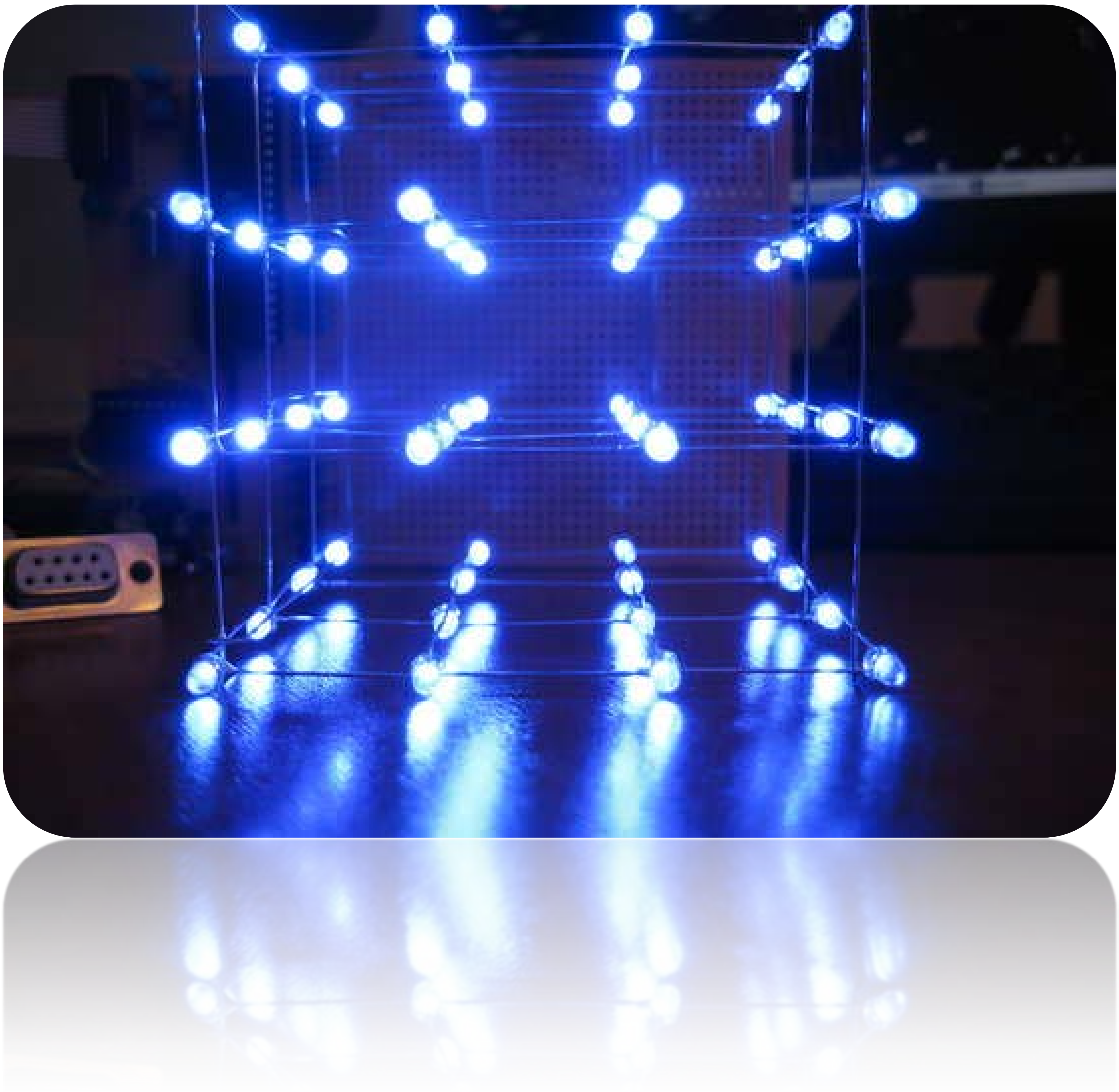
Led Cube 4x4x4

By Bart Brinkman



Supervisor: H. Visch, R. Flipse
Started: 10-01-2010
Finished: 25-05-2010

Table of Contents

Introduction

This project is based on an instructable from the website www.instructables.com it was designed by “CHR” I’ve changed this design to fit my purpose and materials while still being able to keep the same programming.

