Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	16MHz
Connectivity	CANbus, EBI/EMI, UART/USART
Peripherals	DMA, POR, PWM, WDT
Number of I/O	48
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.18x24.18)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/p80c592ffa-00-512

CONTENTS

1 FEATURES
2 GENERAL DESCRIPTION
3 ORDERING INFORMATION
4 BLOCK DIAGRAM
5 PINNING
6 FUNCTIONAL DESCRIPTION
7 MEMORY ORGANIZATION
7.1 Program Memory
7.2 Internal Data Memory
7.3 External Data Memory
8 I/O PORT STRUCTURE
9 PULSE WIDTH MODULATED OUTPUTS (PWM)
9.1 Prescaler frequency control register (PWMP)
9.2 Pulse Width Register 0 (PWM0)
9.3 Pulse Width Register 1 (PWM1)
10 ANALOG-TO-DIGITAL CONVERTER (ADC)
10.1 ADC Control register (ADCON)
11 TIMERS/COUNTERS
11.1 Timer 0 and Timer 1
11.2 Timer T2 Capture and Compare Logic
11.3 Watchdog Timer (T3)
12 SERIAL I/O PORT: SIO0 (UART)
13 SERIAL I/O PORT: SIO1 (CAN)
13.1 On-chip CAN-controller
13.2 CAN Features
13.3 Interface between CPU and CAN
13.4 Hardware blocks of the CAN-controller
13.5 Control Segment and Message Buffer description
13.6 CAN 2.0A Protocol description
14 INTERRUPT SYSTEM
14.1 Interrupt Enable and Priority Registers
14.2 Interrupt Vectors
14.3 Interrupt Priority
15 POWER REDUCTION MODES
15.1 Power Control Register (PCON)
15.2 CAN Sleep Mode
15.3 Idle Mode
15.4 Power-down Mode
16 OSCILLATOR CIRCUITRY
17 RESET CIRCUITRY
17.1 Power-on Reset
18 INSTRUCTION SET
18.1 Addressing Modes
18.2 Instruction Set
19 ABSOLUTE MAXIMUM RATINGS (note 1)
20 DC CHARACTERISTICS
21 AC CHARACTERISTICS
22 CAN APPLICATION INFORMATION
22.1 Latency time requirements
22.2 Connecting a P8xC592 to a bus line (physical layer)
23 PACKAGE OUTLINES
24 SOLDERING
24.1 Introduction
24.2 Reflow soldering
24.3 Wave soldering
24.4 Repairing soldered joints
25 DEFINITIONS
26 LIFE SUPPORT APPLICATIONS

6. Pin 55, REF:
   a) Selection of input resp. output dependent of CAN Control Register bit 5 (CR.5; see Section 13.5.3 Table 32).
   b) If the internal reference is used, then REF should be connected to AV_{SS} via a capacitor with a value of ≥10 nF.
   c) After an external reset (RST = HIGH) the internal $\frac{1}{2}AV_{DD}$ source is activated and, REF is a reference output.
   d) If the CAN-controller is in the reset state, e.g. after an external reset, then the $\frac{1}{2}AV_{DD}$ source is switched off during Power-down mode.
7. CAN-bus line:
   a) CRX0 level > CRX1 level is interpreted as a logic 1 (recessive).
   b) CRX0 level < CRX1 level is interpreted as a logic 0 (dominant).
8. The level of AV_{REF+} must be higher than that of AV_{REF-}.

Table 2 Pin description for pins with **alternative functions** (SOT188-2 and NO330; see note 1)

SYMBOL		PIN	DESCRIPTION
DEFAULT	**ALTERNATIVE**		
Port 4			
P4.0 to P4.7		7 to 14	**8-bit quasi-bidirectional I/O port.**
	CMSR0 CMSR1 CMSR2 CMSR3 CMSR4 CMSR5	7 8 9 10 11 12	**Compare and Set/Reset outputs** for Timer T2.
	CMT0 CMT1	13 14	**Compare and toggle outputs** for Timer T2.
Port 1			
P1.0 to P1.7		16 to 21, 23, 24	**8-bit quasi-bidirectional I/O port.**
	CT0I/INT2 CT1I/INT3 CT2I/INT4 CT3I/INT5	16 17 18 19	**Capture timer inputs** for Timer T2, or **External interrupt inputs**.
	T2	20	**T2 event input** (rising edge triggered).
	RT2	21	**T2 timer reset input** (rising edge triggered).
	CTX0	23	**CAN transmitter output 0** (note 2).
	CTX1	24	**CAN transmitter output 1** (note 2).

6 FUNCTIONAL DESCRIPTION

The P8xC592 functions will be described as shown in the following overview:

- Memory organization
- I/O Port structure
- Pulse Width Modulated outputs
- Analog-to-digital Converter
- Timers/Counters
- Serial I/O Ports
- Interrupt system
- Power reduction modes
- Oscillator circuitry
- Reset circuitry
- Instruction Set.

7 MEMORY ORGANIZATION

The Central Processing Unit (CPU) manipulates operands in three memory spaces (see Fig.4) as follows:

- 16 kbytes internal resp. 64 kbytes external Program Memory
- 512 bytes internal Data Memory MAIN- and AUXILIARY RAM
- up to 64 kbytes external Data Memory (with 256 bytes residing in the internal AUXILIARY RAM).

