

Programming Microcontroller IntroES

First Steps with WIN AVR and AVR Studio

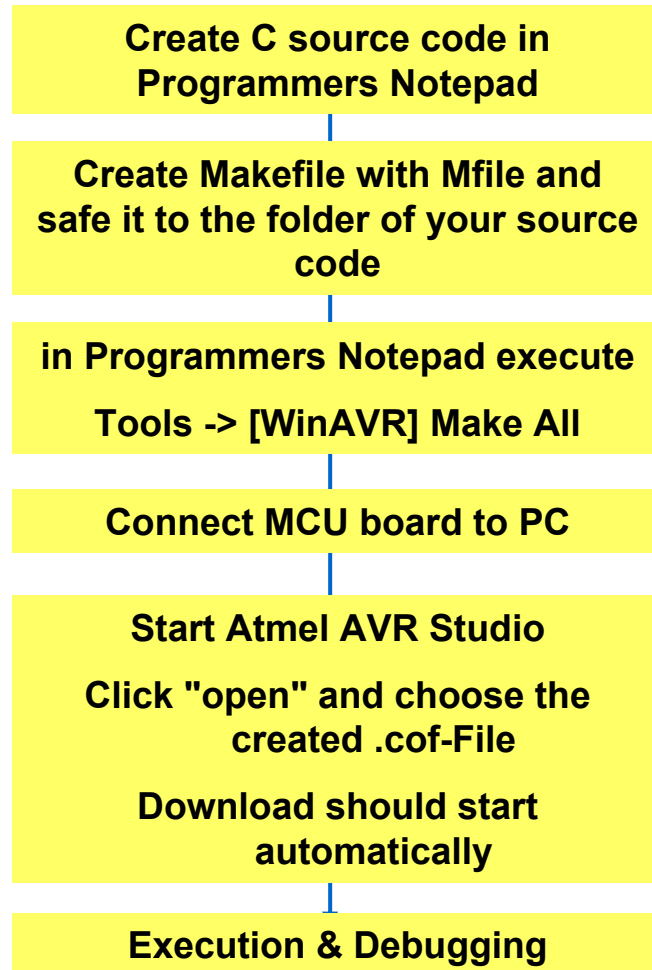
Dipl.- Ing. Falk Salewski

Lehrstuhl Informatik XI
RWTH Aachen

salewski@informatik.rwth-aachen.de

WS 05/06

Overview Design steps



- „Main File Name...“ (Name of the main program without the extension(.c))
- „MCU Type“ (ATmega16)
- “Debug format” (AVR-ext-coff)

Use the following programs:

- Programmers Notepad
- Mfile
- Atmel AVR Studio

Create C Code in Programmers Notepad

- Write your C code in Programmers Notepad
- Save the file with the [extension .c](#)
- Save all your exercises to [Z:\your_folder](#)

Create C Code in Programmers Notepad (2)

Programmers Notepad 2

File Edit View Tools Window Help

C/C++ Find

Projects test.c

Save: name.c

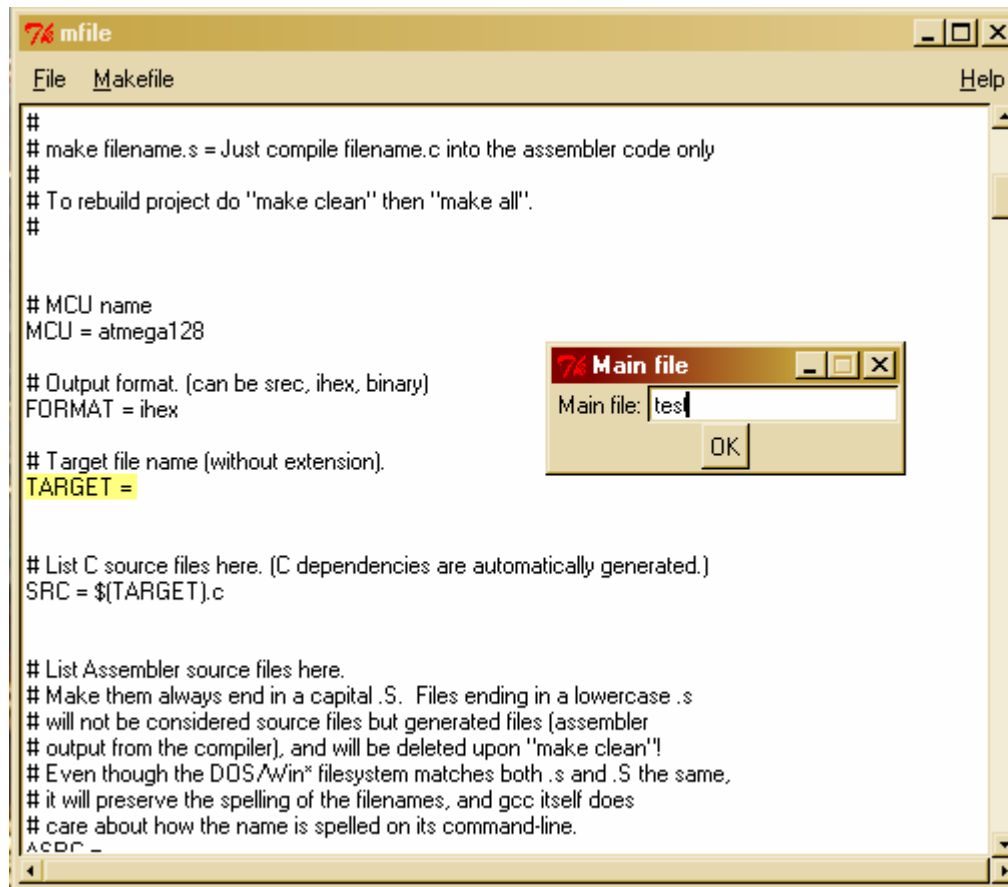
```
//LED to 5V at PINA1; SWITCH to GND at PINA0
#include <avr/io.h>

int main (void)
{
  outp(0xFE,DDRA); //PortA: Pin0: Input, Pin1..7: Output
  outp(0xFF,PORTA); //PortA: Pin0: pull up, Pin1..7: high = LED off
  while(1)
  {
    if(bit_is_set (PINA,0)) //check if PinA0 is high
    {
      cbi(PORTA,1); //clear PinA1 = LED on
    }
    else
    {
      sbi(PORTA,1); //set PinA1 = LED off
    }
  }
}
```

