ODENSE UNIVERSITY COLLEGE OF ENGINEERING

ATMEGA128 ADAPTER USERS GUIDE

Anders Stengaard Sørensen

February 13, 2005

<Photograph missing>

Preliminary documentation:
Although this users guide for the ATmega128 adapter board is not yet complete, it is sufficient to aid students with some electronics experience in assembling and using the adapter board.
This guide contains all relevant technical documentation, but is not yet complete with respect to descriptions, illustrations, reference designs, references etc. Likewise, it has not yet been checked for spelling errors etc.
Please check with `http://www.stengaard.net/anders-s` if a newer version has become available. If you discover errors, or have ideas or suggestions for the content of this document, please send an e-mail to the author: anders-s@stengaard.net

Anders Stengaard Sørensen

Contents

Copyright notice

About the HOPE projects

Hands **O**n **P**rogrammable **E**lectronics — or HOPE , is a series of projects, aimed at promoting the use of programmable electronic components in research, development and students projects, related to Odense University College of Engineering.

While it is good educational practice to teach classical electronic design, based on discrete components and simple integrated circuits, it is also necessary to enable students to gain practical experience with the highly flexible and complicated devices used in practical electronics today.

As I began teaching in 2003, I was surprised to see the complex circuits students were designing with 74.. and 40.. type IC's, to realize registers, counters, decoders and other small digital systems, that could be realized much easier (and cheaper) in a Programmable Logic Device (PLD) or even in a micro controller. I was even more surprised to learn that most of the students had actually followed courses in PLD's and micro controllers, but thought it too abstract or troublesome to transfer their experience with PLD or micro controller demonstration systems to a practical design in its own contexts.

In order to reduce the *entry barrier* towards programmable electronics, I have initiated a number of small projects, resulting in a series of tools, that should make it easier to begin working with selected PLD's, micro controllers etc. I have launched these projects under the common title *Hands On Programmable Electronics*, with subtle reference to the first commandment of the Hacker Ethic:

> *Access to computers — and anything which might teach you something about the way the world works — should be unlimited and total. Always yield to the Hands-On Imperative!* *(MIT students* ~ 1960*)*

It is my HOPE that the tools provided by the HOPE projects will result in increased use of CPLD's, micro controllers, FPGA's, FPAA's and other programmable electronics in students projects, as well as R&D projects in corporation with Odense University College of engineering.

Anders Stengaard Sørensen — 2004

1 Introduction

The ATmega128 adapter described here, will enable you to use the ATmega128 microcontroller with simple (prototyping) Printed Circuit Boards (PCB's), without having to go into the design considerations asociated with Surface Mounted Design (SMD) components.

The adapter simply converts an ATmega128 microcontroller to a component with ordinary pins with the standard 100 mils (2.54mm) spacing used by most classic components. At the same time, the adapter provides decoupling, a quarts oscilator and convenient programming interface for the microcontroller.

The ATmega128 adapter is well suited for rapid prototyping and test circuits, and have been used to great advantage by many students in various projects at Odense University College of engineering, since August 2004.

2 Theory of operation

The ATmega128 adapter is meant to make it possible for students — and others — to use the ATmega128 microcontroller without designing a PCB. It can also be used by less experienced PCB designers that are reluctant to include SMD IC's with high pin density in their design.

In short, the adapter is meant to be used as an 80 pin component. The adapter is menat to be placed on a prototyping PCB, or possibly on a PCB of your own design. In both cases, I will refer to the PCB that holds the adapter as the *host PCB*.

The functionality of the adapter can be broken down into 5 main areas:

- Pin mapping
- Power supply decoupling
- Programming interface
- Quartz oscilator
- RESET pullup

