



ED1021 – I/O Expander with UART interface & analog inputs

== Highlights ==

- 4.5V – 5.5V power supply range.
- 12 GPIOs.
- Up to 40mA maximum current in each output except GPIO8 (up to a total device current of 175mA).
- Most GPIOs can be an input to a 10bit ADC.
- Simple interface to most PC serial ports.
- Uses a common 4.000 MHz crystal.
- Easy to use UART interface at 38400 bps.
- One of a set of 4 GPIO pins can be used as an interrupt request (IRQ) pin.
- High stability internal ADC voltage reference for precise readings.
- Based on ATMEL ATtiny26® AVR®.

== Brief Functional Description ==

The ED1021 is an I/O expander fully controlled and configured through an UART interface. I/O expanders expand the number of pins available to a host system (typically, a microprocessor). Each pin can be configured either as an input, input with pull-up, input to an internal 8 or 10-bit ADC (10 of the 12 GPIOs), push-pull output or disconnected. When a change happens in a GPIO configured as digital input, ED1021 can automatically send a notification to the host system through the UART interface or an IRQ line. The IRQ line can be configured to be any of the GPIO8-11 pins.

Further details on specific hardware characteristics can be found on ATtiny26® datasheet in ATMEL's site <http://www.atmel.com>.

== Pin Description ==

GPIO0 to GPIO10

General-purpose I/O pins. They can be configured as input, input with pull-up, ADC input (GPIO0 to GPIO9 except GPIO3), push-pull output, or high Z (disconnected).

TX

Data transmit pin of the UART interface. Used to send data to the host system.

RX

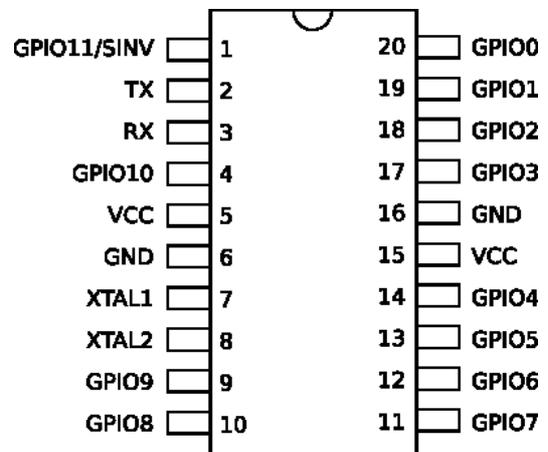
Data receive pin of the UART interface. Used to receive data from the host system.

GPIO11/SINV

If low at power-on, instead of high or floating, the UART signals will be inverted. Although not guaranteed, this typically allows an almost direct connection to most PC COMx serial ports. After power-on, the line works as any other GPIO except that no ADC is supported on this input.

XTAL1 & XTAL2

Connect a 4.000 MHz crystal and load capacitors to these pins, for clock. See below in this document for details.



GND & VCC

The chip's power input pins. Connect both GND pins and both VCC pins. Use one decoupling capacitor on each GND/VCC pair. VCC range is 4.5V to 5.5V.

GPIO8 can only be used to source / sink less than 100uA / 1mA.

= UART Interface =

All chip functions are controlled by registers, which are read/written through the UART interface, which runs at **38400 bps, no parity, 1 stop bit**. There are 3 possible UART transactions:

<i>Register Read</i>	
Host	01rr rrrr
ED1021	vvvv vvvv ... cccc cccc
<i>Register Write</i>	
Host	10rr rrrr vvvv vvvv ... cccc cccc
ED1021	
<i>Change Notification</i>	
ED1021	1000 tttt cccc cccc

The first byte of a read or write message is called the *command byte*, consisting of a 2-bit command and a 6-bit register index (rr rrrr). Then, in the case of a write, 1 or more bytes follow with the value. In the read case, the device answers with 1 or more bytes. “vvvv vvvv“ is a register value, 8-bit sized. An **optional** checksum follows “cccc cccc“, to improve reliability.

