

## Handbuch zur C++ - Lektion "vari v1.032", zu Deutsch "Variablen"

Copyright by Hilfe & Service von EDV-Fachleuten

Dipl.-Phys Harald Schärfer

Reilstr. 22 (Eingang Mozartstr.)

D-06114 Halle (Saale)

Edit v1.020 from 2022-09-15 to 2024-01-17 by Harald Schärfer

Tel. (0345) 521 13 40

Funk (0177) 521 13 40

EM [Vertrieb@HuSvEDVF.de](mailto:Vertrieb@HuSvEDVF.de)

Fax (0345) 521 13 41

Nutzen: Erlernen, wie Variable mit Operationen verarbeitet werden können.

### Inhaltsverzeichnis

<u>1.</u>	<u>Bestandteile</u> .....	2
<u>2.</u>	<u>Systemvoraussetzungen</u> .....	2
<u>3.</u>	<u>Installation</u> .....	2
<u>4.</u>	<u>Benutzung</u> .....	2
<u>4.1</u>	<u>Untersuchung</u> .....	2
	<u>Debuggen</u> .....	3
<u>4.2</u>	<u>Ausgabe</u> .....	4

## 1. Bestandteile

Das Programm wurde mit der Entwicklungsumgebung "Code:Block" entwickelt besteht aus folgenden x Komponenten.

	<u>vari</u>
*.EXE	Das ausführbare Programm "vari".
*.PDF	Dieses Handbuch als "Portable Document Format" (übertragbares Dokumentenformat).
*.TXT	Dieses Handbuch als Text.
*.ZIP	Alle Dateien in eine ZIP - Datei gepackt.

### Work

Change.LOG	Chronologische Liste der Aenderungen des Programms.
main.cpp	Das Hauptprogramm in C++;
main.hpp	Die Kopfdatei zum Hauptprogramm in C++.
Error.TXT	Fehlermeldungen als Liste.
Lizenz.TXT	Lizenzbestimmungen mit Gewährleistungsbeschränkung, Nutzungsrecht und Vertrieb zum Programm.

## 2. Systemvoraussetzungen

Das Programm wurde erfolgreich Ausführung und Nutzung getestet unter

- Code::Block v20.03;
- Windows 7 Prof. 64Bit;
- Windows 10 Prof. 64Bit

## 3. Installation

Wie Code::Blocks oder eine andere Entwicklungsumgebung für C/C++ installiert wird, ist hier nicht das Thema. Kopieren Sie die Dateien in ein Projektverzeichnis mit den Namen "Vari" und benutzen Sie den Quelltext, um das was Sie darin gesucht haben, zu verstehen.

## 4. Benutzung

Hier soll gezeigt werden wie im Quelltext von "main.cpp" und "main.hpp"

- Konstanten und
- Variablen deklariert, beschrieben und benutzt werden.

### 4.1 Untersuchung

Untersucht wurden folgende Typen von Konstanten und Variablen:

Dazu gehören:

- extern int ...;
- char ...;
- unsigned char ...;
- int ...;
- float ...;
- static struct sBITFELD ....
- Character als Zeichen und -kette;
- Integer als Ganzzahl;
- Float als Fließkommazahl. Dabei wird vorab der Tastaturpuffer gelöscht, falls da statt einer Ganzzahl ein Buchstabe eingegeben worden ist, bleibt dieser darin enthalten und überspringt die Eingabe hier;

- Bitfeld als Struktur mit 16Bit fuer Schaltungen;
- Bedingtes Debuggen und Stop wenn das Laufwerk == C ist!
- Operationen mit den Variablen durchgeführt, beschrieben und benutzt werden. Dazu gehören:
  - Plus +, Minus -, Mal \* und Durch /;
  - Potenz;
  - Verschiebeoperation nach Links (\*2) und Rechts (/2);
  - binäres und logisches NOT, UND, OR, XOR.
- Auslagern der Funktionsköpfe der 2 Funktionen in die Header - Datei "main.hpp"!

### Debuggen

Dazu wird jede Variable nach und nach benutzt und der Ablauf kann mit dem Debugger verfolgt werden. Dazu dienen folgende Tasten:

- <F4> Debugger beginnt ab hier.
- <F7> Eine Programmzeile weiter untersuchen.
- <F8> Lauf zum nächsten Breakpoint.