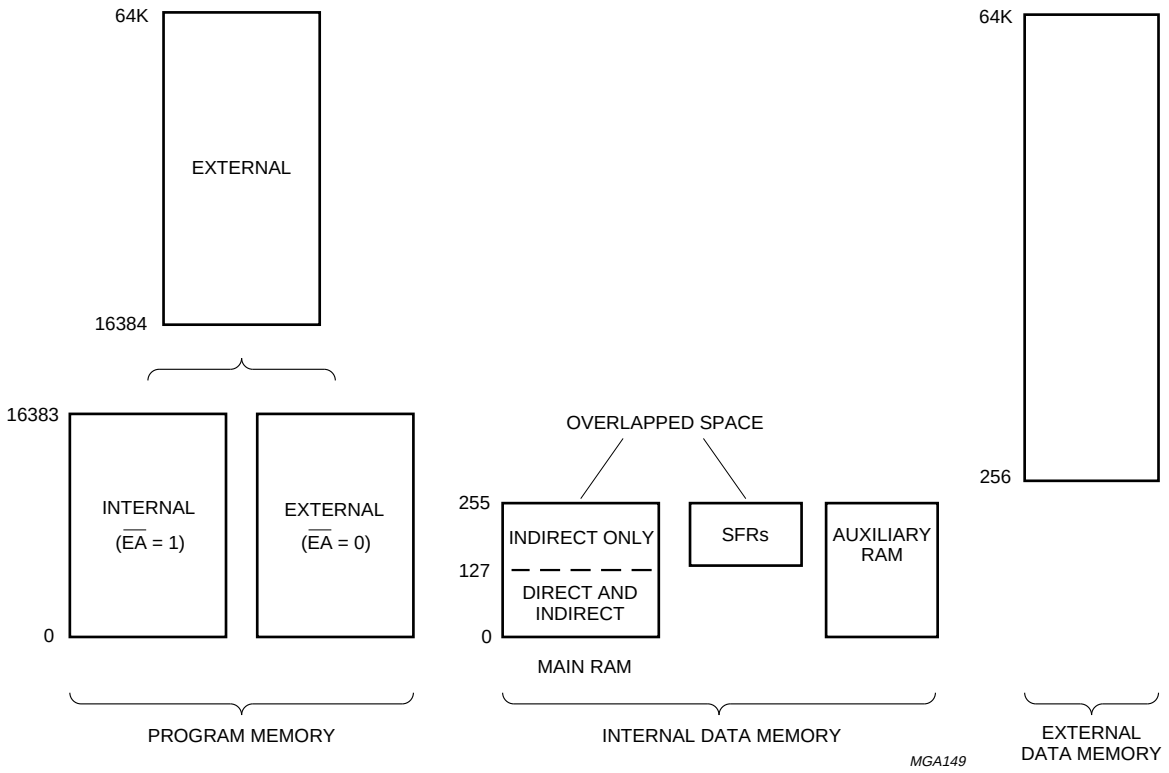


Fig.4 Memory map.

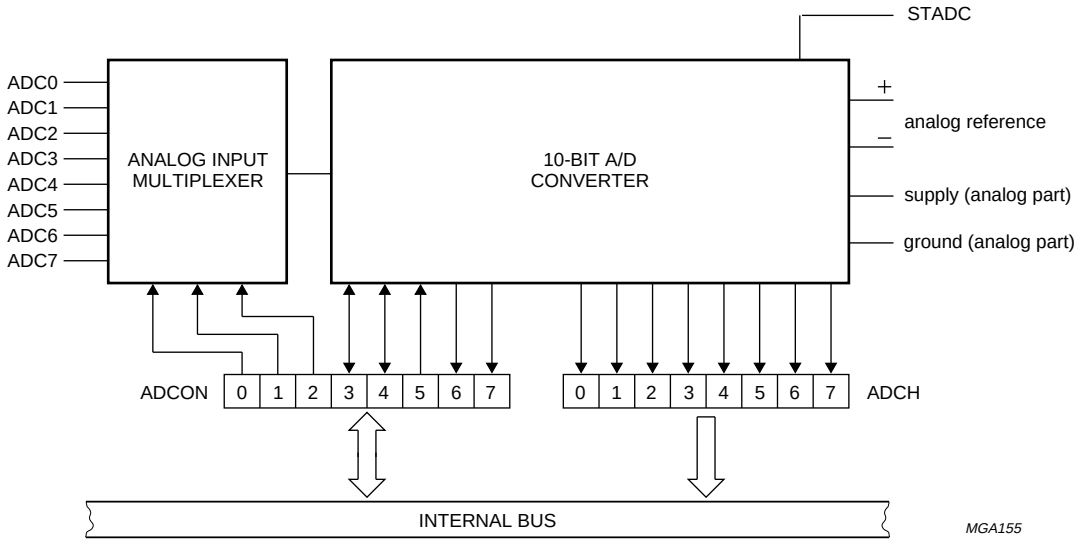


Fig.10 Functional diagram of analog input.

11.2.1 Counter Control register (TM2CON)

Table 16 Counter Control register (address EAH)

7	6	5	4	3	2	1	0
T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0

Table 17 Description of the TM2CON bits

BIT	SYMBOL	FUNCTION
7	T2IS1	Timer 2 16-bit overflow interrupt select.
6	T2IS0	Timer 2 byte overflow interrupt select.
5	T2ER	Timer 2 external reset enable.
4	T2B0	Timer 2 byte overflow interrupt flag.
3	T2P1	Timer 2 prescaler select (see Table 18).
2	T2P0	
1	T2MS1	Timer 2 mode select (see Table 19).
0	T2MS0	

Table 18 Timer 2 prescaler select

T2P1	T2P0	T2 CLOCK
0	0	Clock source
0	1	$\frac{1}{2}$ Clock source
1	0	$\frac{1}{4}$ Clock source
1	1	$\frac{1}{8}$ Clock source

Table 19 Timer 2 mode select

T2MS1	T2MS0	MODE
0	0	Timer T2 is halted
0	1	T2 clock source = $\frac{1}{12}f_{CLK}$.
1	0	Test mode; do not use
1	1	T2 clock source = pin T2

11.2.2 Capture Control register (CTCON)

Table 20 Capture Control register (address EBH)

7	6	5	4	3	2	1	0
CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0

Table 21 Description of the CTCON bits

BIT	SYMBOL	FUNCTION	
		CAPTURE	INTERRUPT ON
7	CTN3	CT3I	negative edge
6	CTP3	CT3I	positive edge
5	CTN2	CT2I	negative edge
4	CTP2	CT2I	positive edge
3	CTN1	CT1I	negative edge
2	CTP1	CT1I	positive edge
1	CTN0	CT0I	negative edge
0	CTP0	CT0I	positive edge

11.2.5 RESET/TOGGLE ENABLE REGISTER (RTE)

Table 26 Reset/Toggle Enable register (address EFH)

7	6	5	4	3	2	1	0
TP47	TP46	RP45	RP44	RP43	RP42	RP41	RP40

Table 27 Description of the RTE bits (note 1)

BIT	SYMBOL	FUNCTION
7	TP47	if HIGH then P4.7 toggles on a match of CM2 and T2
6	TP46	if HIGH then P4.6 toggles on a match of CM2 and T2
5	RP45	if HIGH then P4.5 is reset on a match of CM1 and T2
4	RP44	if HIGH then P4.4 is reset on a match of CM1 and T2
3	RP43	if HIGH then P4.3 is reset on a match of CM1 and T2
2	RP42	if HIGH then P4.2 is reset on a match of CM1 and T2
1	RP41	if HIGH then P4.1 is reset on a match of CM1 and T2
0	RP40	if HIGH then P4.0 is reset on a match of CM1 and T2

Note

1. If RTE.n is LOW then P4.n is not affected by a match of CM1 and T2 or CM2 and T2.
   For more information, refer to the 8051-based *“8-bit Microcontrollers Data Handbook IC20”*.