The *change notification* byte consists of a fixed 4 bits plus a 4-bit *type* field. The following table describes the currently available *type* values.

Type	Description
1111	There's a change in at least 1 digital input.

Transactions are processed sequentially, without overlapping. A new transaction can only be started after the previous one has finished. The device processes transactions one at a time. A new command can only be sent after the previous one has completed.

When enabled by setting to 0 bit CSDI in register CFG1, the device silently ignores any write with a wrong checksum. On the host system side, messages with wrong checksums should be ignored and the request repeated; if the host system is expecting notifications and receives an incomplete or unexpected byte, it should check the STATUS register for changes.

The checksum consists of 0xAA **xor**'ed with all bytes of the request/answer except the checksum itself; for example, the device returns 0xAA **xor** command-byte **xor** answer-byte when reading a configuration register. The change notification is a **xor** of 0xAA with the notification byte.

By default the checksums are enabled.

After the command byte is received, ED1021 implements a timeout of approximately **13.5 ms**

for each of the remaining bytes of the message. On timeout, the message being received is simply discarded and no further processing is performed. This prevents the device from getting stuck waiting for the rest of a message in case of host system failure.

= PC compatibility mode =

Although the RS-232 standard specifies a bus low voltage of -3..-25V and a bus high voltage of +3..+25V, most (if not all) typical PC serial COM ports will work when talking to TTL (0V/5V) devices without any problem.

We have found that even some USB<->COM converters output directly TTL levels and not the typical -12V / +12V.

Based on this fact, ED1021 can be connected to PC COM port through simple series protection resistors, allowing the control of the device directly from a PC. Connecting pin GPIO11/SINV directly to ground at power-up enables inversion of the UART signals. An example circuit is given in the typical applications section of the current document.

GPIO11/SINV is only sampled at power-up.

For the truly RS-232 compliant COM ports, or serious application usage, a protocol converter such as MAX232 is recommended between ED1021 and the host system, letting GPIO11/SINV floating or high at power-up in order to enable the TTL-mode UART.

= Notifications to the Host System =

The host can be notified of changes in **digital inputs** in 2 ways: an IRQ line or a notification message on the UART, consisting of 1 byte (2 when checksums enabled). Both ways can be used at the same time.

Setting field IRQPIN in register CFG0 to a GPIO will set that GPIO to work as the IRQ line and enables IRQ notifications. The line polarity (either active low or active high) is set in field TRIGMODE of the same CFG0 register. The line is kept active until the INTACK register is read. At that point, the line can be kept active if new changes occurred or it can be set inactive. By default IRQ notifications are disabled.

Setting bit UCNE on register CFG1 to 1 enables UART *change notification* messages. Notifications must be acknowledged by reading the INTACK

register, otherwise no new notification messages will ever be sent. Because a change notification can be confused with the answer to a register read, while UCNE is enabled the host should not perform read operations. It should first disable UCNE before a read is performed and re-enable it after; changes that occur meanwhile are not lost. Writes can be performed anytime without ambiguities.

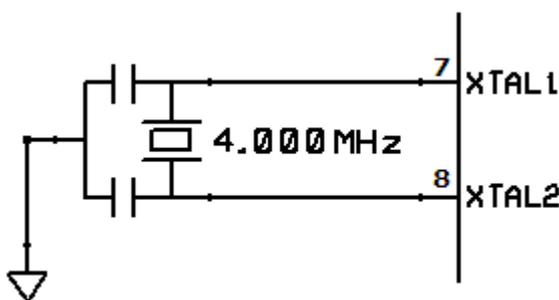
Input pins are sampled at least once every 1ms, when no UART communication is taking place. Transients smaller than this period may not be detected.

In both cases, reading the INTACK register should be the last operation after processing the change. If the state of the inputs changes before the host acknowledges the interrupt, a new interrupt will immediately raise after reading INTACK.

When both notification mechanisms are active, it's enough to perform only 1 read of INTACK.

