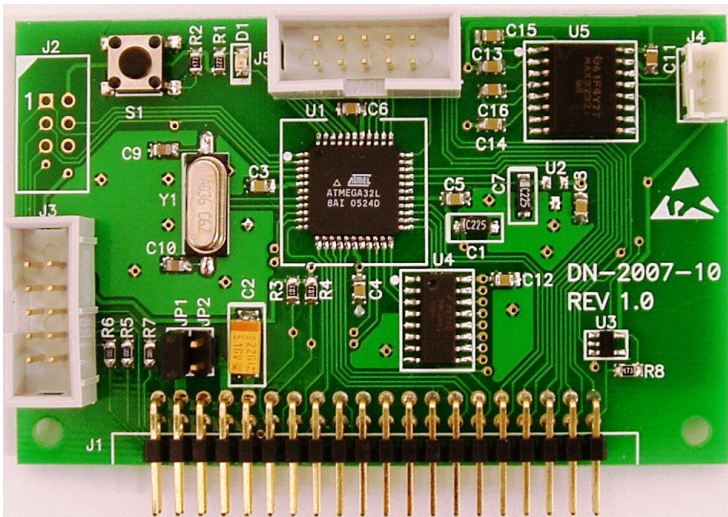


# MICRO-BUS ATMEL AVR Mega32 CPU board

Design Number: DN-2007-10



- ATmega32L CPU
- 32K Flash/2K SRAM
- 1K EEPROM
- 3.3V/5V
- ISP programmable
- RS232 port
- Status LED
- Reset switch
- Analog ADC port