11.3 Watchdog Timer (T3)

In addition to Timer T2 and the standard timers (Timer 0 and Timer 1), a Watchdog Timer (WDT) comprising an 11-bit prescaler and an 8-bit timer (T3) is also provided (see Fig.12).

The timer T3 is incremented every 1.5 ms, derived from the oscillator frequency of 16 MHz by the following formula: $f_{timer} = \frac{f_{CLK}}{12 \times 2048}$

When a timer T3 overflow occurs, the microcontroller is reset and a reset-output-pulse is generated at pin RST. This short output pulse (3 machine cycles) may be suppressed if the RST pin is connected to a capacitor.

To prevent a system reset (by an overflow of the WDT), the user program has to reload T3 within periods that are shorter than the programmed Watchdog time interval.

If the processor suffers a hardware/software malfunction, the software will fail to reload the timer. This failure will produce a reset upon overflow thus preventing the processor running out of control.

The Watchdog Timer can only be reloaded if the condition flag WLE = PCON.4 has been previously set by software. At the moment the counter is loaded the condition flag is automatically cleared.

The timer interval between the timer's reloading and the occurrence of a reset depends on the reloaded value. For example, this may range from 1.5 ms to 0.375 s when using an oscillator frequency of 16 MHz.

In the Idle state the Watchdog Timer and reset circuitry remain active.

The Watchdog Timer (WDT) is controlled by the Enable Watchdog pin ($\overline{EW}$) (see Table 28).

Table 28 $\overline{EW}$ controlling WDT and Power-down mode

PIN $\overline{EW}$	WDT	POWER-DOWN MODE
LOW	enabled	disabled
HIGH	disabled	enabled

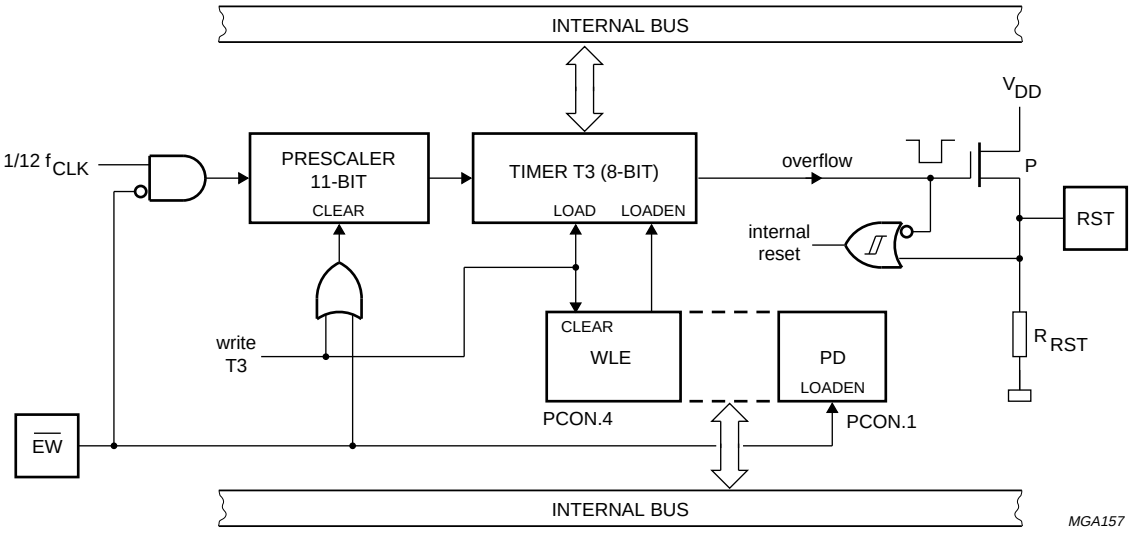


Fig.12 Functional diagram of T3 Watchdog Timer.

13.5.5 STATUS REGISTER (SR)

The contents of the Status Register reflects the status of the CAN-controller. The Status Register appears to the CPU as a read only memory.

Table 36 Status Register (address 2)

7	6	5	4	3	2	1	0
BS	ES	TS	RS	TCS	TBS	DO	RBS

Table 37 Description of the SR bits

BIT	SYMBOL	FUNCTION
7	BS	**Bus Status** (note 1). If the value of BS is: HIGH (Bus-OFF), then the CAN-controller is not involved in bus activities. LOW (Bus-ON), then the CAN-controller is involved in bus activities.
6	ES	**Error Status**. If the value of ES is: HIGH (error), then at least one of the Error Counters (see Section 13.6.10) has reached the CPU Warning limit. LOW (ok), then both Error Counters have not reached the warning limit.
5	TS	**Transmit Status** (note 2). If the value of TS is: HIGH (transmit), then the CAN-controller is transmitting a message. LOW (idle), then no message is transmitted.
4	RS	**Receive Status** (note 2). If the value of RS is: HIGH (receive), then the CAN-controller is receiving a message. LOW (idle), then no message is received.
3	TCS	**Transmission Complete Status** (note 3). If the value of TCS is: HIGH (complete), then last requested transmission has been successfully completed. LOW (incomplete), then previously requested transmission is not yet completed.
2	TBS	**Transmit Buffer Access** (note 3). If the value of TBS is: HIGH (released), then the CPU may write a message into the TBF. LOW (locked), then the CPU cannot access the Transmit Buffer. A message is either waiting for transmission or is in the process of being transmitted.
1	DO	**Data Overrun** (note 4). If the value of DO is: HIGH (overrun), then both Receive Buffers are full and the first byte of another message should be stored. LOW (absent), then no data overrun has occurred since the Clear Overrun command was given.
0	RBS	**Receive Buffer Status** (note 5). If the value of RBS is HIGH (full), then this bit is set when a new message is available. LOW (empty), then no message has become available since the last Release Receive Buffer command bit was set.

13.5.11 Output Control Register (OCR)

The Output Control Register allows, under software control, the set-up of different output driver configurations. This register can be accessed (read/write) if the Reset Request bit is set HIGH (present). If the CAN-controller is in the sleep mode (Sleep = HIGH) a recessive level is output on the CTX0 and CTX1 pins. If the CAN-controller is in the reset state (Reset Request = HIGH) the output drivers are floating.

Tables 50 and 51, show the relationship between the bits of the Output Control Register and the two serial output pins CTX0 and CTX1 of the P8xC592 CAN-controller, connected to the serial bus (see Fig.14).