Create Makefile

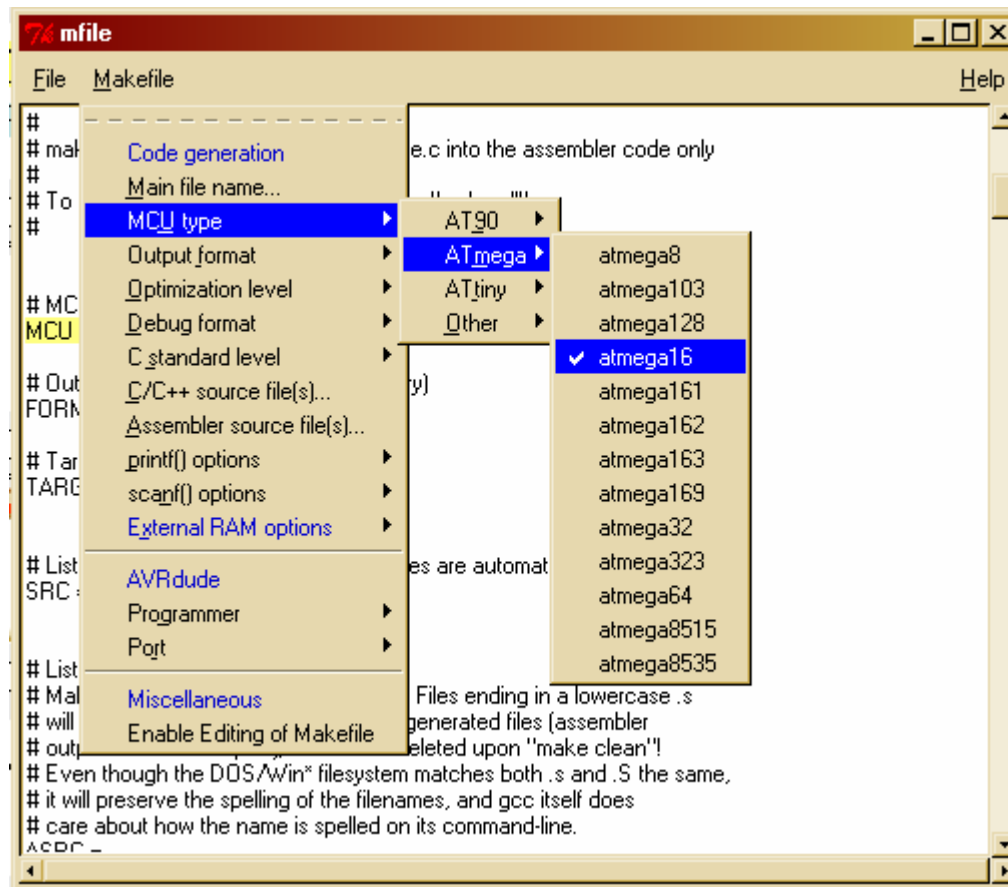
- In order to give the compiler information about the used target device and the desired output format a makefile has to be created.
- This can be done with help of a software tool called Mfile
- Start Mfile
- Follow the steps on the following slides

Using Mfile (1)