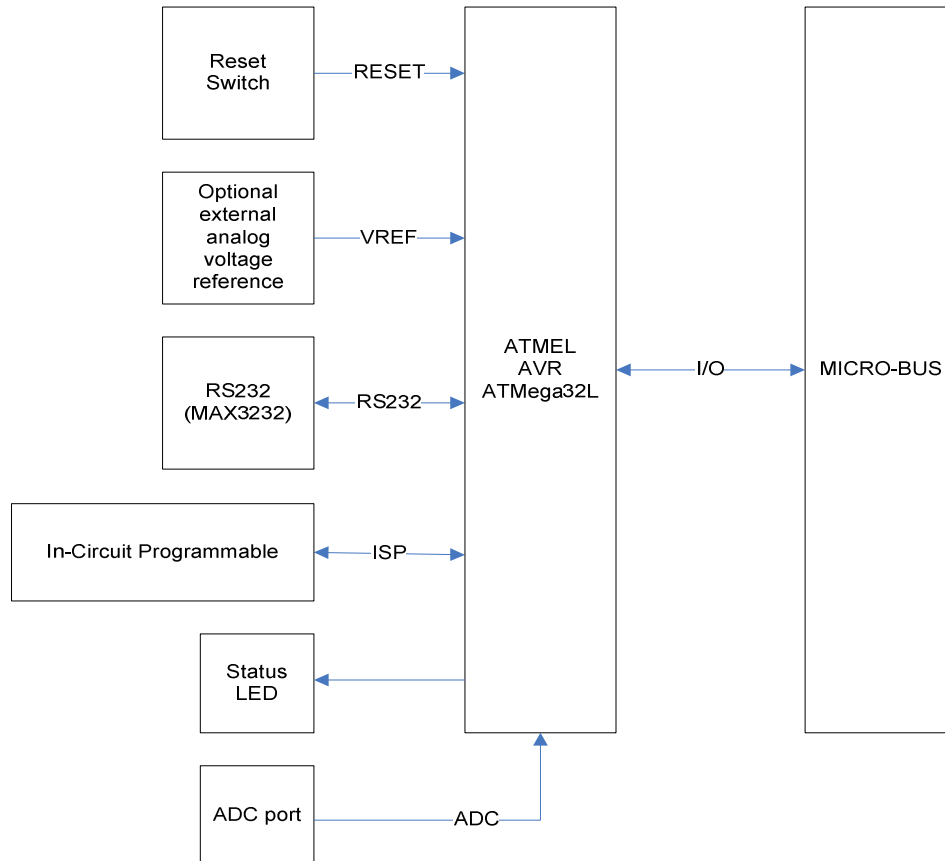


Figure 1 – Functional Block Diagram

# Technical Description, Schematics and Notes

A standard 3.6864MHz crystal is used. This still allows BAUD rates up to 230.4k. If more speed is required, change the crystal to 7.3728MHz if BAUD rate accuracy is required or else go all the way to 8MHz. Since this is the L version of the device 8MHz is the limit.