Table 49 Output Control Register (address 8)

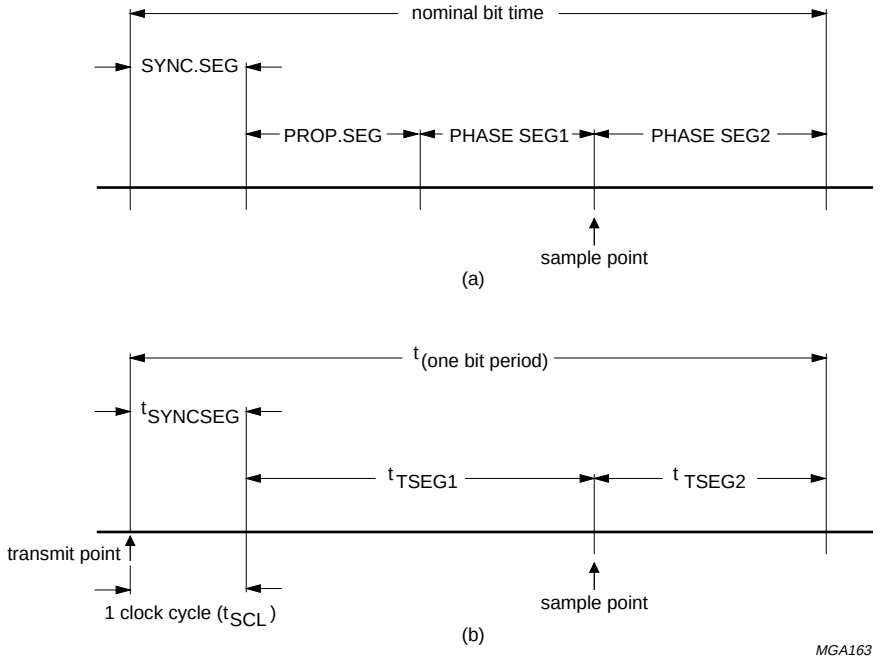
7	6	5	4	3	2	1	0
OCTP1	OCTN1	OCPOL1	OCTP0	OCTN0	OCPOL0	OCMODE1	OCMODE0

Table 50 Description of the OCR bits

BIT	SYMBOL	FUNCTION
7	OCTP1	See Tables 51 and 52.
6	OCTN1	
5	OCPOL1	
4	OCTP0	
3	OCTN0	
2	OCPOL0	
1	OCMODE1	**Output Mode**. These bits select the output mode; see Table 51.
0	OCMODE0	

Table 51 Description of the Output Mode bits

OCMODE1	OCMODE0	DESCRIPTION
1	0	**Normal Output Mode**. The bit sequence (TXD) is sent via CTX0, CTX1. TXD is the data bit to be transmitted. The voltage levels on the output driver pins CTX0 and CTX1 depend on both the driver characteristic programmed by OCTPx, OCTNx (float, pull-up, pull-down, push-pull) and the output polarity programmed by OCPOLx (see Fig.17).
1	1	**Clock Output Mode**. For the CTX0 pin this is the same as in Normal Output Mode (CTX0: bit sequence). However, the data stream to CTX1 is replaced by the transmit clock (TXCLK). The rising edge of the transmit clock (non-inverted) marks the beginning of a bit period. The clock pulse width is t_{SCL}.
0	0	**Bi-phase Output Mode**. In contrast to Normal Output Mode the bit representation is time variant and toggled. If the bus controllers are galvanically decoupled from the bus-line by a transformer, the bit stream is not allowed to contain a DC component. This is achieved by the following scheme. During recessive bits all outputs are deactivated (floating). Dominant bits are sent alternately on CTX0 and CTX1, i.e. the first dominant bit is sent on CTX0, the second is sent on CTX1, and the third one is sent on CTX0 again, etc.
0	1	**Test Output Mode.** For the CTX0 pin this is the same as in Normal Output Mode (CTX0: bit sequence). To measure the delay time of the transmitter and receiver this mode connects the output of the input comparator (COMP OUT) with the input of the output driver CTX1. This mode is used for production testing only.



(a) As defined by the CAN-protocol.
(b) As implemented in the P8xC592's on-chip CAN-controller.

Fig.18 Bit period.

13.5.19.2 Time Segment 1 (TSEG1)

This segment determines the location of the sampling point within a bit period, which is at the end of TSEG1. TSEG1 is programmable from 1 to 16 system clock cycles (see Section 13.5.10).

The correct location of the sample point is essential for the correct functioning of a transmission. The following points must be taken into consideration:

- A Start-Of-Frame (see Section 13.6.2) causes all CAN-controllers to perform a 'hard synchronization' (see Section 13.5.20) on the first recessive-to-dominant edge.
  During arbitration, however, several CAN-controllers may simultaneously transmit. Therefore it may require twice the sum of bus-line, input comparator and the output driver delay times until the bus is stable.
  This is the propagation delay time.
- To avoid sampling at an incorrect position, it is necessary to include an additional synchronization buffer on both sides of the sample point.
  The main reasons for incorrect sampling are:
  - Incorrect synchronization due to spikes on the bus-line
  - Slight variations in the oscillator frequency of each CAN-controller in the network, which results in a phase error.
- Time Segment 1 consists of the segment for compensation of propagation delays and the synchronization buffer segment directly before the sample point (see Fig.18).

13.5.20.1 Synchronization Rules

The synchronization rules are as follows:

- Only one synchronization within one bit time is used.
- An edge is used for synchronization only if the value detected at the previous sample point differs from the bus value immediately after the edge.
- Hard synchronization is performed whenever there is a recessive-to-dominant edge during Bus-Idle (see Section 13.6.6).
- All other edges (recessive-to-dominant and optionally dominant-to recessive edges if the Sync bit is set HIGH (see Section 13.5.3) which are candidates for resynchronization will be used with the following exception:
  - A transmitting CAN-controller will not perform a resynchronization as a result of a recessive-to-dominant edge with positive phase error, if only these edges are used for resynchronization. This ensures that the delay times of the output driver and input comparator do not cause a permanent increase in the bit time.

13.6 CAN 2.0A Protocol description

13.6.1 FRAME TYPES

The P8xC592's CAN-controller supports the four different CAN-protocol frame types for communication:

- Data Frame, to transfer data
- Remote Frame, request for data
- Error Frame, globally signal a (locally) detected error condition
- Overload Frame, to extend delay time of subsequent frames (an Overload Frame is not initiated by the P8xC592 CAN-controller).

13.6.1.1 Bit representation

There are two logical bit representations used in the CAN-protocol:

- A recessive bit on the bus-line appears only if all connected CAN-controllers send a recessive bit at that moment.
- Dominant bits always overwrite recessive bits i.e. the resulting bit level on the bus-line is dominant.