== Clocking ==

A 4.000 MHz crystal and 2 load capacitors are used with a device's internal oscillator. The capacitors should be in the range 12-27pF, depending on your assembly and according to the crystal manufacturer's recommendations (typically around 22p).



== ADC ==

GPIO0 to GPIO2 and GPIO4 to GPIO9 can be configured to be the input to an internal ADC. Reading registers ADCx, where x is the GPIO number, causes a conversion to be performed. Writing to any ADCx register has an undefined behavior and should not be performed.

The ADC conversion is performed against a reference voltage (V_{REF}) of the device's VCC (the power-on default), or an internal voltage source

(V_{INT}) of 2.7V nominal if bit S2V7 in register CFG1 is set to 1.

The internal voltage source's stability is better than 4 mV over the temperature range and 8 mV over the VCC range (4.5V – 5.5V). Its absolute voltage has a variation of up to ±10% over manufacturing differences, but the exact value can be calculated in software, easily allowing the measurement of voltages with a maximum error smaller than ±0.5% of the input range. This calculation is performed reading register VINTC and using the value as a correction delta factor for the internal voltage. The real voltage source value is then calculated as

$$V_{INT} = VINTC \times 0.002 + 2.396$$

Any voltage V being measured at a device's pin with the ADC is then calculated as

$$V = (V_{INT} \times ADC) / 256 \quad (8\text{-bit mode})$$

$$V = (V_{INT} \times ADC) / 1024 \quad (10\text{-bit mode})$$

where ADC is the raw ADC value read from an ADC register.

Our suggestion to obtain a good V_{INT} stability is to use the calculated value and keep VCC constant, within ±200 mV (respecting the VCC range of 4.5V – 5.5V). Also, VCC pin 15 should have a series 10uH inductor before the decoupling capacitor, since this pin is the power feeding the ADC. Good layout techniques and grounding will also considerably help.

To measure voltages higher than the selected voltage reference, an appropriate resistor divider should be applied to the ADC input pin in order to scale the higher voltage range to the device's selected range. Resistors smaller than 10K in total should be used.

Conversions should not be performed on a pin that is configured as the IRQ. Doing so will result in undefined behavior.

== Power-On State ==

At power-on all pins are configured as high Z, therefore disconnected.

= Registers =

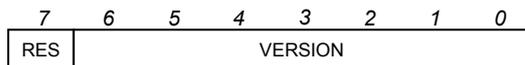
Register Overview

Index	Name	R/w	Description
0x00	VERSION	r	Chip version
0x01	STATUS	r	Current chip status
0x02	VINTC	r	V _{INT} correction delta factor
0x03	CFG0	r/w	Configuration register 0
0x04	CFG1	r/w	Configuration register 1
0x05	INTACK	r/w	Interrupt/notification acknowledge
0x08	DIR0	r/w	Set data direction for GPIOs 0-7
0x09	DIR1	r/w	Set data direction for GPIOs 0-7
0x0A	OC0	r/w	Set open collector for GPIOs 0-7
0x0B	OC1	r/w	Set open collector for GPIOs 8-11
0x0C	IN0	r	Read pin state for GPIOs 0-7
0x0D	IN1	r	Read pin state for GPIOs 8-11
0x0E	OUT0	r/w	Set pin state for GPIOs 0-7
0x0F	OUT1	r/w	Set pin state for GPIOs 8-11
0x20 ... 0x29	ADC0 ... ADC9	r	ADC read for each of the GPIOs

All bits defined as *reserved* or *don't care* should be set to 0 when writing. All register addresses not mentioned are *reserved* and should never be written or read; doing so will result in undefined behavior.

Detailed Description

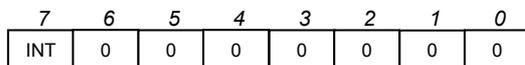
VERSION register, read-only, index 0x00



At reset: 0 x x x x x x x

This register contains the chip's version. Bit 7 is reserved.

