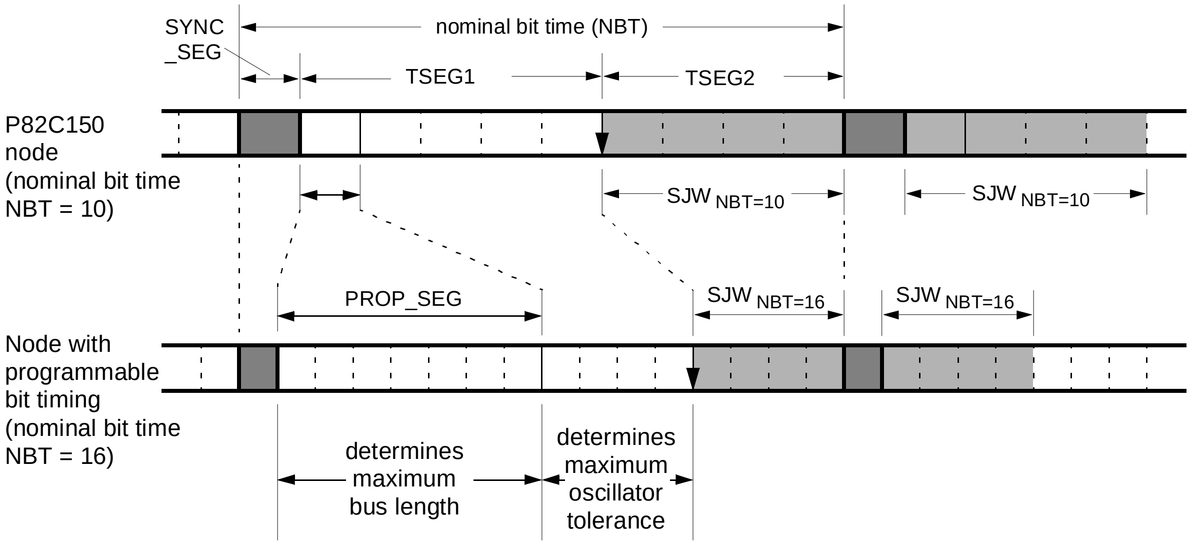


Fig. 25 SJW comparison

7.4 Bus Length with P82C150 Nodes

Due to the fact that the bit timing of P82C150 nodes is optimized for compensating oscillator tolerances only one time quantum is left for propagation delay compensation (PROP_SEG), see Fig. 25. The first time quantum (SYNC_SEG) of the bit time is reserved, because the accuracy of the resynchronization is always limited to one time quantum. The last eight time quanta (PHASE_SEG1 and PHASE_SEG2) are needed to cover the phase error that can have accumulated after a number of bits without resynchronization.

Although the bit timing is fixed and only one time quantum is intended for propagation delay compensation, the maximum achievable bus length can be increased if measures are taken to reduce the possible oscillator tolerance. This means that the resynchronization jump width SJW of 4 is not utilized fully which leaves more room for delay compensation. As was pointed out at the beginning of this chapter, clock frequency variations of the P82C150 are mainly due to changes in supply voltage. Therefore implementing a more stable power supply can increase the usable bus length.

The propagation delay t_{PROP} is the sum of all delays that a signal encounters when travelling from one node to another *and back* (see Fig. 26):

$$t_{PROP} = t_{CON,A} + t_{TRANS,A} + t_{TRANS,B} + t_{CON,B} + 2\ t_{BUS}$$

The abbreviations have the following meaning:

$t_{CON,A}$, $t_{CON,B}$	sum of input and output delay of CAN controller A, B
$t_{TRANS,A}$, $t_{TRANS,B}$	sum of transmit and receive delay of transceiver A, B
t_{BUS}	delay of bus cable (one way)

The allowed bus delay thus is

$$t_{BUS} = (t_{PROP} - t_{CON,A} - t_{CON,B} - t_{TRANS,A} - t_{TRANS,B}) / 2$$

or, if the same transceivers are used,

$$t_{BUS} = (t_{PROP} - t_{CON,A} - t_{CON,B}) / 2 - t_{TRANS}$$

With t_p [ns/m] being the specific line delay of the cable,

the maximum achievable bus length L is found by the formula

$$L = \frac{t_{BUS}}{t_p}$$

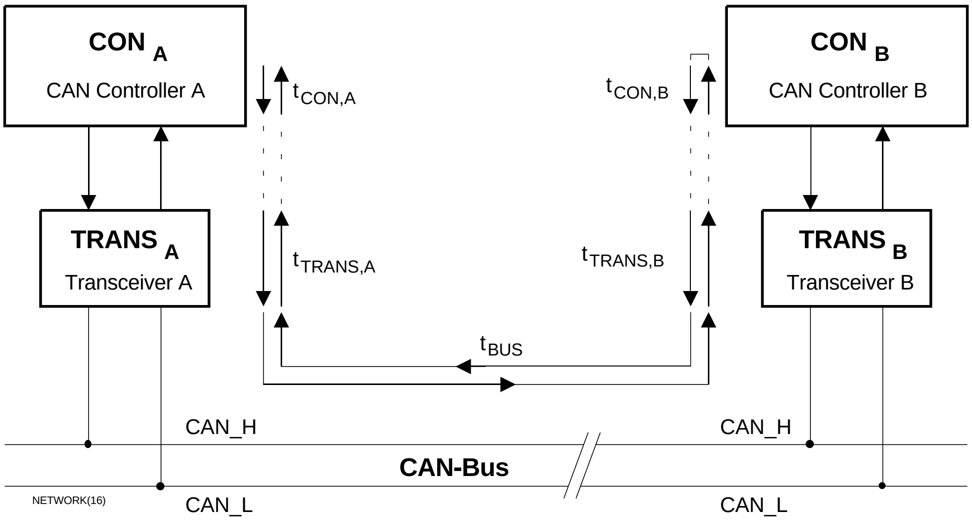


Fig. 26 Propagation delay in a CAN network

Sample bus length calculation

In the succeeding example the maximum bus length for a CAN network containing SLIOs is calculated. The calculation assumes the following maximum delay figures:

$t_{CON,A} = t_{HOST} = 60$ ns

$t_{CON,B} = t_{SLIO} = 60$ ns[8]

$t_{TRANS} = 170$ ns

$t_p = 5$ ns/m

At a bit rate of 100 kbit/s the length of the propagation segment is

$t_{PROP} = 1000$ ns

This gives a maximum bus length of

$$L = \frac{(1000 - 60 - 60)\ \text{ns} / 2 - 170\ \text{ns}}{5\ \text{ns/m}}$$

L = 54 m

[8] Refer to the data sheet of the P82C150 for exact delay data [1]

Optocouplers

Optocouplers can be used with SLIO nodes. For example they can be placed between the protocol controller and the transceiver (see Fig. 11 on page 17). They are used in pairs: the transmitted and the received signal both have to pass an optocoupler. Due to their extra delay these devices directly reduce the maximum achievable bus length.