13.6.2 DATA FRAME

A Data Frame carries data from a transmitting CAN-controller to one or more receiving ones.
A Data Frame is composed of seven different bit-fields:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field (may have a length of zero)
- CRC Field (CRC = Cyclic Redundancy Code)
- Acknowledge Field
- End-Of-Frame.

13.6.2.1 Start-Of-Frame bit

Signals the start of a Data Frame or Remote Frame. It consists of a single dominant bit use for hard synchronization of a CAN-controller in receive mode.

13.6.2.2 Arbitration Field

Consists of the message Identifier and the RTR bit. In the case of simultaneous message transmissions by two or more CAN-controllers the bus access conflict is solved by bit-wise arbitration, which is active during the transmission of the Arbitration Field.

13.6.2.3 Identifier

This 11-bit field is used to provide information about the message, as well as the bus access priority. It is transmitted in the order ID.10 to ID.0 (LSB). The situation that the seven most significant bits (ID.10 to ID.4) are all recessive must not occur.

An Identifier does not define which particular CAN-controller will receive the frame because a CAN based communication network does not differentiate between a point-to-point, multicast or broadcast communication.

13.6.5.2 Overload Delimiter

The Overload Delimiter consists of eight recessive bits and takes the same form as the Error Delimiter. After transmission of an Overload Flag, each CAN-controller monitors the bus-line until it detects a transition from a dominant-to-recessive bit level. At this point in time, every CAN-controller has finished sending its Overload Flag and all CAN-controllers start simultaneously transmitting seven more recessive bits.

13.6.6 INTER-FRAME SPACE

Data Frames and Remote Frames are separated from preceding frames (all types) by an Inter-Frame Space, consisting of an Intermission Field and a Bus-Idle. Error-passive CAN-controllers also send a Suspend Transmission (see Section 13.6.9) after transmission of a message. Overload Frames and Error Frames are not preceded by an Inter-Frame Space.

13.6.6.1 Intermission Field

The Intermission Field consists of three recessive bits. During an Intermission period, no frame transmissions will be started by the P8xC592's on-chip CAN-controller. An Intermission is required to have a fixed time period to allow a CAN-controller to execute internal processes prior to the next receive or transmit task.

13.6.6.2 Bus-Idle

The Bus-Idle time may be of arbitrary length (min. 0 bit). The bus is recognized to be free and a CAN-controller having information to transmit may access the bus. The detection of a dominant bit level during Bus-Idle on the bus is interpreted as the Start-Of-Frame.

13.6.7 BUS ORGANIZATION

Bus organization is based on five basic rules described in the following subsections.

13.6.7.1 Bus Access

CAN-controllers only start transmission during the Bus-Idle state. All CAN-controllers synchronize on the leading edge of the Start-Of-Frame (hard synchronization).

13.6.7.2 Bus Arbitration

If two or more CAN-controllers simultaneously start transmitting, the bus access conflict is solved by a bit-wise arbitration process during transmission of the Arbitration Field.

During arbitration every transmitting CAN-controller compares its transmitted bit level with the monitored bus level. Any CAN-controller which transmits a recessive bit and monitors a dominant bus level immediately becomes the receiver of the higher-priority message on the bus without corrupting any information on the bus. Each message contains an unique Identifier and a RTR bit describing the type of data within the message. The Identifier together with the RTR bit implicitly define the message's bus access priority. During arbitration the most significant bit of the Identifier is transmitted first and the RTR bit last. The message with the lowest binary value of the Identifier and RTR bit has the highest priority. A Data Frame has higher priority than a Remote Frame due to its RTR bit having a dominant level.

For every Data Frame there is an unique transmitter. For reasons of compatibility with other CAN-bus controllers, use of the Identifier bit pattern ID = 1111111XXXXB (X being bits of arbitrary level) is forbidden.

The number of available different Identifiers:

$(2^{11} - 2^4) = 2032$.

13.6.7.3 Coding/Decoding

The following bit fields are coded using the bit-stuffing technique:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field
- CRC Sequence.

When a transmitting CAN-controller detects five consecutive bits of identical polarity to be transmitted, a complementary (stuff) bit is inserted into the transmitted bit-stream.

When a receiving CAN-controller has monitored five consecutive bits with identical polarity in the received bit streams of the above described bit fields, it automatically deletes the next received (stuff) bit. The level of the deleted stuff bit has to be the complement of the previous bits; otherwise a Stuff Error will be detected and signalled (see Section 13.6.8).

The remaining bit fields or frames are of fixed form and are not coded or decoded by the method of bit-stuffing.

The bit-stream in a message is coded according to the Non-Return-to-Zero (NRZ) method, i.e. during a bit period, the bit level is held constant, either recessive or dominant.

13.6.7.4 Error Signalling

A CAN-controller which detects an error condition, transmits an Error Flag. Whenever a Bit Error, Stuff Error, Form Error or an Acknowledgement Error is detected, transmission of an Error Flag is started at the next bit. Whenever a CRC Error is detected, transmission of an Error Flag starts at the bit following the Acknowledge Delimiter, unless an Error Flag for another error condition has already started. An Error Flag violates the bit-stuffing law or corrupts the fixed form bit fields. A violation of the bit-stuffing law affects any CAN-controller which detects the error condition. These devices will also transmit an Error Flag.

An error-passive CAN-controller (see Section 13.6.9) which detects an error condition, transmits a Passive Error Flag. A Passive Error Flag is not able to interrupt a current message at different CAN-controllers but this type of Error Flag may be ignored (overwritten) by other CAN-controllers. After having detected an error condition, an error-passive CAN-controller will wait for six consecutive bits with identical polarity and when monitoring them, interpret them as an Error Flag.

After transmission of an Error Flag, each CAN-controller monitors the bus-line until it detects a transition from a dominant-to-recessive bit level. At this point in time, every CAN-controller has finished transmitting its Error Flag and all CAN-controllers start transmitting seven additional recessive bits (Error Delimiter, see Section 13.6.4).

The message format of a Data Frame or Remote Frame is defined in such a way that all detectable errors can be signalled within the message transmission time and therefore it is very simple for the CAN-controllers to associate an Error Frame to the corresponding message and to initiate retransmission of the corrupted message. If a CAN-controller monitors any deviation of the fixed form of an Error Frame, it transmits a new Error Frame.

13.6.7.5 Overload Signalling

Some CAN-controllers (but not the one on-chip of the P8xC592) require to delay the transmission of the next Data Frame or Remote Frame by transmitting one or more Overload Frames. The transmission of an Overload Frame must start during the first bit of an expected Intermission Field. Transmission of Overload Frames which are reactions on a dominant bit during an expected Intermission Field, start one bit after this event.