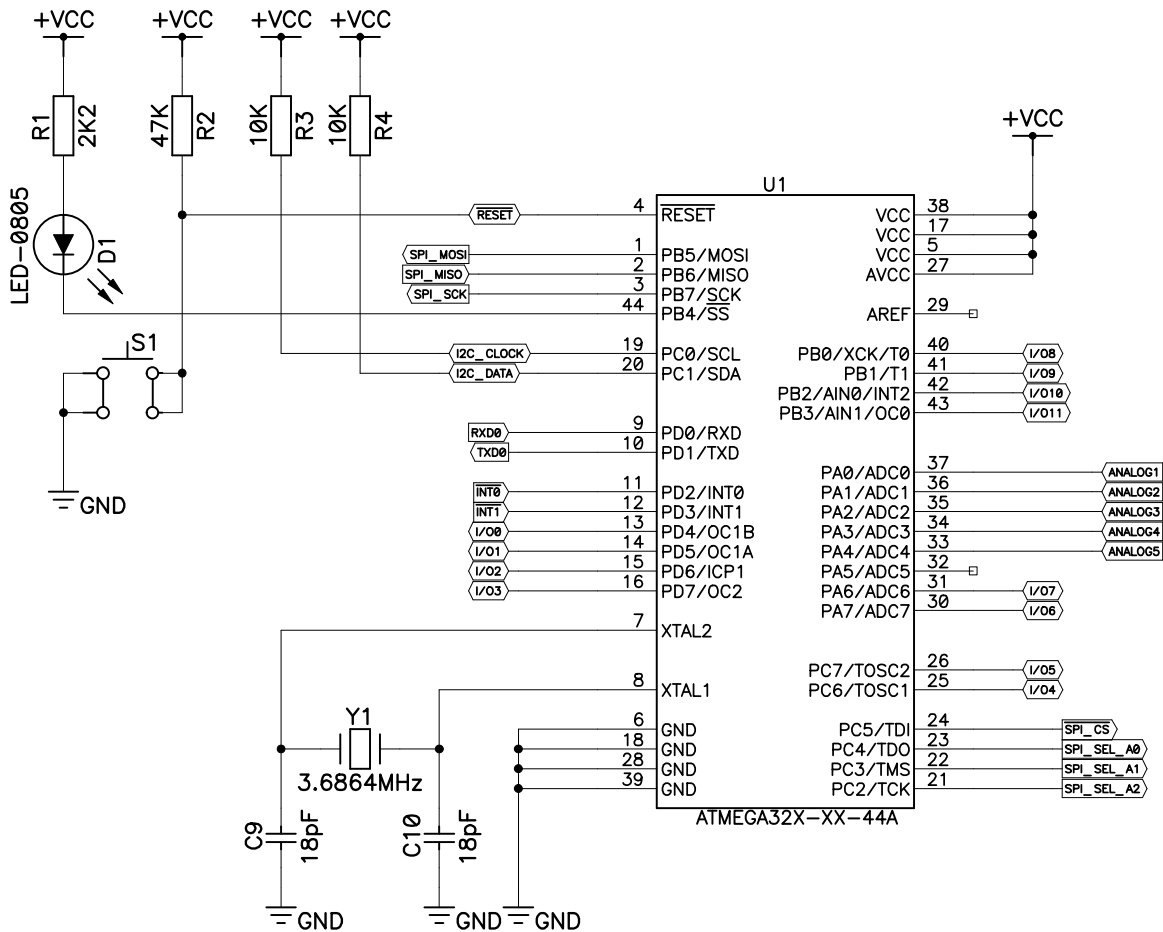


Figure 2 – ATmega32 schematic

## USART interface options

The serial port is wired to a MAX3232 RS232 line driver. This driver will operate from both 3.3V and 5V power supplies using 0.1uF capacitors. An additional logic gate (AND function) allows the CPU to receive data from both bus and RS232 ports. Should data however arrive simultaneously it will be corrupted. See figure 3 on the next page.

