Application Note

1. Introduction

With its *smart* feature (3-wire serial interface), e2v's AT84AD001B dual 8-bit 1 Gsps ADC provides you with digital control over the various functions offered with the dual ADC: calibration, gain and offset adjustments, DMUX ratio selection, analog and clock input mode, and partial or full standby mode.

This digital control via the 3-wire serial interface can be managed using Atmel's ATmega128L AVR. The aim of this application note is to provide you with the relevant information for interfacing these two devices.

2. Driving e2v's AT84AD001B ADC 3-wire Serial Interface with Atmel's ATmega128L AVR

ATmega128L AVR can be used to drive the 3-wire serial interface of the AT84AD001B dual 8-bit 1 Gsps ADC.

This section provides a simple configuration for interfacing the AVR with the ADC.

Note: All the information pertaining to the AVR contained in this document complies with the version available at the date the document was created. Before design, compliance of this information with the current version of the device should be verified.

2.1 AT84AD001B Dual 8-bit 1 Gsps ADC 3-wire Serial Interface

The ATmega128L AVR can drive four signals of the AT84AD001B dual 8-bit 1 Gsps ADC. These are:

- The MODE signal (pin 74 of the 144-LQFP packaged device): used in the ADC to activate the 3-wire serial interface
- The CLK signal (pin 73 of the 144-LQFP packaged device): input clock for the serial interface
- The DATA signal (pin 72 of the 144-LQFP packaged device): input data for the serial interface
- The LDN signal (pin 71 of the 144-LQFP packaged device): beginning and end of the register line for the serial interface

Visit our website: www.e2v.com
for the latest version of the datasheet

0928B–BDC–03/08

The ADC's 3-wire serial interface only accepts 2.25V CMOS digital signals while ATmega128L must be supplied with V_{CC} ranging from 2.7V to 5V.

Hence, it was necessary during the design to add a buffer and line driver with 2.25V to 3.3V tolerant I/Os used as translators in this mixed 2.25V and 3.3V environment.

As the AVR can manage only four of the ADC's signals, a quad buffer is sufficient to translate the four signals from the AVR (WAKEUP, SPICLOCK, SPIDATA, SLE) to the MODE, CLK, DATA and LDN signals (2.25V) of the ADC.

Possible devices enabling a translation between the 3.3V of the AVR and the 2.25V required by the ADC are the 74LCX125 and 74LCX126 low voltage quad buffer and line with 5V-tolerant I/Os or the 74LCX244 low voltage octal buffer and line with 5V-tolerant I/Os available from any digital buffer manufacturer.

The 74LCX125 and 74LCX126 devices have the advantage of using only four inputs. Their main drawback, however, is that all four inputs cannot be connected to the same side of the device for a simplified layout.

Figure 2-1 to Figure 2-3 illustrate the possible application diagrams for the 74LCX125, 74LCX126 and 74LCX244 low voltage buffers with 5V tolerant I/Os.

Figure 2-1. Application Diagram Using the 74LCX125 Buffer

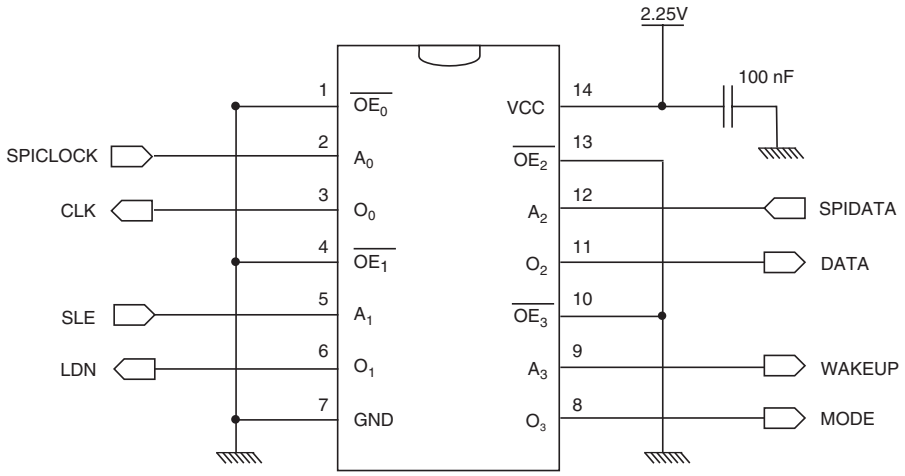


Table 2-1is the truth table of the 74LCX125 device.