The Led Cube 4x4x4 or in short led cube is a relatively simple fully programmable cube based on the Atmega 16 AVR. To be able to build the led cube you need to be able to work with a soldering iron and have a strong sense of persistency this is because you’ll need to redesign the scheme to fit your purpose and any mistake can cause the cube not to react.

By completing a project about led’s and AVR programming, I hope to learn more about the aforementioned things because of an interest in computers as a whole and the possibility of acquiring a future career in that sector.

Goal

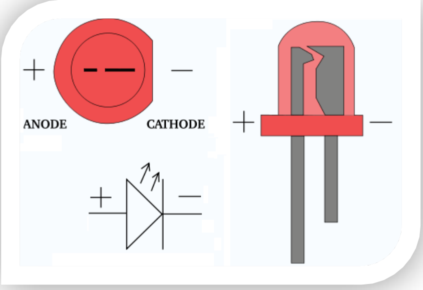
Create a similar led cube to the one from instructables.

Program different behaviour

Make it look as nice as possible.

Theory

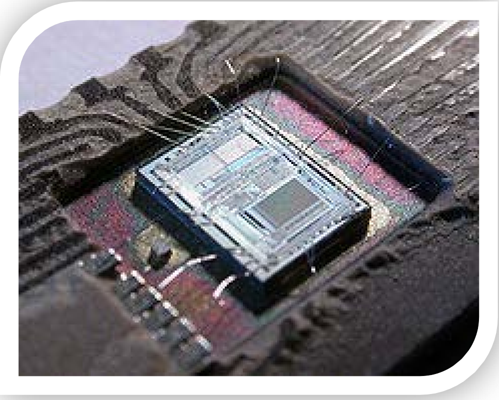
Hardware



led’s

Light emitting diodes are diodes which produce light when current passes through in the right direction. Just like any diode the other direction is completely blocked. Led’s are nowadays commonly used as light sources because they are smaller, very efficient and have a longer lifetime.

Microcontrollers



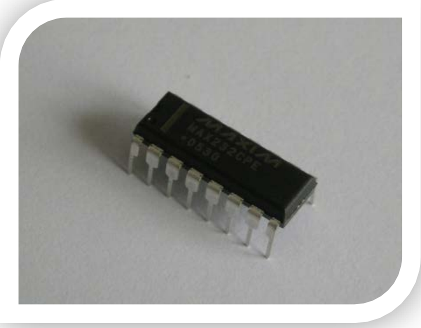
Microcontrollers are small computers in a small package. They contain several things such as a processor, ram and programmable memory and most last but not least inputs and outputs. Microcontrollers are used in most automatically controlled devices e.g. microwaves, cellphones and alarms. They are very convenient to use for products produced in huge quantities because they can be programmed very quickly. And they are very small compared to previous techniques in which processor ram and programmable memory are separated. These computers in a chip are nowadays also used in mp3 players in which another function is very useful, this function is the sleep function, microcontrollers are able to wake up immediately on the touch of a button.

Atmel Atmega 16 AVR

The Atmel Atmega 16 is one such microcontroller.

The Atmega series has become very popular as it was one of the first to use on-chip flash memory which could be rewritten infinite amount of times whereas other microcontrollers used EEPROM (One time programmable ROM). It kept being popular in homebuilt projects which involve microcontrollers. This is mainly because it's still a cheap series of microcontrollers with lots of functions, but it's also because they're relatively easy to program.

I chose this exact microprocessor merely because I could stick to the previously made program. *(at the time being I wasn't sure I was going to be able to rewrite the program, or even design a completely new one)*



MAX232

The MAX232 chip is a chip which functions as an interpreter between the computer and the AVR. It converts the signals from an RS-232 serial port (computer) to the RX, TX, and signals (AVR).

USBasp

The USBasp is another interpreter though this one works over the computers USB port. It is easier than the MAX232 because this is a ready to use solution, whereas the MAX232 still needs to be connected to the rest of the board. The USBasp also has a status led which is very convenient.