2.1 Pin mapping

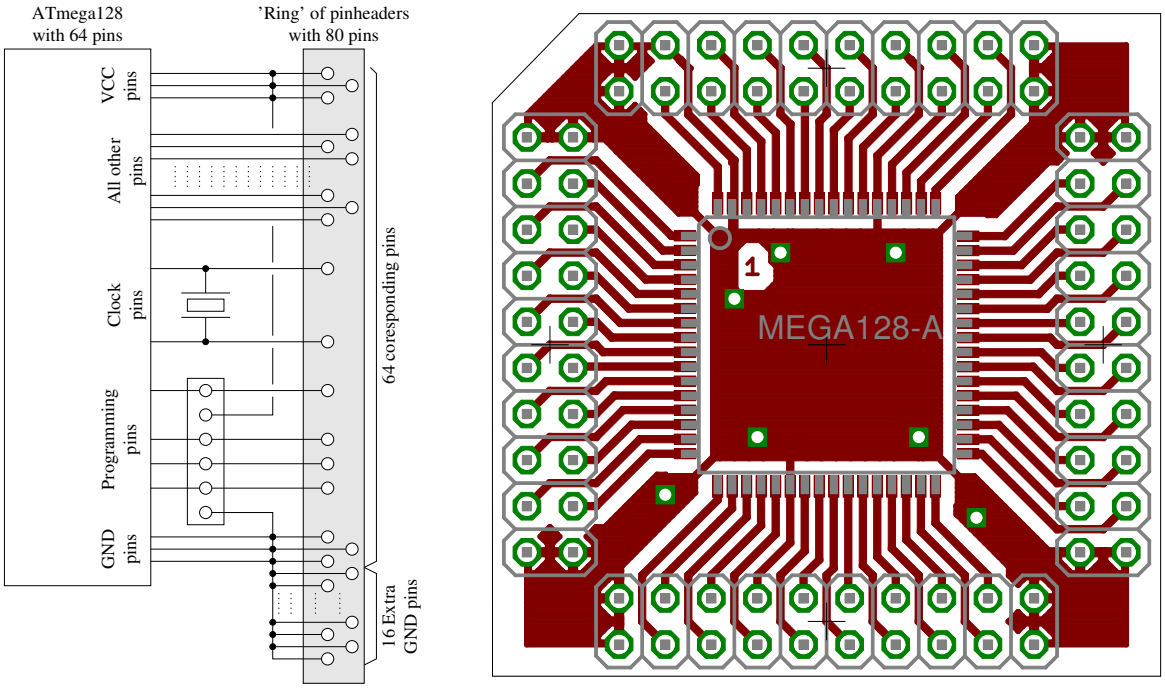
The main point of the ATmega128 adapter is to provide the 64 pin microcontroller SMD IC with a new footprint, supporting through-hole and socket mounting with the classic 100 mils (2.54 mm) spacing.

The actual pin-mapping is implemented on the *microcontroller side* of the PCB, making the PCB layout the best illustration of how the microcontroller pins are mapped to 4 double-row pinheaders placed around the microcontroller, as shown in figure 1-b.

As the figure clearly show, **each of the microcontrollers 64 pins are connected to a corespond-ing header pin.** In addition to these pins, the 4 outer pins of each of the 4 pinheaders provide an additional connection to the reference volatge of the microcontroller, also known as GND or 0V. Note that the microcontrollers GND- as well as VCC-pins are connected to each other on the PCB. This includes a connection between VCC and AVCC.

In some cases, the supply voltage to the analog parts of the microcontroller (AVCC), should be different from the digital supply voltage VCC. This can be achieved by removing the 0Ω resistor R1 (if it was mounted) and breaking the PCB wire located directly beneath it. If you would like to reconnect AVCC and VCC at a later time, simply mount the 0Ω resistor R1.

A quick and easy users guide to the pin mapping is provided in appendix E



(a) Sketch of pin mapping

(b) Microcontroller side of PCB

Figure 1: How the 64 IC pins map to 80 pinheader pins

2.2 Power supply decoupling and indicator

In order to provide adequate decoupling of the ATmega128's 3 power supply pins, the adapter board features a 100nF decoupling capacitor for each of the 3 VCC pins on the microcontroller. The on-board capacitors make it unnecessary for the user to suply HF decoupling external to the adapter board. Although the on-board capacitors provide HF decoupling, I still recommend LF decoupling, e.g. a 33-100μF tantalium or electrolytic capacitor, placed on the host board, near the adapter.

In parallel with the HF decoupling, the adapter is also equiped with a LED in series with a current limiting resistor. The LED indicate that the adapter is supplied with power.

2.3 Programming interface

As indicated in figure 1-a, the programming interface is merely a pinheader with 6 pins, that provide easy access to the signals used when programming the ATmega128 with an AVR-ISP programmer. The pinheader is completely transparent for the microcontroller, and can be left unused or omitted if you prefer to acess the programming signals from the host board.

2.4 Quartz oscilator

The clock frequency of the ATmega128 can be controlled in many different ways. One of the most popular is to connect the two oscilator pins to a quartz crystal, that acts as frequency controlling component for the ATmega's internal oscilator circuit.

I have designed the adapter with such a crystal on-board, as the wires between crystal and microcontroller must be kept as short as possible. If you want to use another way of controlling the clock frequency, e.g. an external TTL clock signal, the crystal and decoupling capacitors can be omitted.

Each pin of the crystal is connected to a small decoupling capacitor, that ensure that the crystal will work on the intended harmonic frequency. Although I recommend mounting these capacitors, many users have reported the adapter to work well without them.

2.5 RESET pullup

The adapter feature a pullup resistor mounted between VCC and the RESET pin of the microcontroller. The resistor keep the RESET pin logic high (not reset), if no external components are driving the RESET pin.

3 Assembling ATmega128 adapter

The directions given below apply to **version 1.1** of the PCB, in all existing revisions. At the time of writing, only revision 1 have been produced, but any future revisions to version 1.1 will only include minor changes, with no or minimal impact on the assembly procedure.

