

Mikrocontroller

Einführung Teil 1



By: David Martens

Gliederung

- ATTINY/ ATMEGA 8, 16
- Vorstellung der PIN-Belegung des ATMEGA 16
 - Vergleich zum Atmega 8
- Die externen Beschaltung
 - Quarz Beschaltung
 - Pull-UP Widerstände
 - Glättungskondensatoren
- Watchdog Timer
- Brown-Out-Detection
- Programmer
 - Programmer Beschaltung



Attiny

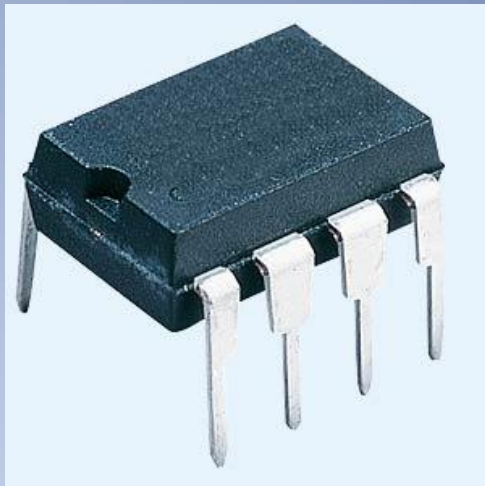
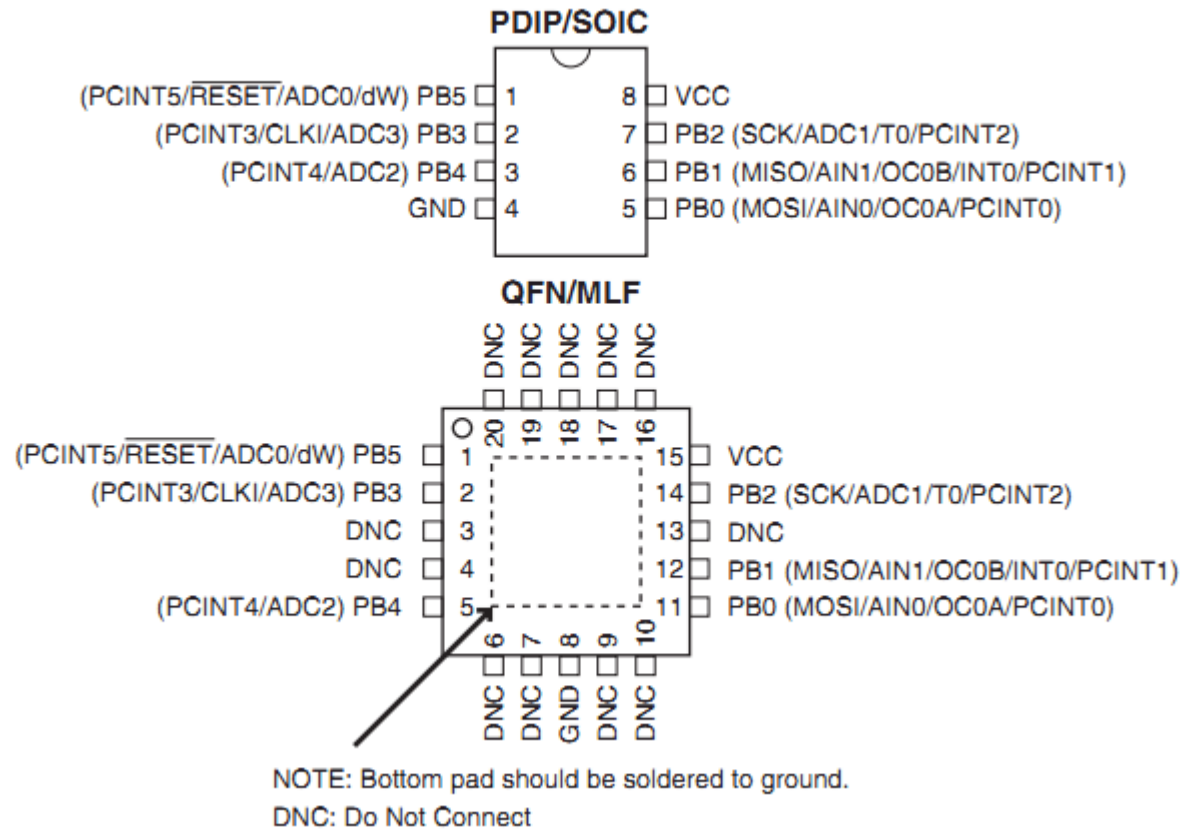


Figure 1. Pinout ATtiny13



Attiny

- Vorteile :
 - Kosten
 - Klein
 - Für einfache Schaltungen ausreichend
- Nachteile:
 - Weniger i/o Ports