STATUS register, read-only, index 0x01

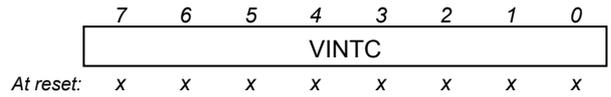


At reset: 0 0 0 0 0 0 0 0

INT [7] (interrupt request)

When an interrupt is raised, this bit is automatically set to 1. The device will keep the interrupt request line active until the interrupt is acknowledged (see register INTACK). This bit always behaves as if interrupts were enabled, even if they are not.

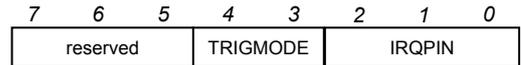
VINTC register, read-only, index 0x02



VINTC [7:0] (V_{INT} correction delta factor)

The value of this register allows correction of manufacturing deviations of the internal voltage reference (V_{INT}) used by the ADC. The corrected value is $V_{INTC} \times 0.002 + 2.396$.

CFG0 register, r/w, index 0x03



At reset: 0 0 0 0 0 0 0 0

TRIGMODE [4:3] (interrupt trigger mode)

Set the interrupt request trigger mode:

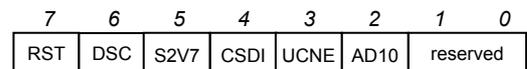
00	Active Low
01	Active high
10	Reserved
11	Reserved

Changing this field may cause the interrupt to rise.

IRQPIN [2:0] (IRQ pin)

Select a pin to function as an interrupt request (IRQ) line. If zero, none is selected and the feature is disabled. Set to 1 for GPIO8, 2 for GPIO9 and so on. This configuration overrides any other GPIO configuration. Changing this field may cause the interrupt to rise.

CFG1 register, r/w, index 0x04



At reset: 0 0 0 0 0 0 0 0

RST [7] (reset)

Setting it to 1 resets all registers to their default values (all pins disconnected). The bit automatically resets to 0. When writing to CFG1 with RST = 1 all other fields in the same write are ignored and set to their default values.

DSC [6] (disconnect)

Setting it to 1 disconnects all pins while retaining all configuration. Setting it back to 0 re-enables the current configuration. While DSC is 1, changing the pins configuration will not immediately take effect; the changes will only take effect when DSC is set back to 0.

0. While in disconnect mode the INx registers always show the true hardware state. If an IRQ pin is configured, it is also disconnected.

S2V7 [5] (Select 2.7V reference)

Setting it to 1 enables the nominal 2.7V internal voltage reference for use with the ADC. By default (S2V7 = 0) the reference is the device's VCC voltage.

CSDI [4] (Checksum Disable)

Setting it to 1 disables checksums. The write transaction that changes this bit should be performed according to the checksum mode in use before the write.

UCNE [3] (UART-Change-Notification enable)

Setting it to 1 enables change notifications through the UART. After the device sends a change notification, no further notifications are sent until the notification is acknowledged, by reading register INTACK.

AD10 [2] (ADC 10-bit mode enable)

Setting it to 1 enables 10-bit ADC conversions. When set to zero (default), conversions are only 8-bit.

INTACK register, r/w, index 0x05

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

At reset: 0 0 0 0 0 0 0 0

This register needs to be read from in order for an active interrupt to be cleared and for new change notification messages to be sent. The INT bit on the STATUS register is cleared and the interrupt request line is de-asserted, unless new changes occurred meanwhile.

DIR0 register, r/w, index 0x08

7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

At reset: 0 0 0 0 0 0 0 0

GPIOx [7:0] (GPIOx input or output select)
Independently set each GPIO to either input (0) or output (1).

DIR1 register, r/w, index 0x09

7	6	5	4	3	2	1	0
0	0	0	0	GPIO11	GPIO10	GPIO9	GPIO8

At reset: 0 0 0 0 0 0 0 0

GPIOx [3:0] (GPIOx input or output select)
Similar to DIR0, but for GPIOs 8 to 11.

IN0 register, r, index 0x0C

7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

At reset: 0 0 0 0 0 0 0 0

GPIOx [7:0] (current GPIOx state)
This register always reflects the current state of the pins, even if a pin is set to output.