Wiring

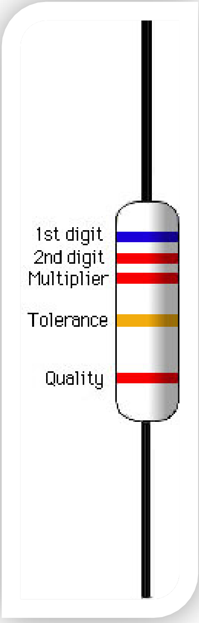
Well wires, these are cables that let through electricity. This can be as a power supply or as a matter of signals. This cable in the picture is the 10 pins connecter used to connect the USBasp with the 10 pins connecter on the protoboard.

Capacitors

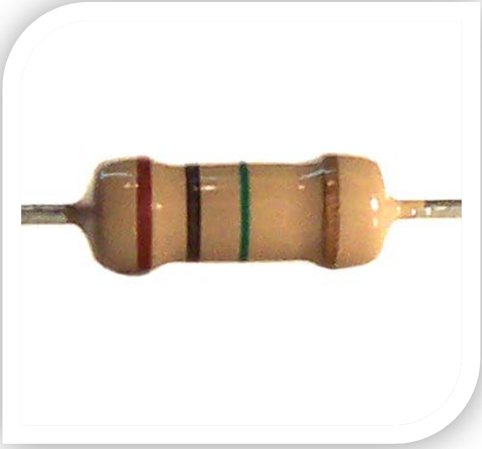
Capacitors are used to for many different things. But in this project they are mainly used as stabilizers for example between a ground and 5v wire. Otherwise the current might blow the microcontroller.

Resistors

A resistor is a component that ‘resists’ the flow of electricity. The flow of electricity is called current. Every resistor has a certain value telling how much it resists the flow.



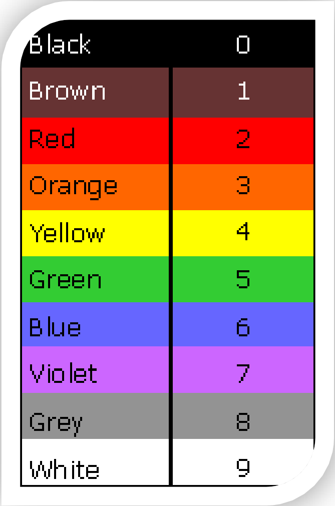
This resistance is called ohm, it’s most commonly shown with the omega Ω.



These resistors are needed on the breadboard to prevent a led from getting to much current. This would kill the LED. It’s the same case with the Infrared LED.

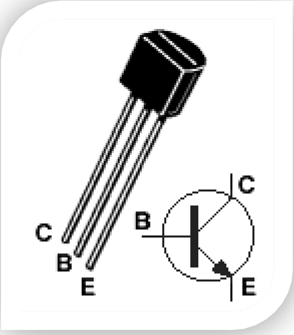
Resistors are color coded. One gold, silver or blank ring indicating the tolerance the resistor can differ from the given value. And 3 rings which together make the value in ohm. The fifth ring isn’t used to often.

Each color is a number varying from 1 to 9. See the table on the right.
For example when a resistor is in the above case blue, red, red it means the first digit is “6” the second digit is “2” and it must be multiplied by “2 x 10” or a hundred which would make it 6200 Ω.



Black	0
Brown	1
Red	2
Orange	3
Yellow	4
Green	5
Blue	6
Violet	7
Grey	8
White	9

Once such a resistor is made it will not change its value. The only thing which could happen is it gets burned when its limit is overridden by supplying a high enough voltage killing the entire thing. When this happens the resistor won’t do anything anymore and can be thrown away.



Transistor

In this project the transistor is used merely as a switch to allow more current to pass through the circuit bypassing the AVR which cannot take this much current.

Crystal oscillator

Crystals are used together with microcontrollers to make them run faster. The standard clock speed (cycles per second) of the Atmega 16 series is 1 MHz but with our crystal oscillator it can run at 14.7456 MHz which makes it look much smoother.