Atmega 16

- ATMEGA 16

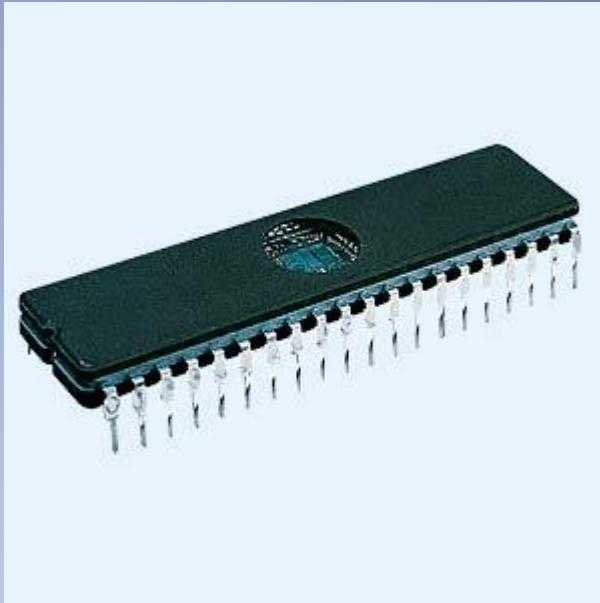


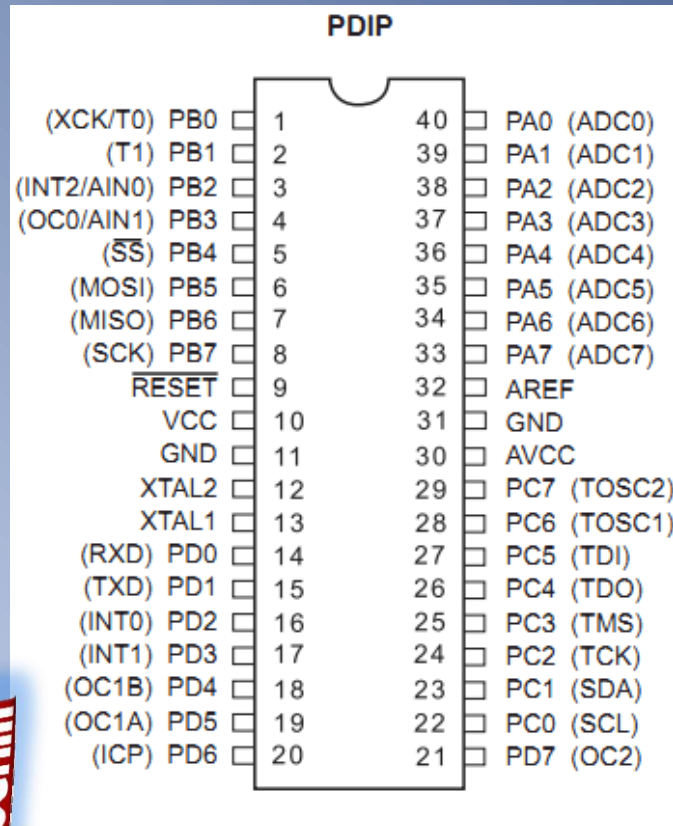
Figure 1. Pinouts ATmega16

		PDIP	
(XCK/T0)	PB0	1	40
(T1)	PB1	2	39
(INT2/AIN0)	PB2	3	38
(OC0/AIN1)	PB3	4	37
(SS)	PB4	5	36
(MOSI)	PB5	6	35
(MISO)	PB6	7	34
(SCK)	PB7	8	33
RESET		9	32
VCC		10	31
GND		11	30
XTAL2		12	29
XTAL1		13	28
(RXD)	PD0	14	27
(TXD)	PD1	15	26
(INT0)	PD2	16	25
(INT1)	PD3	17	24
(OC1B)	PD4	18	23
(OC1A)	PD5	19	22
(ICP)	PD6	20	21
			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)

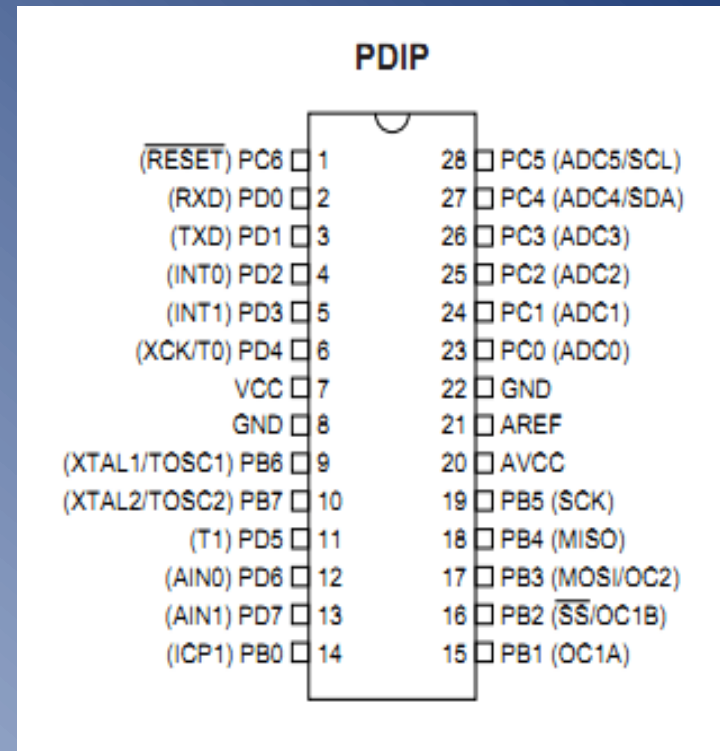


Atmega 8 und 16

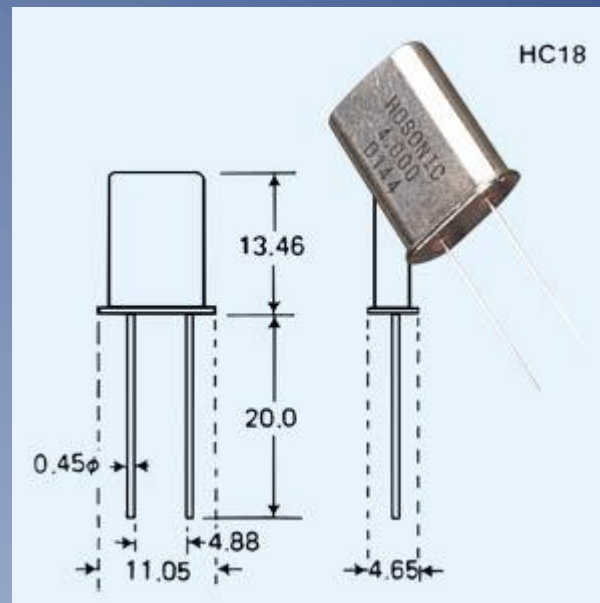
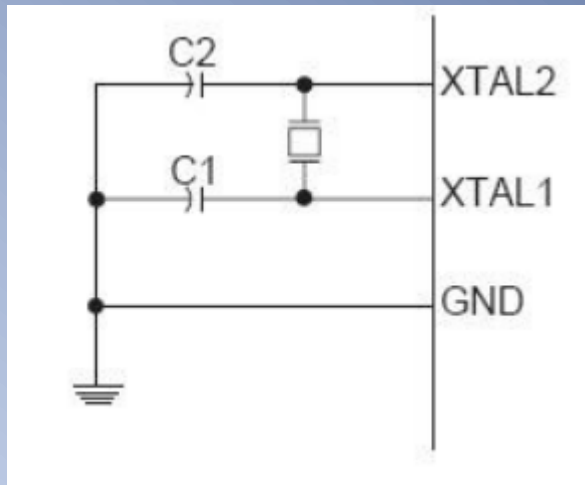
Atmega 16