Table 2-1. 74LCX125 Truth Table

Inputs		Outputs
$\overline{OE}_n$	A_n	O_n
L	L	L
L	H	H
H	Z	Z

Figure 2-2. Application Diagram Using the 74LCX126 Buffer

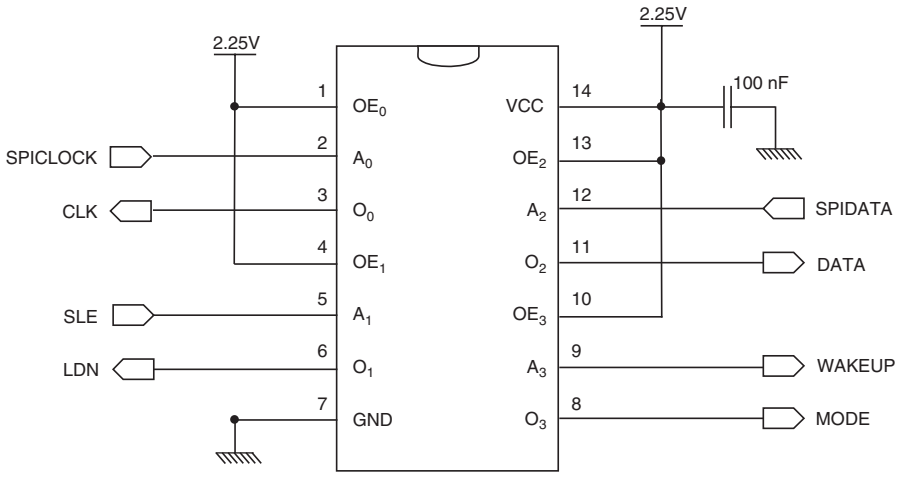
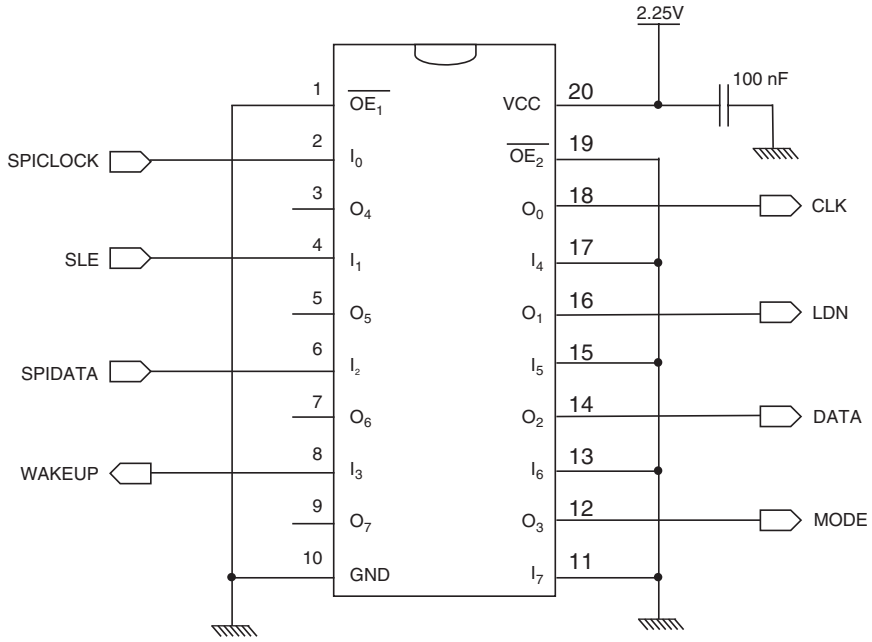


Table 2-2 is the truth table of the 74LCX126 device.

Table 2-2. 74LCX126 Truth Table

Inputs		Outputs
OE_n	A_n	O_n
H	L	L
H	H	H
L	Z	Z

Figure 2-3. Application Diagram Using the 74LCX244 Buffer



Note:
1. It is highly recommended the unused inputs of the octal buffer be connected to ground (directly to ground or via a 0Ω resistor in case future signal access is required) so that the buffers will never toggle and consequently dissipate power while they are not in use.
2. It may be useful to connect the OE signals to ground (74LCX125 and 74LCX244 devices) or to 2.25V (74LCX126) via a 0Ω resistor in case the application's signal level needs to be changed.
3. The WAKEUP signal is considered here as an input for both the 74LCX device and the AVR. When connected to 2.25V, the ADC's serial interface becomes active. When connected to ground, the ADC's serial interface is disabled. In the case of a demonstrator design, the serial interface can be useful to connect the WAKEUP signal to a LED to indicate that the 3-wire serial interface has been activated (LED lit) and to a push button (between ground and 3.3V), as illustrated in Figure 2-5 on page 6.

Table 2-3 is the truth table of the 74LCX244 device.

Table 2-3. 74LCX244 Truth Table

Inputs		**Outputs (O_0, O_1, O_2, O_3)**
$\overline{OE1}$	I_n	
L	L	L
L	H	H
H	X	Z
Inputs		**Outputs (O_4, O_5, O_6, O_7)**
	I_n	
L	L	L
L	H	H
H	X	Z

Figure 2-4. Global Schematic Using AT84AD001B ADC with ATmega128 AVR and 74LCX244 Buffer