IN1 register, r, index 0x0D

7	6	5	4	3	2	1	0
0	0	0	0	GPIO11	GPIO10	GPIO9	GPIO8

At reset: 0 0 0 0 0 0 0 0

GPIOx [3:0] (GPIOx pull-up enable)
Similar to IN0, but for GPIOs 8 to 11.

OUT0 register, r/w, index 0x0E

7	6	5	4	3	2	1	0
GPIO7	GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

At reset: 0 0 0 0 0 0 0 0

GPIOx [7:0] (GPIOx output or pull-up state)
When a pin direction is set to output (DIR = 1), this register is used to set its state. When the direction is input, setting a bit in this register to 1 enables the internal pull-up resistor.

OUT1 register, r/w, index 0x0F

7	6	5	4	3	2	1	0
0	0	0	0	GPIO11	GPIO10	GPIO9	GPIO8

At reset: 0 0 0 0 0 0 0 0

GPIOx [3:0] (GPIOx output or pull-up state)
Similar to OUT0, but for GPIOs 8 to 11.

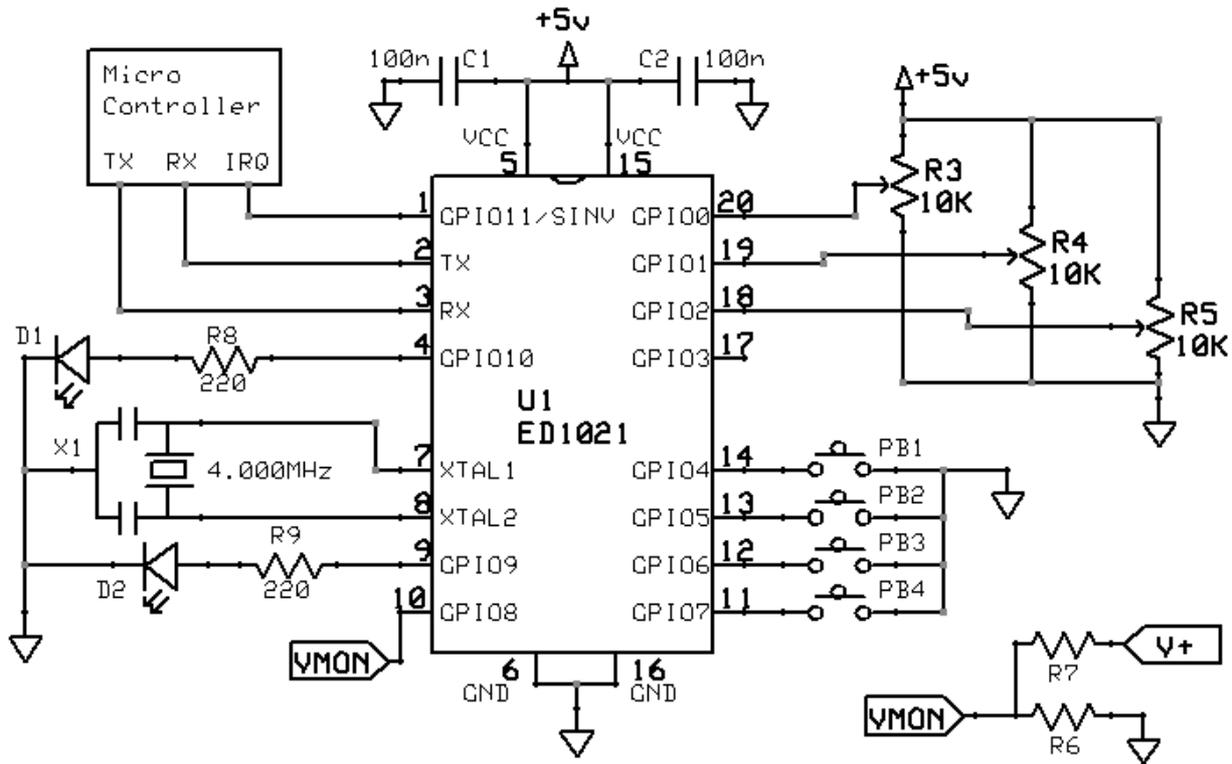
ADC0..9 registers, r/w, index 0x20..0x29