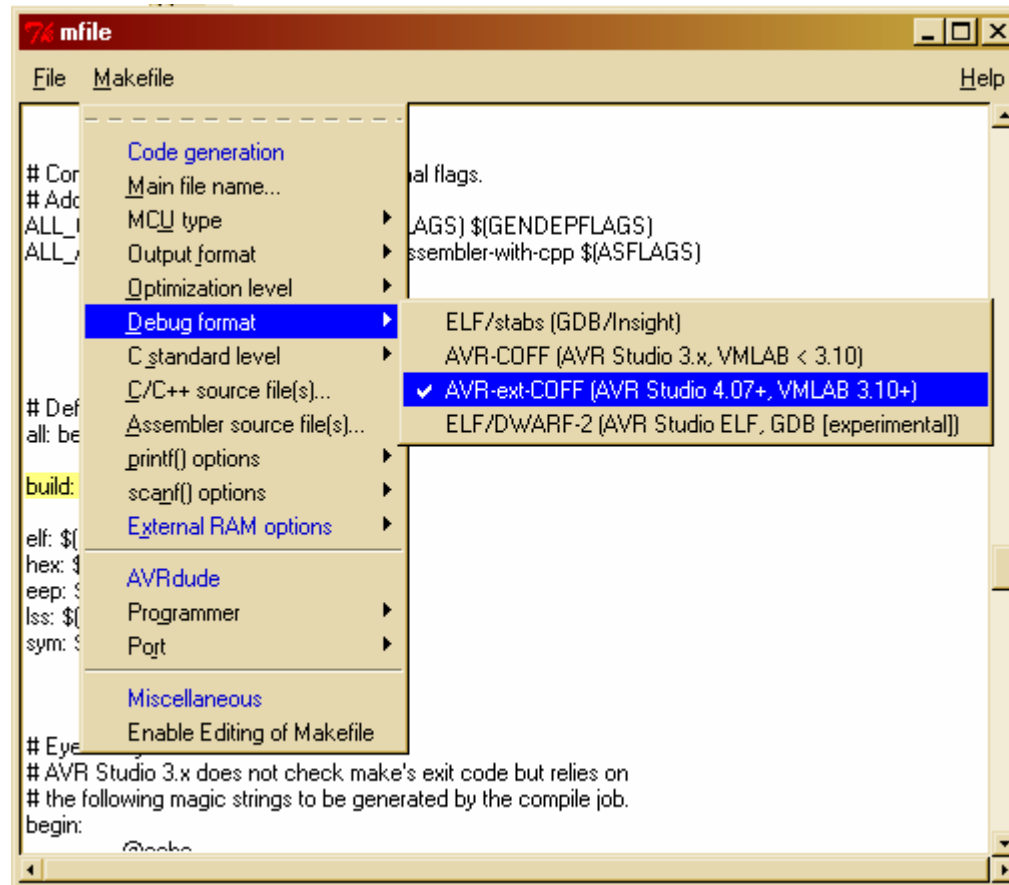
- Click Makefile > Main file name...
- Type in the name of your C-file **without the extension .c**

Using Mfile (2)



- Choose the MCU type (ATmega16 in this case)

Using Mfile (3)