If the delay of a *single* optocoupler is t_{OPTO}, using *one pair* of them in a node reduces the bus length by

$$L_R = \frac{t_{OPTO}}{t_p}$$

With a specific cable delay of t_p = 5 ns/m the achievable bus length is reduced by 1 m for every 5 ns of optocoupler delay, for example.

Instead of placing optocouplers between the transceiver and the controller, a galvanic isolation may also be done by placing isolating elements between the power supply of the application or the application inputs and the CAN controller. This has the advantage, that the achievable bus length is not reduced.

8. DIGITAL PORT FUNCTIONS

The P82C150 provides 16 digital I/O port pins. By means of several 16-bit-registers each pin can be individually configured, see Table 7.

Table 7 I/O Register map (digital outputs)

Register	Address	MSB	...	LSB
Data input register	0	DI15	...	DI0
Positive edge register	1	PE15	...	PE0
Negative edge register	2	NE15	...	NE0
Data output register	3	DO15	...	DO0
Output enable register	4	OE15	...	OE0

The data input register contains the logic levels of the port pins. Reading it by CAN bus is done by sending a remote frame or a data message addressing the data input register to the P82C150 which is answered by a data frame containing the contents of the data input register (see also Fig. 19 on page 27). If any bit of the positive or negative edge register is set, a change on any one of these selected pins will automatically send a data frame with the contents of the data input register. Data intended to be output is loaded into the data output register. However, it is output only on those pins whose corresponding bits are set to ‘1‘ in the output enable register.

Several port pins have additional functions: they can also be used as analog I/O pins or they monitor the output of the built-in comparators, see Fig. 27. If these ports are used as digital I/O the additional functions must be disabled: the bits M3 ... M1 and SW3 ... SW1 of register 5 (analog configuration) and all bits of registers 6 and 7 (DPM1 and DPM2) must be set to zero, see Table 8.

Table 8 I/O Register map (analog outputs)

		MSB															LSB
Register	**Address**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Analog config.	5	ADC	OC3	OC2	OC1	0	M3	M2	M1	SW3	SW2	SW1	0	0	0	0	0
DPM1	6	DP9	DP8	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0	0	0	0	0	0	0
DPM2	7	DQ9	DQ8	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0	0	0	0	0	0	0
A/D	8	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

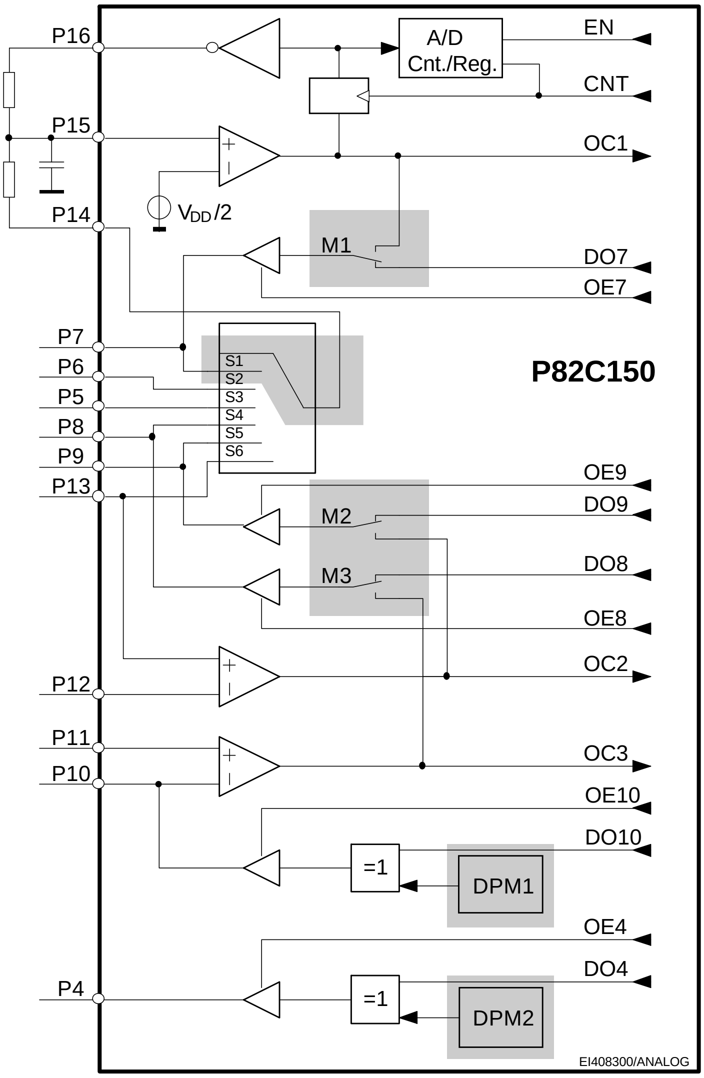


Fig. 27 Additional functions to be disabled for digital I/O

9. ANALOG PORT FUNCTIONS

Besides functioning as digital I/O ports (chapter 8) the port pins may be used for several analog functions.

- Input comparators for analog signals (differential or single ended).
  Application examples:
  - Minimum or maximum supervision of analog signals [1].
  - Window comparator for an analog signal [1].
- Digital-to-Analog converters using the algorithm of the distributed pulse modulation (DPM).
- Analog-to-Digital converter.
  One separate input comparator is available for the A/D conversion task. An implemented multiplexer allows for the extension to a maximum of 6 multiplexed A/D-input channels.

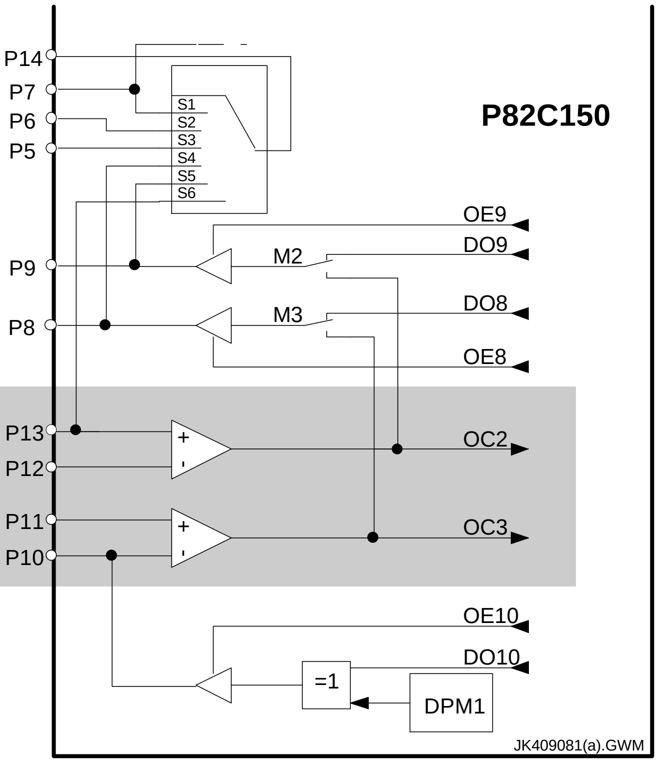


Fig. 28 Input comparators at ports P10/P11 and P12/P13

Table 9 I/O register content if both input comparators are used

		I/O Register Content															
I/O Register	**Addr.**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Output enable		OE(n)															
	4	X	X	0	0	0	0	X	X	X	X	X	X	X	X	X	X