A Crystal oscillator is an electronic circuit that uses mechanical resonance of a vibrating crystal to create an electrical signal with one exact frequency.

Method

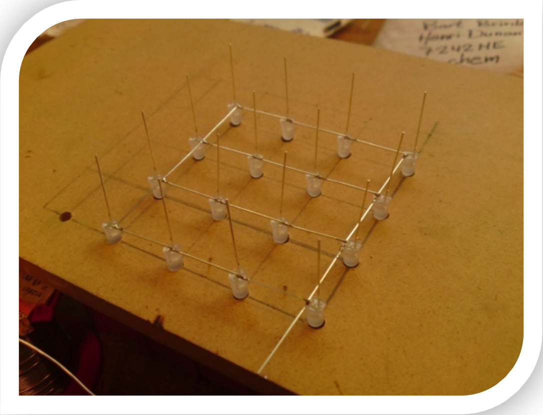
List of Materials

- Protoboard.
- Atmel AVR Atmega16 microcontroller
- AVR programmer (USBasp)
- 64 Blue Led's (some extra for backup)
- 1 Green Led
- 1 Red Led
- Max232 rs-232 chip
- 16x resistor 220 Ohm
- 2x resistor 470 Ohm
- 1x resistor 10 kOhm
- 4x resistor 2.2 kOhm
- 4x NPN transistor BC337
- 10uF capacitor
- 1000uF capacitor
- 6x 100nF ceramic capacitor
- 2x 22pF ceramic capacitor
- Crystal 14.7456 MHz
- 2x push button
- connector for 5v power (Screw terminal)
- 5v power supply (a cell phone charger)
- Sockets for each chip (Atmega16, Max232, 74HCT244)
- Soldering station
- Wires
- 10 pin header
- 10 pin flat cable
- Tin

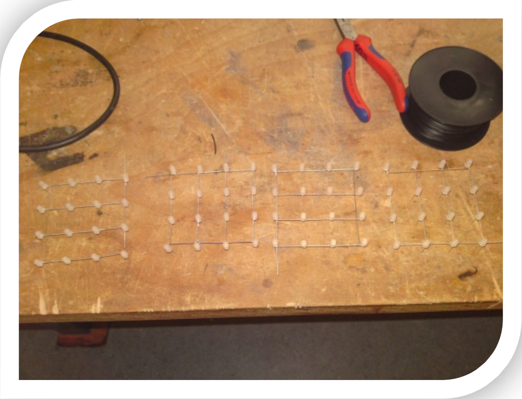
I'd advice to buy one more from all the cheap small parts, just in case you might lose or kill one.

Assembly of the led's

Take a piece of wood and make holes of the size of your led's, in my case 5mm place the holes at pins length of each other. Do this very accurately! If you have one hole in the wrong place you'll have it wrong in four different layers.



Put your led's one by one in the holes bend one leg so it touches the leg of the other led and solder them together. Do this for every led till you've got all four levels finished.