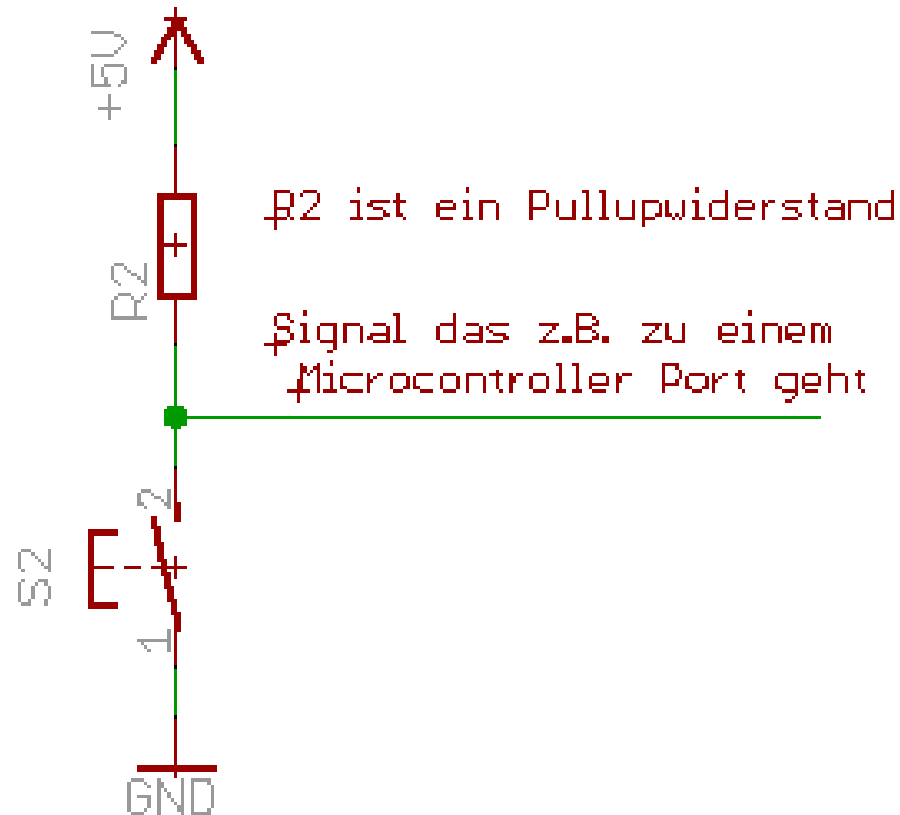
Atmega 8



quarzoszillator

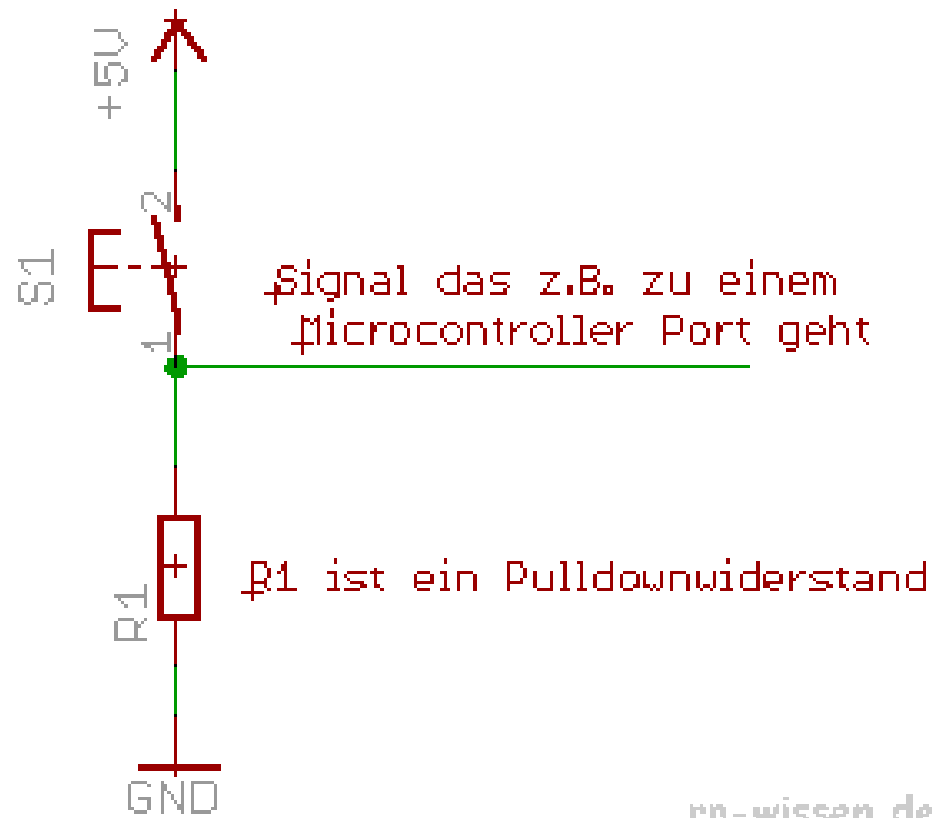


Pull up

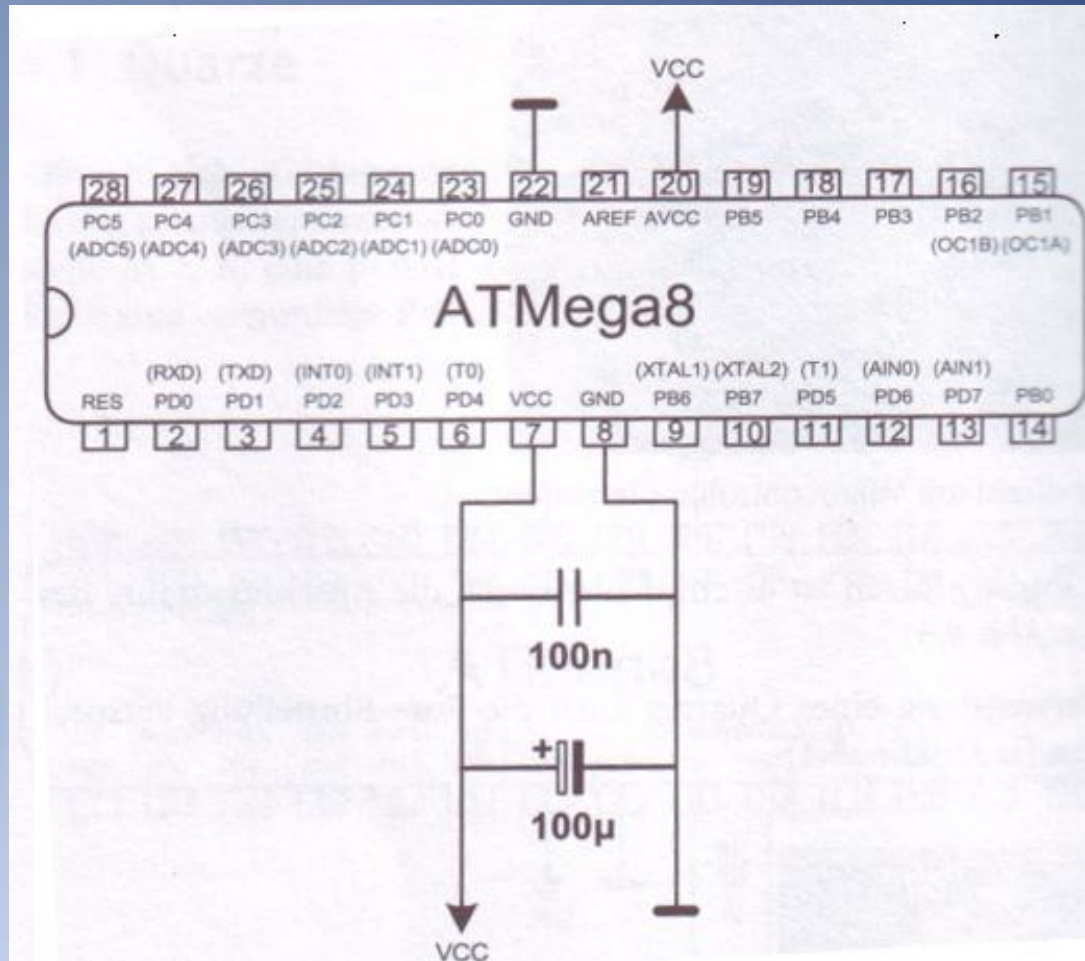


rn-wissen.de

Pull down



Stütz- und abblockkondensatoren