(X = not relevant for this configuration)

9.1 Comparators

The P82C150 provides two general purpose comparators (see Fig. 28). The inputs are provided at the ports P10/P11 and P12/P13. Table 9 shows the setting of the I/O register control bits, if the comparators are used in an application. The outputs of the ports P10 through P13 must be disabled, which means that P10 can not serve as DPM1 output at the same time either.

For the electrical characteristics of the input comparators see the data sheet [1].

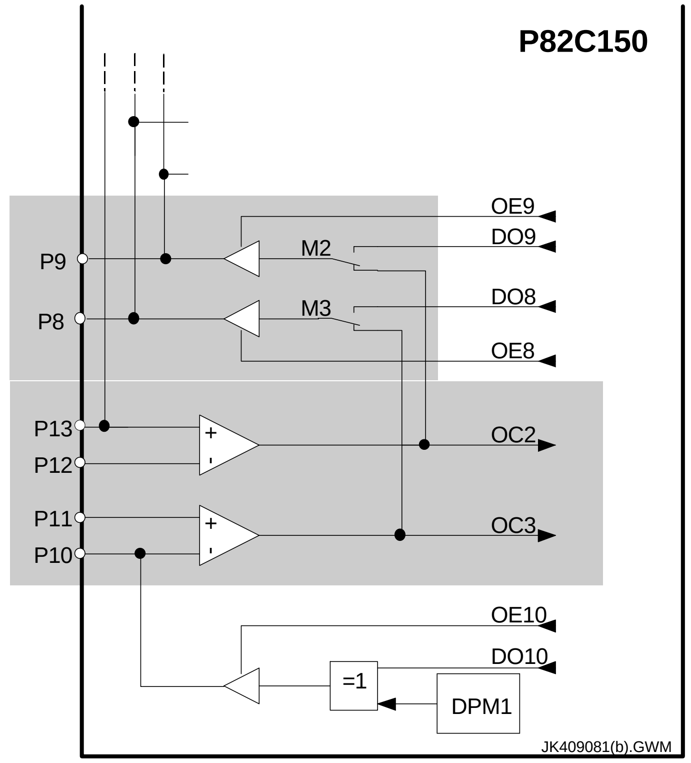


Fig. 29 Comparator outputs connected to ports P8 and P9

Table 10 I/O register content for connecting the comparator outputs to ports P8 and P9

		I/O Register Content															
I/O Register	**Addr.**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Output enable	4	OE(n)															
		X	X	0	0	0	0	1	1	X	X	X	X	X	X	X	X
Analog configuration	5	ADC	OC3	OC2	OC1		M3	M2	M1	SW3	SW2	SW1					
		X	X	X	X	0	1	1	0	X	X	X	0	0	0	0	0
Positive edge	1	PE(n)															
		X	X	X	X	X	X	0/1	0/1	X	X	X	X	X	X	X	X
Negative edge	2	NE(n)															
		X	X	X	X	X	X	0/1	0/1	X	X	X	X	X	X	X	X

(X = not relevant for this configuration, 0/1 = depends on application)

The comparator output is switched to HIGH, if input ‘plus‘ > input ‘minus‘. In sleep mode the output is forced HIGH regardless of the input levels. The logical result of the comparison is placed into the analog configuration register (bits OC3 and OC2) and may be read by addressing this register in a data frame on the CAN-bus.

The logical result of the comparison may be switched to an output port (P8 for OC3 and P9 for OC2) for being used in the connected application (Fig. 29). Table 10 shows the setting of the I/O register control bits for this case (M2 and M3 in the analog configuration register is set to ‘1‘). Furthermore this opens the possibility of reading the comparator output values via the digital input port (data input register) and being able to use the event capture facility of the ports for an automatic transmission of the input data via the CAN-bus, if the PE8/NE8 and PE9/NE9 are set properly in the I/O registers 1 (positive edge trigger) and 2 (negative edge trigger).

9.2 Distributed Pulse Modulation (DPM) Outputs

9.2.1 General Description of the DPM

The P82C150 provides two output channels (DPM1 at port P10 and DPM2 at port P4, Fig. 32) for converting a 10 bit digital value into a quasi-analog output signal - ‘Distributed Pulse Modulation‘. The function of the modulator may be explained with the help of Fig. 30, which shows a simplified functional block diagram of a DPM-circuit and a PWM-circuit with a resolution of 3 bit. The value to be converted (value ‘B‘) is compared with the output (PWM) or modified output (DPM) (value ‘A‘) of a continuous running counter. If the value ‘A‘ is smaller than the value ‘B‘, the pulse output is switched HIGH else it stays LOW. The resulting output pulses are given in Fig. 31. If the value to be converted (‘B‘) is increased continuously, the PWM generates a pulse with a growing mark to space ratio, whereas the DPM generates more equally spaced pulses. Further information may be found in [7].

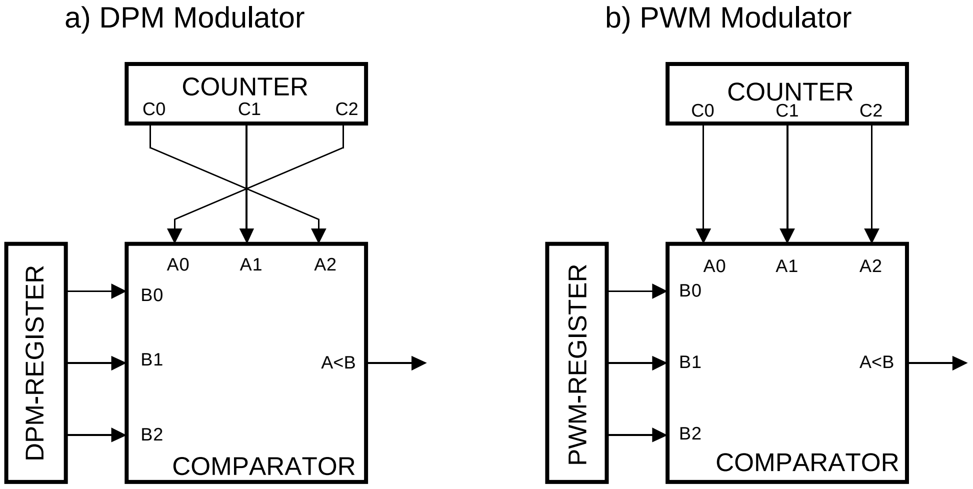


Fig. 30 Functional block diagrams of DPM- and PWM-modulators (3 bit resolution)