7	6	5	4	3	2	1	0
ADC							

At reset: 0 0 0 0 0 0 0 0

ADC [7:0] (ADC value and control)
Each register is associated with a GPIO pin, for the 1st 10 GPIOs, by number. Reading from these registers cause a conversion to be performed and the result to be returned. A conversion takes priority over **any** other configuration for the pin, during the conversion period. The pin is forced as an ADC input and after conversion the original pin's configuration is restored. If AD10 is set on CFG1, 2 bytes are returned, MSB 1st.

= Typical Applications =



Crystal capacitors are 22pF

ED1021 Example Application
<http://EmbeddedDreams.Com>

Circuit A – Microprocessor I/O expander

= Circuit A =

An ED1021 is used to expand the I/O capabilities and add new peripherals to a microprocessor core block. The microprocessor has now the ability to control 2 extra Leds, 4 push buttons (PB1-PB4), 3 analog dials (R3-R5) and monitor an external voltage level higher than ED1021's VCC (through resistor divider R6/R7). One extra pin is still free for future use and the interrupt feature is being used to notify the microprocessor of changes in the push buttons.

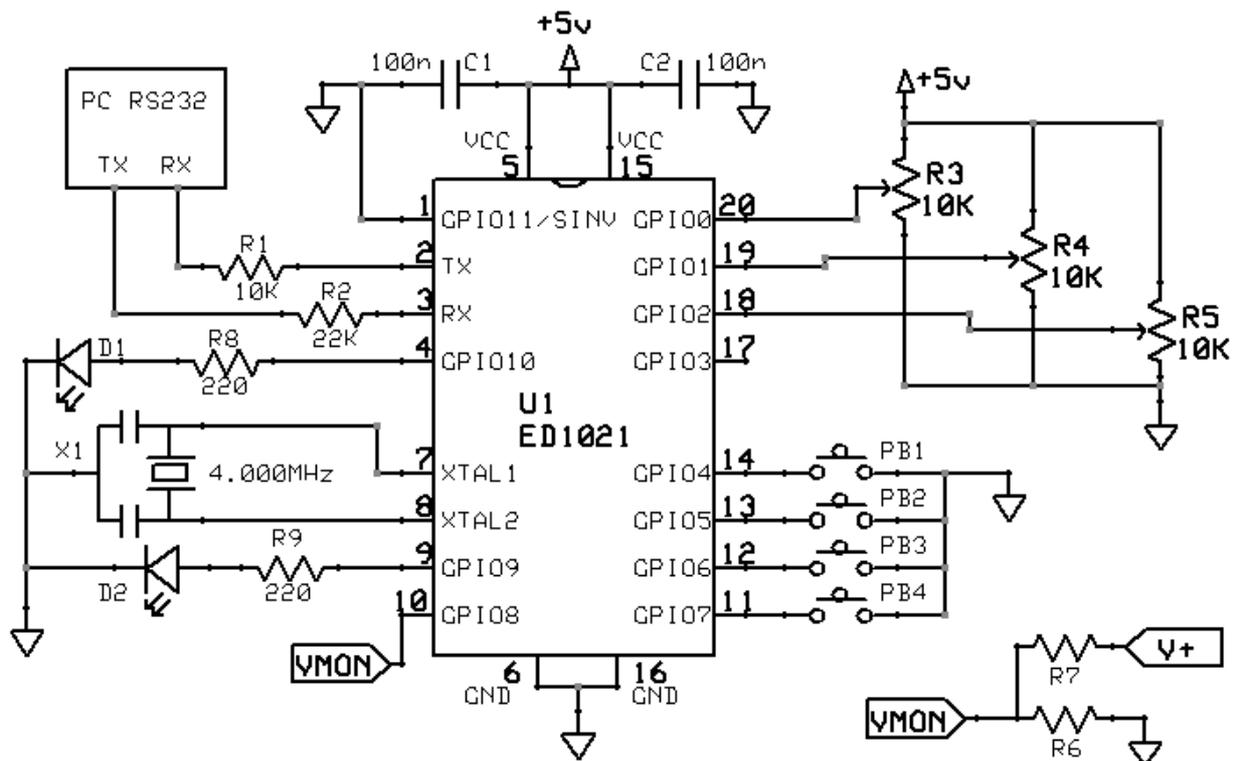
The Led GPIOs are configured as outputs (DIR1[2:1] = 11b); the push buttons as inputs with pull-ups enabled (DIR0[7:4] = 0000b, OUT0[7:4] = 1111b). GPIO0 to GPIO4 are configured as ADC inputs. GPIO11 was enabled in register INTCFG, field IRQPIN, to work as an interrupt request line. GPIO3 is left unused.