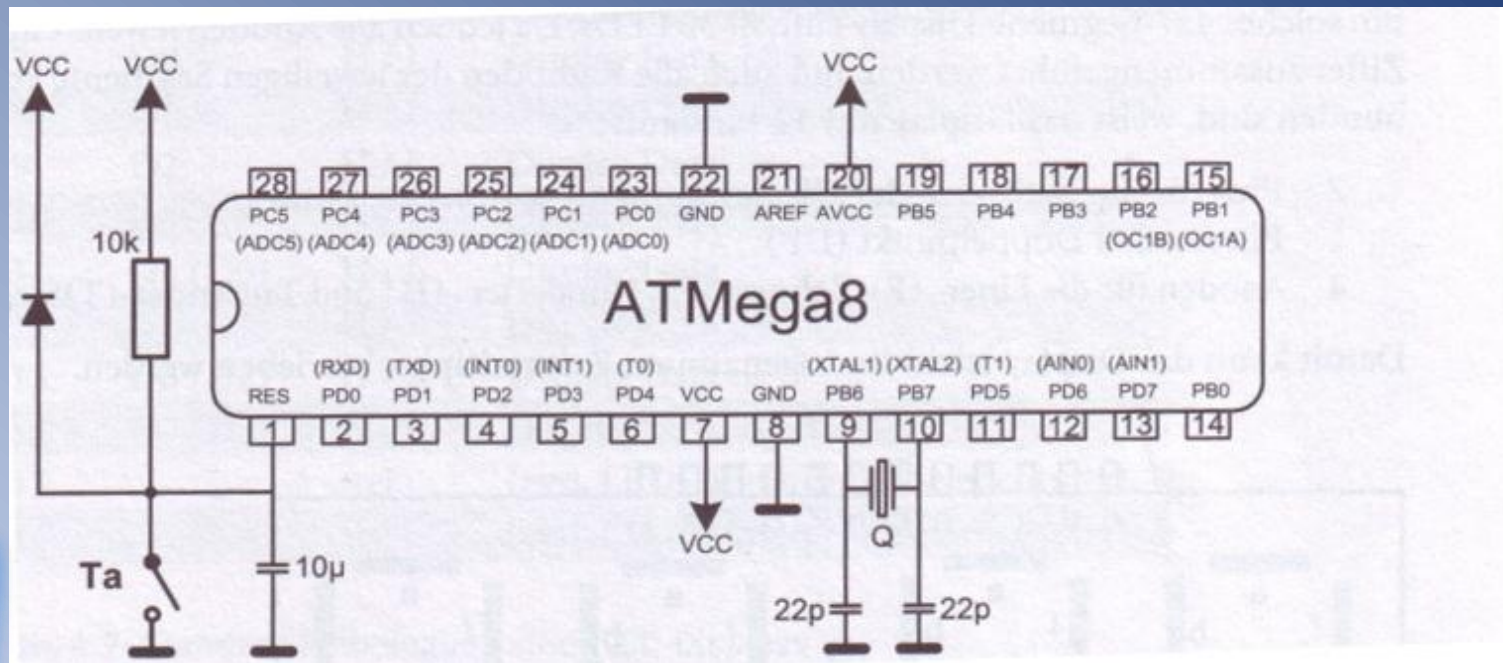
Watchdog timer

- Überwacht den Programmablauf
- Resetet bei definierten Zeiten
- Interne Software Aktivierung /Deaktivierung
- Externes Bauteil incl. Brown-Out-Detection



