

# REM

## 0. Erste Schritte

---

**Jürgen Mäter**

Arbeitsgruppe Modellierung und Simulation

Lehrstuhl für Informatik 4  
TU Dortmund



Juergen.Maeter@udo.edu  
Campus Süd R402/GB5  
Tel.: 0231 755 2411

<http://ls12-www.cs.tu-dortmund.de/%7Emarwedel/eda/08-eda-index.html>

---

# REM

## 0. Erste Schritte

---

**Übungsgruppenleiter:**

- \* SystemC: Jürgen Mäter**
  - \* VHDL/FPGAs: Sascha Plazar**
  - \* Synthese: Daniel Cordes**
-



# REM

## 0. Erste Schritte

---

- \* **SystemC: Installation**
  - \* **Übungsblatt 1**
-

---

# Rechnergestützter Entwurf von Mikroelektronik

## Präsenzübung

# Literatur

---

## ■ SystemC

- SystemC Language Reference Manual, <http://www.SystemC.org>
- H. Müller et al.: C++ für Java-Programmierer
- Black, Donovan: SystemC: From the ground up, *Springer*, 2004 (Präsenzbestand BI)

## ■ FPGAs

- <http://www.xilinx.com/univ/index.htm>
- Reichard, Schwarz: VHDL-Synthese, *Oldenbourg*, 2003

## ■ Syntheseverfahren

- P. Marwedel: Synthese und Simulation von VLSI-Systemen, *Hanser-Verlag*, 1993
- P. Marwedel: Skript der Vorgängervorlesung, 2001

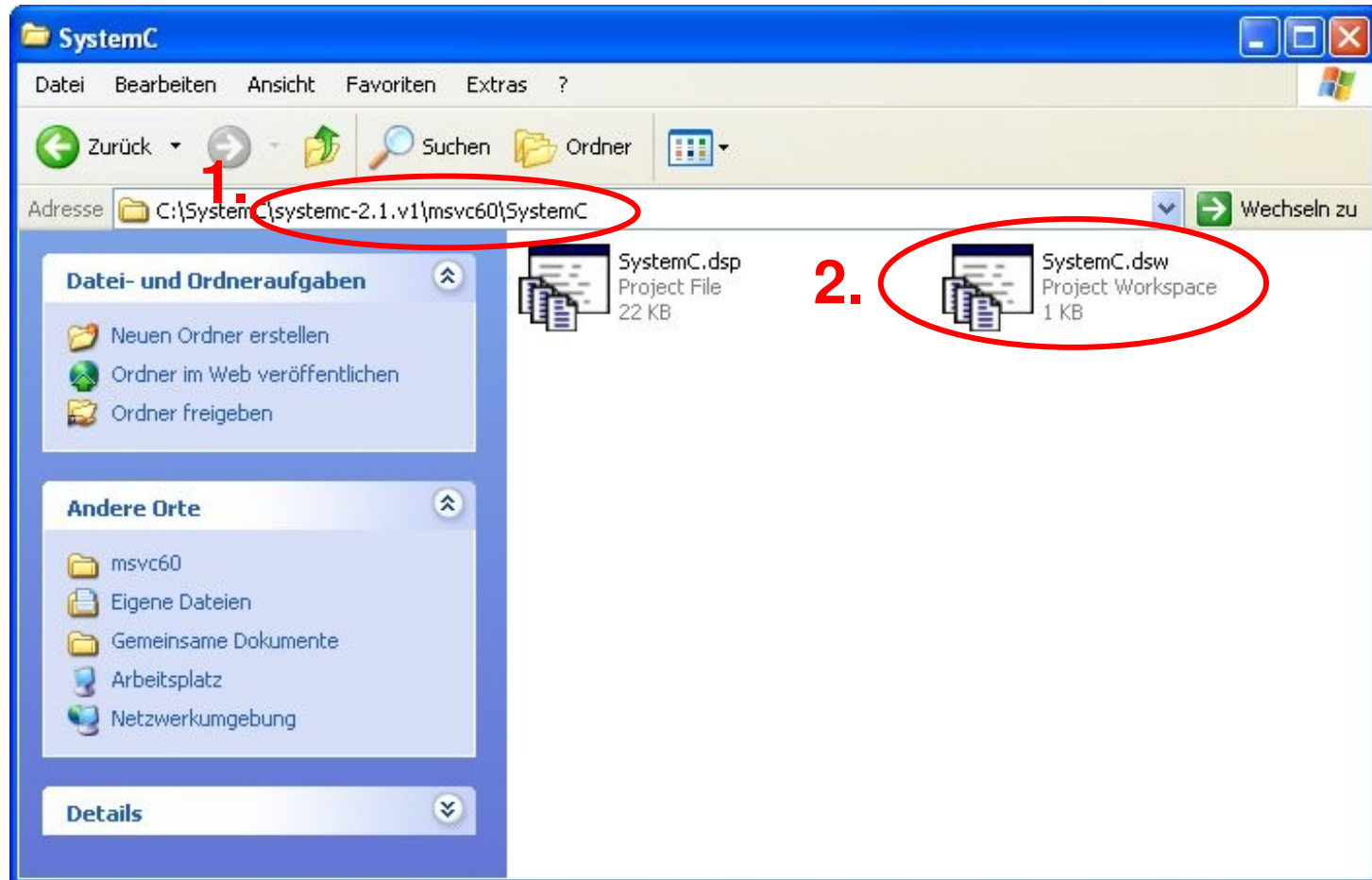
# SystemC - Installationsvorbereitungen

---

- SystemC Library:
  - auf [www.SystemC.org](http://www.SystemC.org) registrieren und einloggen
  - Download der Bibliothek und des Manuals
  - Bibliothek entpacken
  
- Windows:
  - Visual Studio 6.0 herunterladen (=> IRB)
  - Installieren
  
- Linux:
  - Makefile erstellen

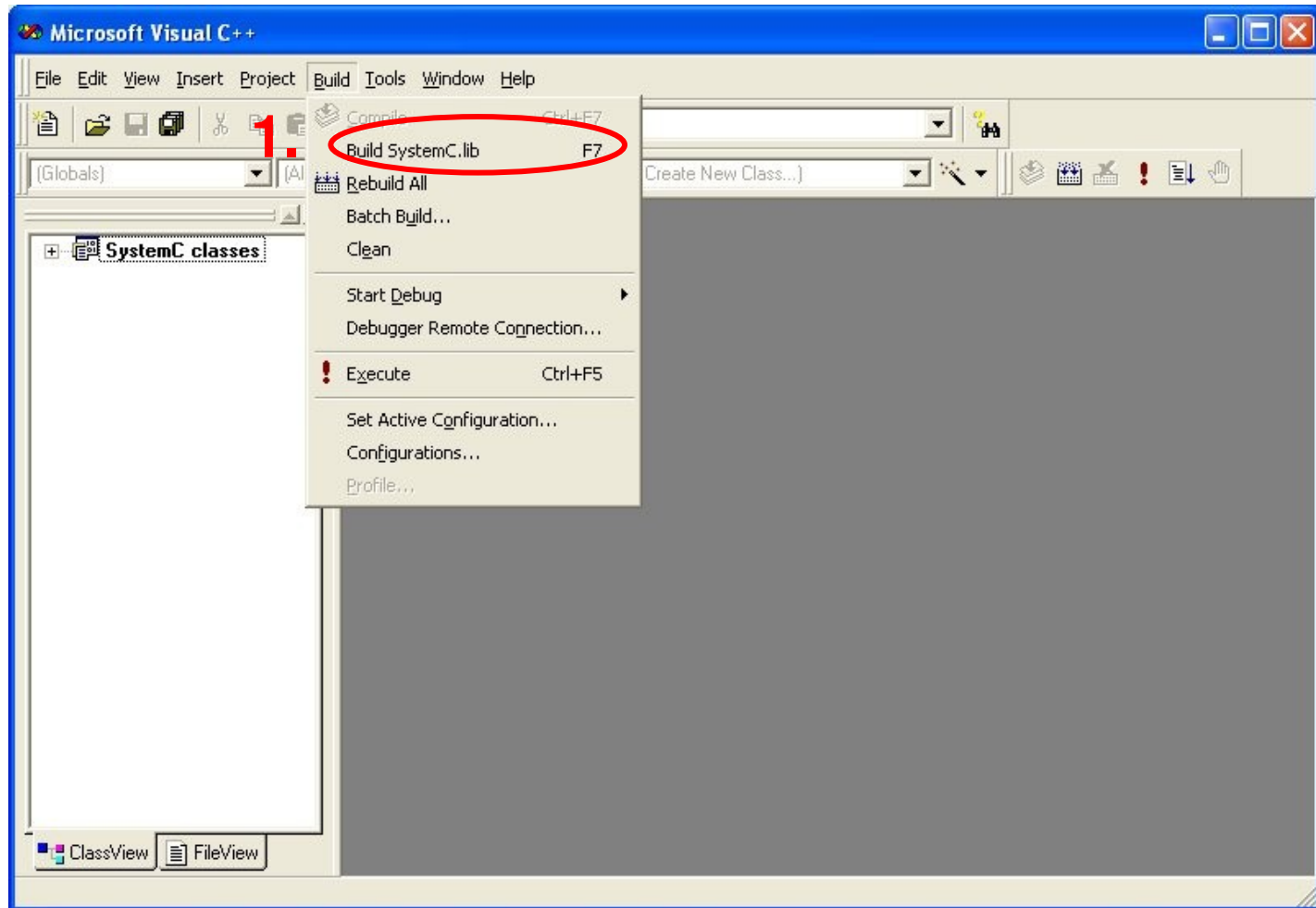
# SystemC – Installation unter Windows (1)

- Workspace öffnen:



# SystemC – Installation unter Windows (2)

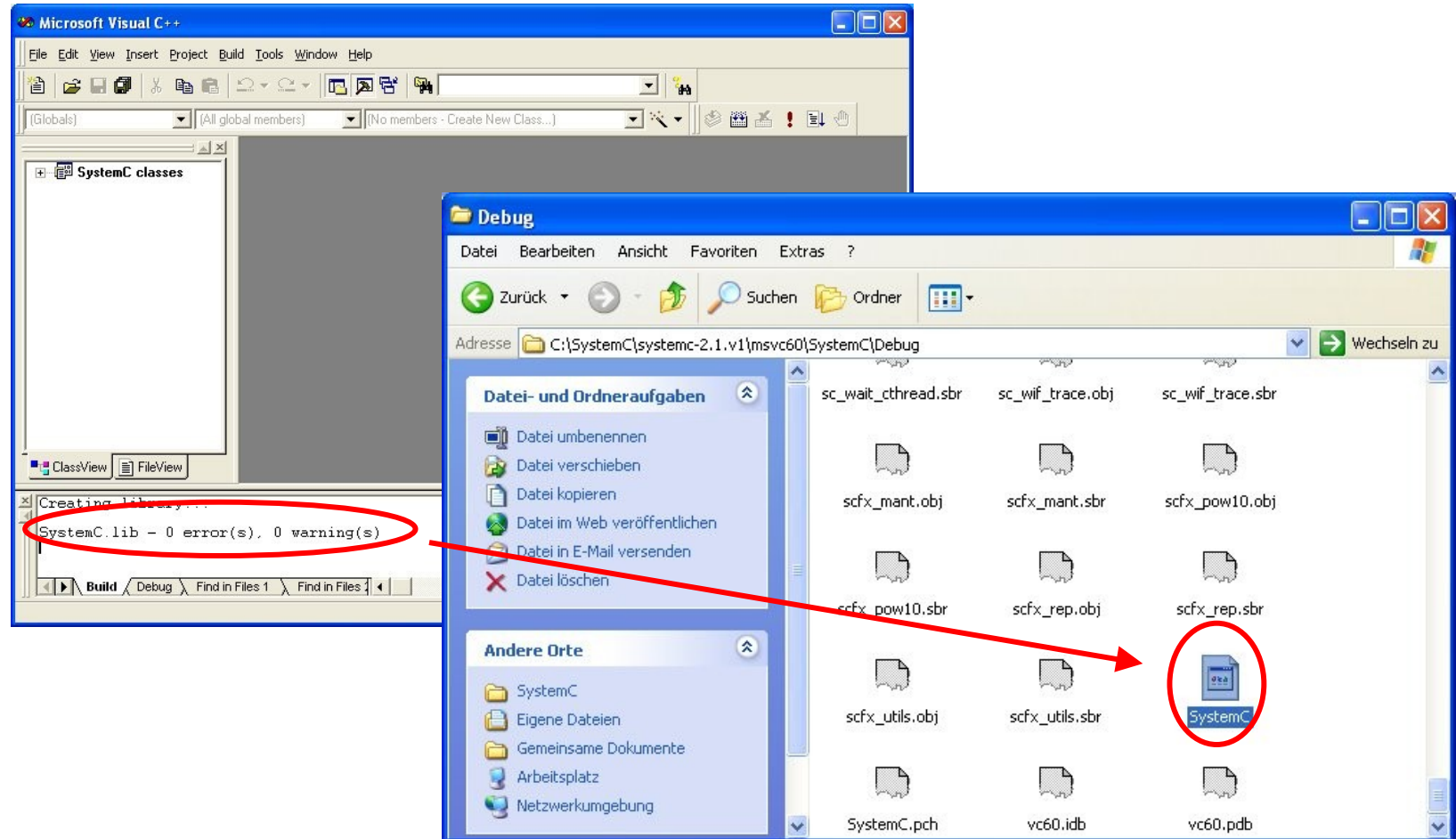
- SystemC übersetzen („F7“):





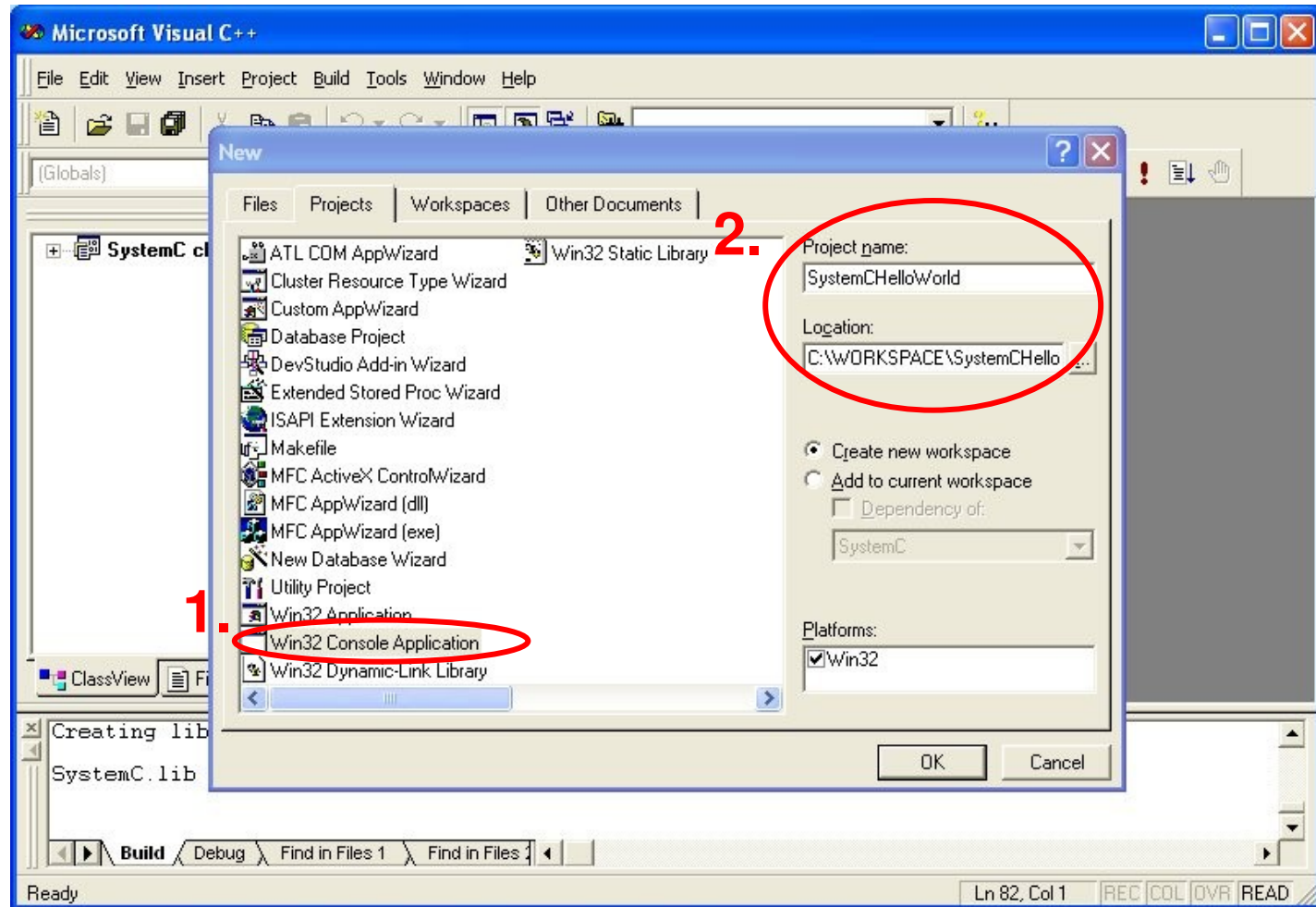
# SystemC – Installation unter Windows (3)

- Bibliothek wurde erzeugt:



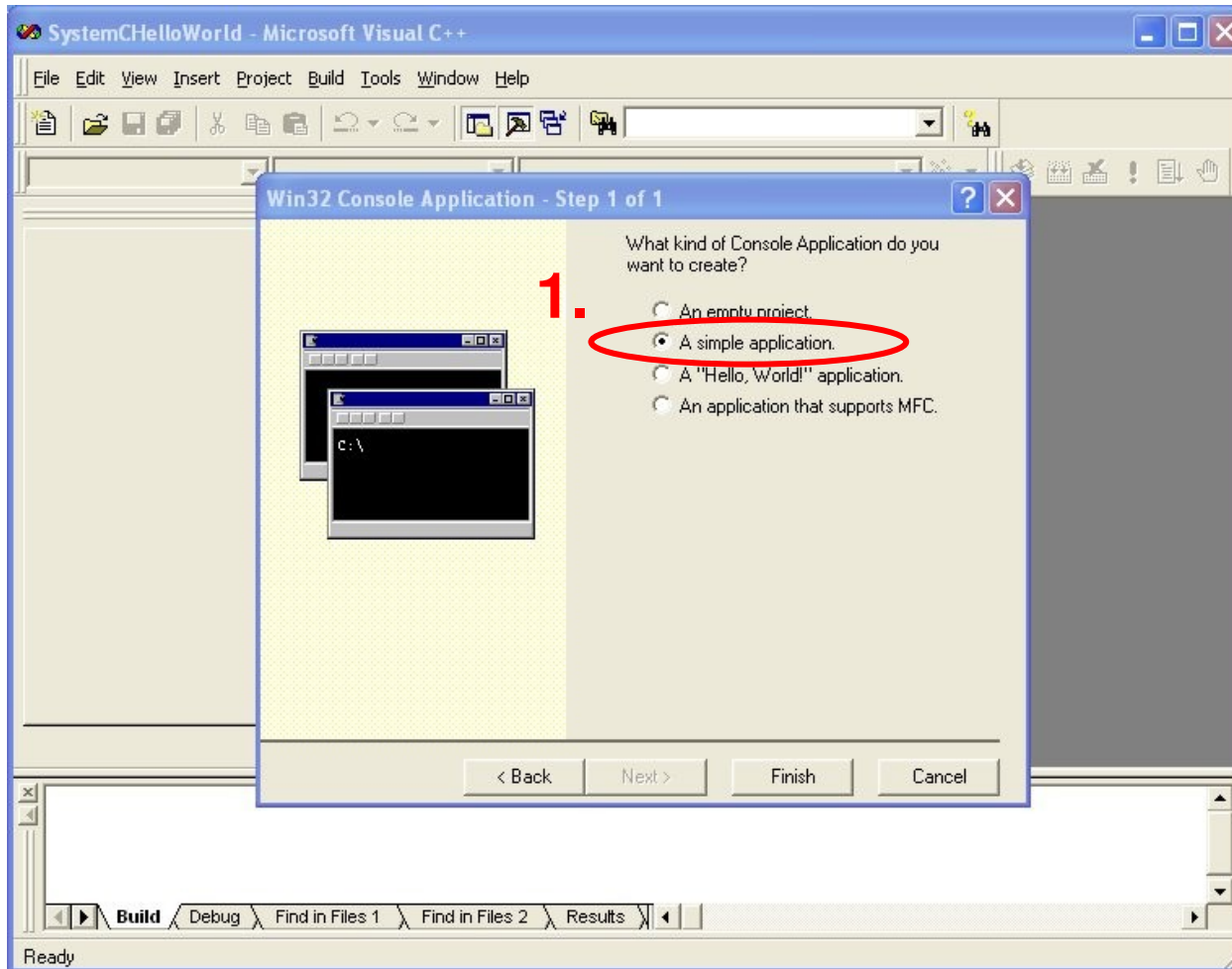
# SystemC – 1. Projekt unter Windows (1)