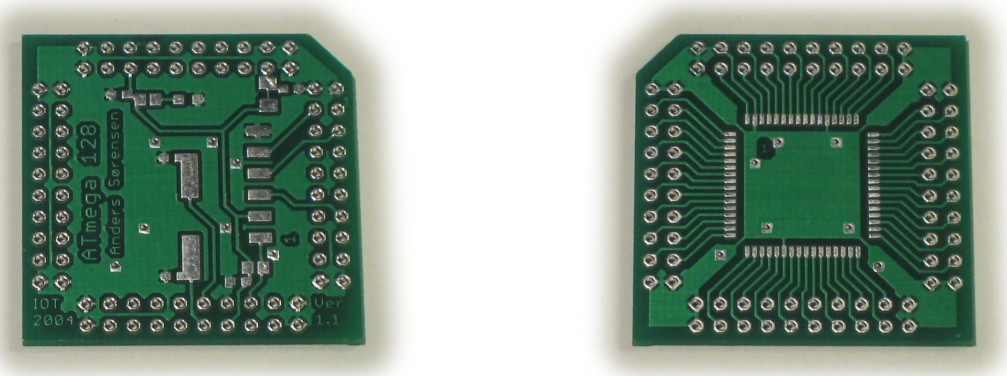
3.1 Identifying the PCB

As indicated by figure 2, the PCB has a quadratic outline, with a width of approximately 37mm. The PCB have a missing corner, to enable easy reckognition of the PCB orientation. The top side of the PCB is clearly marked with the text *"ATmega 128"*, *"Anders Stengaard Sørensen"*, as well as *"IOT 2004" and "Ver 1.1"*

Revision 1 is marked **Ver 1.1**. Future revisions of version 1.1 will be marked Ver 1.1.x, where x will be the revision number.

3.2 Identifying the components

The components needed to assemble the adapter are listed in appendix C, and shown in figure 3



(a) Top side (b) Bottom side

Figure 2: The ATmega128 adapter Printed Circuit Board (PCB)

ATmega128: is a black plastic IC, with quadratic outline, featuring 16 pins on each side, fitting the footprint on the PCB bottom side. It is clearly marked with the text:*"ATMEGA128"* or *"ATMEGA128L"*. When the printing on the IC is readable left to right, pin 1 is in the left-upper corner.

Capacitors: The three 100nF and two 22pF capacitors used in the circuit, are all the 0603 SMD type, which are rather small. They have no distinguishing features of any kind, so care must be taken to avoid mixing them.

Resistors: All resistors are either in 0805 or 1206 SMD houses. Both types have their value printed on them. Typically as tree digits: `XYZ`, which should be read as $XY \times 10^Z$, e.g. `000` (0Ω), `471` (470Ω), `103` ($10k\Omega$). The print can be somewhat fine, so a magnifying glass or an ohmmeter can come in handy.

Crystal: The crystal usually come as an oval metallic package, with the crystal frequency printed on top. Be sure to use a crystal frequency within the permitted range for the exact version of the ATmega128 you are using. Usually, a 16MHz crystal is used, but for some applications (supply voltages below 5V), a lower crystal frequency should be used. Refer to the data sheet of the microcontroller for further information.

Pinheaders: The pinheaders are single/double rows of (golden) pins with a 2.54mm (1/10 inch) spacing. The single row pinheader used should feature pins with a $90\circ$ bend, so they can be surface mounted.