Brown out detection

- Reset Beschaltung



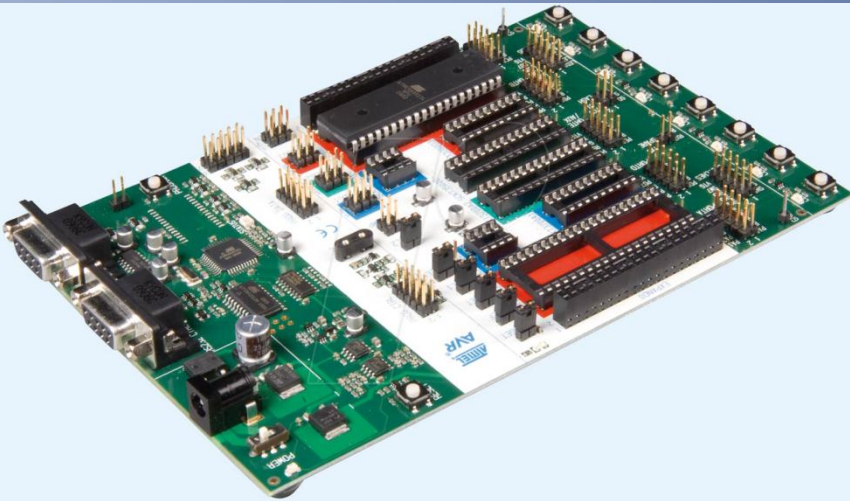
programmer

STK500

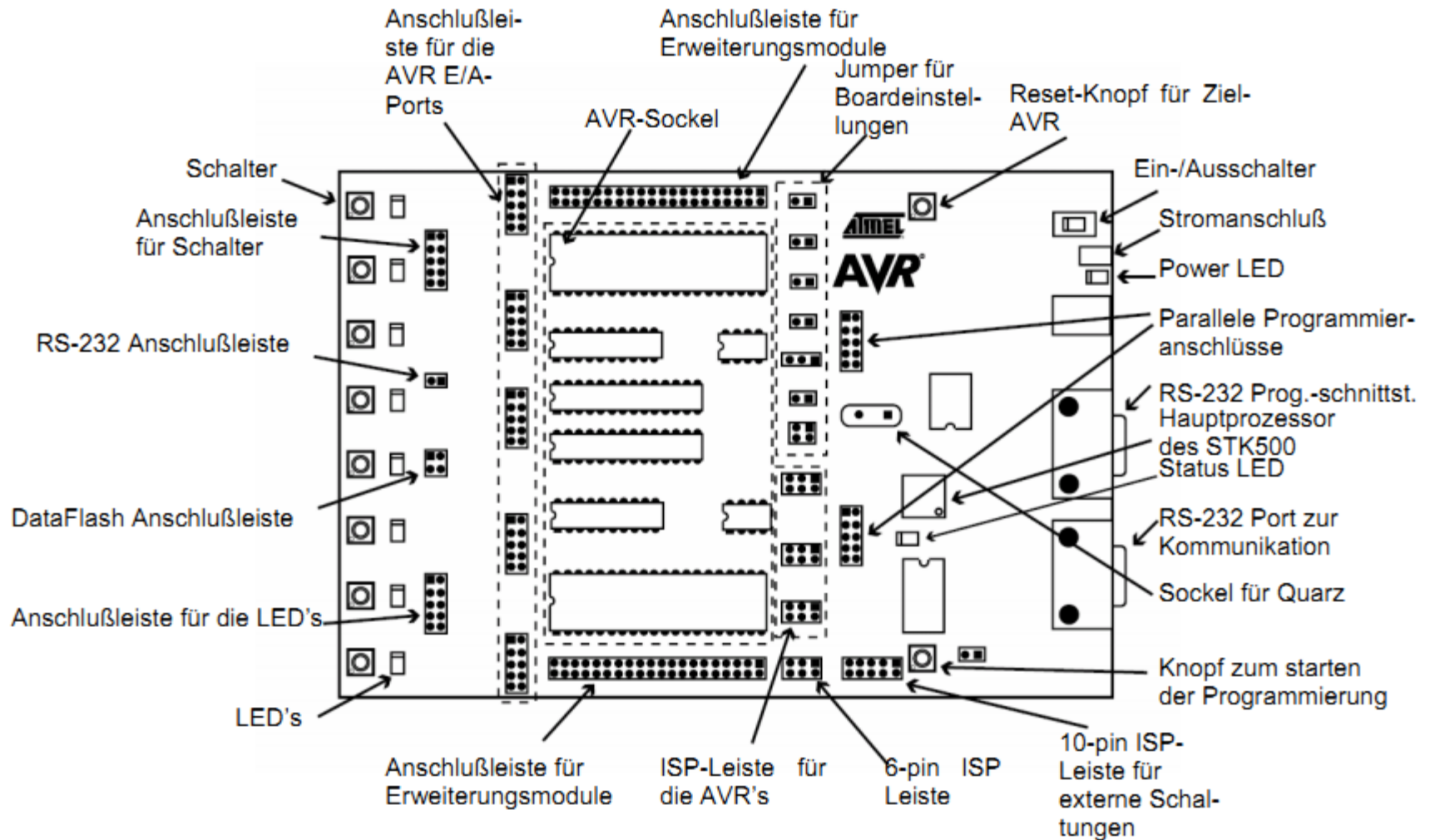
45-80Euro

Diamex-Prog-S

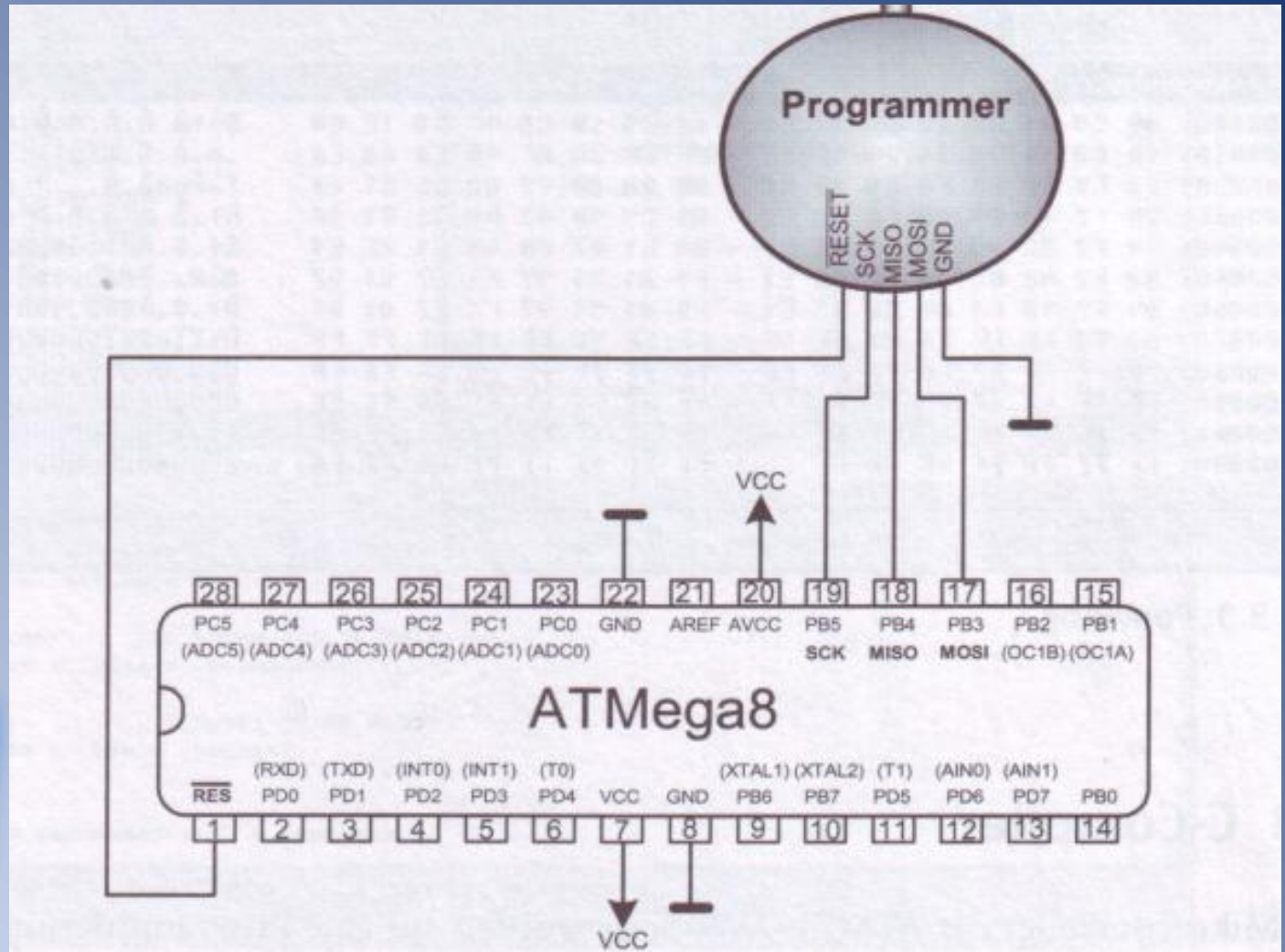
Ab 5 Euro



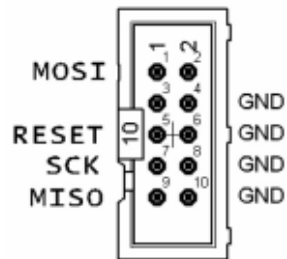
stk500



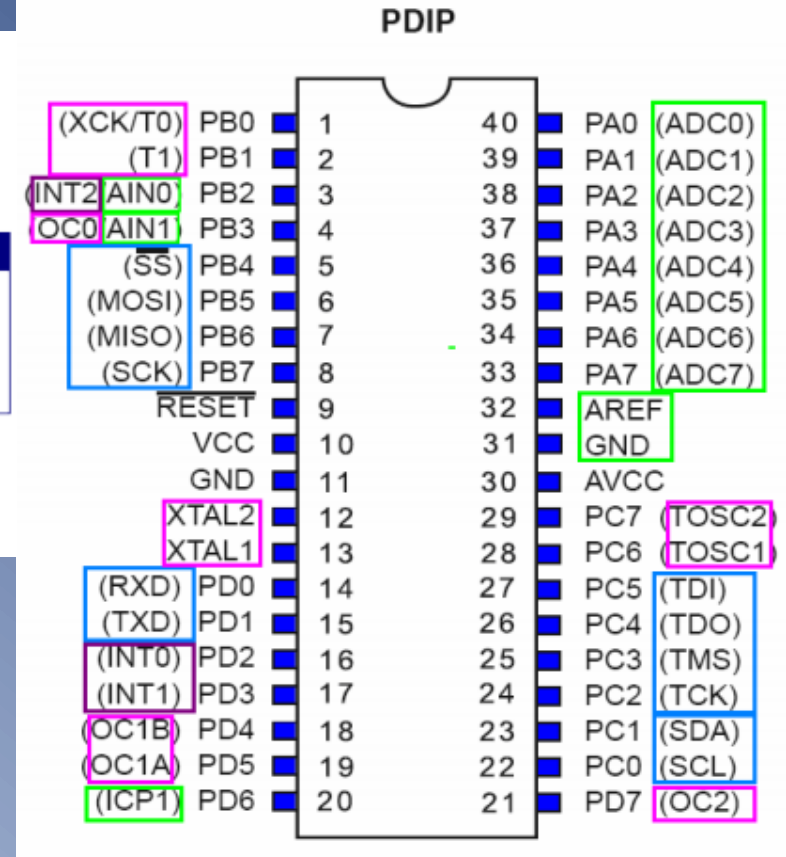
programmer



Diamex – prog - s



10-pol. Stifteleiste	AVR-Controller
PIN1 (MOSI)	MOSI oder PDI
PIN5 (RESET)	RESET