Und oder bedingt ('C' == cZ) in der Zeile cZ=cLW aufgerufen werden.

```
for (cLW='\0'; cLW < 'Z';
cLW++)
  cZ=cLW;
```

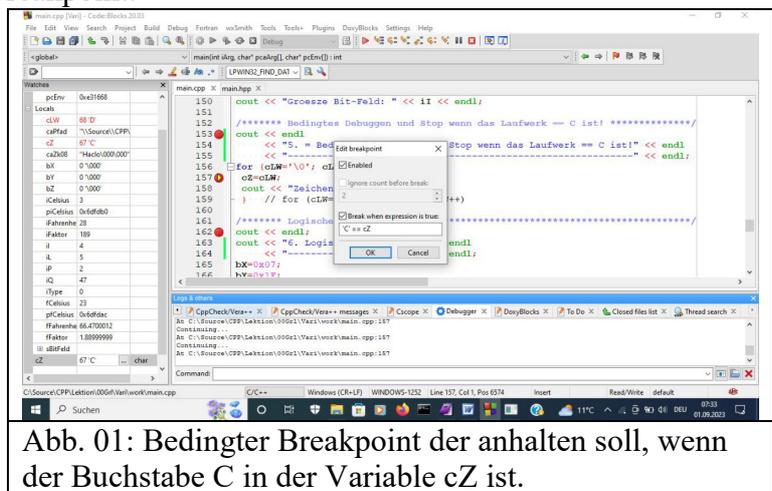


Abb. 01: Bedingter Breakpoint der anhalten soll, wenn der Buchstabe C in der Variable cZ ist.

Das Programm ist auch in der ausführbaren Version "vari.exe" beigelegt und deren Ausgaben können am Bildschirm betrachtet werden.



- §
- ?
- ?
- ?
- ?
- ?
- ?
- ?
- ?
- ?
- ?
- 
- !
- "
- #
- \$
- %
- &
- '
- (
- )
- \*
- +
- ,
- -
- .
- /
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- :
- ;
- <
- =
- >
- ?
- @
- A
- B
- C
- D
- E
- F
- G
- H
- I
- J
- K
- L
- M
- N
- O
- P
- Q
- R
- S
- T
- U
- V

- W
- X
- Y

## 7. Logisch binaire Operationen

-----

Gegeben:

- bX = 07H = 0000 0111B
- bY = 1fH = 0001 1111B

Aufgabe und Ergebnis:

- NOT bX = 1111 1000B;
- NOT bY = 1110 0000B;
- bX UND bY = 0000 0111B;
- bX OR bY = 0001 1111B;
- bX XOR bY = 0001 1000B;

## 8. Verschiebe Operationen

-----

Gegeben:

- bX = 07H = 0000 0111B
- bY = 1fH = 0001 1111B

Aufgabe und Ergebnis:

- Links Schieben der Bits um 3 Stellen mit (bX << 3) = 0011 1000B;
- Rechts Schieben der Bits um 2 Stellen mit (bY >> 2) = 0000 0111B;

## 9. Logische Operationen

-----

Gegeben:

1. positive Zahl iI = 9
2. NULL-Zahl iJ = 0
3. negative Zahl iL = -4

Aufgabe und Ergebnis:

- Logisches NOT von
  - iI = 0
  - iJ = 1
  - iL = 0
- Logisches UND von
  - iI UND iJ = 0
  - iI UND iL = 1
  - iJ UND iL = 0
- Logisches OR von
  - iI OR iJ = 1
  - iI OR iL = 1
  - iJ OR iL = 1
- Logisches XOR von
  - iI XOR iJ = 1
  - iI XOR iL = 1
  - iJ XOR iL = 1
  - iL XOR iL = 0

10. Programm schliessen und pruegen ob ein Fehler aufgetreten ist!  
Fehler-Nummer eingeben: 2

Ein Fehler ist aufgetreten!  
2: Unbekannter Fehler! Bitte melden.

Process returned 2 (0x2) execution time : 93.680 s  
Press any key to continue.

\*\*\*\*\* EOF \*\*\*\*\*

**Viel Erfolg beim studieren!**

\*\*\*\*\* EOF \*\*\*\*\*