The repetition time of the pulse stream (one DPM cycle) is defined by the resolution of the modulator and the width of a single pulse, which again is related to the clock frequency of the counter. The DPM of the P82C150 has a resolution of 10 bit and the width of a single pulse is equal to four cycles of the internal oscillator clock. Thus the repetition time may be calculated from the formula:

$$T_{DPM} = 1024 \times 4\ T_{CLK}$$

Refer to [1] for more information about the electrical characteristics.

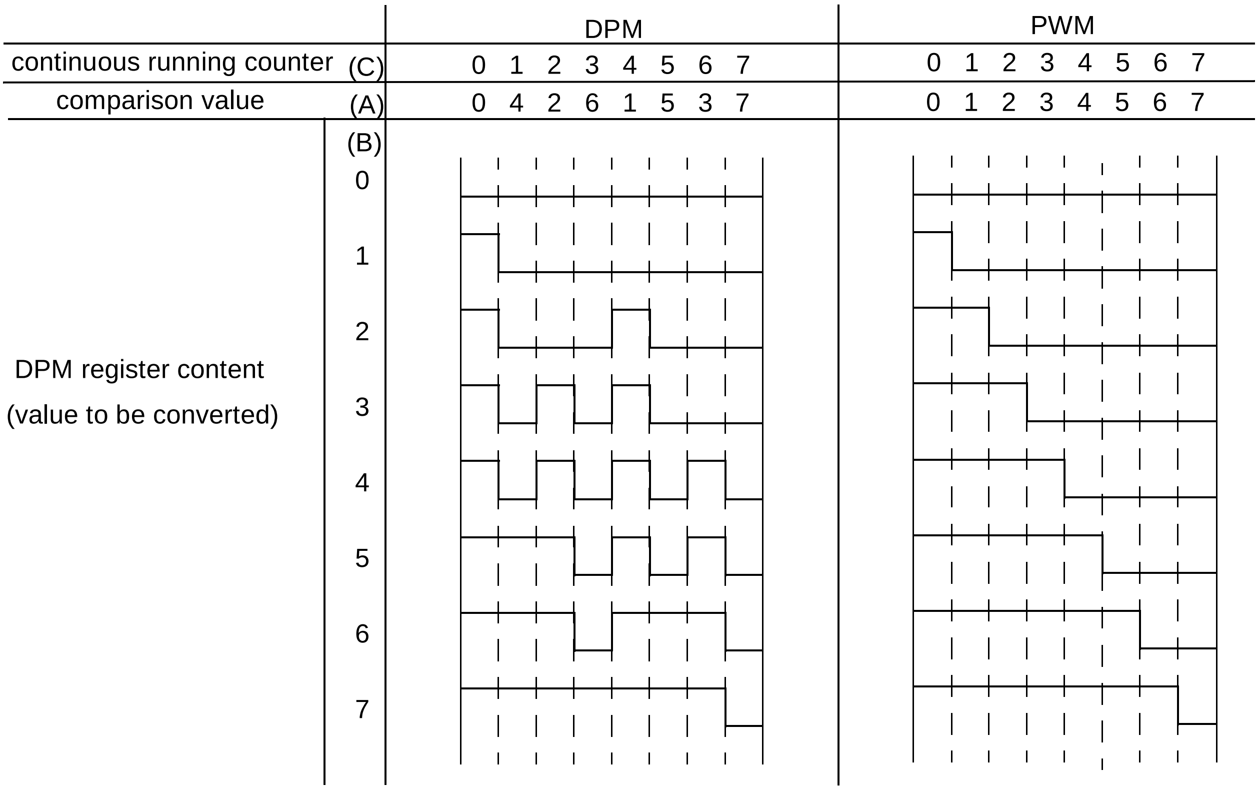


Fig. 31 DPM- and PWM- output signals (3 bit resolution)

In order to use the quasi-analog output signal of the DPM in an application the I/O control registers have to be set according to the information given in Table 11. The output enable bits have to be set for port P4 and P10.

The pulse trains at the outputs P4 and P10, which are generated from the latched DPM value, are inverted, if the corresponding bits in the data output register (DO4 and DO10) are set to ‘1‘, as may be seen from the logic diagram given in Fig. 32.

If the DPM1 output is used, the general comparator (OC3), which has its input ‘minus‘ connected to P10, can not be used.

9.2.2 Digital to Analog Conversion using the DPM

The simplest way to generate an analog voltage from the quasi-analog signal is to apply an external low pass filter at the DPM output. One possible implementation would be a RC-filter of first order as indicated in Fig. 32.

Regarding the selection of the time constant (edge frequency) of this filter, a trade-off between minimizing the ripple voltage for maximum accuracy and minimizing the settling time has to be considered. The following examples are a guideline how to select the components.

Example 1:

- 10 bit accuracy; settling time <30 ms; a ripple voltage <5 mV_{pp} (<0,1%, which means less than the resolution) may be achieved by $\tau > 2 \times$ period of the pulse stream [7]:
- $\tau = R \times C \geq 4$ ms (recommended)
  - R = 1 kΩ; C = 4,7 µF or R = 10 kΩ, C = 470 nF

Example 2:

- 8 bit accuracy; settling time = 6 ms; a ripple voltage <20 mV_{pp} (<0,4%, which means less than the resolution):
- $\tau = R \times C \geq 1$ ms (recommended)
    - R = 1 kΩ; C = 1 µF or R = 10 kΩ, C = 100 nF

NOTE 1: If the output is loaded by a resistive load, the accuracy will be decreased due to the voltage drop across the series resistor. In these cases a low value for the series resistor should be chosen. However, it is recommended to provide a minimum series resistor of 1 kΩ for protection.

NOTE 2: The external RC-filter should be placed close to the corresponding DPM output pin in order to minimize the radiated emission of this output.