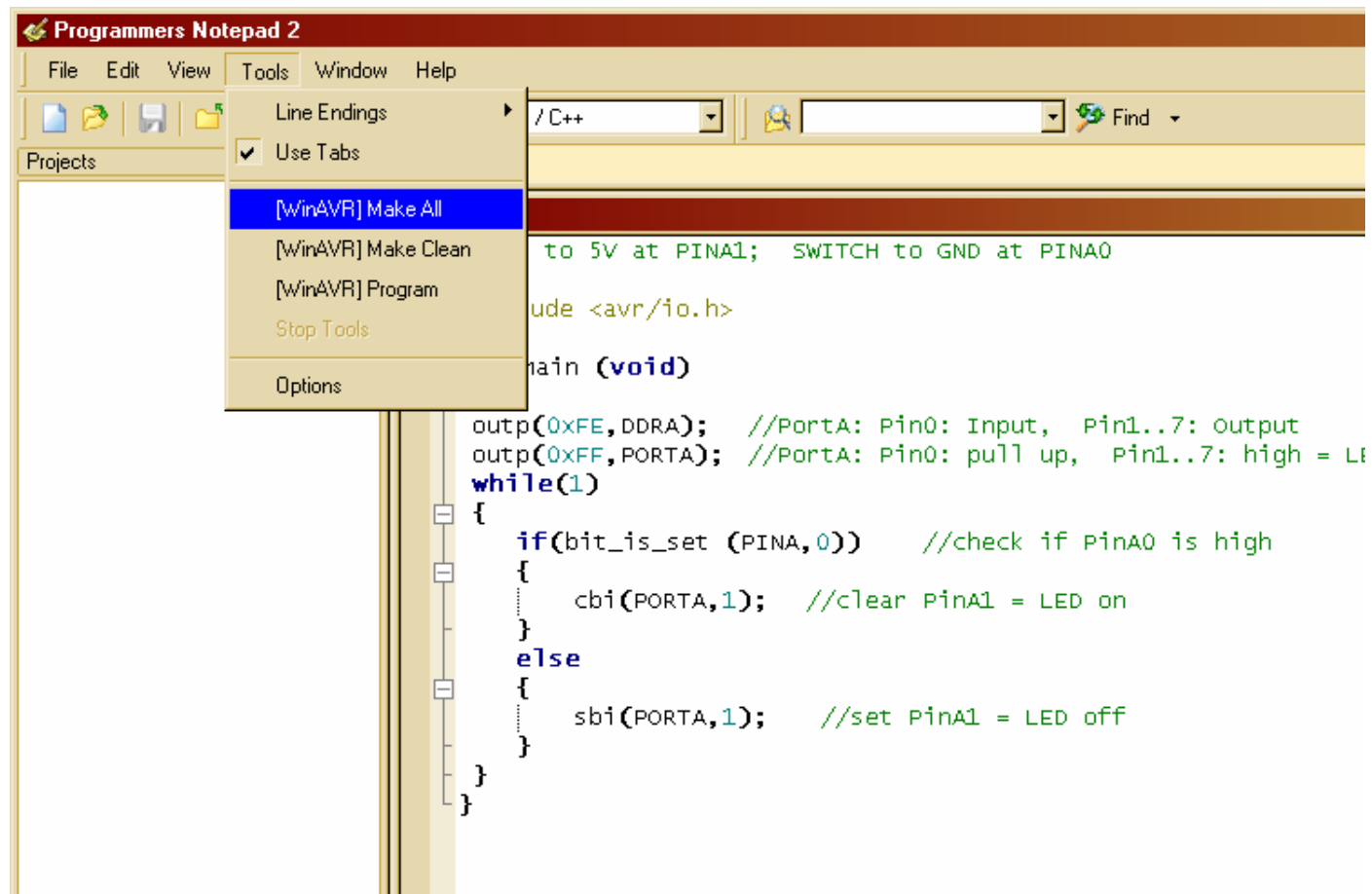


- Choose the compiler output format

Using Mfile (4)

- Now save the makefile in the same folder as your C-code by using **Save As...**
- Do not change the name of the makefile!
- Close Mfile.

Compile your C-code



- Compile your C-file by clicking on Make All

Using AVR Studio

During compilation several output files are generated.

The file *name.cof* will be used as input file for the AVR studio.

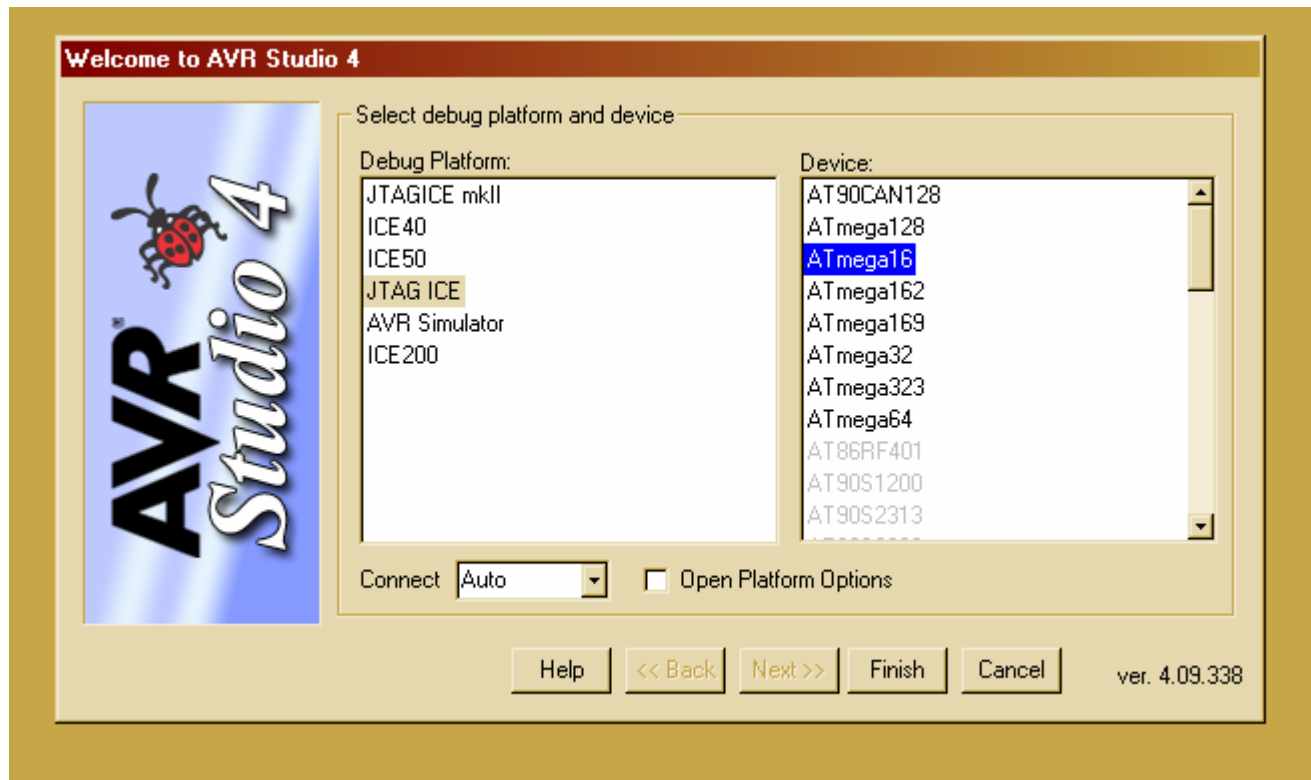
Now start AVR studio

Using AVR Studio (2)



- Open the according .cof file (test.cof in this case)

Using AVR Studio (3)



- Choose JTAG ICE and the device (ATmega16 in this case)
- Click **Finish** then...