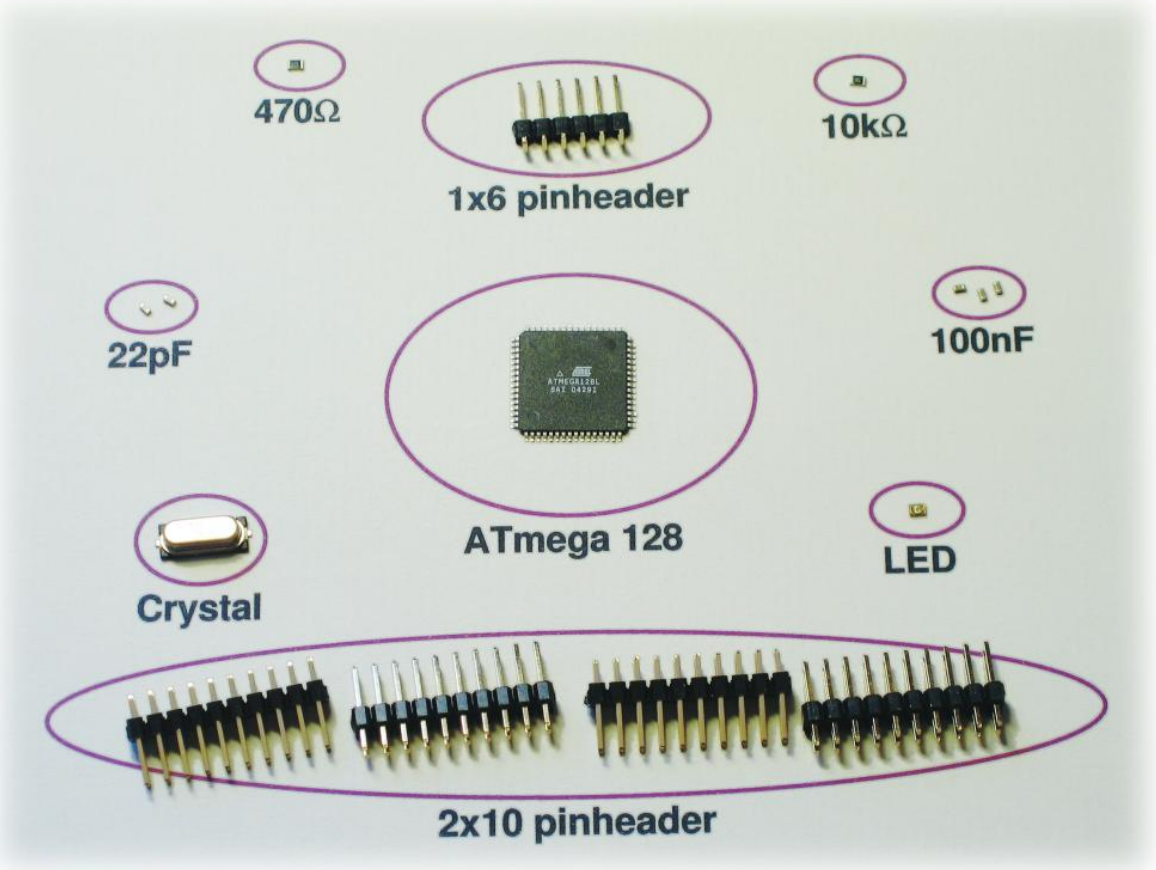


Figure 3: Photograph of the necassary components

3.3 Mounting the components

As many of the components for the adapter are surface mount devices (SMD), some special tools, combined with a little experience is necessary to mount them. If you do not have experience with soldering SMD components like these, I suggest that you seek advice from someone who has.

Mounting the microcontroller

Although the microcontroller can indeed be soldered the traditional way — one pin at a time — the traditional method pose some difficulties because of the narrow spacing of the pins.

A more feasible alternative to the traditional method, is to use a special hollow soldering tip, that utilizes the surface tension of the melted solder to avoid leaving droplets between the pins. The procedure for using the hollow tip methis is as follows:

1. Place the microcontroller in the correct position, and solder two diagonal pins with the traditional method, in order to hold it in place.
2. Place the PCB on a flat surface, and apply a lot of liquid fluss to the PCB below the microcontroller. All pins of the microcontroller should be thouroghly drenched in fluss. You should rather use too much than too little.
3. Clean the hollw solder tip thouroghly with a damp sponge.
4. Melt a small amount of solder on the hollw tip, so it forms a drop which is held in place on the hollow tip by surface tension.
5. Drag the hollow tip along the pins of the microcontroller, one edge at a time. The hollow tip should be held in a way so the solder drop and the solder tip comes in contact with both the pins on the microcontroller and the pads on the PCB. Move the solder tip at a slow pace, taking 3-4 seconds to move it along each edge of the microcontroller.
6. Inspect the solderings visually — using a strong lens. In case of short circuits or poor soldering, reapply plenty of fluss and resolder the row of pins with the hollow tip.
7. When satisfied with the soldering, remove surpluss fluss with a suitable cleaning agent.

The hollow tip will dispence just the amount of solder that can be held in place by the surface tension of each individual pin-pad pair, removing any surpluss solder, which will be sucked into the droplet formed by the surface tension at the solder tip. The fluss cleans the soldering surfaces, and creates a vapor pressure that will inhibit the creation of “solder bridges” between microcontroller pins.

Mounting the resistors and capacitors

Compared to mounting the microcontroller, the resistors and capacitors should be quite easy. You need a pointed soldering tip, a set of pointed tweezers, some thin solder, and possibly a magnifying glass. The procedure goes like this:

1. Place a small drop of solder on one of the pads.
2. Pick up the component with the tweezers.
3. While melting the small drop of solder, move one end of the component into the liquid solder-drop. Make sure you hold the component flat down toward the PCB.
4. Remove the soldering iron, while holding the component in place with the tweezers. Hold the component in place till the solder drop solidifies.
5. Solder the other pad without using the tweezers, and wait for the solder to solidify.
6. Resolder the first pad, applying a bit more solder if necessary. This will remove any mechanical stress that might be present in the first soldering due to strain or vibrations

from holding the component.

Mounting the LED

I have chosen a LED with only two terminals, making it unfortunately easy to mount it the wrong way, as it can be quite difficult to determine which terminal is the anode and which is the cathode by simply looking at the LED.

Consult the datasheet of the specific LED you are using to get directions for visual identification of the terminals, or simply test it by applying a voltage supply in series with a suitable resistor (eg. 5V in series with 330Ω). When a current flows from anode to cathode the LED should light up.

The LED is soldered in the same way as the capacitors/resitors, only a bit more care must be taken, as the LED's metal terminals do not extend to the upper side of the component.

Mounting the Crystal

If you use a SMD crystal, the easiest way to solder it is to:

1. Place a drop of solder on both pads.
2. Hold the crystal in place on both pads.
3. Press the soldering iron to an area of the pads beside the crystal, using heat conduction to melt the solder below the crystal.