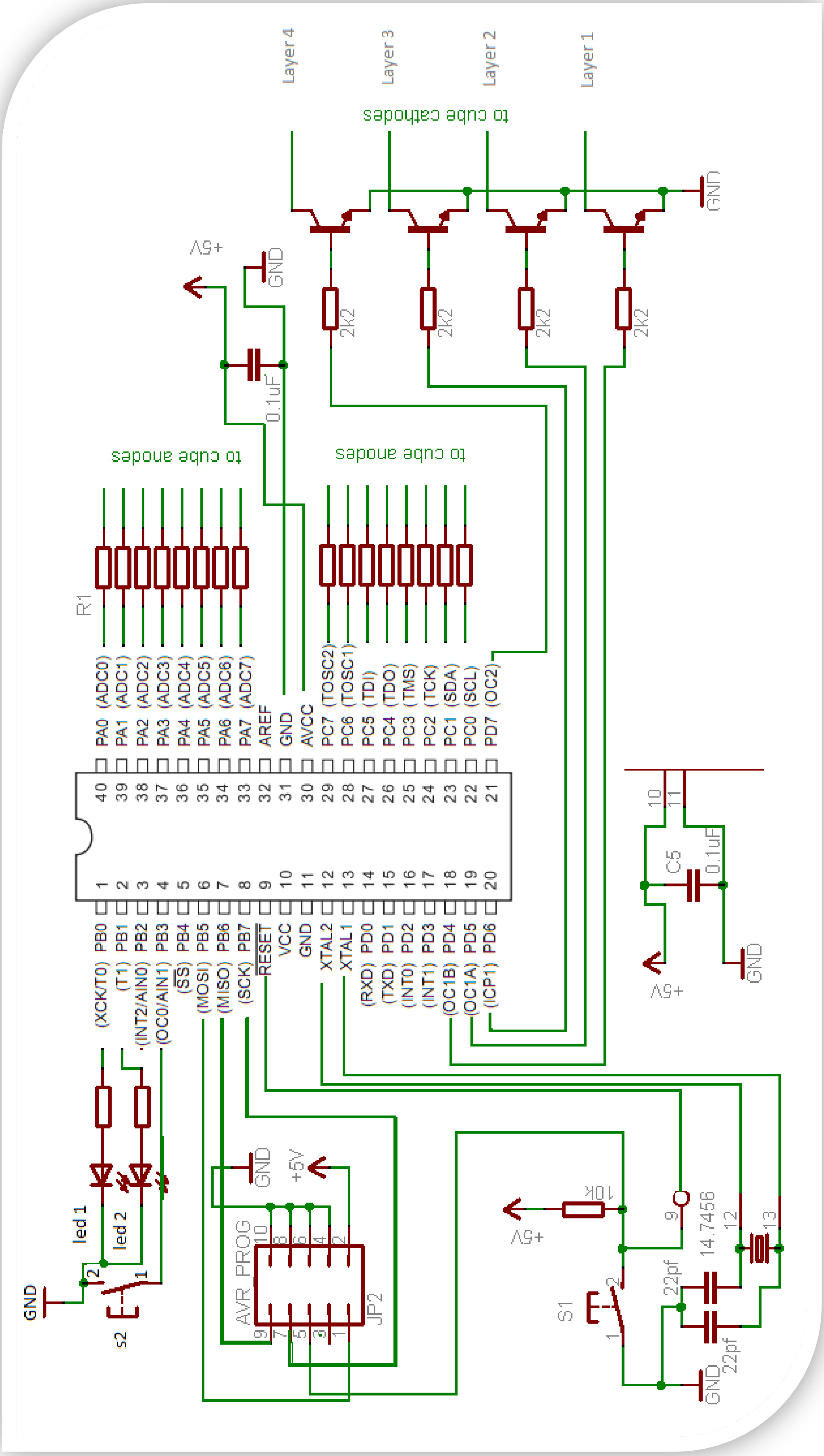


When you've got them all it should look something like this. Then you'll have to solder the different layers together finishing the cube itself.

Now that we've done the easy things were going to focus on the real electronics behind this cube. I'm not going to describe how I've connected each and every wire because that would be boring. What you're trying to do is connect every wire to the correct pin. The correct way is shown on the next page.

Important is that before you start this you should make sure your desk is clean and tidy! Otherwise you'll soon lose track.

The circuit shown on the next page was made by me. I did this as this uses the physical connection of the AVR whereas the other circuit showed the schematic connections (see appendix)


Layer 4
Layer 3
Layer 2
Layer 1
to cube cathodes
GND
+5V
GND
0.1uF
2k2
2k2
2k2
2k2
to cube anodes
to cube anodes
R1
PA0 (ADC0)
PA1 (ADC1)
PA2 (ADC2)
PA3 (ADC3)
PA4 (ADC4)
PA5 (ADC5)
PA6 (ADC6)
PA7 (ADC7)
AREF
GND
AVCC
PC7 (TOSC2)
PC6 (TOSC1)
PC5 (TDI)
PC4 (TDO)
PC3 (TMS)
PC2 (TCK)
PC1 (SDA)
PC0 (SCL)
PD7 (OC2)
(XCK/T0) PB0
(T1) PB1
(INT2/AIN0) PB2
(OC0/AIN1) PB3
(SS) PB4
(MOSI) PB5
(MISO) PB6
(SCK) PB7
RESET
VCC
GND
XTAL2
XTAL1
(RXD) PD0
(TXD) PD1
(INT0) PD2
(INT1) PD3
(OC1B) PD4
(OC1A) PD5
(ICP1) PD6
C5
0.1uF
+5V
GND
10
11
GND
led 1
led 2
s2
AVR PROG
JP2
GND
+5V
10K
+5V
S1
14.7456
22pf
22pf
GND
9
12
13