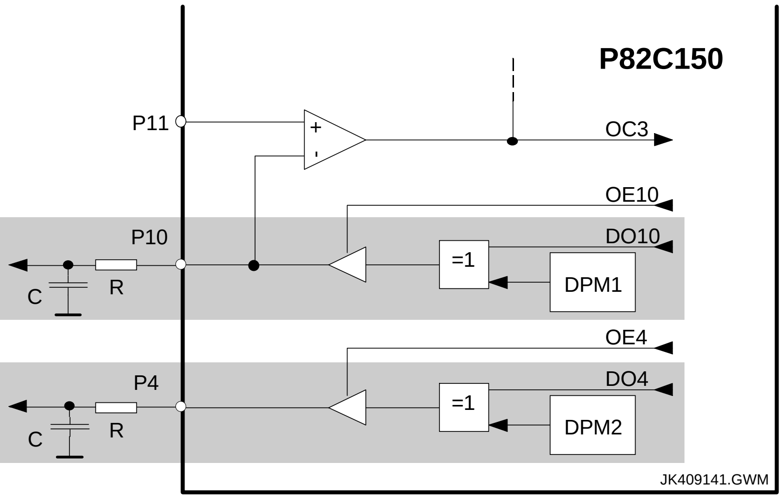


Fig. 32 Digital-to-Analog converter outputs (DPM1 and DPM2)

Table 11   I/O register content if using the D/A-converter outputs at ports P4 and P10

		I/O Register Content															
I/O Register	**Addr.**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Output enable		OE(n)															
	4	X	X	X	X	X	1	X	X	X	X	X	1	X	X	X	X
Analog configuration	5	ADC X	OC3 X	OC2 X	OC1 X	0	M3 X	M2 X	M1 X	SW3 X	SW2 X	SW1 X	0	0	0	0	0
DPM1 or DPM2	6 or 7	DP.. or DQ.. (= current digital value)															
		Dx9	Dx8	Dx7	Dx6	Dx5	Dx4	Dx3	Dx2	Dx1	Dx0	0	0	0	0	0	0
Data output	3	DO(n)															
		X	X	X	X	X	0/1	X	X	X	X	X	0/1	X	X	X	X

(X = not relevant for this configuration
DO(n) = 0: DPM output is not inverted, DO(n) = 1: DPM output is inverted)

9.2.3 Generation of Clock Pulses using the DPM-Outputs

The DPM-outputs may also be used as clock outputs in systems, where the connected application needs an external clock e.g. for driving a display. Not all digital values of the DPM-register content are qualified for such an application as not all generated pulse streams do have equally spaced pulses. Table 12 gives suitable values together with the generated frequencies. The values in the right part of the table may also be achieved by inverting the DPM-output by setting the corresponding data output bits in the control register (address 3).

Table 12 Generated clock frequencies for different DPM values

	Clock			Clock	
DPM-value	**Frequency**	**mark-to-space ratio**	**DPM-value**	**Frequency**	**mark-to-space ratio**
0000 0000 01	f_{CLK} / 4096	1 : 1023	1111 1111 11	f_{CLK} / 4096	1023 : 1
0000 0000 10	f_{CLK} / 2048	1 : 511	1111 1111 10	f_{CLK} / 2048	511 : 1
0000 0001 00	f_{CLK} / 1024	1 : 255	1111 1111 00	f_{CLK} / 1024	255 : 1
0000 0010 00	f_{CLK} / 512	1 : 127	1111 1110 00	f_{CLK} / 512	127 : 1
0000 0100 00	f_{CLK} / 256	1 : 63	1111 1100 00	f_{CLK} / 256	63 : 1
0000 1000 00	f_{CLK} / 128	1 : 31	1111 1000 00	f_{CLK} / 128	31 : 1
0001 0000 00	f_{CLK} / 64	1 : 15	1111 0000 00	f_{CLK} / 64	15 : 1
0010 0000 00	f_{CLK} / 32	1 : 7	1110 0000 00	f_{CLK} / 32	7 : 1
0100 0000 00	f_{CLK} / 16	1 : 3	1100 0000 00	f_{CLK} / 16	3 : 1
1000 0000 00	f_{CLK} / 8	1 :1			

f_{CLK} = system clock frequency [1]

9.3 Analog-to-Digital Converter

9.3.1 General Description of the ADC

The P82C150 provides one Analog-to-Digital converter on-chip (‘bit-stream‘ or ‘sigma-delta‘ conversion).

A ‘bit-stream‘ converter (Fig. 33) includes a feedback loop consisting of

- an adder, which subtracts the output signal from the input signal to determine the approximation error
- a loop filter (e.g of 1st order, integrator), which extracts the low frequency content of the approximation error
- a 1-bit quantizer (comparator) and a sampling flipflop, which stores the comparator output for one period of the sampling clock
- a 1-bit Digital-to-Analog converter (e.g. an inverter), which converts the 1-bit data stream into a quasi analog signal to be compared with the analog input signal.

This type of converter produces a 1-bit code, which is converted to a n-bit code by the digital decimation filter.

As indicated in Fig. 33, the comparator and the 1-bit DAC of the ADC of the P82C150 are on-chip and the adder and the integrator are to be connected externally (see also Fig. 34). The analog input signal is fed via the adder and integrator (external RC-combination) to the input of the comparator at port 15. The output of the comparator is clocked into a flipflop (CNT = f_{CLK}/4 with f_{CLK} = frequency of the internal oscillator) and fed back via an inverter (1-bit DAC and inversion for the subtraction) to the output port P16, which is connected to the adder.

The behaviour of the feedback loop may be explained with an example:

The start-up situation may be: A small analog input signal is applied (near to 0 V).

Assuming that a voltage above the upper switch-over voltage of the comparator is present at port P15, then the output at P16 is set to a voltage near to 0 V (Fig. 35). Now both the input and the feedback signal are discharging