If you use a through-hole crystal:

1. Bend the legs of the component to a 90° angle close to the component body, so that they point straight away from each other.
2. Cut off the component legs so they will stay within the area of the solder pads.
3. Solder the stumps to the PCB in the same way as described above.

Mounting the pinheaders

The 1×6 pinheader used for the programming interface should be of the 90° type, where the pins are bent to a right angle. The bent end of the pins can be surface mounted to the 6 coresponding pads on the PCB. Some times the bent end of the pins are a bit too long to fit on the pads, so they should be cut a bit shorter before soldering.

The 4 2×10 pinheaders must be mounted from the *microcontroller* side of the PCB. Take care to mount them completely straight, or the adapter wont fit a socket or PCB.

4 Using the ATmega128 adapter

`This section has yet to be written`

4.1 Test circuit

4.2 Development tools

4.3 The microcontroller configuration bits

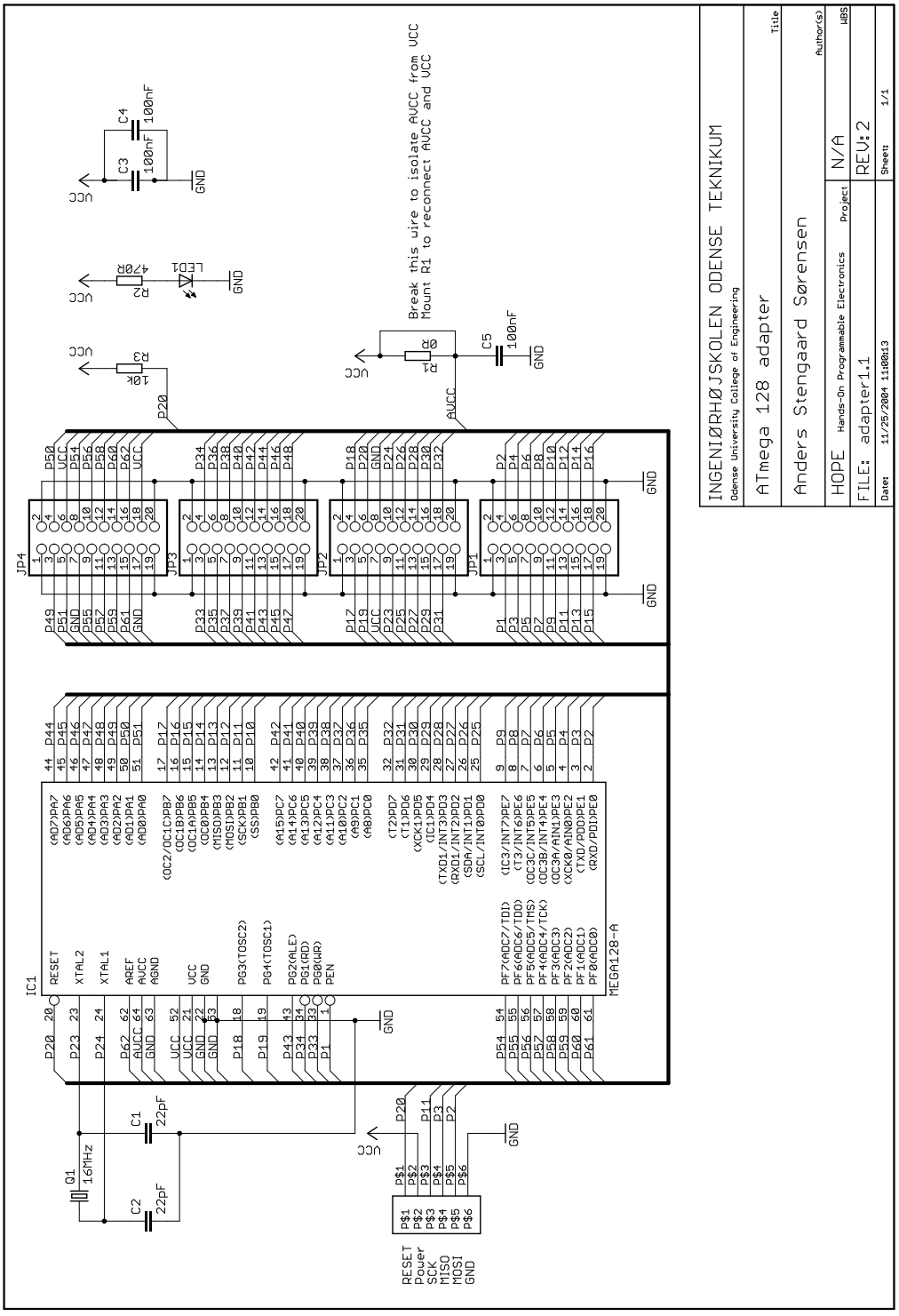
4.4 A blinking diode

4.5 Using the serial port

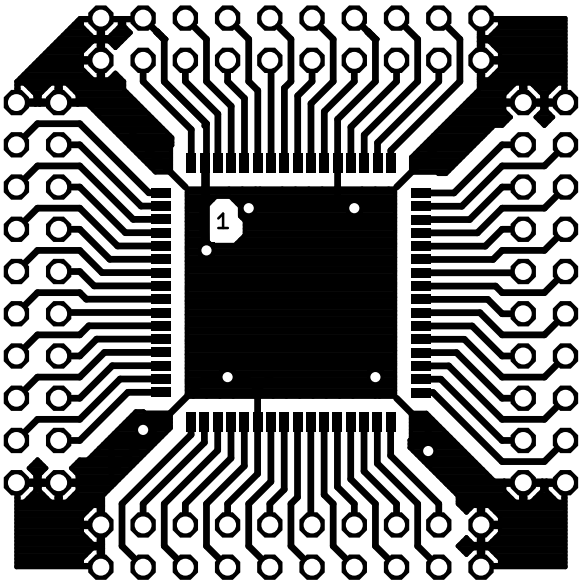
5 Pitfalls and common problems

- Mounting the LED the wrong way.
- Disabling the oscilator.

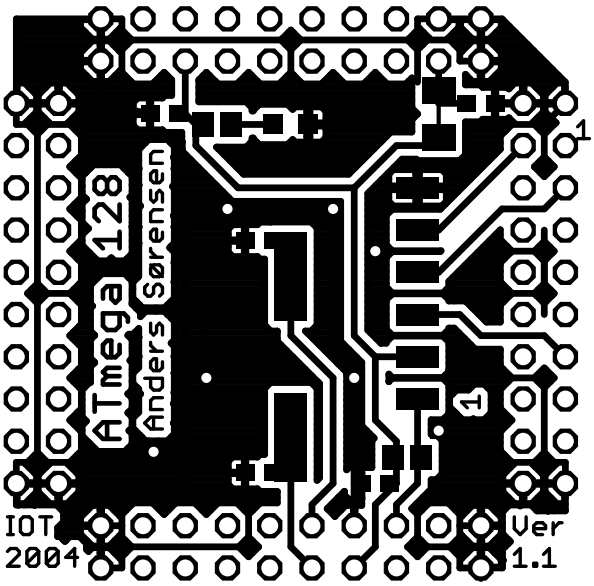
A Schematic


INGENIØRHØJSKOLEN ODENSE TEKNIKUM
Odense University College of Engineering
ATmega 128 adapter
Anders Stengaard Sørensen
HOPE
Hands-On Programmable Electronics
N/A
REV: 2
FILE: adapter1.1
Date: 11/25/2004 11:00:13
Sheet: 1/1
Break this wire to isolate AVCC from VCC
Mount R1 to reconnect AVCC and VCC
MEGA128-A
RESET
Power
SCK
MISO
MOSI
GND