Though the format of Overload Frame and Error Frame are identical, they are treated differently. Transmission of an Overload Frame during Intermission Field does not initiate the retransmission of any previous Data Frame or Remote Frame. If a CAN-controller which transmitted an Overload Frame monitors any deviation of its fixed form, it transmits an Error Frame.

13.6.8 Error Detection

The processes described in Sections 13.6.8.1 to 13.6.10.3 are implemented in the P8xC592's on-chip CAN-controller for error detection.

13.6.8.1 Bit Error

A transmitting CAN-controller monitors the bus on a bit-by-bit basis. If the bit level monitored is different from the transmitted one, a Bit Error is signalled.
The exceptions being:

- During the Arbitration Field, a recessive bit can be overwritten by a dominant bit. In this case, the CAN-controller interprets this as a loss of arbitration.
- During the Acknowledge Slot, only the receiving CAN-controllers are able to recognize a Bit Error.

13.6.8.2 Stuff Error

The following bit fields are coded using the bit-stuffing technique:

- Start-Of-Frame
- Arbitration Field
- Control Field
- Data Field
- CRC Sequence.

There are two possible ways of generating a Stuff Error:

- A disturbance generates more than the allowed five consecutive bits with identical polarity. These errors are detected by all CAN-controllers.
- A disturbance falsifies one or more of the five bits preceding the stuff bit. This error situation is not recognized as a Stuff Error by the receivers. Therefore, other error detection processes may detect this error condition such as:
  - CRC check, format violation at the receiving CAN-controllers, or
  - Bit Error detection by the transmitting CAN-controller.

15.4 Power-down Mode

The instruction that sets bit PCON.1 to HIGH, is the last one executed before entering the Power-down mode. In Power-down mode the oscillator of the P8xC592 is stopped. If the CAN-controller is in use, it is recommended to set it into Sleep mode before entering Power-down mode. However, setting PCON.1 to HIGH also sets the Sleep bit (CAN-controller Command Register bit 4) to HIGH.

The P8xC592 leaves Power-down mode either by a hardware reset or by a CAN Wake-Up interrupt (due to activity on the CAN-bus), if the SIO1 (CAN) interrupt source is enabled (contents of register IEN0 = 1X1XXXXXB).

A hardware reset affects the whole P8xC592, but leaves the contents of the on-chip RAM unchanged (CAN-controller-and CPU's SFRs are reset, see Section 13.5.2, Chapter 17 and Table 40). A CAN Wake-Up interrupt during Power-down mode causes a reset output pulse with a width of 6144 machine cycles (4.6 ms with f_{CLK} = 16 MHz). All hardware except that for the CAN-controller of the P8xC592 is reset (i.e. the contents of all CAN-controller registers are preserved).

A capacitance connected to the RST pin can be used to lengthen the internally generated reset pulse. If the pulse exceeds 8192 machine cycles, the CAN-controller part is reset too.

Table 82 Status of external pins during Idle and Power-down modes

MODE	PROGRAM	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1(1)	PORT2	PORT3	PORT4	PWM0/ PWM1
Idle	internal	1	1	port data	port data	port data	port data	port data	1
	external	1	1	floating	port data	address	port data	port data	1
Power-down	internal	0	0	port data	port data	port data	port data	port data	1
	external	0	0	floating	port data	port data	port data	port data	1