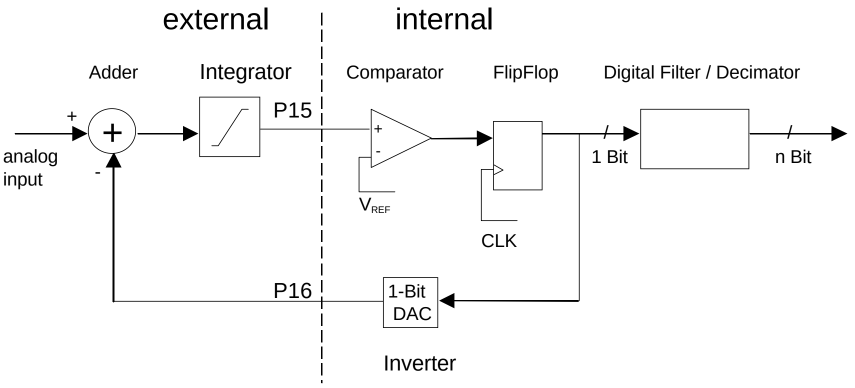


Fig. 33 Principle of ‘bit-stream‘ conversion

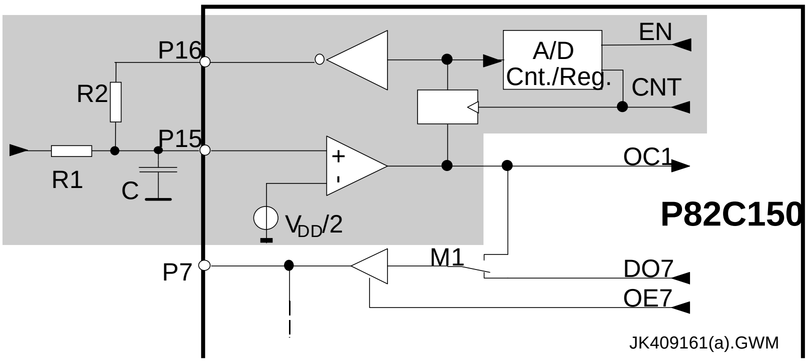


Fig. 34 Analog-to-Digital converter of P82C150

the capacitor. As long as the voltage at P15 stays above the lower switch-over voltage of the comparator, a ‘1‘ is clocked into the flipflop, holding P16 ‘LOW‘. If the lower switch-over voltage is reached, the comparator output switches to ‘LOW‘. With the next active edge of the sampling clock the flipflop will store a ‘0‘ and the output P16 will be switched to ‘HIGH‘ (a voltage near to V_{DD}). Now the capacitor will be loaded and the voltage at P15 increases again. A ‘0‘ will be stored as long as the voltage stays below the upper switch-over voltage of the comparator. When the upper switch-over voltage is reached, the comparator output is set to ‘HIGH‘ and the port output P16 is switched to 0 V again. Here the cycle starts again.

This example clarifies that the voltage at the output P16 has to compensate fully the low voltage introduced by the input signal in order to being able to increase the voltage at port P15. The same applies for a high input voltage, but here the voltage at the output of P16 has to be low enough for the compensation.

The generated bit stream is fed to a counter, which counts the ‘HIGH‘ states of the flipflop during a period of 1024 clocks, which corresponds to a duration of 1024×4 t_{CLK} (t_{CLK} = repetition time of the internal oscillator). The content of the counter/register corresponds to the achieved digital value.

A conversion is started either by setting the bit ‘ADC‘ in the ‘analog configuration register‘ to ‘1‘ or by reading the ‘A/D register‘. Upon receiving a conversion request, the P82C150 is waiting for the current cycle of 1024 clocks to be finished. Before enabling the counter another cycle of 1024 clocks is waited for to allow for settling of the analog input voltage. This is important, if the analog voltage has been switched through to the input of the ADC via the multiplexer by setting the switches SW1...SW3 together with the conversion request bit (ADC).

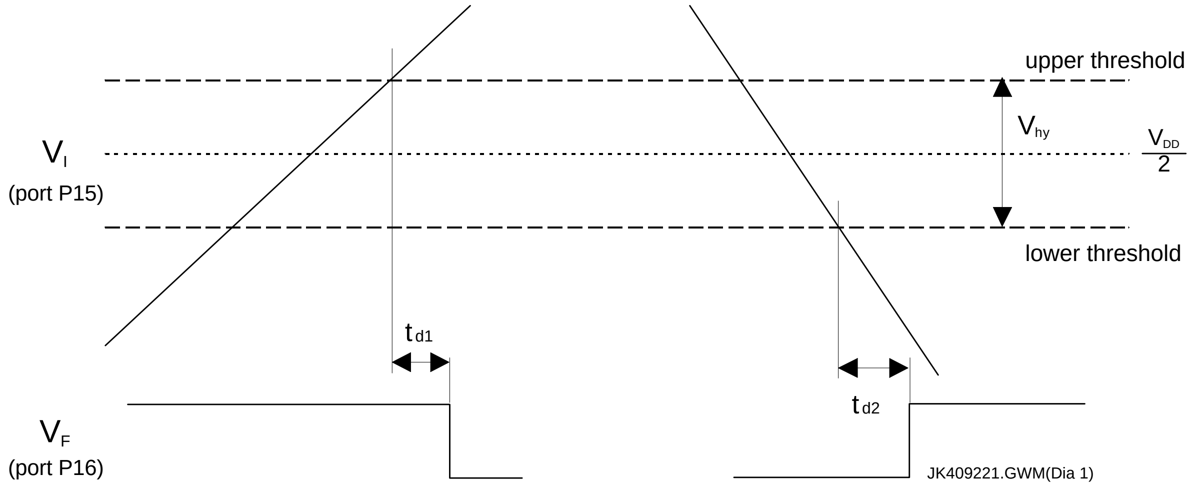
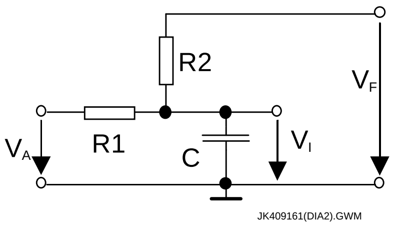


Fig. 35 ADC: switch-over points of the input comparator

9.3.2 The external Circuit of the ADC

The external circuit is shown in Fig. 34 together with the internal circuit closing the loop for the A-to-D conversion process. Whereas Fig. 36 shows the external circuit only, together with charging and discharging equations for the capacitor if the loop is opened.



Charging the capacitor (V_F switches from V_{FOL} to V_{FOH}):

$$v_I(t) = \frac{R1 \times R2}{R1 + R2}\left(\frac{V_{FOH}}{R2} + \frac{V_A}{R1} - \frac{1}{R2}(V_{FOH} - V_{FOL})e^{-t/\tau}\right)$$

Discharging the capacitor (V_F switches from V_{FOH} to V_{FOL}):

$$v_I(t) = \frac{R1 \times R2}{R1 + R2}\left(\frac{V_{FOL}}{R2} + \frac{V_A}{R1} + \frac{1}{R2}(V_{FOH} - V_{FOL})e^{-t/\tau}\right)$$

V_A :	analog input voltage
$V_I, v_I(t)$:	input voltage at port P15
V_F, V_{FOL}, V_{FOH} :	output voltage at port P16 (LOW, HIGH)

$$\tau = \frac{R1 \times R2}{R1 + R2} \times C$$

Fig. 36 Open loop equations for the ADC input voltage

From the equations given in Fig. 36 V_{Imin} and V_{Imax} are calculated for $t \rightarrow \infty$:

$$V_{Imin} = \frac{R1 \times R2}{R1 + R2}\left(\frac{V_{FOL}}{R2} + \frac{V_{Amin}}{R1}\right) \quad \text{and} \quad V_{Imax} = \frac{R1 \times R2}{R1 + R2}\left(\frac{V_{FOH}}{R2} + \frac{V_{Amax}}{R1}\right) \qquad (1)$$

V_{Imin} is achieved with V_{Amin} and V_{Imax} is achieved with V_{Amax}.