Using AVR Studio (4)



- If you are using a USB-to-Serial converter you will get this message.
- Click on Select Port and choose COM5 or COM6 and retry
- You should get a screen like presented on the following slide



Trace Disabled

Workspace

Name	Value	Bits
Register 0-15		
Register 16-31		
Processor		
Stack Monitor		
I/O ATMEGA16		

Project I/O Info

Disassembler

+00001BF9:	FFFF	???	Data or unknown c
+00001BFA:	FFFF	???	Data or unknown c
+00001BFB:	FFFF	???	Data or unknown c
+00001BFC:	FFFF	???	Data or unknown c
+00001BFD:	FFFF	???	Data or unknown c
+00001BFE:	FFFF	???	Data or unknown c
+00001BFF:	FFFF	???	Data or unknown c
+00001C00:	FFFF	???	Data or unknown c
+00001C01:	FFFF	???	Data or unknown c
+00001C02:	FFFF	???	Data or unknown c
+00001C03:	FFFF	???	Data or unknown c
+00001C04:	FFFF	???	Data or unknown c
+00001C05:	FFFF	???	Data or unknown c
+00001C06:	FFFF	???	Data or unknown c
+00001C07:	FFFF	???	Data or unknown c
+00001C08:	FFFF	???	Data or unknown c
+00001C09:	FFFF	???	Data or unknown c
+00001C0A:	FFFF	???	Data or unknown c
+00001C0B:	FFFF	???	Data or unknown c
+00001C0C:	FFFF	???	Data or unknown c
+00001C0D:	FFFF	???	Data or unknown c
+00001C0E:	FFFF	???	Data or unknown c
+00001C0F:	FFFF	???	Data or unknown c
+00001C10:	FFFF	???	Data or unknown c
+00001C11:	FFFF	???	Data or unknown c
+00001C12:	FFFF	???	Data or unknown c
+00001C13:	FFFF	???	Data or unknown c
+00001C14:	FFFF	???	Data or unknown c
+00001C15:	FFFF	???	Data or unknown c
+00001C16:	FFFF	???	Data or unknown c
+00001C17:	FFFF	???	Data or unknown c
+00001C18:	FFFF	???	Data or unknown c

Disassembler

Memory

Program 8/16 abc. Address: 0x0 Cols: Auto

000000	0C 94 2A 00 0C 94 45 00 0C 94	. * . E .
000005	45 00 0C 94 45 00 0C 94 45 00	E . E . E .
00000A	0C 94 45 00 0C 94 45 00 0C 94	. E . E .
00000F	45 00 0C 94 45 00 0C 94 45 00	E . E . E .

Output

Loaded plugin STK500
 Loaded partfile: C:\Programme\Atmel\AVR Tools\PartDescriptionFiles\ATmega16.xml
 Loaded objectfile: test.cof

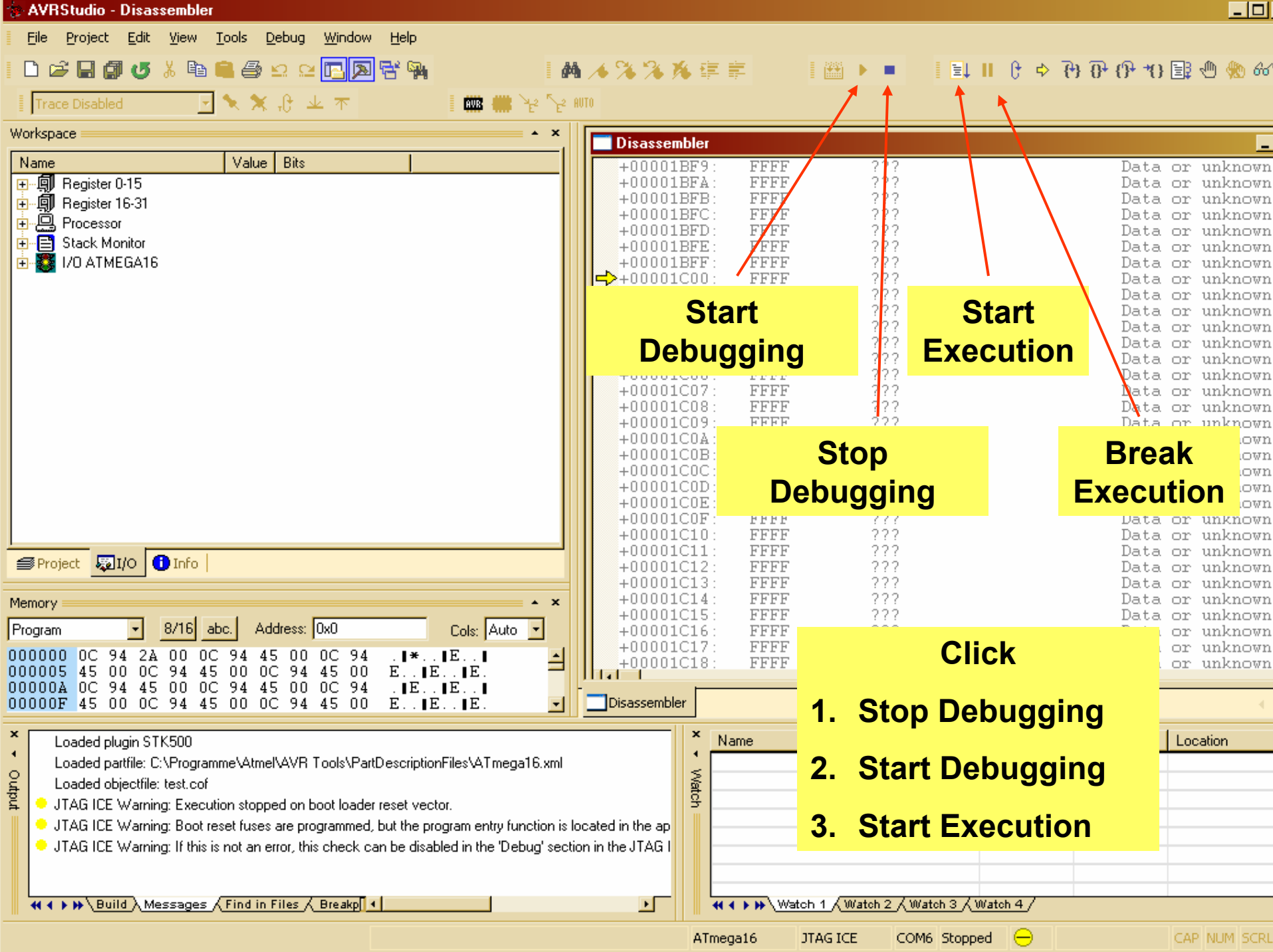
- JTAG ICE Warning: Execution stopped on boot loader reset vector.
- JTAG ICE Warning: Boot reset fuses are programmed, but the program entry function is located in the ap
- JTAG ICE Warning: If this is not an error, this check can be disabled in the 'Debug' section in the JTAG I