Wiring is done

What to do now? Well that's easy you've probably discovered that your led cube is not yet lighting up. Well that's because you'll need a 5v power supply. This could be anything, I've used a cell phone charger as this was available to me.

You take cut of the connecter and strip the wire for about 2 cm's. Depending on the brand of charger you'll find several wires. Most of the time you'll gonna want to use the black and red one. Being the black ground and the red one the 5V. Connect these two to the corresponding place on the protoboard and now it should be finished.

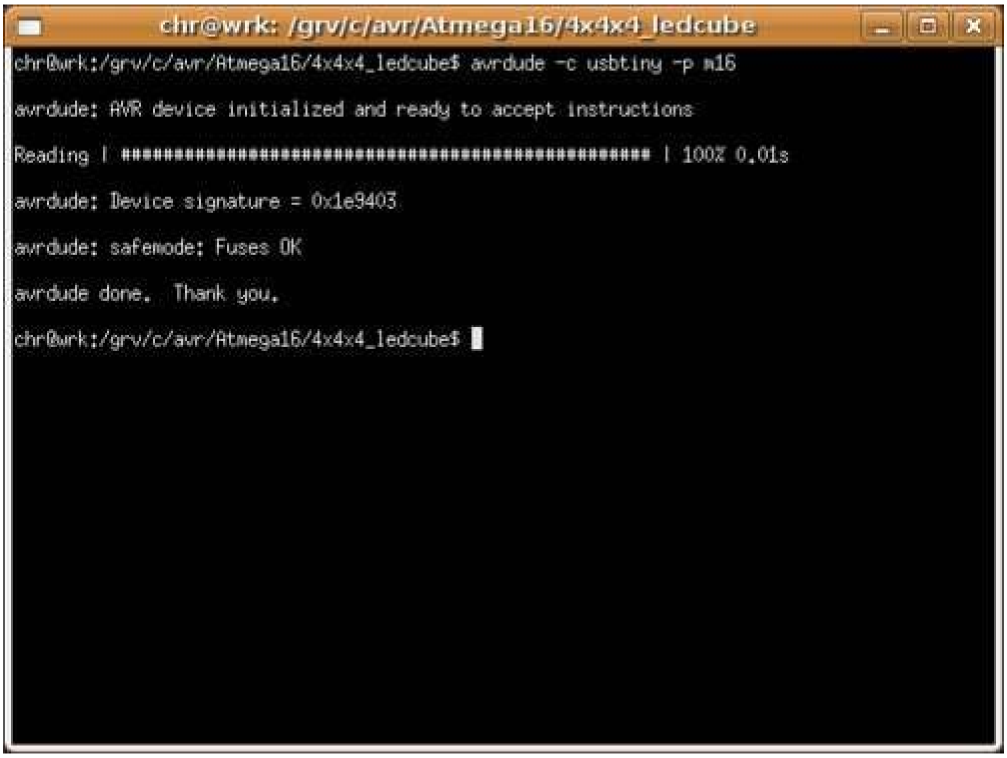
Mind you there is always the troubleshooting. In most cases you will have connected one wire the wrong way. Or not at all go through all the wires step by step and part by part if the cube won't react when you try to reach it in the next step.

Programming

Programming takes up lots of time. You'll need to think of exactly how you want the program to run. And you'll need to tell (write down) every step you want it to do. This can be done in multiple languages though "C" is the easiest for our purpose.

I've not yet learned to master this language and therefor it was impossible to design my own program. I was however able to edit the current program more to my liking. I'll soon go on and will make my own patterns. It will need some time tough as there is a current problem programming through USBasp with a 64-bit operating system.