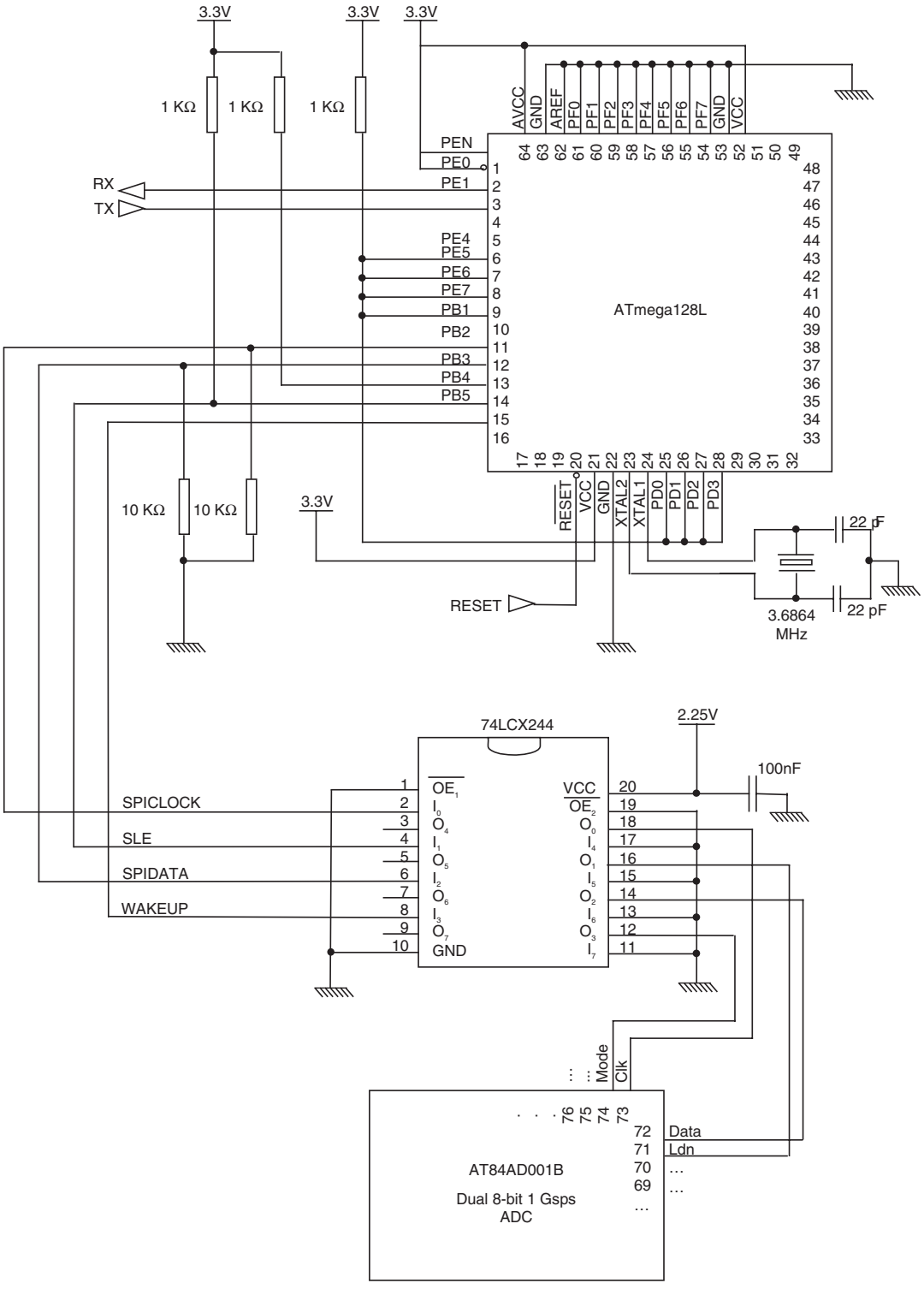
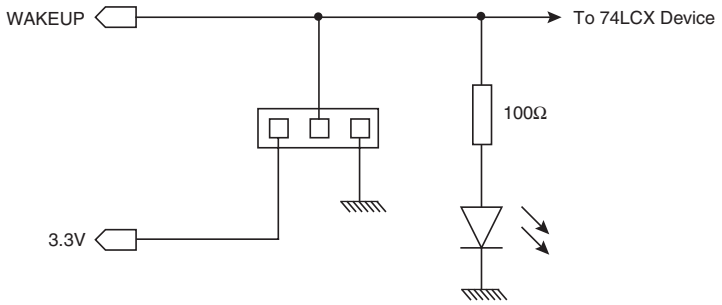


Figure 2-5. Manual Management of the WAKEUP Signal (as in a Demonstrator Design, for example)



3. ATmega128L 8-bit Microcontroller In-System Programmable Flash

On the AVR side, eight bidirectional I/O ports are provided but only four bits of one port are used for the interface between the 74LCX device and the AVR (for the WAKEUP, SPICLOCK, SPIDATA and SLE signals).

Because Port B provides the pins for the SPI channel, this is the port chosen for the four previously-mentioned signals:

- SPICLOCK: PB1 (SCK = SPI bus serial clock)
- SPIDATA: PB2 (MOSI = SPI bus Master Output/Slave Input)
- SLE: PB4 (OC0 = Output Compare and PWM Output for Timer/Counter0)
- WAKEUP: PB5 (OC1A = Output Compare and PWM Output A for Timer/Counter1)

The other pins PB0 ($\overline{SS}$), PB6 (OC1B) and PB7 (OC2/OC1C) can be left floating (open). Pin PB3 (MISO = SPI Bus Master Input/Slave Output) must be pulled up to 3.3V via a
1 KΩ resistor so as to be forced to a high level and not left open.