Build Messages Find in Files Break

Watch

Name	Value	Type	Location

Watch 1 Watch 2 Watch 3 Watch 4



**Start
Debugging**

**Start
Execution**

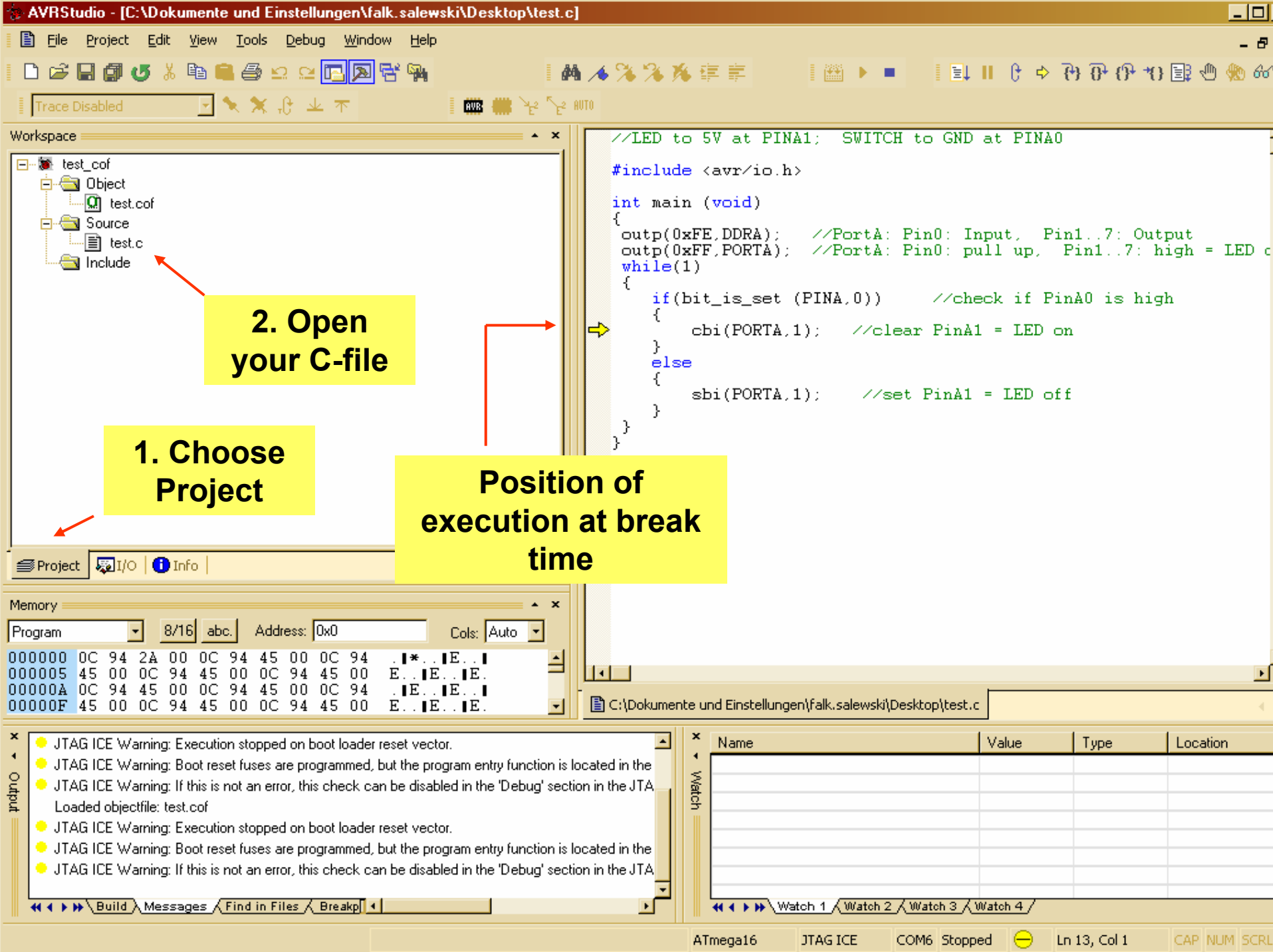
**Stop
Debugging**

**Break
Execution**

- Click**
- 1. Stop Debugging**
 - 2. Start Debugging**
 - 3. Start Execution**

Using AVR Studio (5)