The range of the input voltage V_A (analog voltage to be converted) is restricted due to two conditions:

1. The voltage level at the input of the ADC (port P15) must not exceed the 'DC input voltage on any pin' of the 'limiting values' given in the data sheet [1]. This leads to the following two relations:

   $V_{Imin} \geq 0V$ and $V_{Imax} \leq V_{DD}$

   During operation the voltage at the input of port P15 is controlled by the feedback loop to approximately $V_{DD}/2$ i.e. it will not exceed the limiting values. But if the loop is opened — as it happens during sleep mode, where the internal oscillator is stopped — the worst case value for V_I has to be considered. The output voltage at port P16 (the feedback voltage V_F) is either ~0 V (V_{FOL} L) or ~V_{DD} (V_{FOH}) dependent on the last cycle of the ADC-operation before the oscillator did stop.

   With the above range for $V_{I,}$ V_{FOL} and V_{FOH}
   the input range for V_A is calculated from equation (1): $0V < V_A < V_{DD}$ (2)

   The input range for the voltage V_A is independent of the value of the resistors.

2. For a proper operation of the ADC, the voltage V_I at port P15 must at least reach the upper and lower threshold voltage (see Fig. 35) during the charging and discharging cycles. This condition leads to the following relations:

   Under the worst case condition for **charging** the capacitor ($V_F = V_{FOH}$ and $V_A = V_{Amin}$) the relation

   $V_{Iend}(t \to \infty) > \frac{V_{DD} + V_{hy}}{2}$ must be fulfilled (the output port P16 is set to 'LOW' again: $V_F = V_{FOL}$).

   Under the worst case condition for **discharging** the capacitor ($V_F = V_{FOL}$ and $V_A = V_{Amax}$) the relation

   $V_{Iend}(t \to \infty) < \frac{V_{DD} - V_{hy}}{2}$ must be fulfilled (the output port P16 is set to 'HIGH' again: $V_F = V_{FOH}$).

   (V_{hy} = hysteresis of the switch-over point of the input comparator, see Fig. 35).

   With the above relations for V_I and the mentioned worst case conditions the following relation for V_A is received from the equations given in Fig. 36 on page 46:

$$\left(\frac{R1 + R2}{2 \times R2}(V_{DD} + V_{hy}) - \frac{R1}{R2}V_{FOH}\right) < V_A < \left(\frac{R1 + R2}{2 \times R2}(V_{DD} - V_{hy}) - \frac{R1}{R2}V_{FOL}\right) \quad (3)$$

With the ratio R1/R2 = 1 a conversion is achieved, which best transforms the voltage range of V_A into the whole range of achievable digital values, where 1 digital step corresponds to ΔV_A ~5 mV.

With $\frac{R1}{R2} = 1$ and $V_{FOH} = V_{DD} - \Delta V_{FOH}$ and $V_{FOL} = \Delta V_{FOL}$

the relation (3) is reduced to: $(V_{hy} + \Delta V_{FOH}) < V_A < (V_{DD} - (V_{hy} + \Delta V_{FOL}))$

which is even more restrictive than the relation (2). More information on the characteristics of the input port P15 if used for the ADC and of the output port P16 is given in the data sheet [1].

Ratios >1 (R1 > R2) would allow for a wider range of the analog input voltage V_A, but according to the previous mentioned restriction (2) only the range between ~0 V and ~V_{DD} can be converted. So the analog voltage has to be transformed into the allowed range of V_A before it may be applied to the external circuit of the ADC.

Ratios <1 (R1 < R2) result in a reduced voltage range of V_A according to the relation (3).

Example 1

If a ratio of R1/R2 = 1/2 is chosen the achievable range for V_A is calculated from (3) to

$$\frac{V_{DD} - V_{hy} - 2\Delta V_{FOH}}{4} + V_{hy} + \Delta V_{FOH} < V_A < (V_{DD} - (V_{hy} + \Delta V_{FOL})) - \frac{V_{DD} - V_{hy} - 2\Delta V_{FOL}}{4}$$

Compared with the result for R1/R2 = 1 the voltage range for V_A is reduced with ~V_{DD}/4 at both ends.

The resulting range for V_I, which determines the digital representation of the analog signal is calculated from (1) to:

$$V_{Imin} = \frac{V_{DD} + 3V_{hy} + 2\Delta V_{FOH} + 2\Delta V_{FOL}}{6} \quad \text{and} \quad V_{Imax} = V_{DD} - \frac{V_{DD} + 3V_{hy} + 2\Delta V_{FOH} + 2\Delta V_{FOL}}{6}$$

As it is seen from this example, there exists a possibility of transforming a small range of V_A (<V_{DD}) into a wider range of V_I, but it is still smaller than V_{DD}.

Example 2

Given is a range for the input voltage of $1{,}5\ V < V_A < 3{,}5\ V$ which corresponds to

$$\frac{3}{10}V_{DD} < V_A < V_{DD} - \frac{3}{10}V_{DD}, \quad \text{if} \quad V_{DD} = 5V.$$

Which ratio (R1/R2) have to be selected transforming the range into the widest possible range for V_I?

The ratio is calculated using the relations for V_{Amin} or V_{Amax} given in equation (3).

The result is $$\frac{R1}{R2} = \frac{2V_{DD} + 5V_{hy}}{5(V_{DD} + V_{hy} + 2\Delta V_{FOH})} \approx \frac{2}{5} \quad \text{or} \quad \frac{R1}{R2} = \frac{2V_{DD} + 5V_{hy}}{5(V_{DD} - V_{hy} - 2\Delta V_{FOL})} \approx \frac{2}{5}$$

The resulting range for V_I, which determines the digital representation of the analog signal is calculated from (1) to:

$$V_{Imin} = \frac{3V_{DD} + 4\Delta V_{FOL}}{14} \approx 1,07V \quad \text{and} \quad V_{Imax} = V_{DD} - \frac{3V_{DD} + 4\Delta V_{FOH}}{14} \approx V_{DD} - 1,07V$$

Due to the equation (3) the small range of V_A cannot be transformed into the maximum range for V_I, which would make use of the whole range of possible digital values.

The value to be chosen for the resistors R1 and R2 depends on

- the leakage current of the input at port P15 (ADC comparator input) [1] and
- the ability of the analog source (V_A) to sink or source current.

It is up to the designer to decide upon this value for the resistors, as the trade-off between a low value for reducing the influence of the leakage current and a high value for reducing the load of the output of the analog source must be considered. The values given in the data sheet [1] are recommended as a first approach. If the analog multiplexer is used, the on-resistance of the analog switches has to be taken into account for the calculation of the external components.