- Neues Projekt anlegen („Strg + N“):



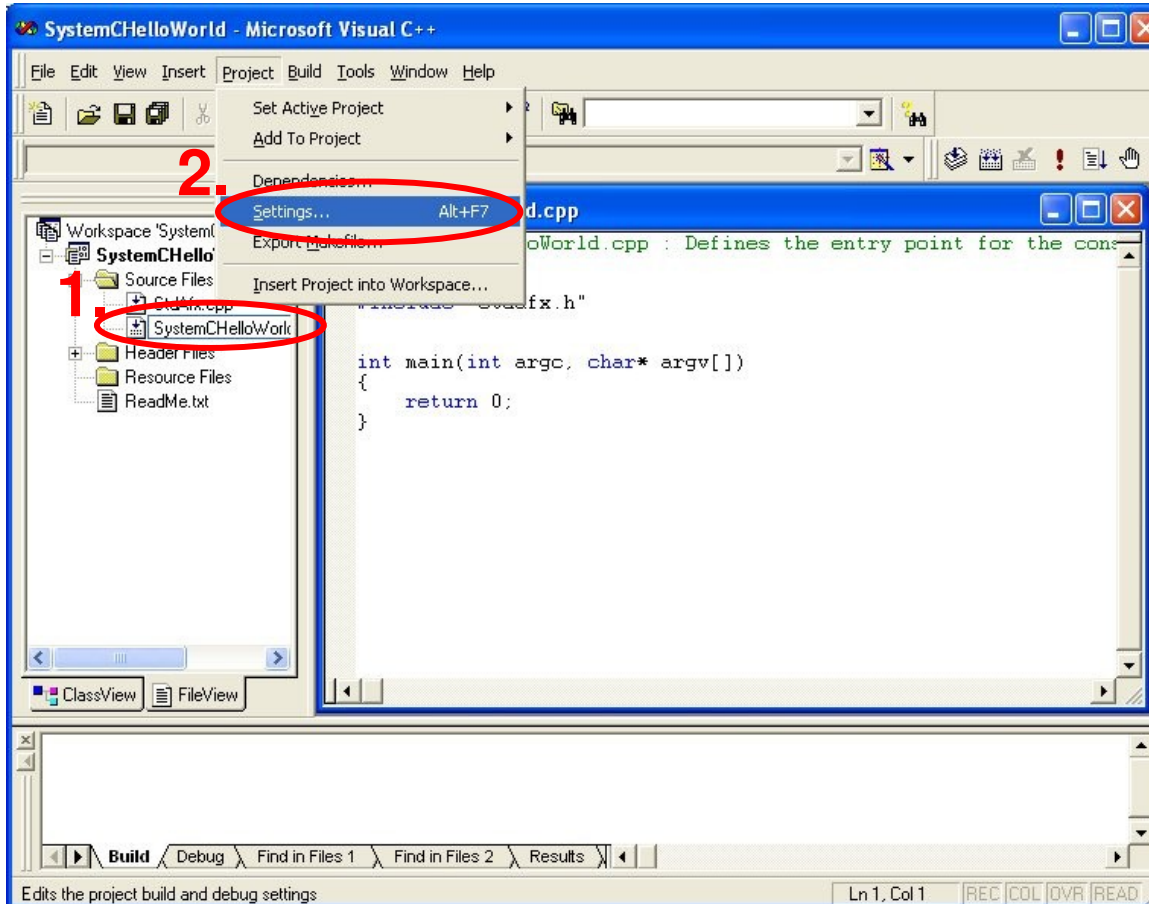
# SystemC – 1. Projekt unter Windows (2)

- Neues Projekt anlegen:



# SystemC – 1. Projekt unter Windows (3)

- Projekt anpassen („Strg + F7“):

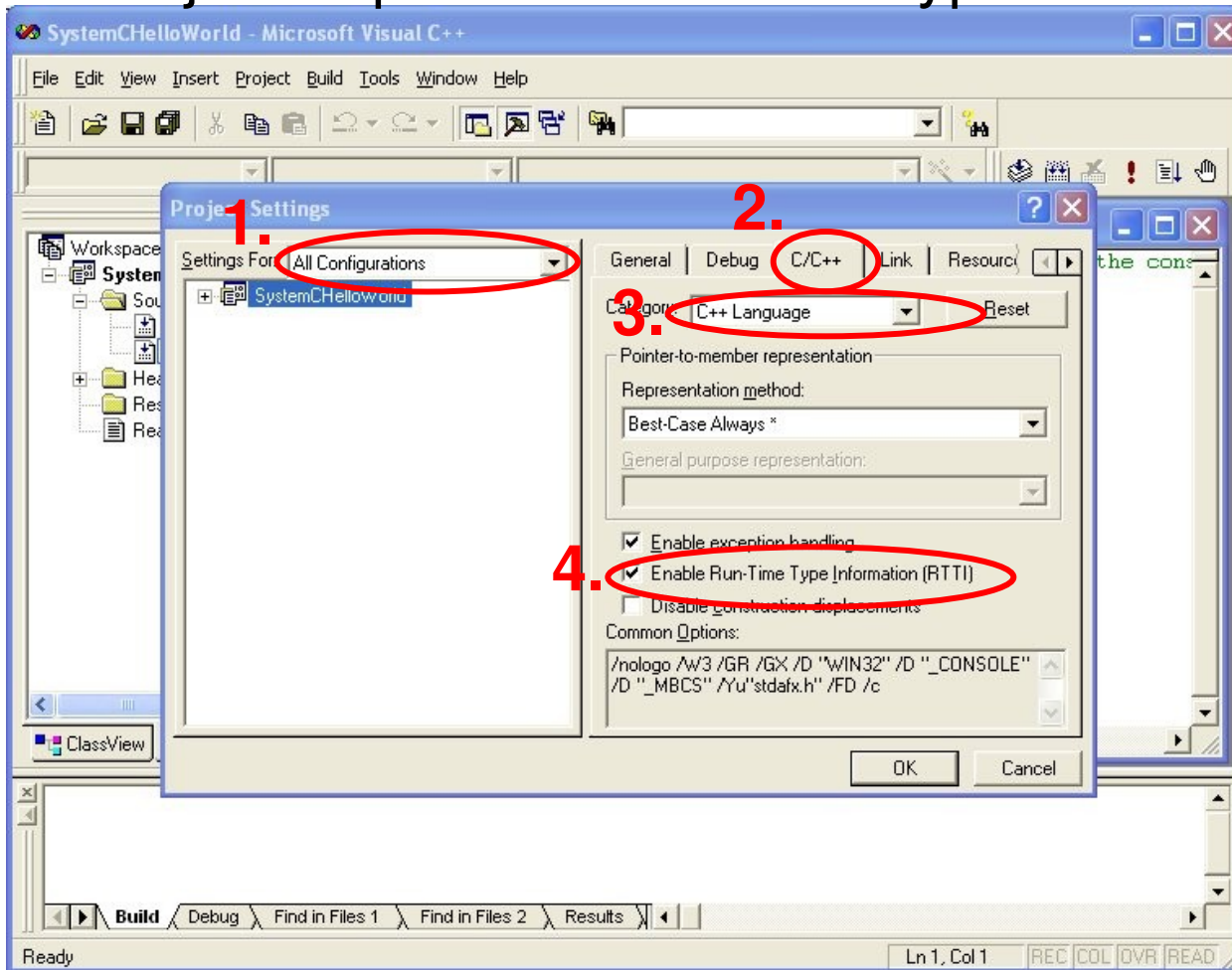


1. SourceFile  
*SystemCHelloWorld.cpp*  
öffnen

5. *Project -> Settings...*

# SystemC – 1. Projekt unter Windows (4)

- Projekt anpassen - Laufzeit-Typinfo einschalten:



1. Settings For:  
*AllConfigurations*

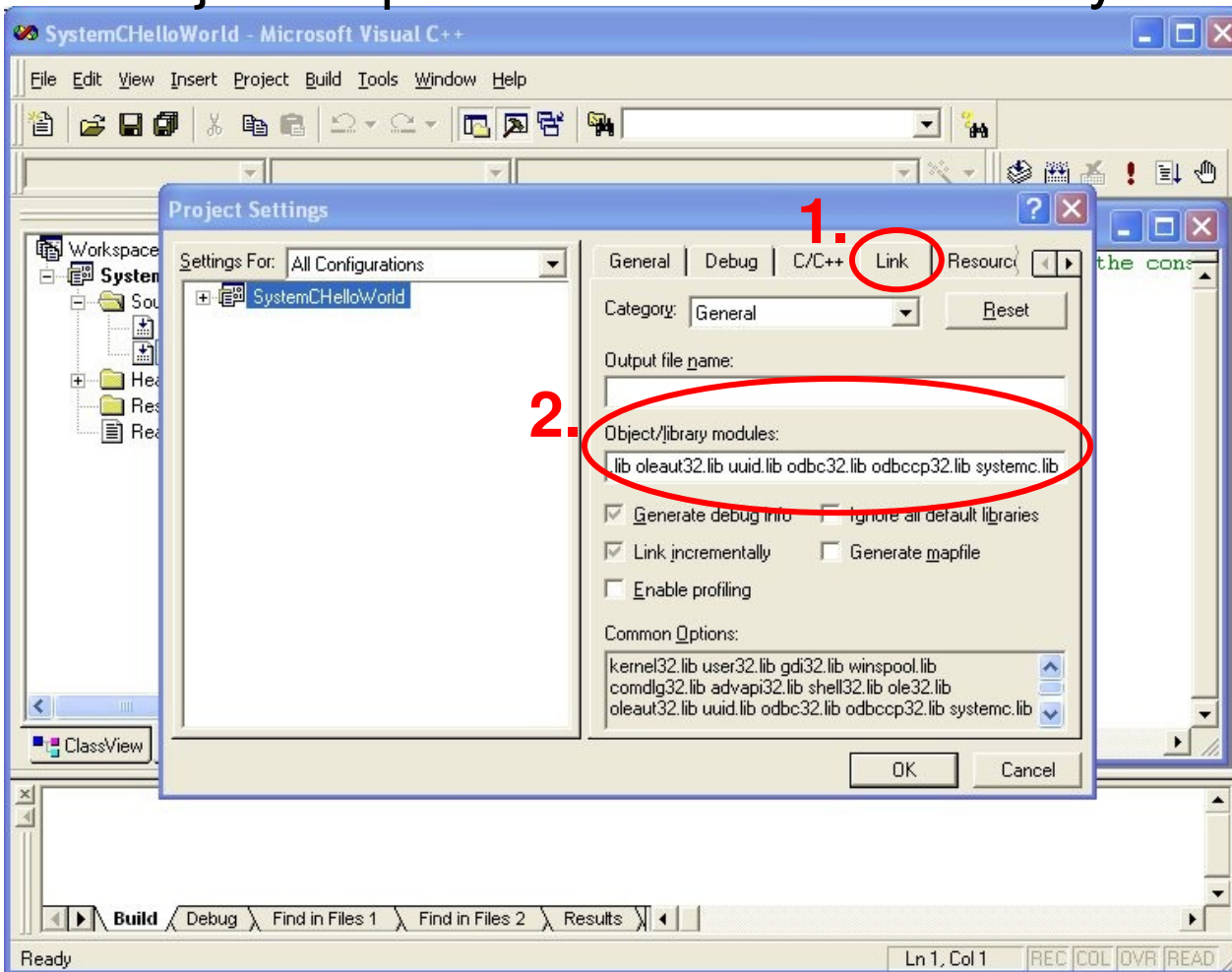
4. C/C++

6. Category:  
*C++ Language*

9. *Enable RTTI*

# SystemC – 1. Projekt unter Windows (5)

- Projekt anpassen - Einbinden der SystemC-Library:

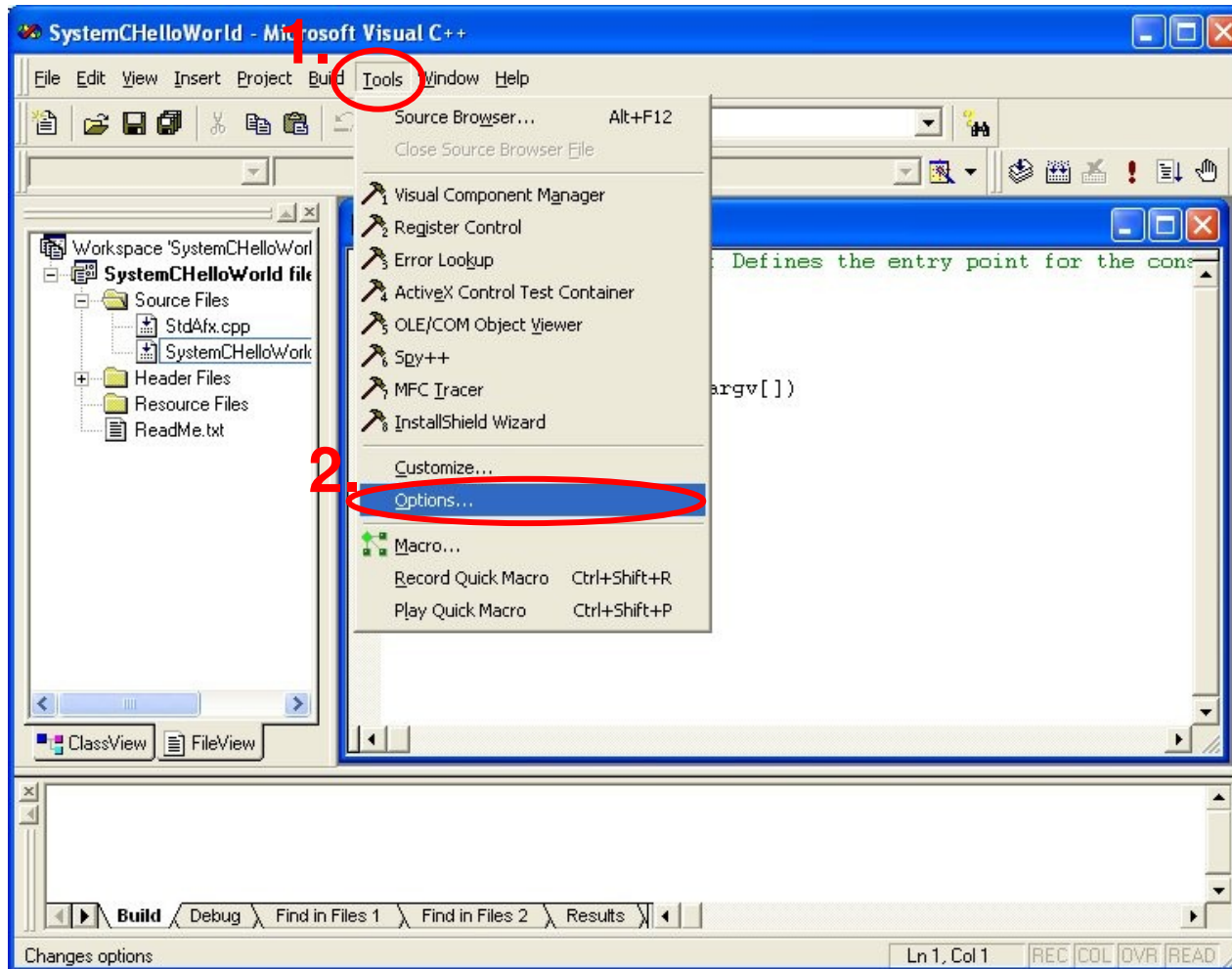


1. Link

3. Object/library modules:  
*systemc.lib*  
hinzufügen

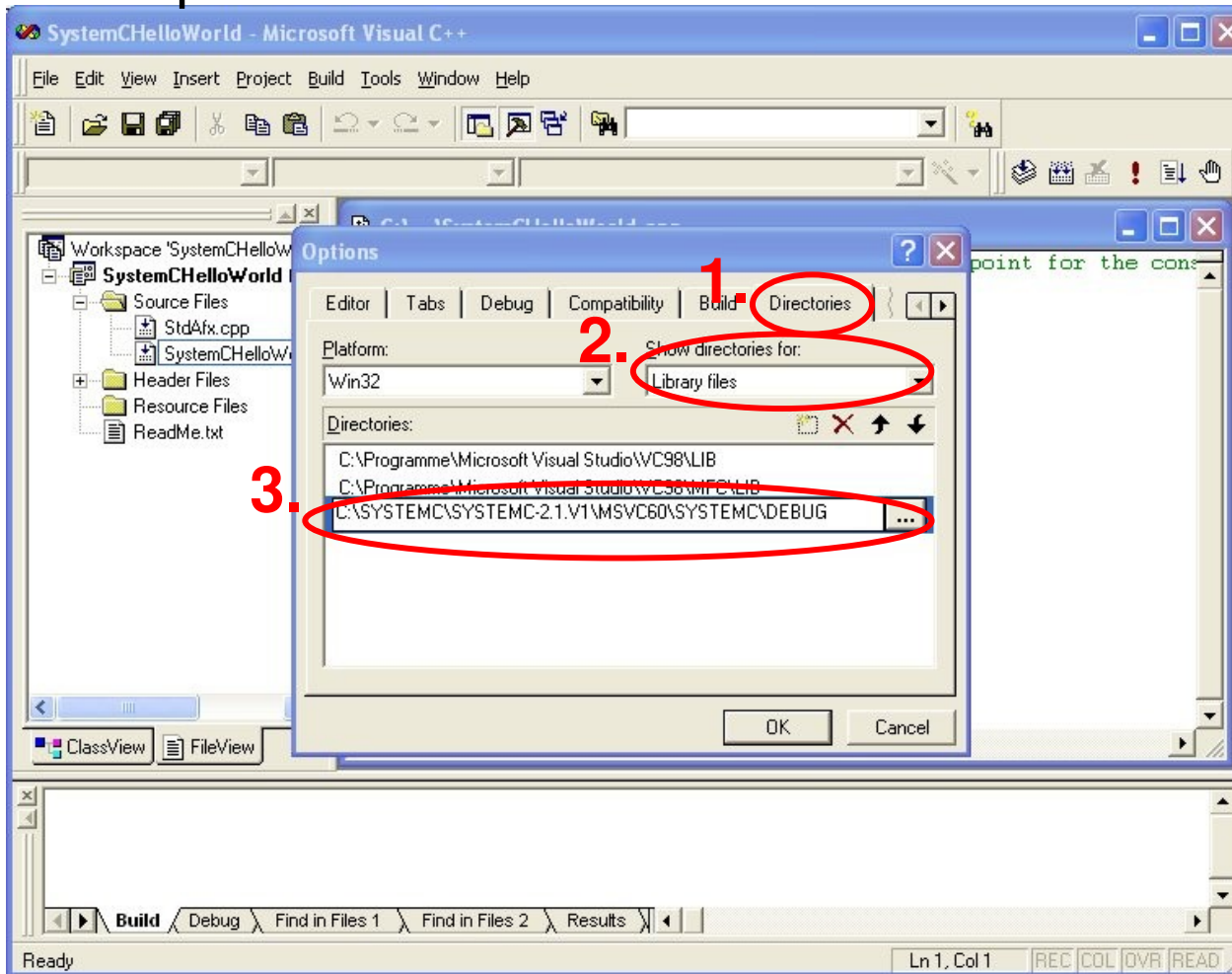
# SystemC – 1. Projekt unter Windows (6)