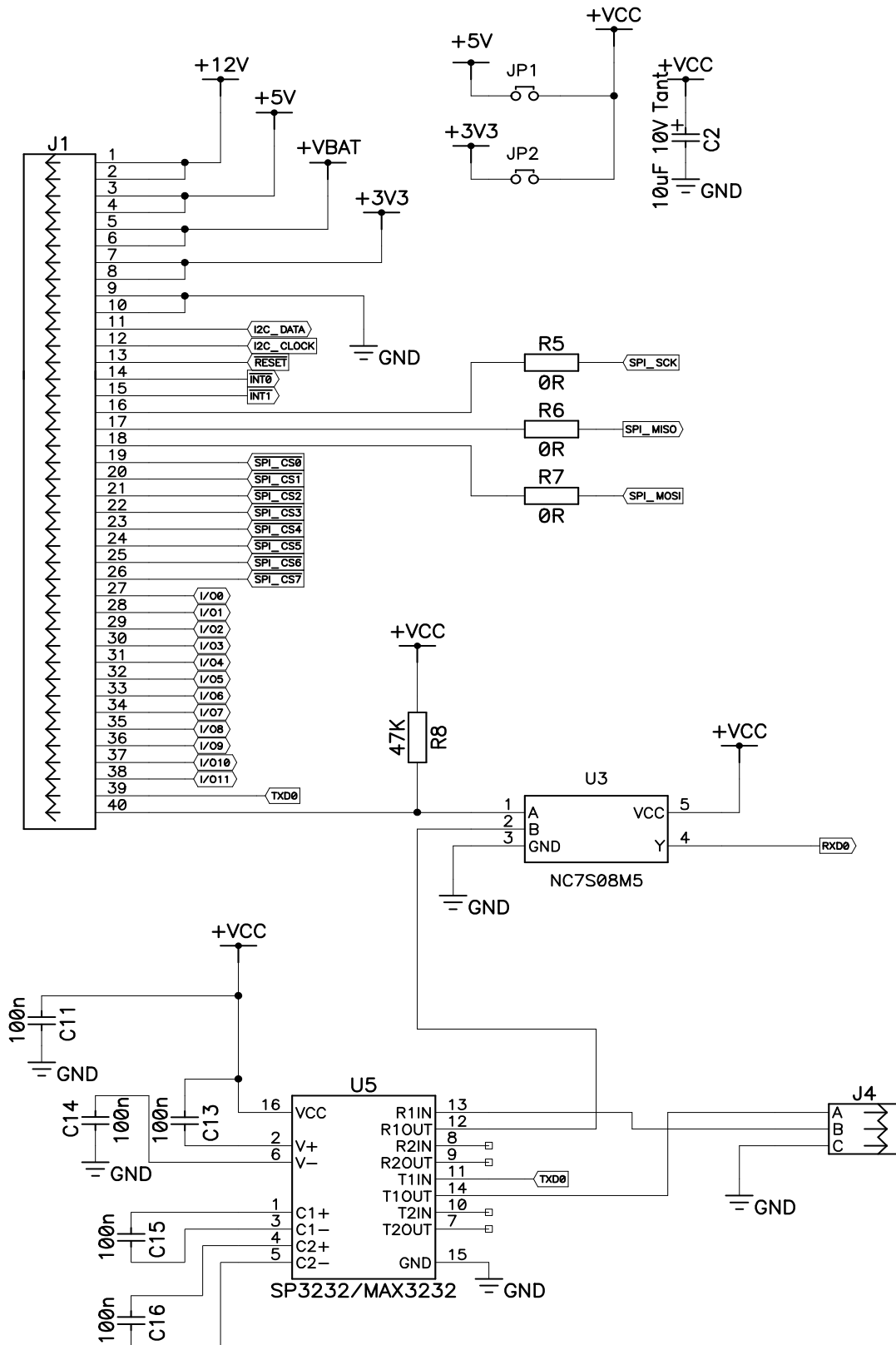


Figure 3 RS232 and Micro-bus Interface

## SPI bus chip select expansion

The SPI bus can select a total of 8 devices using a 3-8 line decoder as shown in figure 4 below. To select none of eight simply hold the #SPI\_CS pin (#E1) high and all chip selects go high.

**A cautionary note, if the CPU is stopped and programmed with an active chip select and a device is associated with that chip select then the CPU may fail to program as the active device is interfering with the ISP programmer. This is a temporary state and can be avoided by making sure that the chip select lines are always returned to the inactive state in software. This does not however guarantee that this situation will not occur again as the CPU could be stopped in the middle of SPI communication, but it does reduce the occurrence of the problem. Simply resetting the CPU and re-programming often resolves the problem. If all else fails then unplug the module(s) that are affecting the ISP programmer. (This however is a pain. Is the CPU reset active while programming? Maybe reset could be used to drive another enable pin on the 3-8 line decoder and force the chip selects to the inactive state while programming. We will consider this for the next revision.)**