- Now the code is running on the microcontroller.
- If you click on Break you stop the execution
- During break the following can be done
 - Check where in the code you stopped the execution (yellow arrow)
 - Look at all registers of the MCU
 - Add certain variables to watch window
- The code mentioned above is assembly code
- You can also debug the according C-code. This is described on the following slide



2. Open your C-file

1. Choose Project

Position of execution at break time

C:\Dokumente und Einstellungen\falk.salewski\Desktop\test.c

Output

- JTAG ICE Warning: Execution stopped on boot loader reset vector.
- JTAG ICE Warning: Boot reset fuses are programmed, but the program entry function is located in the
- JTAG ICE Warning: If this is not an error, this check can be disabled in the 'Debug' section in the JTA
- Loaded objectfile: test.cof
- JTAG ICE Warning: Execution stopped on boot loader reset vector.
- JTAG ICE Warning: Boot reset fuses are programmed, but the program entry function is located in the
- JTAG ICE Warning: If this is not an error, this check can be disabled in the 'Debug' section in the JTA

Watch

Name	Value	Type	Location

Workspace

Name	Value	Bits	A...
I/O ATMEGA16			
AD_CONVERTER			
ANALOG_COMPARATOR			
BOOT_LOAD			
CPU			
EEPROM			
EXTERNAL_INTERRUPT			
JTAG			
PORTA			
PORTA	0xFD	■■■■■■■□■	0x1B (0x3B)
DDRA	0xFE	■■■■■■■□■	0x1A (0x3A)
PINA	0xFD	■■■■■■■□■	0x19 (0x39)
PORTB			
PORTC			
PORTD			
SPI			
TIMER_COUNTER_0			
TIMER_COUNTER_1			
TIMER_COUNTER_2			
TWI			

```
//LED to 5V at PINA1; SWITCH to GND at PINA0
#include <avr/io.h>

int main (void)
{
  outp(0xFE,DDRA); //PortA: Pin0: Input, Pin1..7: Output
  outp(0xFF,PORTA); //PortA: Pin0: pull up, Pin1..7: high = LED c
  while(1)
  {
    if(bit_is_set (PINA,0)) //check if PinA0 is high
    {
      cbi(PORTA,1); //clear PinA1 = LED on
    }
    else
    {
      sbi(PORTA,1); //set PinA1 = LED off
    }
  }
}
```

Examine dedicated registers

Memory

Program 8/16 abc Address: 0x0 Cols: Auto

000000	0C 94 2A 00 0C 94 45 00 0C 94	..*.. E..
000005	45 00 0C 94 45 00 0C 94 45 00	E.. E.. E..
00000A	0C 94 45 00 0C 94 45 00 0C 94	..E.. E..
00000F	45 00 0C 94 45 00 0C 94 45 00	E.. E.. E..
000014	0C 94 45 00 0C 94 45 00 0C 94	..E.. E..

Read all memory locations

Watch variables

Output

- JTAG ICE Warning: Execution stopped on boot loader reset vector.
- JTAG ICE Warning: Boot reset fuses are programmed, but the program entry function is located in the
- JTAG ICE Warning: If this is not an error, this check can be disabled in the 'Debug' section in the JTA
- Loaded objectfile: test.cof
- JTAG ICE Warning: Execution stopped on boot loader reset vector.

Watch

Name	Value	Type	Location

Watch 1 Watch 2 Watch 3 Watch 4

File Project Edit View Tools Debug Window Help

Trace Disabled

AVR AUTO

Debugger toolbar with icons for Run, Step Over, Step Into, etc.

Workspace

Name	Value	Bits	A...
I/O ATMEGA16			
AD_CONVERTER			
ANALOG_COMPARATOR			
BOOT_LOAD			
CPU			
EEPROM			
EXTERNAL_INTERRUPT			
JTAG			
PORTA			
PORTA	0xFD	■■■■■■■□	0x1B (0x3B)
DDRA	0xFE	■■■■■■■□	0x1A (0x3A)
PINA	0xFD	■■■■■■■□	0x19 (0x39)
PORTB			
PORTC			
PORTD			
SPI			
TIMER_COUNTER_0			
TIMER_COUNTER_1			
TIMER_COUNTER_2			
TWI			

```

//LED to 5V at PINA1; SWITCH to GND at PINA0

#include <avr/io.h>

int main (void)
{
  outp(0xFE,DDRA); //PortA: Pin0: input, Pin1..7: Output
  outp(0xFF,PORTA); //PortA: Pin0: pull up, Pin1..7: high = LED on
  while(1)
  {
    if(bit_is_set (PINA,1)) //PinA0 is high
    {
      cbi(PORTA,1); //set PinA1 = LED on
    }
    else
    {
      sbi(PORTA,1); //set PinA1 = LED off
    }
  }
}

```

Reset (yellow box pointing to the start of main)

Step-wise execution (yellow box pointing to the while loop)

Memory

Program 8/16 abc. Address: 0x0 Cols: Auto

000000	DC 94 2A 00 0C 94 45 00 0C 94	. * . . E . .
000005	45 00 0C 94 45 00 0C 94 45 00	E . . E . . E . .
00000A	DC 94 45 00 0C 94 45 00 0C 94	. E . . E . .
00000F	45 00 0C 94 45 00 0C 94 45 00	E . . E . . E . .
000014	DC 94 45 00 0C 94 45 00 0C 94	. E . . E . .