B PCB layout



Bottom side



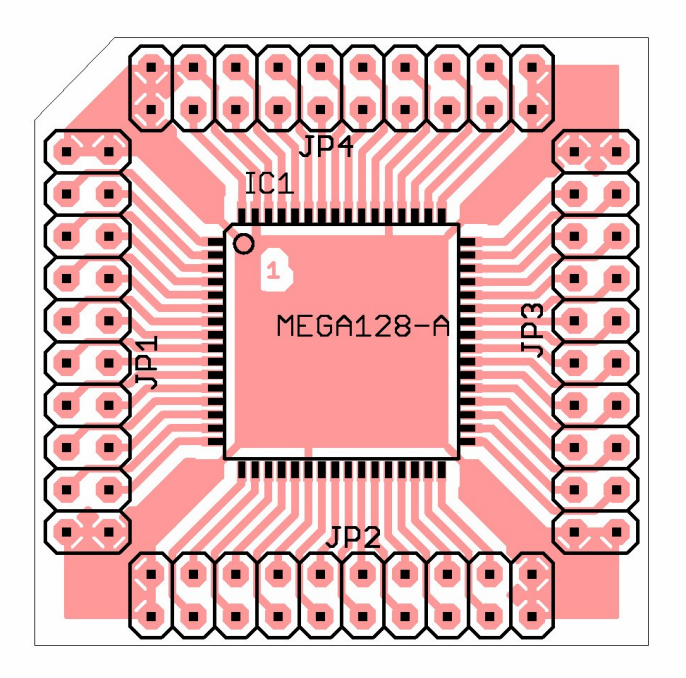
Top Side

C Bill of materials

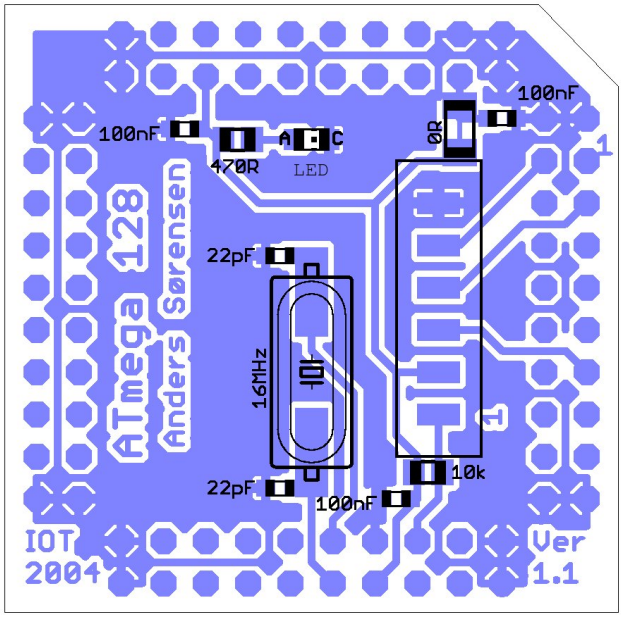
Qty	Type	Value	Footprint	Parts
1	Microcontroller	$ATmega128$	PQFP-64	IC1
1	Resistor	$10k\Omega$	SMD-0805	R3
1	Resistor	470Ω	SMD-0805	R2
3	Capacitor	$100nF$	SMD-0603	C3, C4, C5
2	Capacitor	$22pF$	SMD-0603	C1, C2
1	Crystal	$16MHz$	HC49UP	Q1
1	LED	N/A	SMD-0805	LED1
4	Pinheader	2×10	10x2x100mil PINHD	JP1, JP2, JP3, JP4
1	Pinheader	$1 \times 6, 90°$	6x1x100mil PINHD SMD	CON1
(1) *	Conductor	0Ω	SMD-1206	R1

*: The conductor R1 is only needed if you have disconnected the PCB connection between VCC and AVCC