- Update der Include- und Bibliotheksverzeichnisse:



# SystemC – 1. Projekt unter Windows (7)

- Update der Include- und Bibliotheksverzeichnisse:

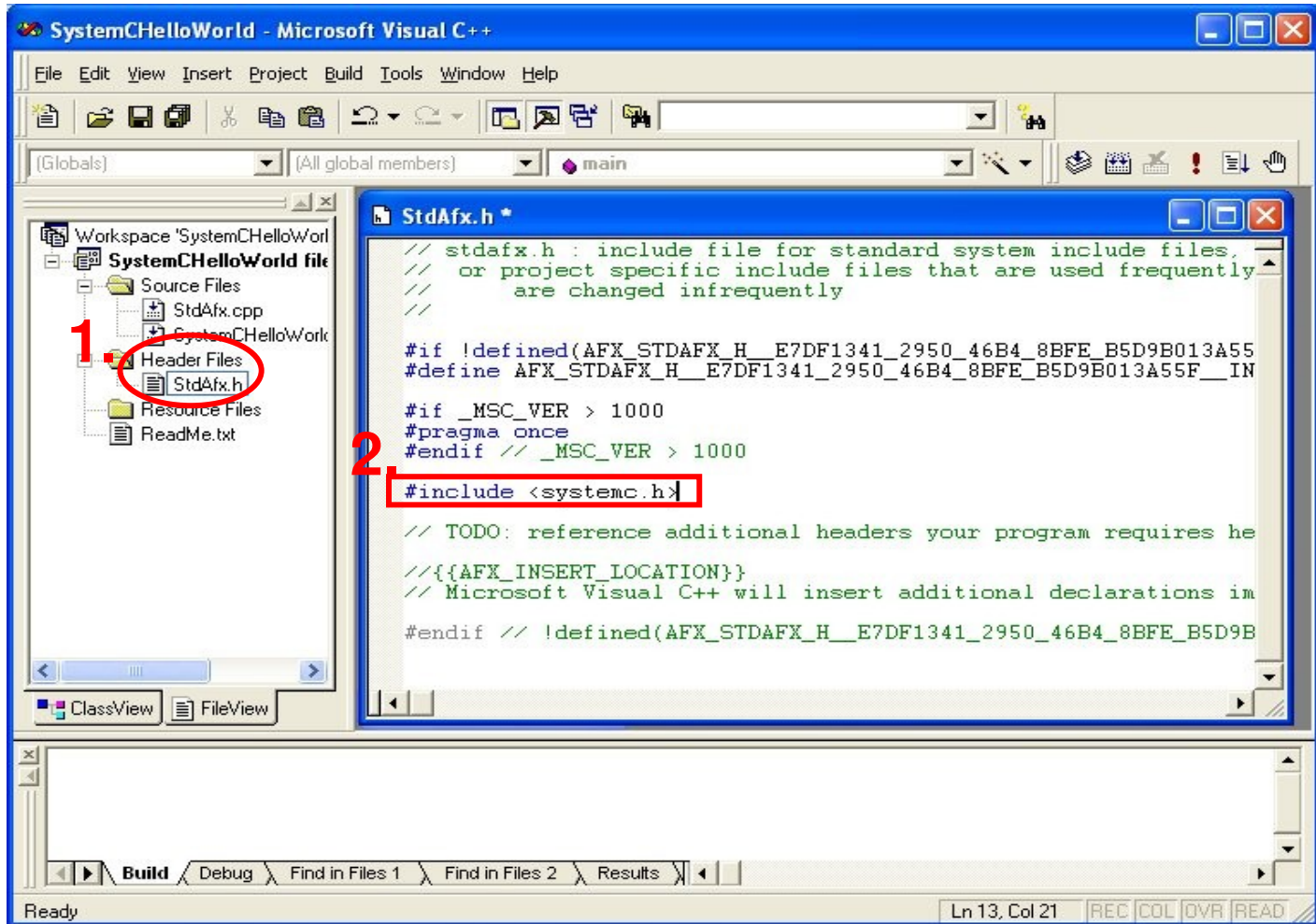


- Directories
- Show directories for: *Library files*
- Neuer Pfad: *... \MSVC60\SystemC\Debug*
- Show directories for: *Include files*
- Neuer Pfad: *... \SystemC\src* (Wurzelverzeichnis von SystemC!!!)



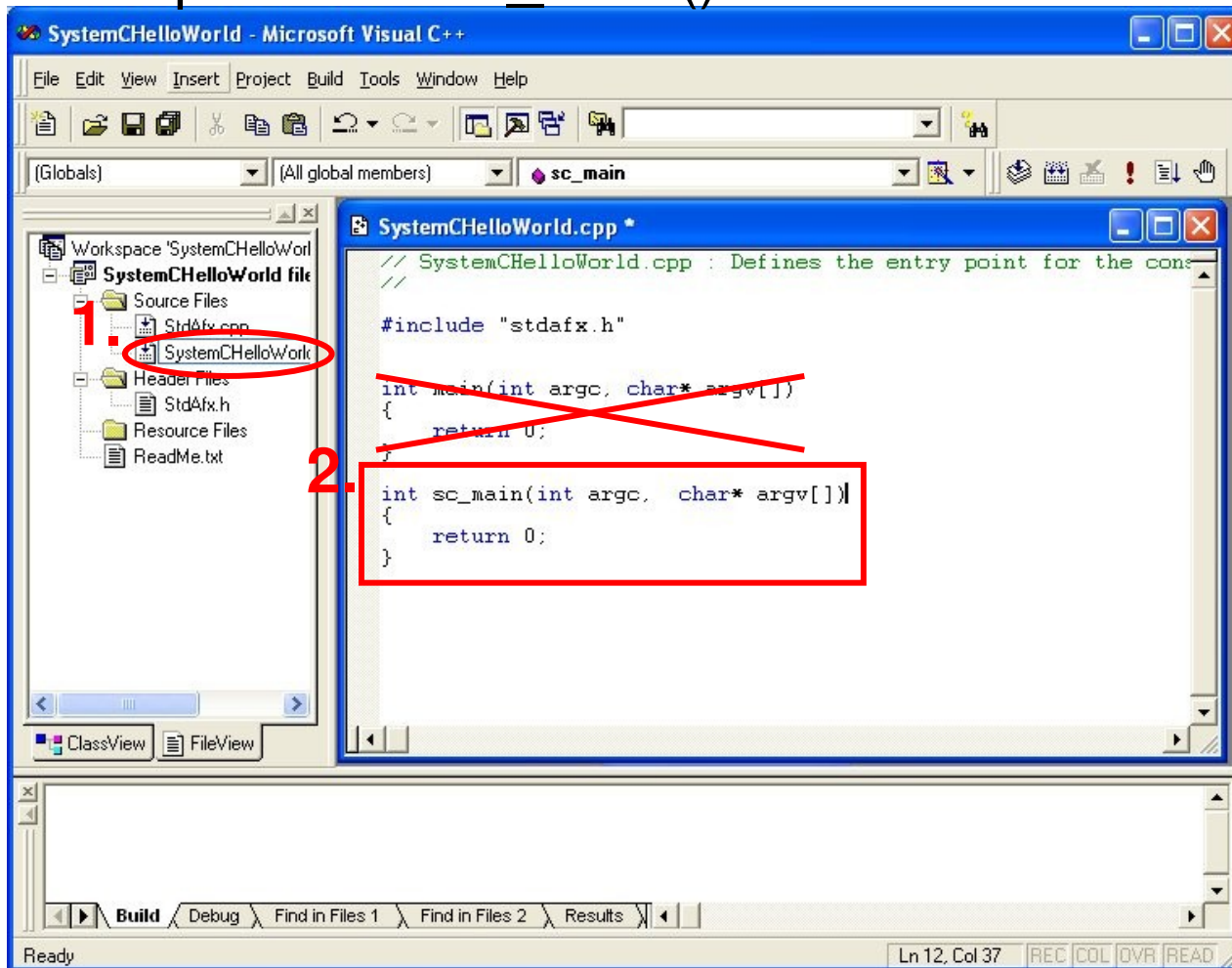
# SystemC – 1. Projekt unter Windows (8)

- Projekt anpassen - SystemC-Header einbinden:



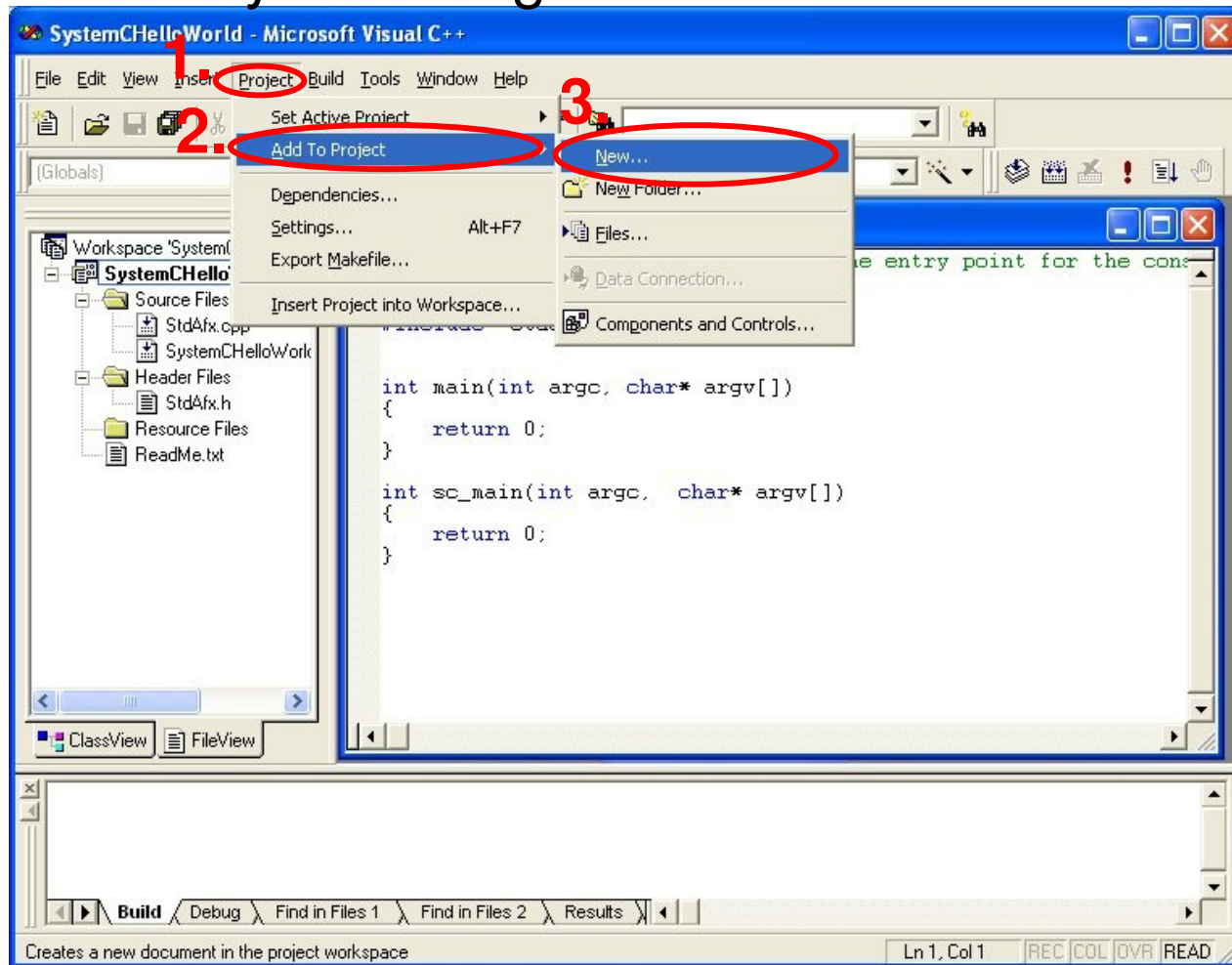
# SystemC – 1. Projekt unter Windows (9)

- Projekt anpassen – `sc_main()` erstellen:



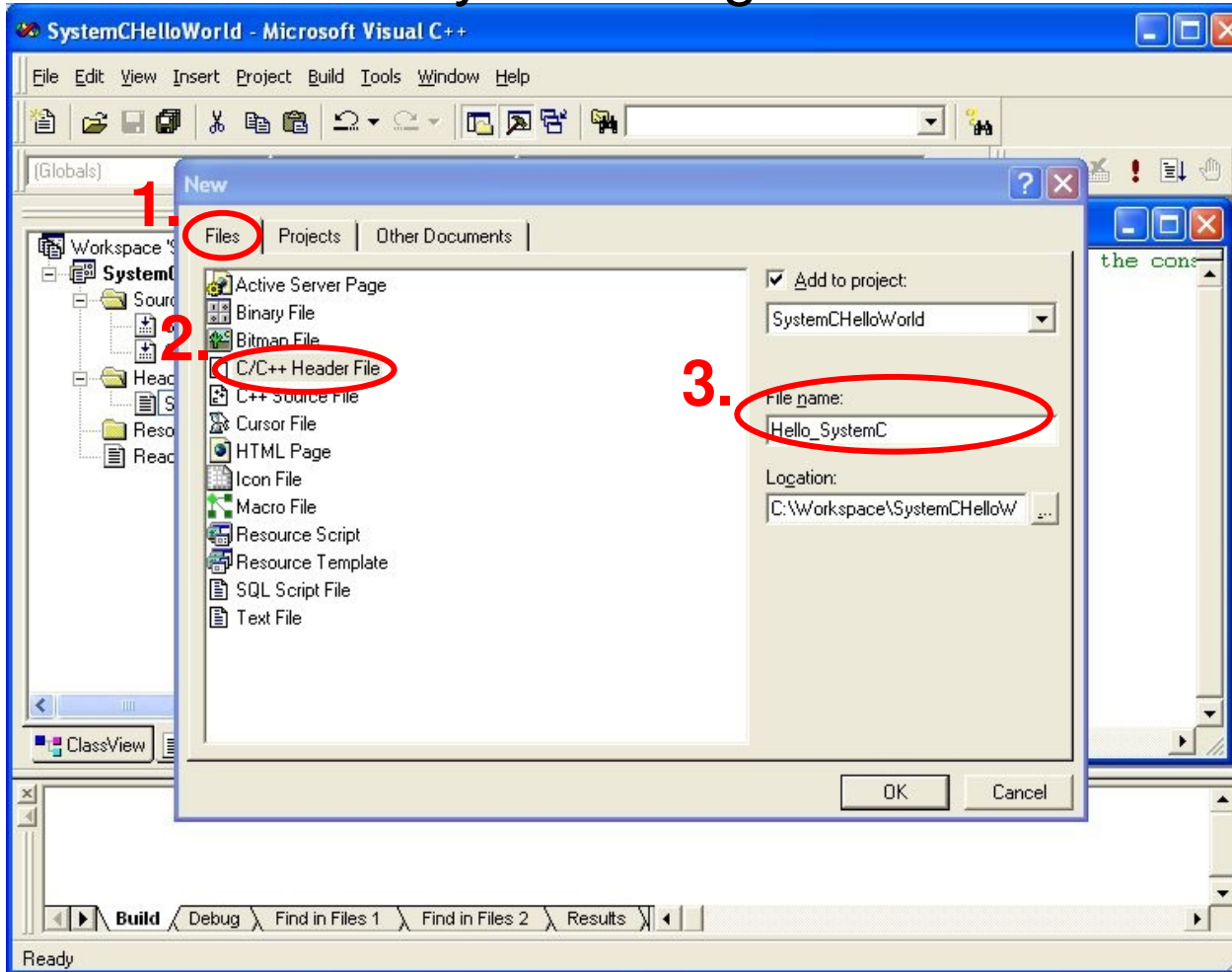
# SystemC – 1. Projekt unter Windows (10)

- HelloWorld System eingeben:



# SystemC – 1. Projekt unter Windows (11)

- HelloWorld System eingeben – Header-/Source-File:



- C/C++ Header File

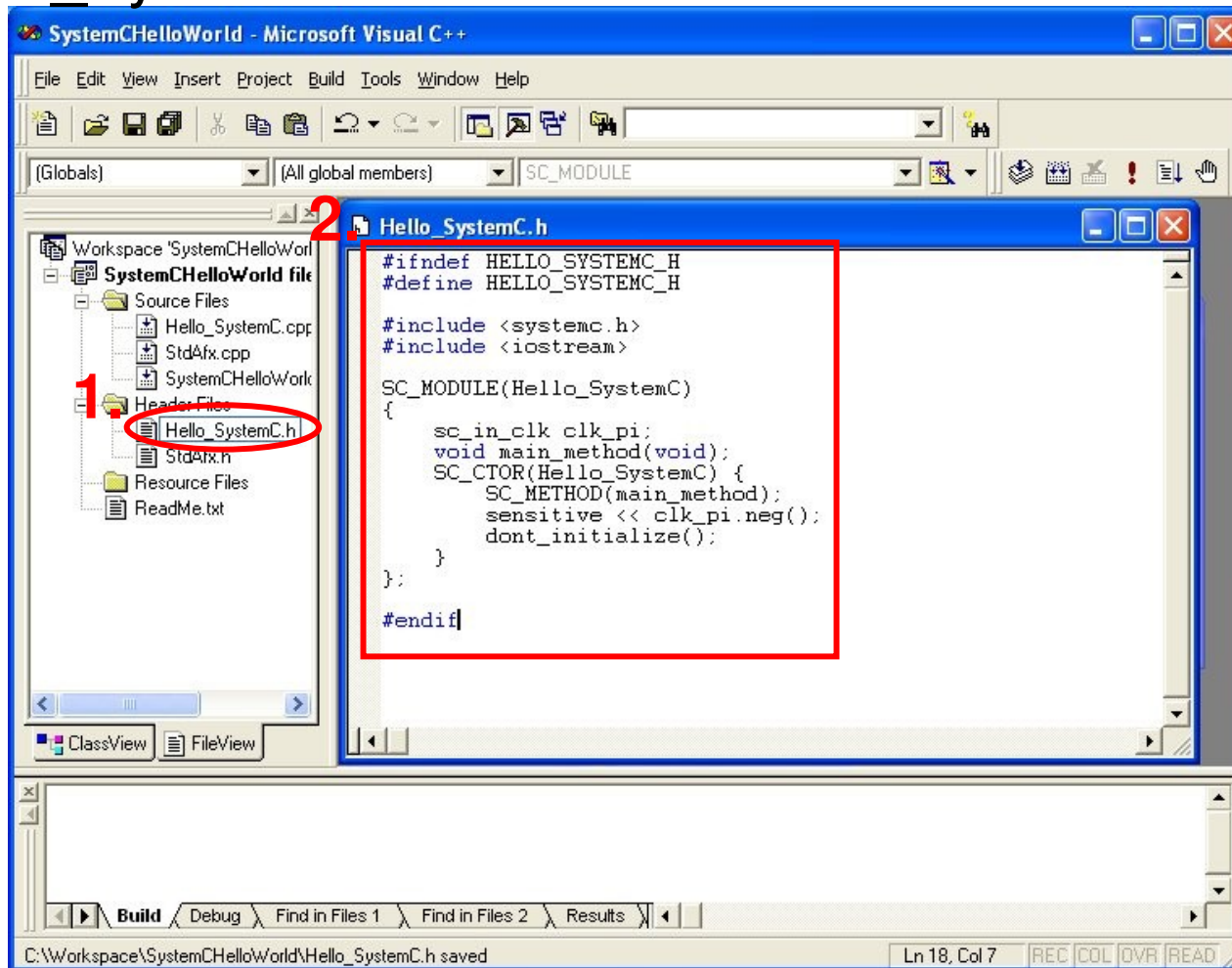
Analog:

- C++ Source File

=> Dateiendung wird automatisch ergänzt

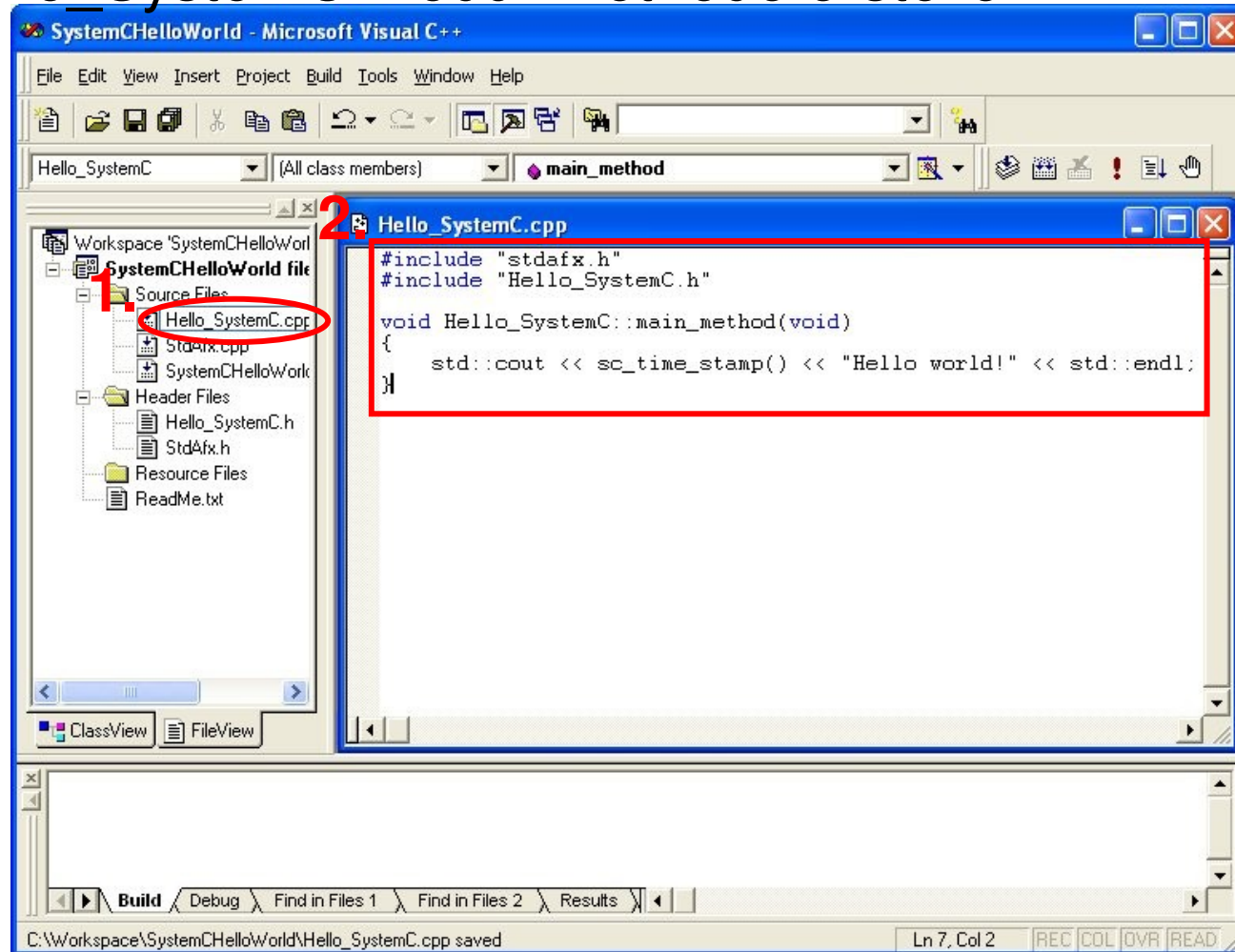
# SystemC – 1. Projekt unter Windows (12)

- Hello\_SystemC Modul erstellen:



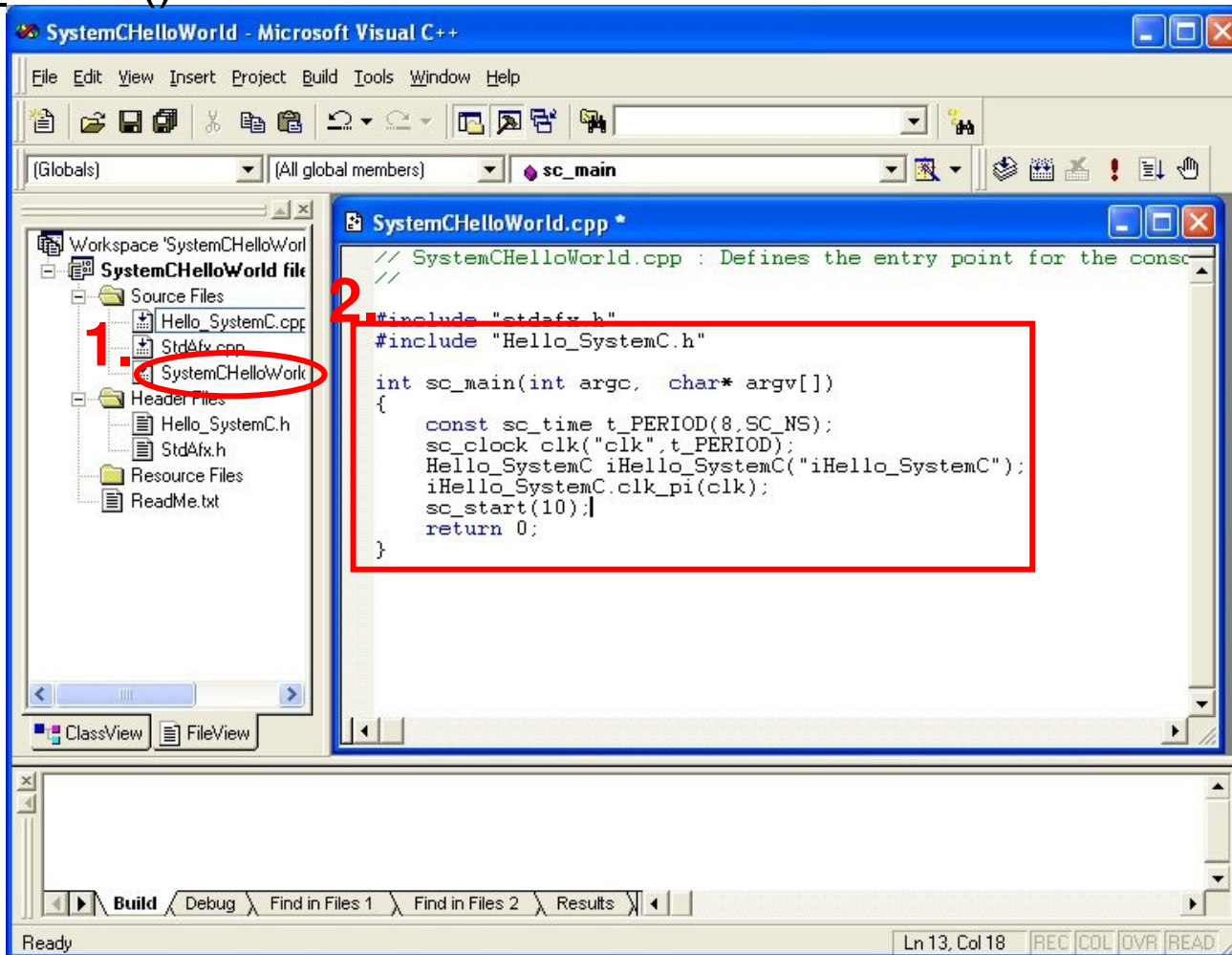
# SystemC – 1. Projekt unter Windows (13)

- Hello\_SystemC Modul-Methode erstellen:



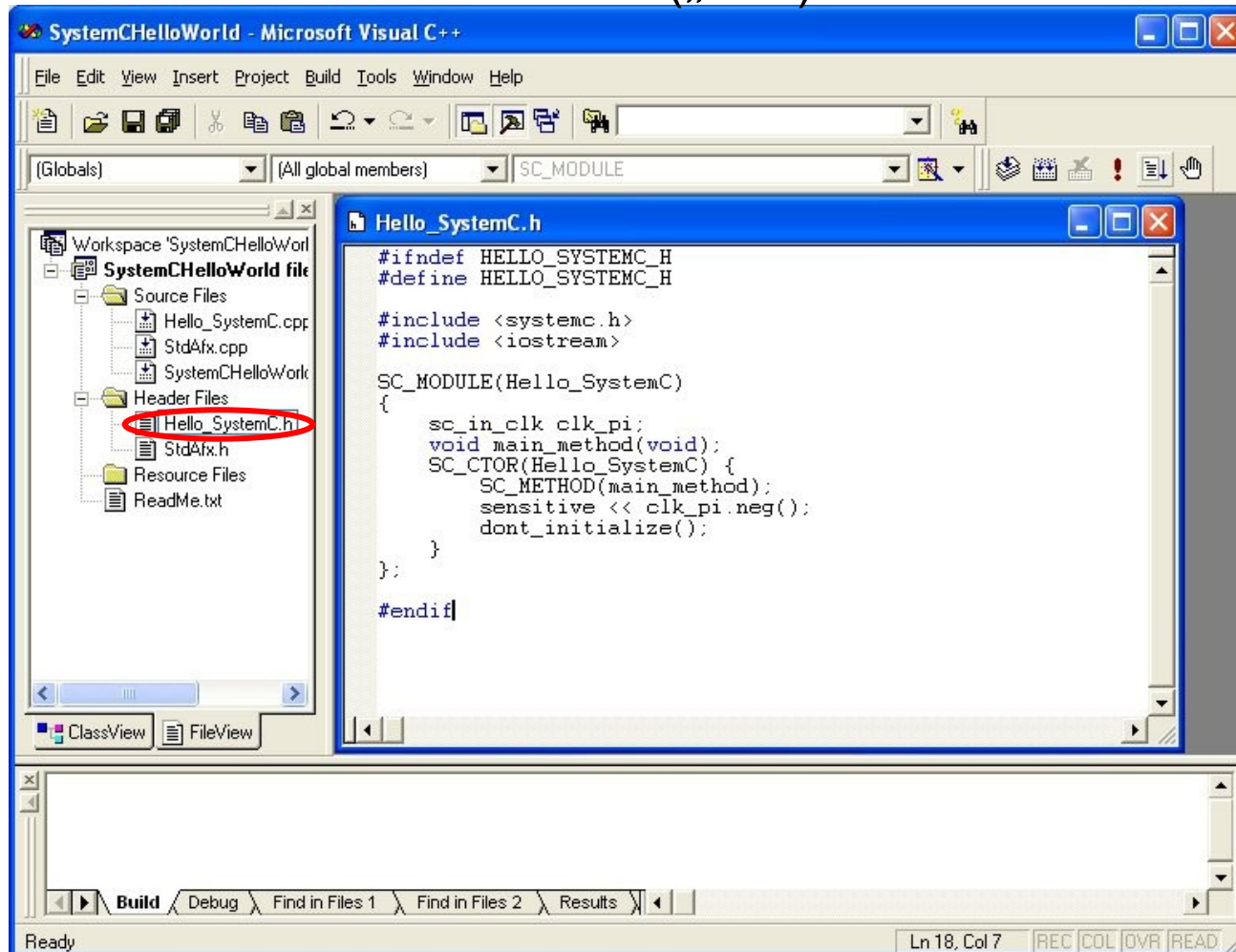
# SystemC – 1. Projekt unter Windows (14)