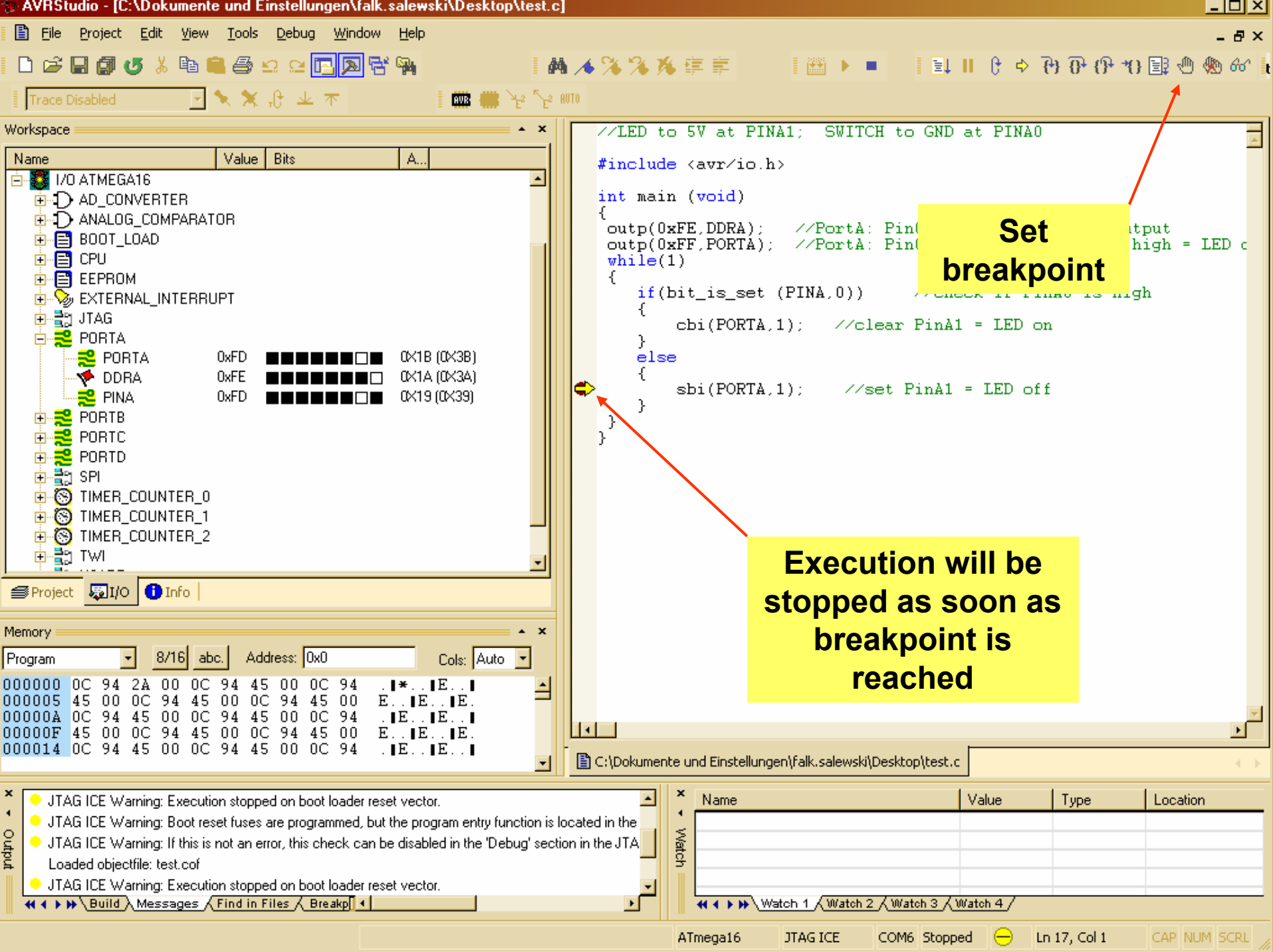
Output

- JTAG ICE Warning: Execution stopped on boot loader reset vector.
- JTAG ICE Warning: Boot reset fuses are programmed, but the program entry function is located in the
- JTAG ICE Warning: If this is not an error, this check can be disabled in the 'Debug' section in the JTAG Loaded objectfile: test.cof
- JTAG ICE Warning: Execution stopped on boot loader reset vector.

Watch

Name	Value	Type	Location

Watch 1 Watch 2 Watch 3 Watch 4



More Information

- http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725
- <http://sourceforge.net/projects/winavr>
- <http://www.mikrocontroller.net/> (GERMAN only)