Pins SPICLOCK = PB1 and SPIDATA = PB2 need to be pulled down to ground via a 10 KΩ resistor to be forced to a low level (inhibition of the SPI during reset of the microcontroller).

Pin SLE = PB4 (OC0 = Output Compare and PWM Output for Timer/Counter0) must be pulled up to 3.3V via a 3.3 KΩ (or 1 KΩ if the power consumption is not critical) resistor in order to protect the line during reset of the microcontroller (during which phase the signal becomes an input).

Ports A and C of the AVR can be left floating (open) but must be internally configured with pull-ups.

For Port D, pins PD7, PD6, PD5 and PD4 can be left unused (open) but must be internally configured with pull-ups. Pins PD3, PD2, PD1 and PD0 have to be pulled up in order to inhibit the external interrupts.

For port E, pins PE3 and PE2 can be left unused (open) but must be internally configured with pull-ups. Pins PE7, PE6, PE5 and PE4 must be pulled up to 3.3V via a 3.3 KΩ (or 1 KΩ if the power consumption is not critical) resistor in order to inhibit the external interrupts.

PE1 and PE0 can be used as the Programming Data Output (TX) and Input (RX) to be connected to the TX and RX of the system (in the case of the AT84AD001-EB evaluation board, these signals are sent to the PC via an RS-232 port).

All the Port F pins must be connected to ground so that they are in a known fixed state (no internal pull-down is available for these pins).

All the Port G pins can be left floating (open).

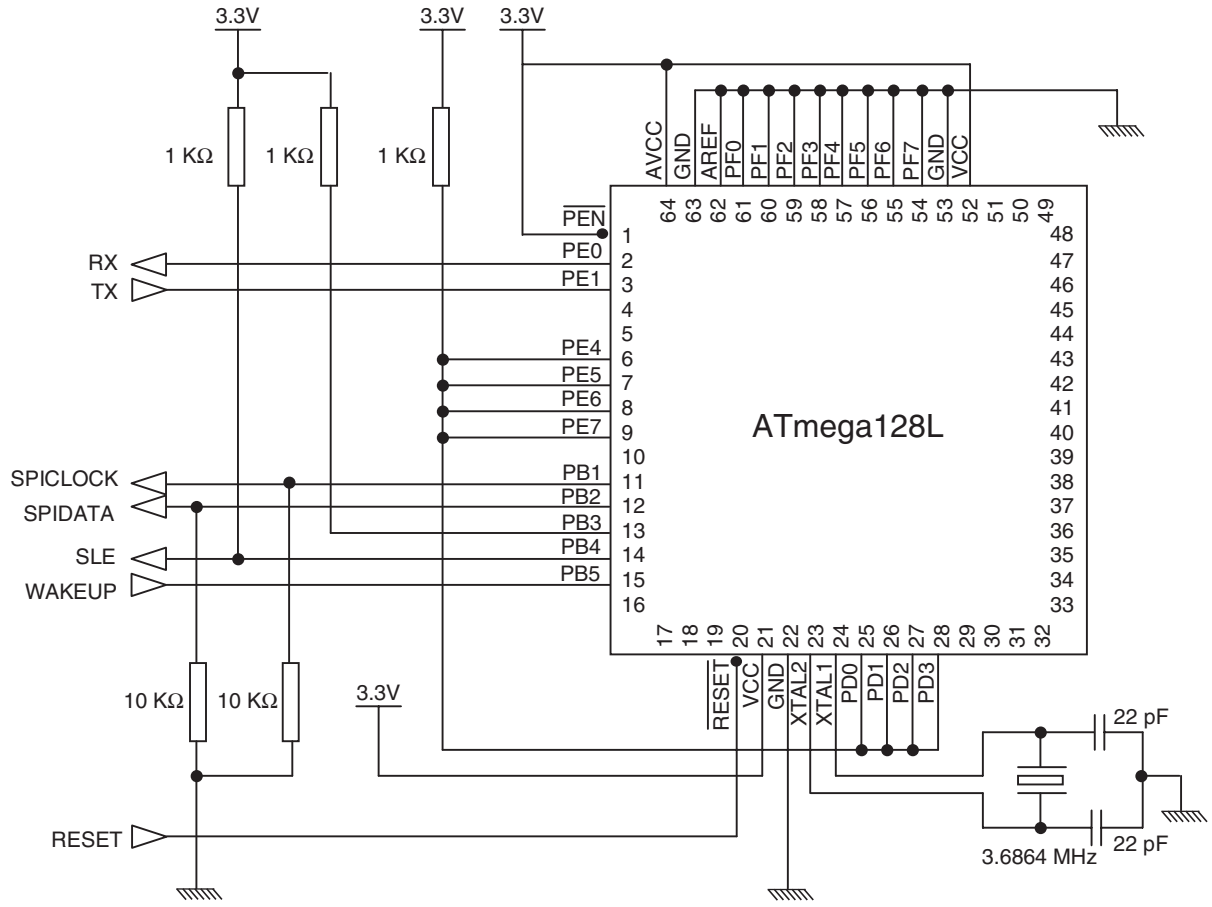
Finally, the five remaining signal pins are to be connected as follows:

- $\overline{PEN}$: Programming Enable pin for the SPI serial programming mode, to be connected to V_{CC}, set to 3.3V, to activate the SPI programming mode
- $\overline{RESET}$: master reset of the AVR, to be connected to a microcontroller supervisory circuit (for example and for information only: MCP809 from Microchip® – one possible configuration is given in the next section)
- XTAL1 and XTAL2: input and output to and from the inverting oscillator amplifier
- AREF: analog reference for the A/D internal converter

Finally, V_{CC} and AV_{CC} must be connected to a 3.3V source and GND must be connected to ground.

This gives the configuration depicted in Figure 3-1 on page 7 (AVR only).

Figure 3-1. ATmega128L Application Diagram (for Use with e2v's AT84AD001B Dual 8-bit 1 Gsps ADC)



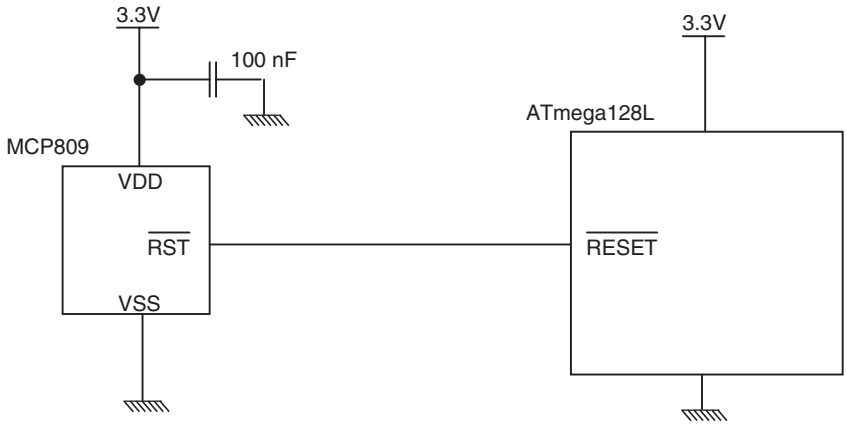
Note: Only the connected pins are shown (the unused pins are left open).

Reset of the ATmega128L AVR can be controlled through a *voltage supervisory circuit* comparable to the MCP809 device from Microchip® (for information only). Such a device allows you to keep the microcontroller in reset until the system voltage has reached its final level. It also ensures that the microcontroller is reset whenever a power drop occurs.

Any voltage supervisory circuit compliant with V_{CC} set to 3.3V and with a reset pulse longer than a 50 ns width (minimum/active low) would work.

In Figure 3-2 on page 8, the reset voltage level of MicroChip's supervisory device is set to 3V with a pulse of 350 ms.

Figure 3-2. Typical Application Diagram for the $\overline{\text{RESET}}$ Circuit



4. Programming ATmega128L AVR

ATmega128L AVR can be programmed through the AVR ISP (In-System Programmer) tool using AVR Studio®, Atmel's Integrated Development Environment (IDE) for code writing and debugging. The programming software can be controlled from both the Windows® environment and a DOS command line interface.

A 6- or 10-pin ISP connector is required to program the AVR.

For this application, an HE10 6-pin connector is used:

- Pin 1: Pdo, Avr Programming Data Out
- Pin 2: Avr Target Application Card Power Supply (3.3v)
- Pin 3: Sck, Avr Programming Clock
- Pin 4: Pdi, Avr Programming Data In
- Pin 5: Rst_isp, Avr Programming Reset
- Pin 6: Ground

Note: 1. The ISP card's power supply comes from the AVR card (3.3V). No additional power supply is required.
2. The AVR is programmed in serial mode.

The RST_ISP signal is used to set the AVR to programming or SPI mode.

This signal is sent to the AVR's $\overline{\text{RESET}}$ so that:

- When RST_ISP is set to 0, $\overline{\text{RESET}}$ = 0 also and the AVR is in reset (ISP) mode, PE0 is used as the Data In for programming of the AVR, PE1 is the Data Out and PB1 is the programming clock
- When RST_ISP is set to 1, $\overline{\text{RESET}}$ = 1 also and the AVR is in normal mode, PE0 = RX, PE1 = TX and PB1 = SPICLOCK