To flash the program to the led cube you'll need a program that allows you to write data to the flash memory of the atmega16 on the led cube. This can be done by using AVRdude. AVRdude is a command line tool which takes some time to get used to.



```
chr@wrk:/grv/c/avr/Atmega16/4x4x4_ledcube$ avrdude -c usbtiny -p m16

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.01s

avrdude: Device signature = 0x1e9403

avrdude: safemode: Fuses OK

avrdude done.  Thank you.

chr@wrk:/grv/c/avr/Atmega16/4x4x4_ledcube$ 
```

Conclusion

I’ve created a led cube which works and does exactly what the one on instructables.net does. I think it looks pretty too. I’m quite happy with it even though I’ve not been able to program different behaviour yet. I hope to be able to do this later.

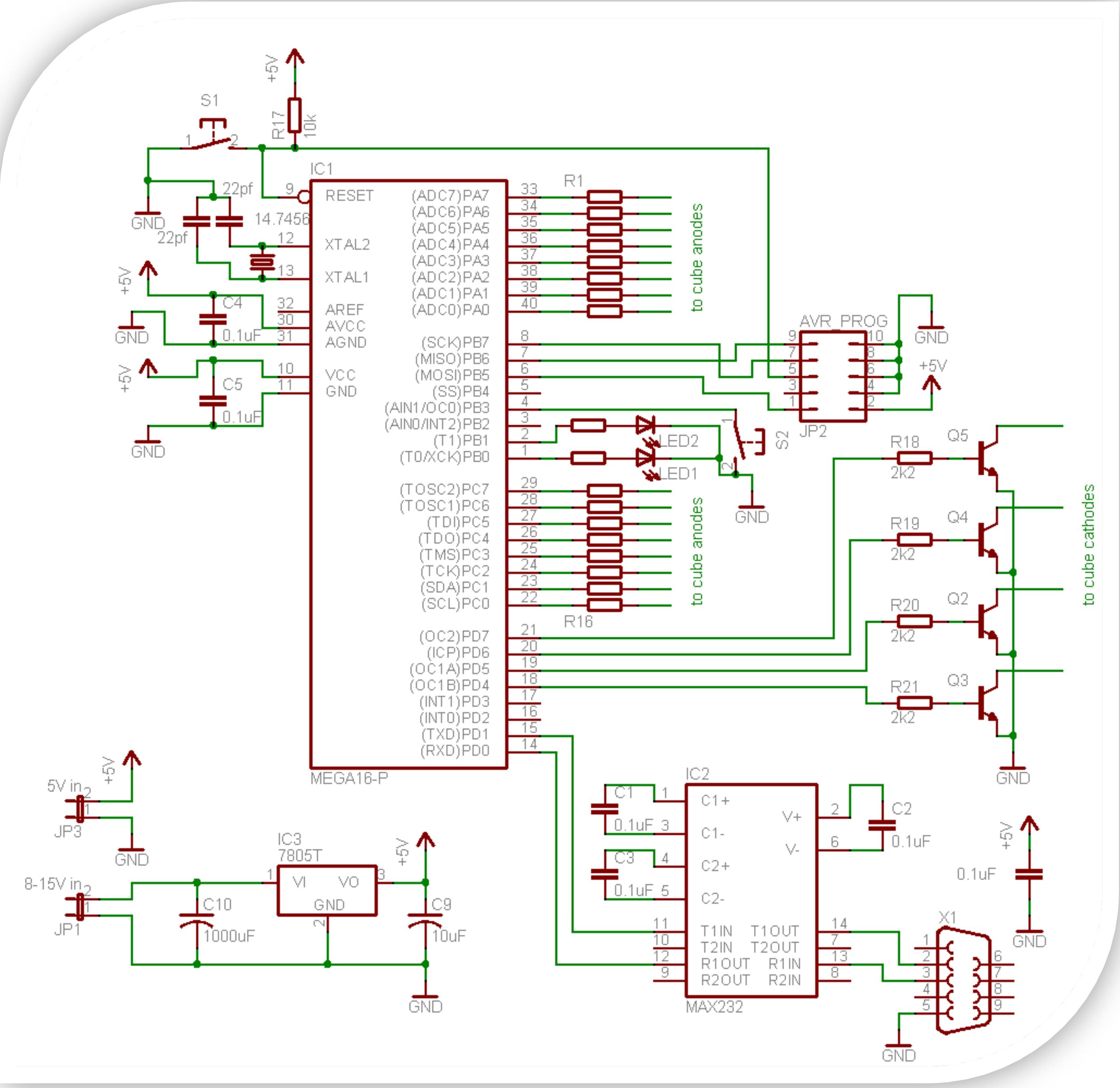
Discussion

Even though I'm completely happy about the whole project I reckon there are things I could've done better.

Like:

- I could've let someone recheck the way I soldered the lets together, this would have saved me multiple hour of work and troubleshooting.
- I should have used multiple colors of wires making troubleshooting easier.
- I should have bought the USBasp directly as the serial ports are really out-dated and are therefore not supported anymore.
- I should have made the report directly after the project or even while working on the project.

Appendix


+5V
S1
1
2
R17
10k
IC1
GND
22pf
22pf
14.7456
9
RESET
12
XTAL2
13
XTAL1
+5V
C4
0.1uF
32
30
31
AREF
AVCC
AGND
GND
+5V
C5
0.1uF
10
11
VCC
GND
GND
(ADC7)PA7
(ADC6)PA6
(ADC5)PA5
(ADC4)PA4
(ADC3)PA3
(ADC2)PA2
(ADC1)PA1
(ADC0)PA0
33
34
35
36
37
38
39
40
R1