- `sc_main()` modifizieren:



# SystemC – 1. Projekt unter Windows (15)

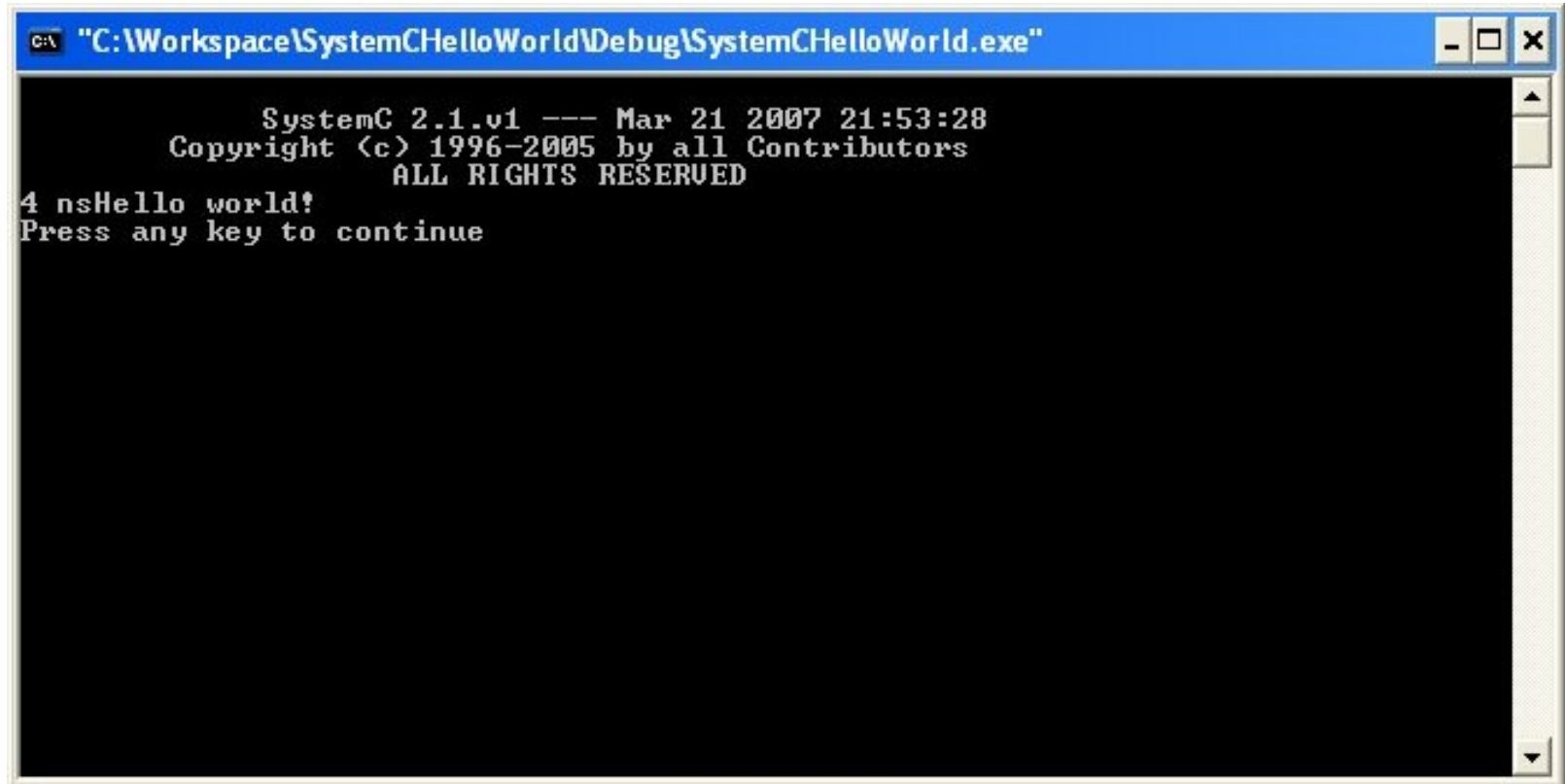
- Übersetzen und ausführen („F5“):





# SystemC – 1. Projekt unter Windows (16)

- Hello\_World!



```
C:\Workspace\SystemCHelloWorld\Debug\SystemCHelloWorld.exe

SystemC 2.1.v1 --- Mar 21 2007 21:53:28
Copyright (c) 1996-2005 by all Contributors
ALL RIGHTS RESERVED
4 nsHello world!
Press any key to continue
```



GESCHAFFT!!!

# SystemC – 1. Projekt unter Linux (1)

---

- Sourcen entpacken und übersetzen:

```
~> gunzip systemc-2.1.v1.tgz
~> tar xfv systemc-2.1.v1.tar
...
~> cd systemc-2.1.v1
~> mkdir objdir; cd objdir
~> ../configure
~> make
...
~> make install
```

# SystemC – 1. Projekt unter Linux (2)

- Header-Datei erstellen:

Mit beliebigem Editor: „Hello\_SystemC.h“

```
#ifndef HELLO_SYSTEMC_H
#define HELLO_SYSTEMC_H

#include <systemc.h>
#include <iostream>

SC_MODULE(Hello_SystemC)
{
    sc_in_clk clk_pi;
    void main_method(void);
    SC_CTOR(Hello_SystemC){
        SC_METHOD(main_method);
        sensitive << clk_pi.neg();
        dont_initialize();
    }
};

#endif
```

# SystemC – 1. Projekt unter Linux (3)

- SystemC-Code erzeugen:

Mit beliebigem Editor: „Hello\_SystemC.cpp“

```
#include <Hello_SystemC.h>

void Hello_SystemC::main_method(void)
{
    std::cout << sc_time_stamp () << „Hello World! << std::endl;
}
```

# SystemC – 1. Projekt unter Linux (4)

- `sc_main()` erzeugen:

Mit beliebigem Editor: „systemchelloworld.cpp“

```
#include <systemc.h>
#include „Hello_SystemC.h“

int sc_main(int argc, char* argv[])
{
    const sc_time t_PERIOD(8, SC_NS);
    sc_clock clk(„clk“, t_PERIOD);
    Hello_SystemC iHello_SystemC(„iHello_SystemC“);
    iHello_SystemC.clk_pi(clk);
    sc_start(10);
    return 0;
}
```

# SystemC – 1. Projekt unter Linux (5)

- HelloWorld übersetzen:

Übersetzen:

```
~> cd systemchelloworld
```

```
~/systemchelloworld> make
```

```
....
```

```
~/systemchelloworld> ./systemchelloworld
```

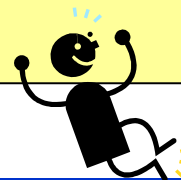
```
SystemC 2.1.v1 - - - Mar 21 2007 22:15:27
```

```
Copyright (c) 1996-2005 by all Contributors
```

```
ALL RIGHTS RESERVED
```

```
4 ns Hello world!
```

```
~/systemchelloworld>
```



**GESCHAFFT!!!**

## Lehrstuhl Informatik XII

Jürgen Mäter (Informatik IV)

Übungen zur Vorlesung  
„Rechnergestützter Entwurf  
von Mikroelektronik (REM)“  
Sommersemester 2008

Mittwoch, 09.04.2008

### Übungsblatt 1

- Abgabe:** Bis Mittwoch, 16.04.2008, 16:00 Uhr; in der Vorlesung oder in den Übungsgruppen oder per Email an den jeweiligen Übungsgruppenleiter.
- Hinweise:** Gruppenarbeit von bis zu drei Personen aus der gleichen Übungsgruppe ist ausdrücklich erwünscht.  
Bitte vergessen Sie nicht, Ihre Namen, Ihre Matrikelnummern auf die Lösung zu schreiben:  
Übungstermin: Mittwoch, 10-12 Uhr, OH16/E07  
Anwesenheitspflicht in den Übungen!  
Bitte Krankmeldungen an den Übungsgruppenleiter.

#### **Kriterien für die Scheinvergabe:**

- Es gibt 12 Übungszettel
- Insgesamt 50% der Punkte müssen erreicht werden
- Die Ergebnisse müssen vor der Gruppe präsentiert werden können

#### **Aufgabe 1.1 (4 Punkte)**

Beschreiben Sie kurz das Anwendungsgebiet von SystemC!

#### **Aufgabe 1.2 (3 Punkte)**

Nennen Sie drei Erweiterungen von SystemC gegenüber Standard C++.

(Bitte wenden)

### **Aufgabe 1.3 (4 Punkte)**

- a) Zählen Sie alle Möglichkeiten (mit Codebeispielen) auf, die SystemC bietet, um einen 37bit integer zu deklarieren.
- b) Zählen Sie alle Möglichkeiten (mit Codebeispielen) auf, die SystemC bietet, Zahlen mit Nachkommastellen zu deklarieren.

### **Aufgabe 1.4 (5 Punkte)**

Betrachten Sie folgendes SystemC-Codefragment:

```
sc_logic x,y,a,b;  
x = '0';  
y = 'X';  
a = x | y;  
b = x & y;
```

Welche Werte haben a und b nach der Ausführung?