C1 and C2 should be as close as possible to the VCC/GND pin pairs of the device. This is recommended for all applications that make use of the ADC.

= Circuit B =

This circuit is similar to circuit A, except that pin SINV is grounded and therefore the UART signals will be inverted which allows an almost direct connection to a typical PC serial port (COMx port). Most PC serial ports and USB-to-Serial converters will tolerate the lower voltage levels provided by ED1021. Some USB-to-Serial converters even use 5V TTL signals, which make them just perfect for interfacing with this I/O expander. R1 and R2, together with the chip's internal diodes, protect the device from voltages outside the range 0..5V. However, for this protection to be effective, the device **must** be powered up before connecting it to the RS-232 master (like a PC's serial port).

Correct inter-work with all PC serial ports is not, however, guaranteed, since it works outside of RS-232 specification. If you need to make sure the connection works and in all situations, use a RS-232/TTL converter such as the well-known MAX232, and don't ground SINV at power-up.



Crystal capacitors are 22pF

ED1021 Example Application
<http://EmbeddedDreams.Com>

Circuit B – PC serial I/O Expander

= Revision History =

16-01-2008: Revision A. Initial draft.

27-03-2008: Revision B. Production ready.

17-04-2008: Revision C. Corrected internal voltage reference value and tolerance according to tests.

24-05-2008: Revision D. Added command to return internal voltage reference deviation from nominal.

Improved the internal voltage source description and ADC usage.

27-05-2008: Revision E. Added possibility to use the ADC in 10-bit mode and changed text accordingly.

Added reference to base micro-controller.

16-06-2008: Revision F. Updated disclaimer. Clarified some aspects with feedback from reviews.

25-10-2008: Revision G. Corrected bugs in the example applications and changed the text accordingly. Added more info on GPIO8's current capability.

28-12-2008: Revision H. Added info about correct handling when connecting to an RS-232 master device.

03-04-2009: Revision I. Changed Vcc range to the correct values supported by ATtiny26. Improved information on crystal capacitors.

*Copyright © Embedded Dreams 2007-2008. All rights reserved.
Further information, software and support at <http://www.EmbeddedDreams.com/>.*

Disclaimer

This document has the only and sole purpose of describing the features, functionality and usage of the product it refers to. Except as set forth in Embedded Dreams's terms and conditions of sale located in Embedded Dreams's web site, no warranty is given concerning its merchantability or fitness for a particular purpose. Embedded Dreams will in no event be liable for any kind of direct or indirect problem or issue arising from the usage of this document, the information contained in this document nor the product it describes.

Both this document and the product it describes may be changed, in any aspect and at any time, without any notice. Embedded Dreams tries really hard to keep the information in this document correct, but does not make any commitment to its accuracy nor to update the information in it.

Embedded Dreams products are not intended, authorized, or warranted for use in applications: intended to support or sustain life; military applications; applications to be used in any context that violates Human or Animal rights.

ATtiny26® and AVR® are trademarks of Atmel Corporation.