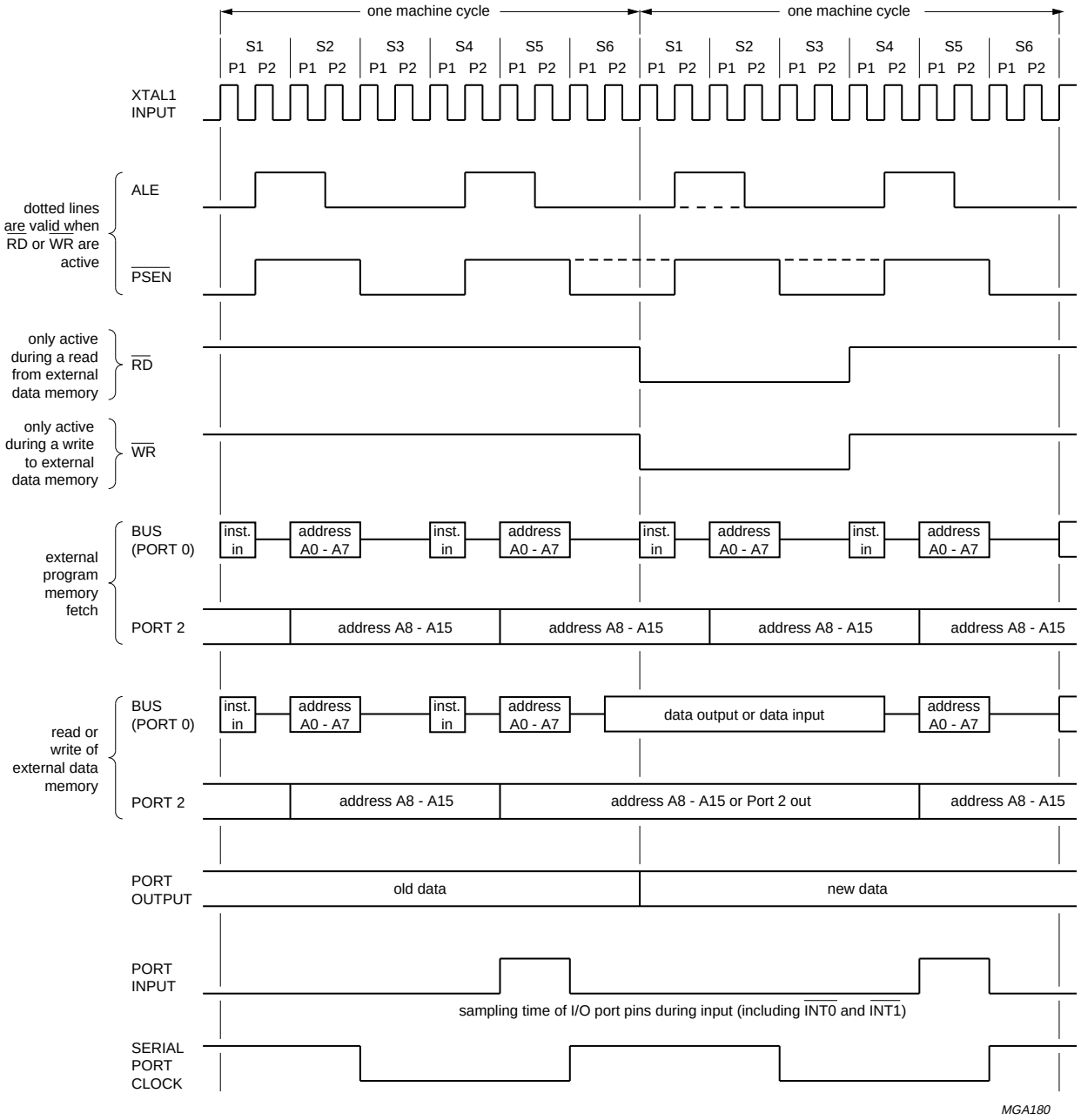
Note

1. If the port pins P1.6 and P1.7 are used as the CAN transmitter outputs (CTX0 and CTX1), then during Sleep and Power-down mode these pins output a ‘recessive’ level (see Sections 13.5.2 and 13.5.11).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
CAN output driver (V_{DD} = 5 V ± 5%)					
V_{OLT}	LOW level output voltage (CTX0 and CTX1)	I_o = 1.2 mA; note 15 I_o = 10 mA	– –	0.1 0.6	V V
V_{OHT}	High level output voltage (CTX0 and CTX1)	I_o = –1.2 mA; note 15 I_o = –10 mA; note 16	V_{DD} – 0.1 V_{DD} – 0.6	– –	V V
Reference (AV_{DD} = 5 V ± 5%)					
V_{REFOUT}	REF output voltage	–0.1 mA < I_L < 0.1 mA; C_L = 10 nF; note 15; bit Reference Active = HIGH	½AV_{DD}–0.1	½AV_{DD}+0.1	V
I_{REFIN}	REF input current	1.5 V < V_{REFIN} < AV_{DD}–1.5 V; bit Reference Active = LOW	–	±10	µA

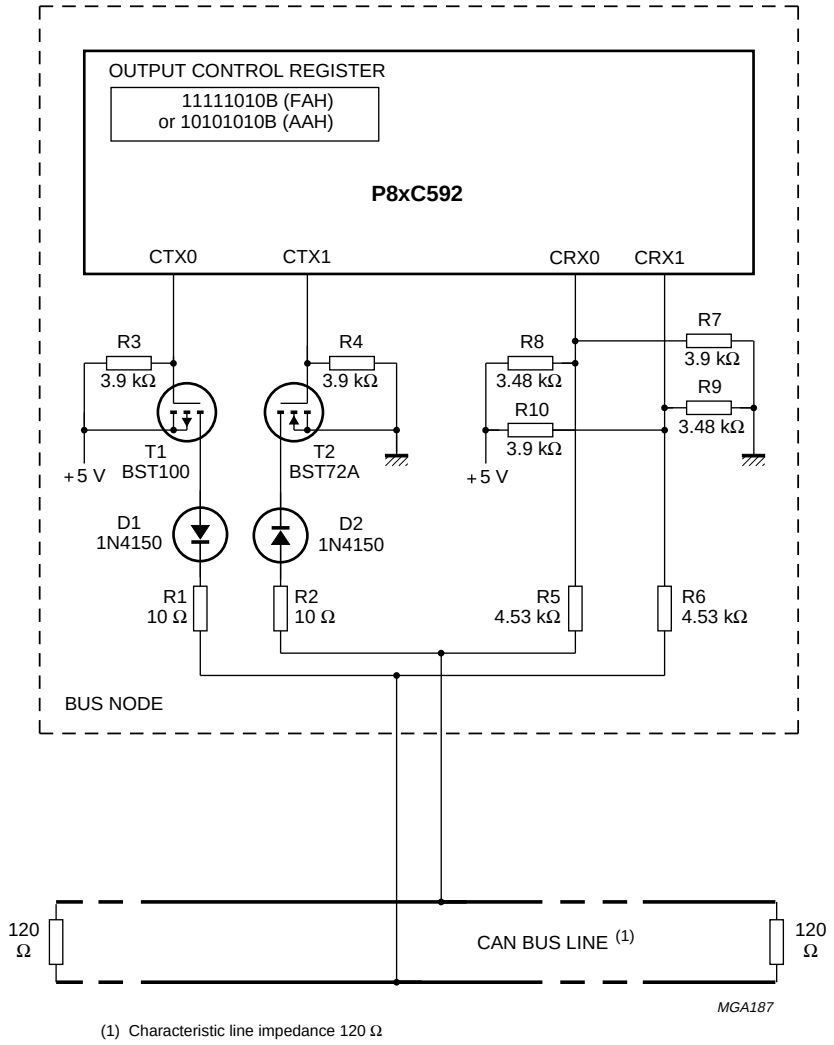
Notes to the DC characteristics

1. Conditions for:
   a) The **digital** operating current measurement: all output pins disconnected; XTAL1 is driven with $t_r = t_f$ = 10 ns; $V_{IL} = V_{SS}$ + 0.5 V; $V_{IH} = V_{DD}$ – 0.5 V; $\overline{EA}$ = RST = Port 0 = P1.6 = P1.7 = $\overline{EW}$ = V_{DD}; STADC = V_{SS}; CRX0 = 2.7 V; CRX1 = 2.3 V.
   b) The **analog** operating current measurement: Port 5 = AV_{DD}; CAN: register 6: = 00H; load current reference voltage source 100 µA.
2. Conditions for:
   a) The **digital** Idle mode supply current measurement: all output pins disconnected; XTAL1 is driven with $t_r = t_f$ = 10 ns; $V_{IL} = V_{SS}$ + 0.5 V; $V_{IH} = V_{DD}$ – 0.5 V; Port 0 = P1.6 = P1.7 = $\overline{EW}$ = V_{DD}; $\overline{EA}$ = RST = STADC = V_{SS}; CRX0 = 2.7 V; CRX1 = 2.3 V.
   b) The **analog** Idle mode current measurement: Port 5 = AV_{DD}; CAN: register 6: = 00H; load current reference voltage source 100 µA.
3. Conditions for:
   a) The **digital** Idle and Sleep mode supply current measurement: all output pins disconnected; XTAL1 is driven with $t_r = t_f$ = 10 ns; $V_{IL} = V_{SS}$ + 0.5 V; $V_{IH} = V_{DD}$ – 0.5 V; Port 0 = P1.6 = P1.7 = $\overline{EW}$ = CRX0 = V_{DD}; $\overline{EA}$ = RST = STADC = CRX1 = V_{SS}; CAN: register 6: = 00H, register 7: = 12H, register 8: = 02H, register 0: = 20H, wait 15t_{CY}, register 1: = 10H, wait for bit Sleep = 1.
   b) The **analog** Idle and Sleep mode current measurement: Port 5 = AV_{DD}; load current reference voltage source 100 µA.
4. Window devices have to be covered. Conditions for:
   a) The **digital** Power-down mode supply current measurement: all output pins and Port 5 disconnected; Port 0 = P1.6 = P1.7 = $\overline{EW}$ = CRX0 = V_{DD}; $\overline{EA}$ = RST = STADC = CRX1 = XTAL1 = AV_{REF+} = AV_{REF-} = CV_{SS} = V_{SS}; AV_{DD} = V_{DD}, but current into AV_{DD} pin is not comprised in digital Power-down current.
   b) The **analog** Power-down mode supply current measurement: Port 5 = AV_{DD}.
5. Capacitive loads on Port 0 and Port 2 may degrade the LOW level output voltage of ALE, Port 1 and Port 3. During a HIGH-to-LOW transition on the Port 0 and Port 2 pins and a capacitive load >100 pF, the ALE LOW level may exceed 0.8 V. In the case that it is necessary to connect ALE to a Schmitt trigger input respectively use an address latch with a Schmitt trigger STROBE input.