to cube anodes
(SCK)PB7
(MISO)PB6
(MOSI)PB5
(SS)PB4
(AIN1/OC0)PB3
(AIN0/INT2)PB2
(T1)PB1
(T0/XCK)PB0
8
7
6
5
4
3
2
1
AVR_PROG
9
7
5
3
1
10
8
6
4
2
JP2
GND
+5V
LED2
LED1
S2
1
2
GND
(TOSC2)PC7
(TOSC1)PC6
(TDI)PC5
(TDO)PC4
(TMS)PC3
(TCK)PC2
(SDA)PC1
(SCL)PC0
29
28
27
26
25
24
23
22
R16
to cube anodes
R18
2k2
Q5
R19
2k2
Q4
R20
2k2
Q2
R21
2k2
Q3
to cube cathodes
GND
(OC2)PD7
(ICP)PD6
(OC1A)PD5
(OC1B)PD4
(INT1)PD3
(INT0)PD2
(TXD)PD1
(RXD)PD0
21
20
19
18
17
16
15
14
MEGA16-P
5V in
2
1
JP3
+5V
GND
IC3
7805T
VI
VO
GND
1
3
2
+5V
8-15V in
2
1
JP1
C10
1000uF
C9
10uF
GND
IC2
C1
0.1uF
C3
0.1uF
1
3
4
5
C1+
C1-
C2+
C2-
V+
V-
2
6
C2
0.1uF
11
10
12
9
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
14
7
13
8
MAX232
+5V
0.1uF
GND
X1
1
2
3
4
5
6
7
8
9
GND

Date	Activity	Where	Time
01-10-2010	Finding the Instructable and reading it over and over till I understand.	Home	120
01-14-2010	Making a list of things to buy	Home	120
01-15-2010	Making sure I could do this as a school project	School	50
01-16-2010	Ordering the leds and other hardware	School	50
01-20-2010	Soldering the cube itself together (calling the shop for missing parts.)	School	200
01-20-2010	Report +Print drawing	Home	100
01-22-2010	Soldering at home (bad solder.)	Home	20
01-25-2010	Soldering		
01-26-2010	Soldering	School	100
01-28-2010	Soldering	School	50
01-29-2010	Soldering	School	100
	I stopped writing down my hours.		
Total:			**14**

Sources

http://www.instructables.com/id/LED-Cube-4x4x4/

http://www.dickbest.nl/webshop/index.php

http://www.topled.nl/

http://www.circuitsonline.net/schakelingen/112/computer-en-microcontroller/avr-programmer.html

http://www.zegeniestudios.net/ldc/