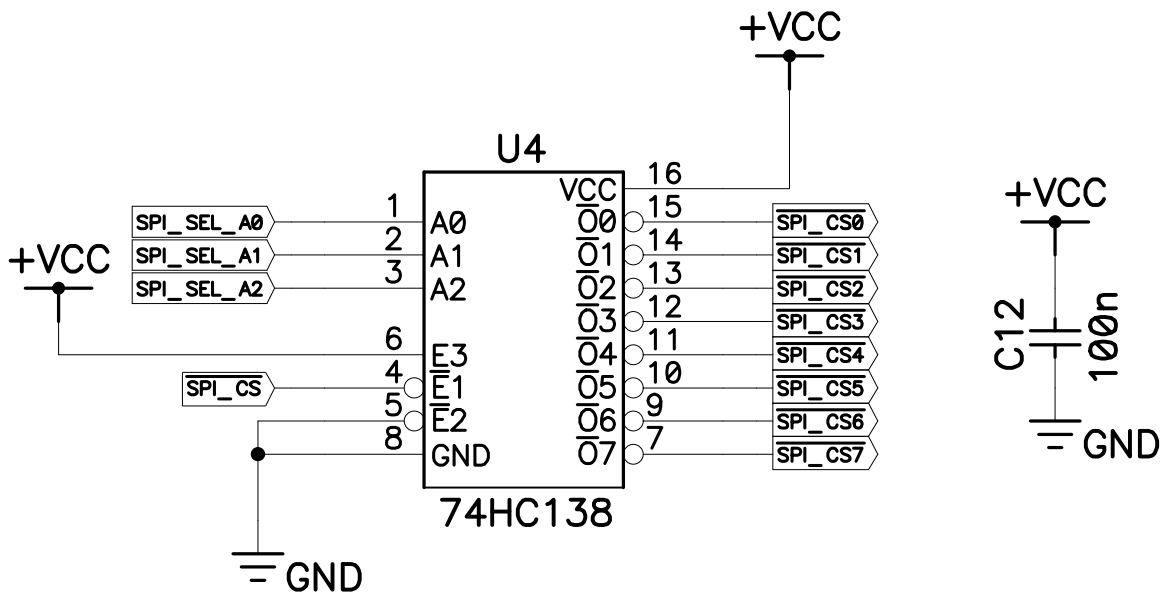


Figure 4 – SPI bus chip select expansion

## SPI bus device selection

---

SPI_SEL_A2	SPI_SEL_A1	SPI_SEL_A0	SPI_CS	SPI device selected
0	0	0	0	SPI_DEVICE_0
0	0	1	0	SPI_DEVICE_1
0	1	0	0	SPI_DEVICE_2
0	1	1	0	SPI_DEVICE_3
1	0	0	0	SPI_DEVICE_4
1	0	1	0	SPI_DEVICE_5
1	1	0	0	SPI_DEVICE_6
1	1	1	0	SPI_DEVICE_7
X	X	X	1	SPI_DEVICE_NONE

X = don't care

## Jumper Settings

---

Jumper	Description
JP1	VCC = +5V
JP2	VCC = +3.3V

## CPU I/O to bus mapping

---

BUS PIN	CPU PIN	CPU I/O	Description
11	20	PC1	I <sup>2</sup> C DATA
12	19	PC0	I <sup>2</sup> C CLOCK
14	11	PD0	Active LO interrupt 0
15	12	PD1	Active LO interrupt 1
16	3	PB7	SPI_CLOCK
17	2	PB6	SPI_DATA_IN(MISO)
18	1	PB5	SPI_DATA_OUT(MOSI)
19			SPI_CS0 (see SPI Device selection)
20			SPI_CS1
21			SPI_CS2
22			SPI_CS3
23			SPI_CS4
24			SPI_CS5
25			SPI_CS6
26			SPI_CS7
27	13	PD4	I/O0 (general purpose I/O)
28	14	PD5	I/O1
29	15	PD6	I/O2
30	16	PD7	I/O3
31	25	PC6	I/O4
32	26	PC7	I/O5
33	30	PA7	I/O6
34	31	PA6	I/O7
35	40	PB0	I/O8
36	41	PB1	I/O9
37	42	PB2	I/O10
38	43	PB3	I/O11
39	10	PD1	SERIAL 5V TX
40	9	PD0	SERIAL 5V RX

## Other I/O to bus mapping

---

BUS PIN	CPU PIN	CPU I/O	Description
13	4		RESET

## Analog Voltage Reference

---

The ATmega32 has an on board 2.56V voltage reference. Allowance for an external reference has been made using a Microchip MCP1525T 2.5V reference. This reference is not fitted by default and may be fitted by the user.

## Analog I/O

---

A conscious decision was made not to pass analog signals down the bus for several reasons,

1. Signal noise pickup from adjacent bus traces and potentially long traces.
2. The promotion of I2C and SPI analog-to-digital converters.

In very simple applications 1 above may not be a problem. 2 above can be a problem in some applications as it adds cost and complexity.

Since almost all modern micro-controllers offer ADC ports a separate connector was added to accommodate some of the ADC ports. See figure 5. There are five ADC ports available on J5 interleaved with ground. This port could connect to another Micro-bus module or a separate signal conditioning board. Care should be taken to avoid ground loops when using this port.

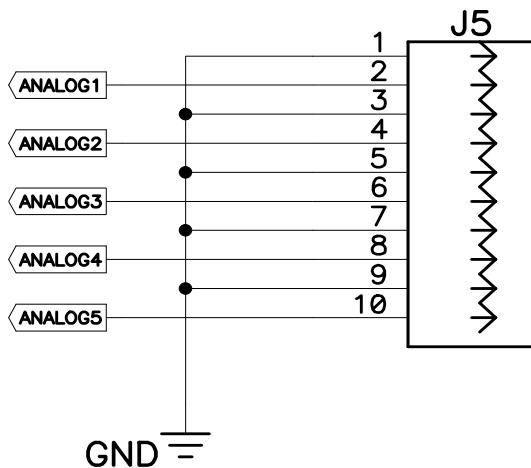


Figure 5 – ADC port

## **ISP programming**

---

In-Circuit programming is done directly from AVR studio software. This module supports both the 10-pin and 6-pin programming header. We have at this time only used and tested the 10-pin header which is fitted by default.