PIN7 (SCK)	SCK
PIN9 (MISO)	MISO oder PDO



Quellen

- <http://www.reichelt.de/>
- <http://www.mikrocontroller.net/>
- <http://www.rn-wissen.de/>
- <http://www.wikipedia.org/>



Mikrocontroller

Einführung Teil 2



By: Immanuel Reuter

Gliederung

- I. Entwicklungstools
 - I. Hardware
 - II. Software

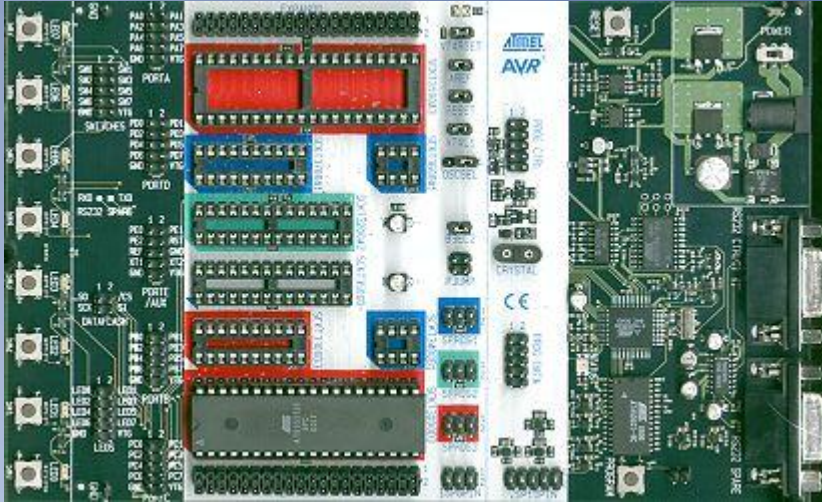
- II. μ C Programmierung
 - I. Setzen von fuses
 - II. Write to μ C
 - III. interessantes