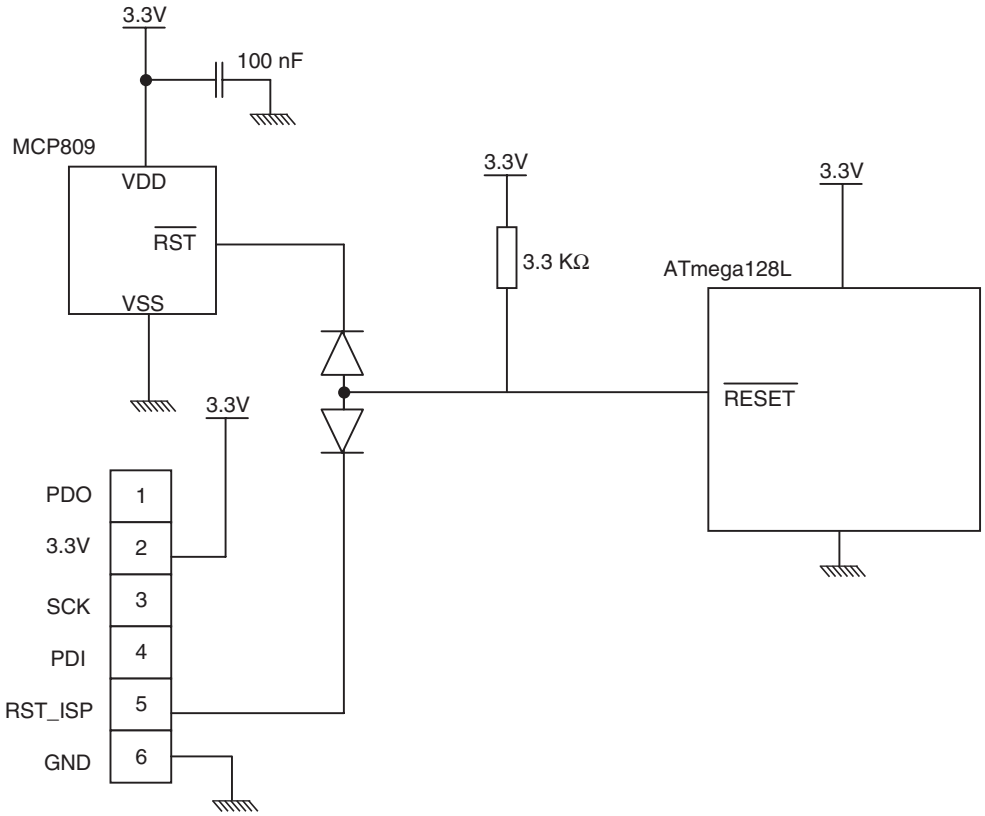
The three AVR signals mentioned previously (PE0, PE1 and PB1) therefore have two functions, both of which are controlled by the RST_ISP signal. Caution should be taken when implementing these signals, series resistors on the SCK, PDO and PDI data may be needed to manage possible conflicts, please see below and Figure 4-2 on page 10.

Similarly, the $\overline{\text{RESET}}$ signal has two possible sources:

- The Signal Generated By The Microcontroller Supervisory Device, And
- The Rst_isp Signal From The Isp

To control this signal and in a case where the microcontroller supervisory device is not configured with an open collector (ex. MCP809 device), two head-to-tail diodes are required, as illustrated in Figure 7. The line going to the $\overline{\text{RESET}}$ signal of the AVR is then in open-collector mode and a pull-up resistor (3.3 KΩ) to 3.3V is required.

Figure 4-1. Typical Application Diagram for the $\overline{\text{RESET}}$ Circuit with the ISP Connector