D Component placement



Bottom side



Top Side

E Pin overview

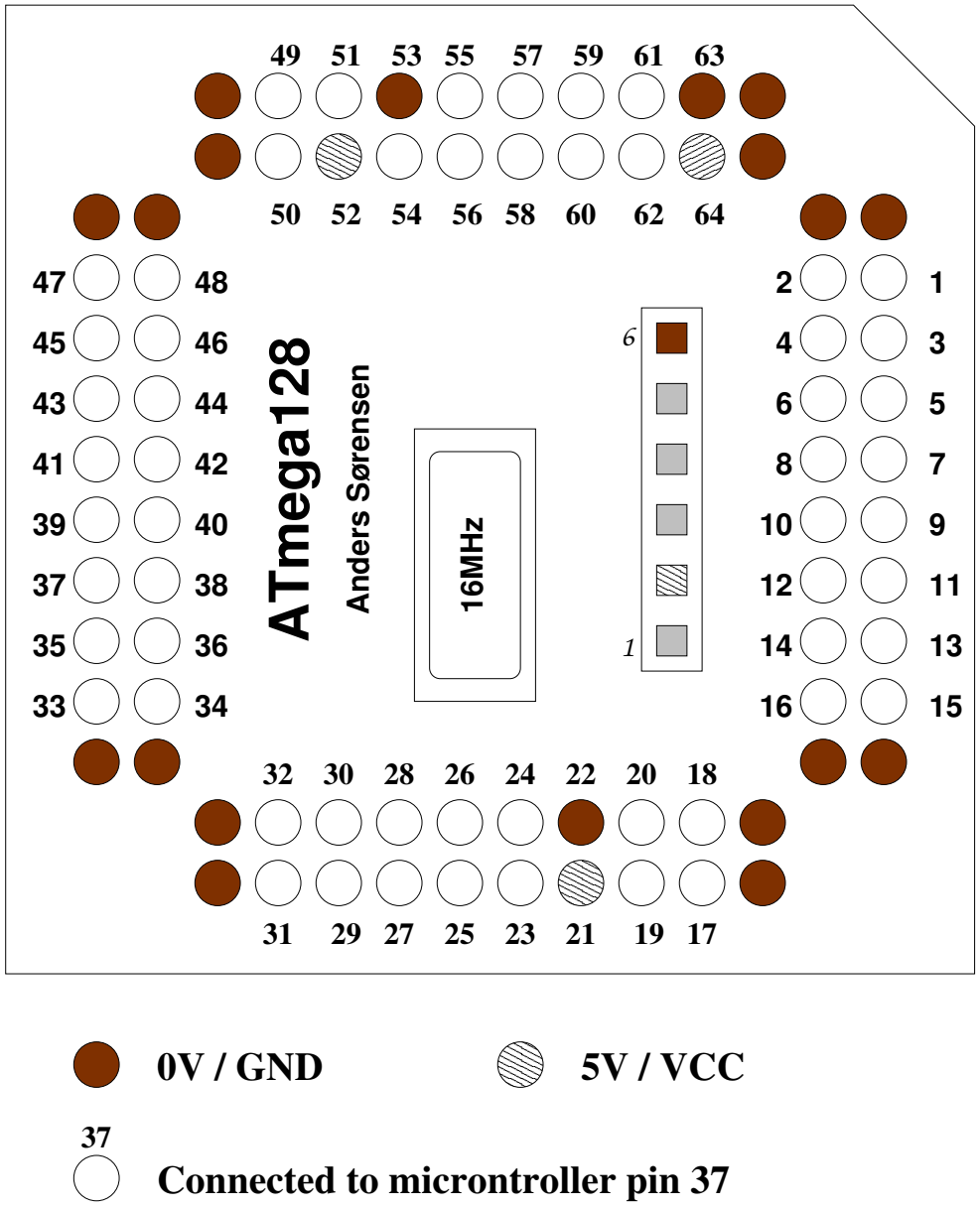


Figure 4: Top view of the pin connections

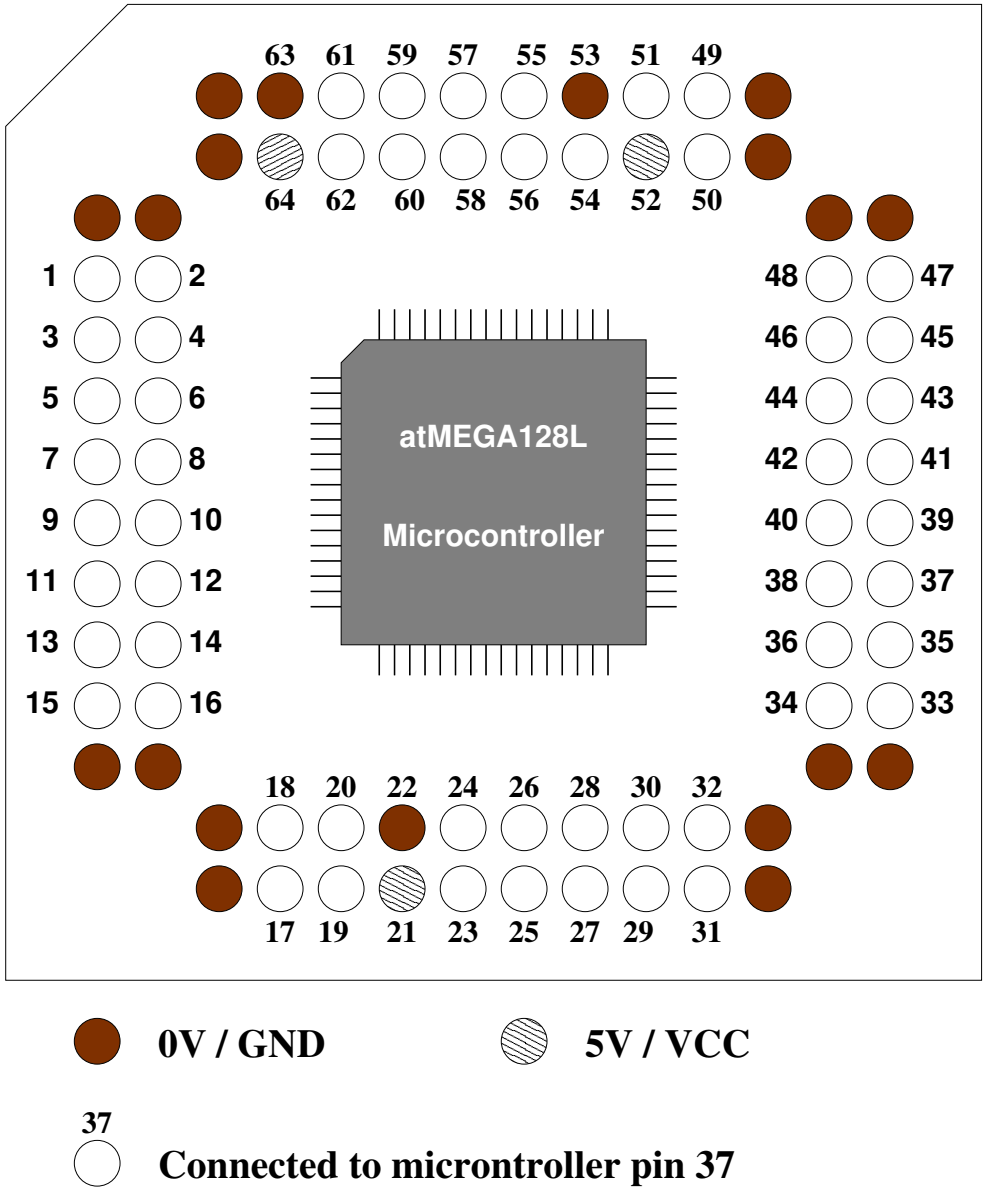


Figure 5: Bottom view of the pin connections

F Known problems and annoyances

This section cover some of the experience already gathered at the point of writing (February 13, 2005). Future versions of the ATmega128 adapter will take the problems stated below into account.

- It is hard to determine the polarization of LED1, when mounting the components. In future versions, LED1 should have a footprint without rotational symmetry, e.g. SOT-23.
- The $6 \times 1, 90^\circ$ pinheader used for the programming interface is not a genuine SMD component. The design should be changed to suport a through-hole pinheader, or I should use a genuine SMD connector.

G Test circuit

H Further reading

- Datasheets
- AVRfreaks
- Other HOPE documents

References