NOTE: The components of the external circuit should be placed close to the corresponding pins they are connected to in order to avoid the introduction of noise.

9.3.3 Controlling the Operation of the ADC

If the ADC is used the digital output at port P15 must not be enabled. The settings of the I/O registers are shown in Table 13.

Table 13 I/O register content if the Analog-to-Digital converter is used

		I/O Register Content															
I/O Register	**Addr.**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Output enable		OE(n)															
	4	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Analog configuration		ADC	OC3	OC2	OC1		M3	M2	M1	SW3	SW2	SW1					
	5	X	X	X	X	0	X	X	0	X	X	X	0	0	0	0	0

(X = not relevant for this configuration)

In case a node application needs to convert more than one analog signal a 6:1-analog multiplexer is added on chip (Fig. 37). The analog inputs are provided at the ports P5 through P9 and P13, the output at P14. For the use together with the ADC port P14 has to be connected to R1 of the external circuit of the ADC. The input for an analog voltage is selected by setting the analog switch bits ‘SW1‘...‘SW3‘ of the ‘analog configuration‘ register appropriately. For the coding see the data sheet [1]. For each port used for an analog input signal the digital output driver must not be enabled. The setting of the I/O control registers is shown in Table 14.

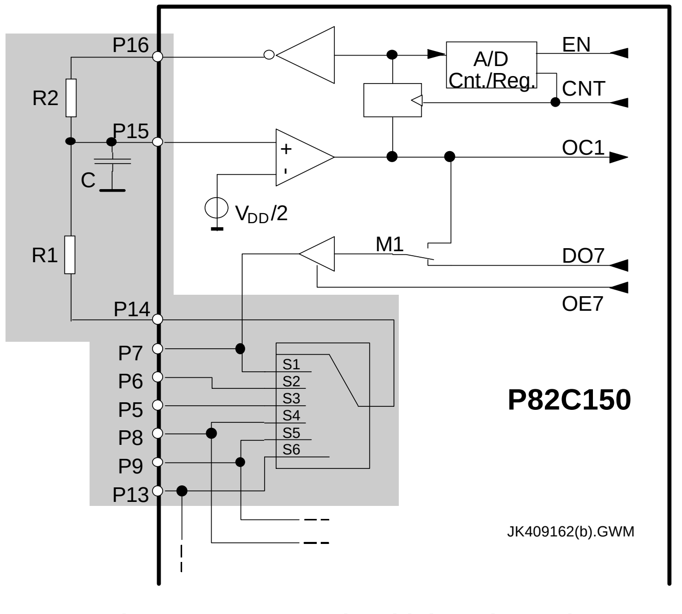


Fig. 37 ADC: Internal multiplexer is used

The output of the ADC comparator may be switched to the output of port P7 by setting the bit ‘M1‘ in the ‘analog configuration‘ register (see Fig. 37 and Table 15). The digital output of port P7 has to be enabled for this case and can thus not be used as analog input. Furthermore it is possible to automatically generate transmissions via the CAN-bus if the input of port P7 changes either from ‘LOW‘ to ‘HIGH‘ or from ‘HIGH‘ to ‘LOW‘ due to the comparator output level. In this case the bits for positive or negative edge triggering for this port have to be set in the respective registers.

Table 14 I/O register content if the Analog-to-Digital converter is used together with the multiplexer*

		I/O Register Content															
I/O Register	**Addr.**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Output enable		OE(n)															
	4	0	0	0	X	X	X	0	0	0	0	0	X	X	X	X	X
Analog configuration		ADC	OC3	OC2	OC1		M3	M2	M1	SW3	SW2	SW1					
	5	X	X	X	X	0	X	X	X	0/1	0/1	0/1	0	0	0	0	0

(X = not relevant for this configuration, 0/1 = depends on application)

* This multiplexer may also be used independent of the ADC.

Table 15 I/O register content for connecting the comparator output of the ADC to port P7

		I/O Register Content															
I/O Register	**Addr.**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Output enable		OE(n)															
	4	0	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X
Analog configuration		ADC	OC3	OC2	OC1		M3	M2	M1	SW3	SW2	SW1					
	5	X	X	X	X	0	X	X	1	X	X	X	0	0	0	0	0
Positive edge		PE(n)															
	1	X	X	X	X	X	X	X	X	0/1	X	X	X	X	X	X	X
Negative edge		NE(n)															
	2	X	X	X	X	X	X	X	X	0/1	X	X	X	X	X	X	X

(X = not relevant for this configuration, 0/1 = depends on application)

10. CONCLUSION

The P82C150 (SLIO, Serial Linked I/O device) may be used in a CAN-network as a low cost I/O controlling various functions of an application.

It may be prepared to take over several different tasks as

- switching digital signals,
- monitoring digital signals incl. edge trigger function,
- comparing analog signals with each other or with a reference,
- generating quasi-analog signals (DPM, distributed pulse modulated output signal) usable in Digital-to-Analog converters,
- converting an analog signal to a digital value (ADC).

With its ability to adapt its timing to the bitrate on the bus the P82C150 may be implemented easily in systems. The device gets all configuration information (except the message identifier) via the bus line. Nodes equipped with the P82C150 may be plugged or unplugged in a running system, if a proper network management is provided.

11. LIST OF REFERENCES

[1] P82C150 Data Sheet - Can serial linked I/O device (SLIO) with digital and analog port functions, Philips Semiconductors (latest version)

[2] CAN Bus Specification, Philips Semiconductors, 1994, 12NC: 9398 706 64011

[3] Data Sheet / Specification of the P82C250, Philips Semiconductors (latest version)

[4] Road vehicles - Interchange of digital information - Controller area network (CAN) for highspeed communication.
Document ISO 11898, International Standardization Organization 1993

[5] Road vehicles - Low-speed serial datacommunication.
Document ISO 11519-2, International Standardization Organization 1994

[6] OM4272 -SLIO Evaluation Board with the 82C150 and 82C250 CAN ICs, Philips Semiconductors
User Manual: SLIO P82C150 Demonstration Software.

[7] Michael Heiss, Wolfgang Dittrich: ‘Pulsanzahlmodulator als D/A-Umsetzer‘, Elektronik, H.19, 15.9.89

[8] Harald Eisele: ‘CAN Physical Layer Concepts for the P8x592 Microcontroller‘, Application Note, Philips Semiconductors PCALH, Report No.: HKI/AN 91 027, Date: 92-07-03

[9] PCF 1252-Family Data Sheet, Power Failure Detector and Reset Generator, Philips Semiconductors, March 1990

[10] Jens-Ulf Pehrs: ‘CAN Bus Failure Management Using the P8xC592 Microcontroller‘, Application Note, Philips Semiconductors PCALH, Report No.: HKI/AN 91 020, Date: 91-10-16