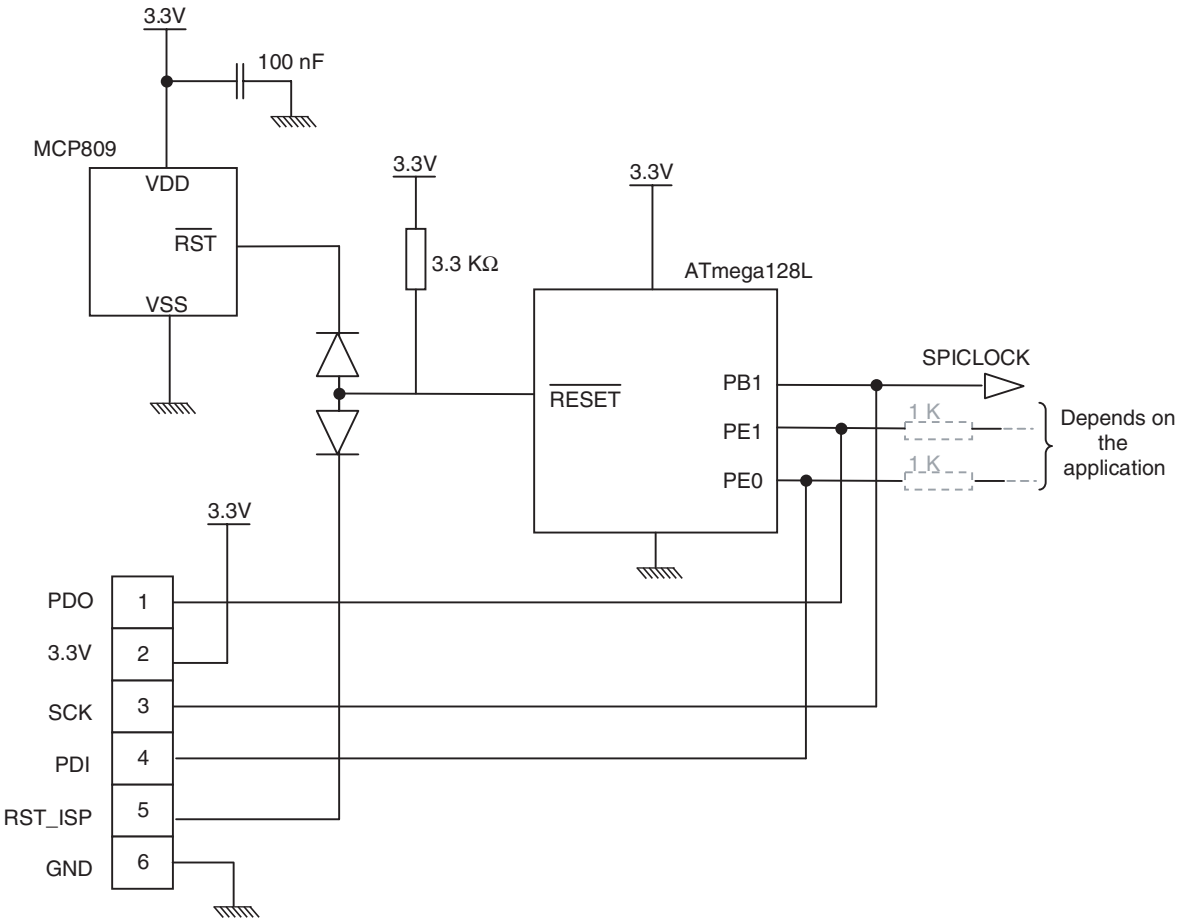


A basic diagram illustrating the interface between the ISP connector and the AVR is depicted in Figure 4-2 on page 10.

In this general case, PE1 and PE0 interconnections are left to the user's responsibility. If these signals conflict (for example PE1 is driven by both PDO and another signal), it may be necessary to add a 1 KΩ resistor in series so that any voltage difference is dissipated in this resistor.

No additional protection is required on the AVR PB1 signal if there is no conflict between SCK and SPICLOCK. It is nevertheless recommended the ADC be set to standby mode or the 3-wire serial interface be disabled by using the MODE bit during programming of the AVR.

Figure 4-2. General Application Diagram for the ISP Connector and the AVR



If the RX and TX signals are to be connected to a transceiver (RS-232 connector to a PC, for example), a low voltage buffer/line driver with a 3-state output device can be used to multiplex the AVR's signals (PE0, PE1 and PB1) between the ISP and the RX signals and between the TX and SPICLOCK signals. The 74LVQ241 devices are well-suited for this application (clock driver and bus-oriented transmitter or receiver).

The 74LVQ241 device has eight inputs and eight corresponding outputs and two 3-state output enable inputs. The latter (3-state output enable inputs) can be managed by the RST_ISP signal:

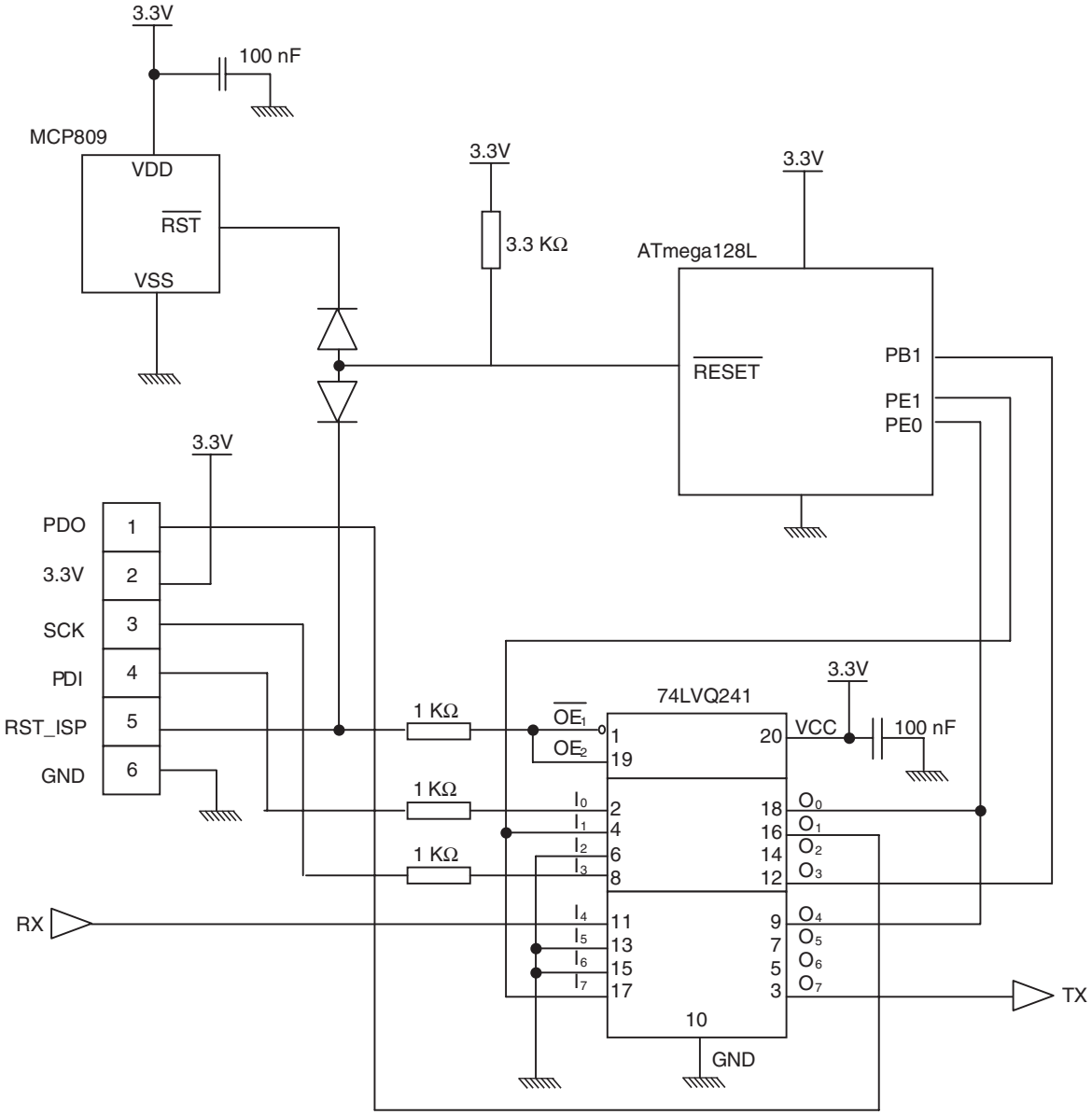
- When RST_ISP is set to 0, $\overline{OE1}$ and OE2 = 0 and O_0 to O_3 are low and O_4 to O_7 are in high impedance
- When RST_ISP is set to 1, $\overline{OE1}$ and OE2 = 1 and O_0 to O_3 are in high impedance and O_4 to O_7 are low

Table 4-1 is the truth table of the 74LVQ241 device:

Table 4-1. 74LVQ241 Truth Table

Inputs		**Outputs (O_0, O_1, O_2, O_3)**
OE1	I_n	
L	L	L
L	H	H
H	X	Z
Inputs		**Outputs (O_4, O_5, O_6, O_7)**
OE2	I_n	
L	X	Z
H	L	L
H	H	H

Figure 4-3. Typical Application Diagram for the 74LVQ241 Device



Note:
1. The unused inputs are connected to ground to prevent them from toggling.
2. $\overline{OE1}$ and OE2 are connected together and to RST_ISP via a 1 KΩ resistor.
3. SCK, RST-ISP and PDI are connected to I_3, $\overline{OE1}$ and OE2, and I_0 respectively via 1 KΩ resistors in order to manage possible conflicts on the signals when the connector is used to program several AVRs.
4. PE0 is connected to both O_0 and O_4, which are respectively the inputs for SCK and RX. PE0 is generated by either SCK or RX, depending on the mode.
5. PE1 is connected to both I_1 and I_7, which are the outputs for PDO and TX respectively. PE1 is generated by either PDO or TX, depending on the mode.

Programming of the AVR itself as well as the connections of the RX and TX signals are not described in this application note as they depend on the final application.

How to reach us

Home page: www.e2v.com

Sales Office:

Northern Europe

e2v ltd

106 Waterhouse Lane

Chelmsford

Essex CM1 2QU

England

Tel: +44 (0)1245 493493

Fax: +44 (0)1245 492492

E-Mail: enquiries@e2v.com

Southern Europe

e2v sas

16 Burospace

F-91572 Bièvres

Cedex

France

Tel: +33 (0) 1 60 19 55 00

Fax:+33 (0) 1 60 19 55 29

E-Mail: enquiries-fr@e2v.com

Germany and Austria

e2v gmbh

Industriestraße 29

82194 Gröbenzell

Germany

Tel: +49 (0) 842 410 570

Fax:: +49 (0) 842 284 547

E-Mail: enquiries-de@e2v.com

Americas

e2v inc.

4 Westchester Plaza

Elmsford

NY 10523-1482

USA

Tel: +1 (914) 592 6050

Fax:: +1 (914) 592-5148

E-Mail: enquiries-na@e2v.com

Asia Pacific

e2v ltd

11/F,

Onfem Tower,

29 Wyndham Street,Central,

Hong Kong

Tel: +852 3679 364 8/9

Fax: +852 3583 1084

E-Mail: enquiries-ap@e2v.com

Product Contact:

e2v

Avenue de Rochepleine

BP 123 - 38521 Saint-Egrève Cedex

France

Tel: +33 (0)4 76 58 30 00

Hotline:

hotline-bdc@e2v.com