- III. Quellen



Entwicklungstools

Stk 500

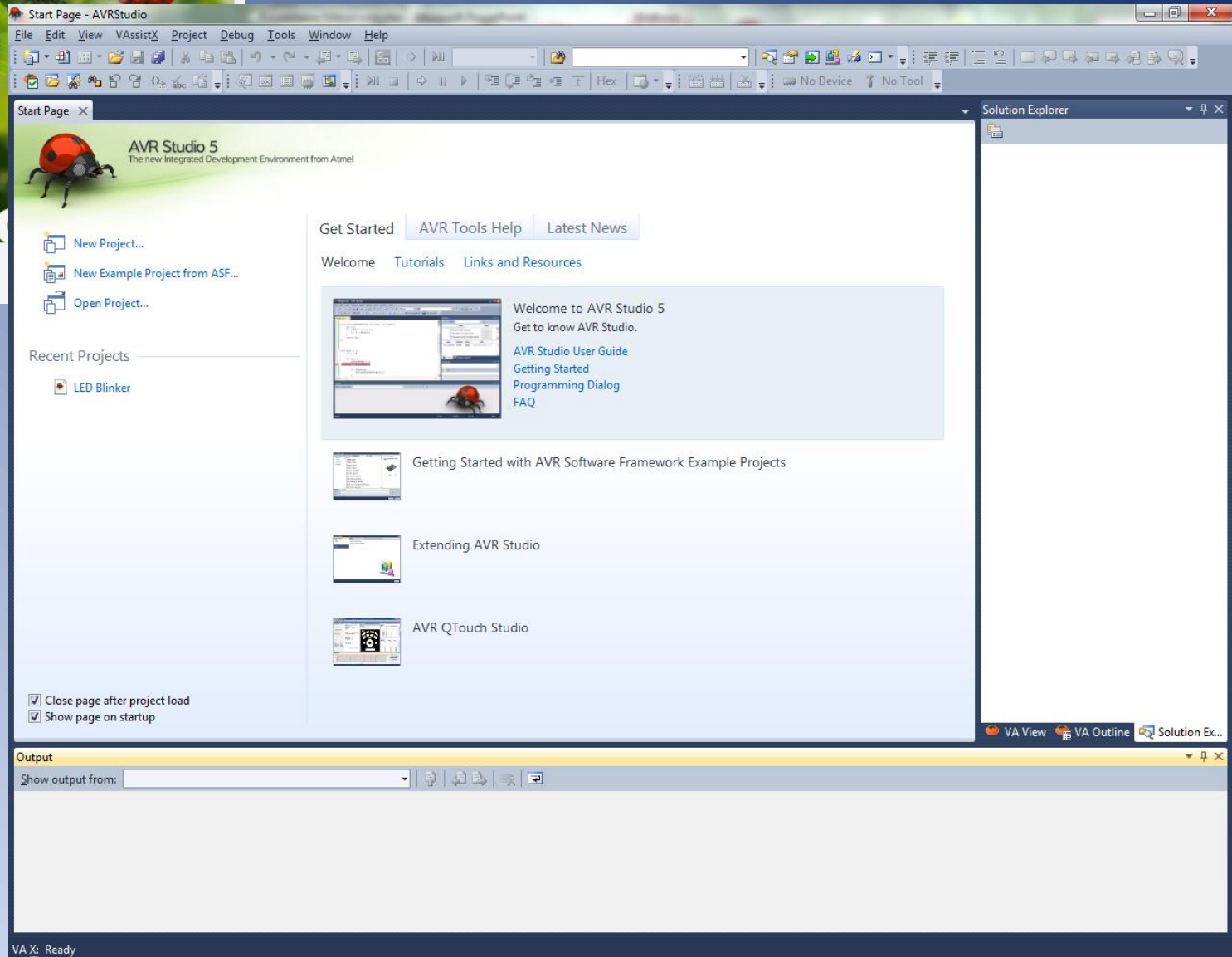
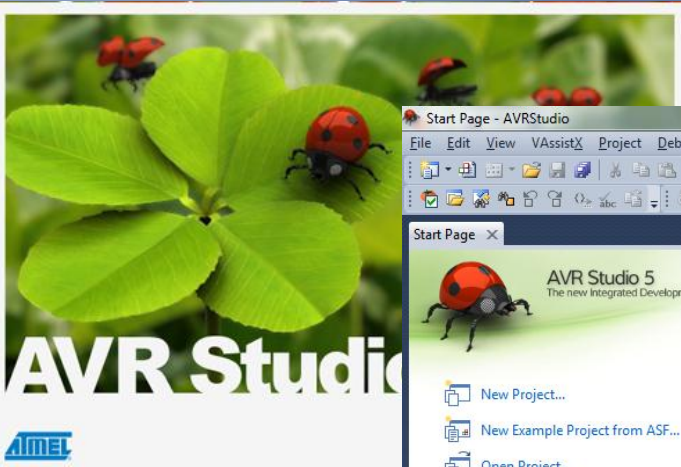