The Port 5 input buffers have a maximum propagation delay of 300 ns. As a result Port 5 sample time begins 300 ns before state S5 and ends when S5 has finished.

Fig.30 Instruction cycle timing.



(1) Characteristic line impedance 120 Ω

Fig.38 In-vehicle Transceiver.

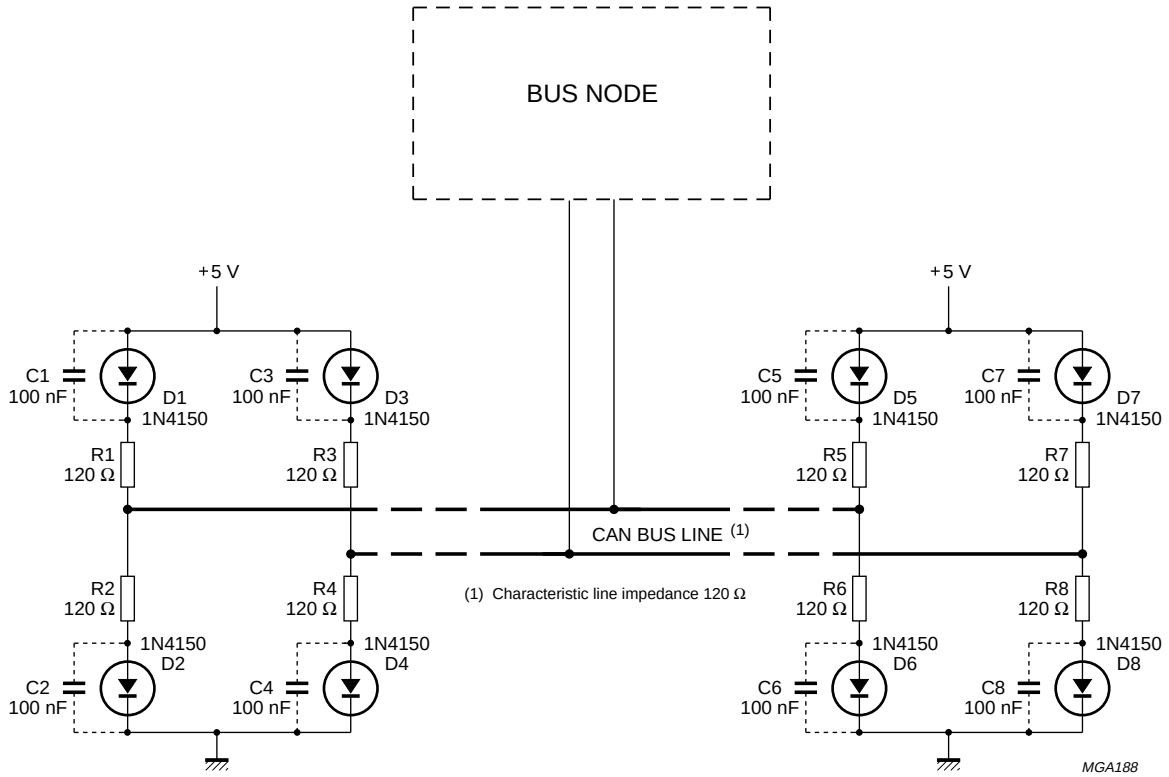


Fig.39 Bus termination with decoupled wires.

```
LOC   OBJ      LINE  SOURCE
00A0           107          ; determine the destination address in data-memory for the
00A1           108          ; message's Data-Field
      54E0     109          ANL     A, #ID2_0_MASK      ; use ID.2 ... ID.0 only
      C4       110          SWAP    A
      03       111          RR      A                   ; A = 4*ID.2 + 2*ID.1 + ID.0
               112          ; this value is used as an index for an array of 8 bytes
               113          ; containing the destination-addresses for the 8 different
               114          ; messages. Note, that the #RX_ARRAY_OFFSET is due to the
00A2           115          ; program counter-relative access to the array.
      2415     116          ADD     A, #RX_ARRAY_START – RX_ARRAY_OFFSET
      83       117          MOVC    A, @A + PC
               118          RX_ARRAY_OFFSET:
               119
00A5           120          ; if a message passes the acceptance-filter of the CAN
00A7           121          ; Controller, but the CPU doesn't need it, the array
               122          ; entry's value may be set to zero indicating this.
               123          ; The following <jz> instruction cares for this.
      6007     124          JZ      CAN_RX_READY
00A9           125
00AB           126          ; now copy the Data-Field (only) from CAN- to CPU memory
00AD           127          ; with the aid of the DMA-logic. Note, that a TX-DMA is
               128          ; performed when writing 8AH (DMA + address 10) into CANADR
               129          ; and a RX-DMA is performed when writing 94H (DMA + address 20)
               130          ; ... 9DH (DMA + address 29) into CANADR. Here address 22 is
               131          ; used to copy just the Data-Field.
      F5D8     132          MOV     CANSTA, A           ; data-memory address
      75DB96   133          MOV     CANADR, #CAN_RX_DMA ; starts RX-DMA at address 22
               134
00AE           135          ; the DMA-transfer is done in at maximum 2 instruction cycles.
00AF           136          ; During the transfer, neither the data-memory (RAM) nor one
               137          ; of the SFRs CANADR, CANDAT, CANCON and
00B0           138          ; CANSTA may be accessed by the CPU.
               139          ; For simplicity, two NOPs are used here.
      00       140          NOP
      00       141          NOP
00A0           142
```

NOTES