oder andere programmer plus steckboard

Avr Mk2 Programmer



Entwicklungstools



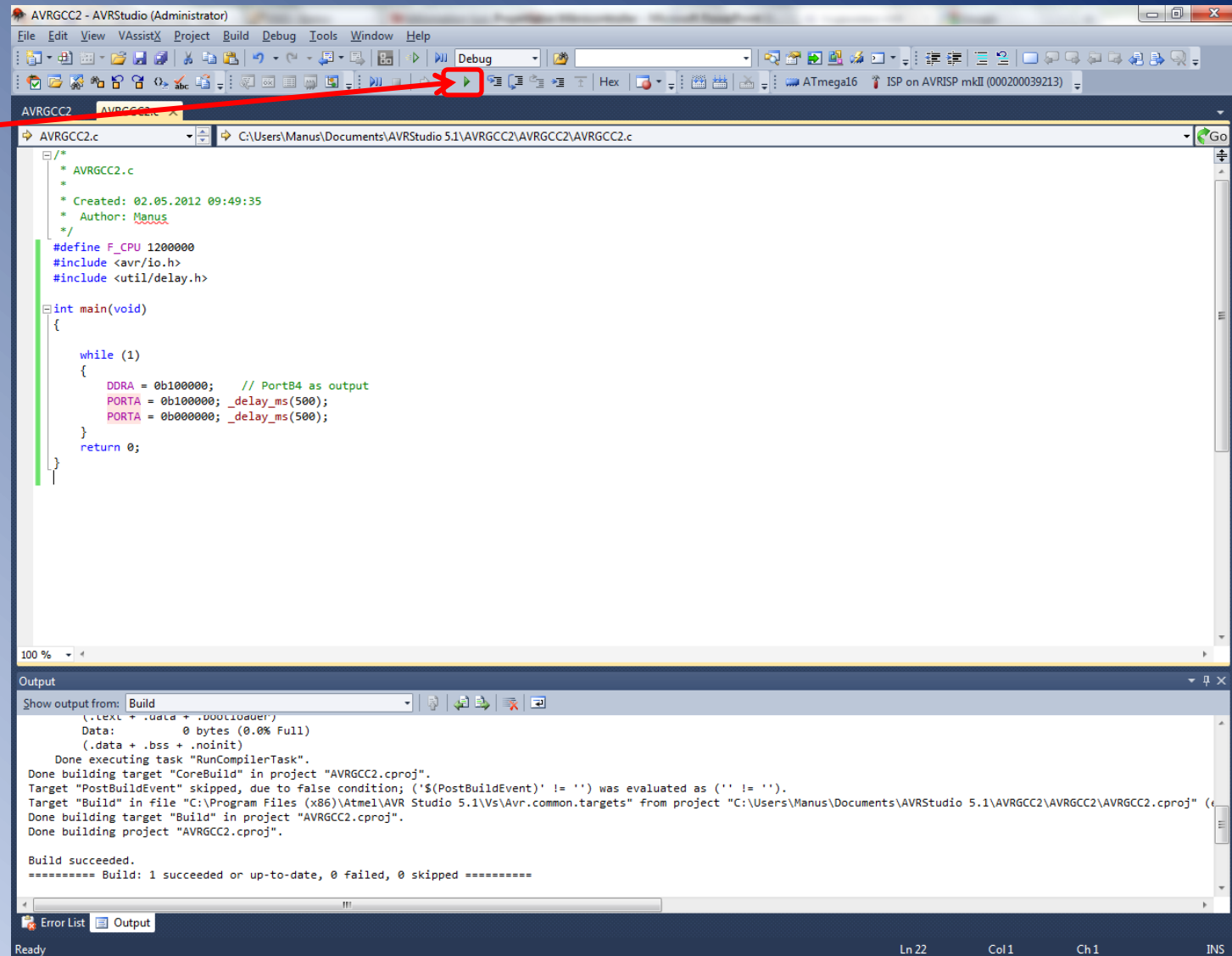
μC Programmierung

- Setzen von fuses
 - Was sind fuses ?
 - Eng: „Sicherungen“
 - Bezug μC: hardware einstellungen
 - Wozu ?
 - Hardware ändern z.b.:
 - External clock
 - Reset pin zu i/o Pin
 - Enable jtag



µc programmierung

- Speichern
- run



The screenshot shows the AVRStudio 5.1 interface. The main window displays the source code for AVRGCC2.c. The code includes comments about its creation and author, and defines F_CPU as 1200000. It includes <avr/io.h> and <util/delay.h>. The main function contains a while loop that sets DDRA to 0b100000 (PortB4 as output), writes 0b100000 to PORTA, and delays for 500ms. The output window at the bottom shows the build process, including the execution of RunCompilerTask, building the target 'CoreBuild', and the final build success message.

```
/*  
 * AVRGCC2.c  
 * Created: 02.05.2012 09:49:35  
 * Author: Manus  
 */  
  
#define F_CPU 1200000  
#include <avr/io.h>  
#include <util/delay.h>  
  
int main(void)  
{  
    while (1)  
    {  
        DDRA = 0b100000; // PortB4 as output  
        PORTA = 0b100000; _delay_ms(500);  
        PORTA = 0b000000; _delay_ms(500);  
    }  
    return 0;  
}
```

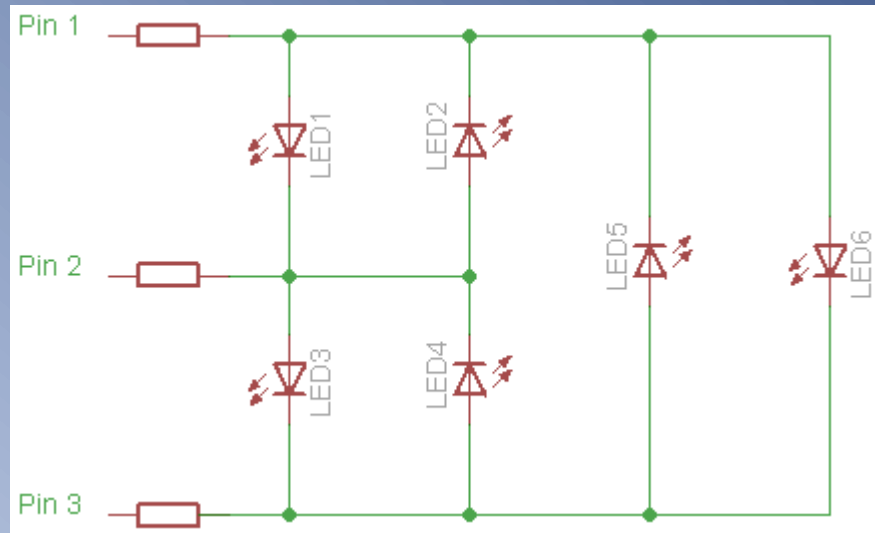
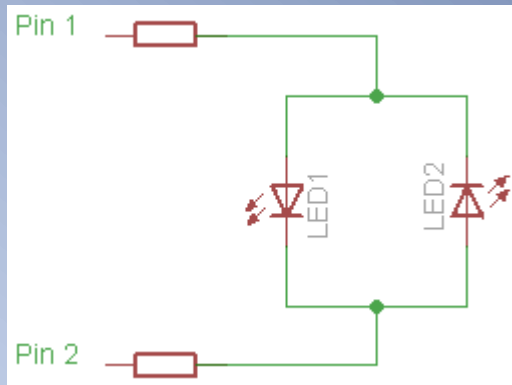
Output
Show output from: Build
(.text + .data + .bss) = 0 bytes (0.0% Full)
Data: 0 bytes (0.0% Full)
(.data + .bss + .noinit)
Done executing task "RunCompilerTask".
Done building target "CoreBuild" in project "AVRGCC2.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('\$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\AVR Studio 5.1\Vs\Avr.common.targets" from project "C:\Users\Manus\Documents\AVRStudio 5.1\AVRGCC2\AVRGCC2\AVRGCC2.cproj" (C:\Users\Manus\Documents\AVRStudio 5.1\AVRGCC2\AVRGCC2\AVRGCC2.cproj) (0 seconds)
Done building target "Build" in project "AVRGCC2.cproj".
Done building project "AVRGCC2.cproj".

Build succeeded.
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====



interessantes

- Hyperplexing oder charlieplexing
 - DEF: beschreibt die methode mit wenigen i/o ports einen größeren array von leds zu betreiben



Pins	LEDs
1	0
2	2
3	6
4	12
5	20
6	30
7	42
8	56
9	72
10	90
n	$N * (N - 1)$



Quellen

- <http://www.reichelt.de/>
- <http://www.mikrocontroller.net/>
- <http://www.the-powl.de/knowhow/charlieplexing/index.php>
- <http://www.wikipedia.org/>



Danke fürs
zuhören

