K&A Laborgeraete GbR

Software manual

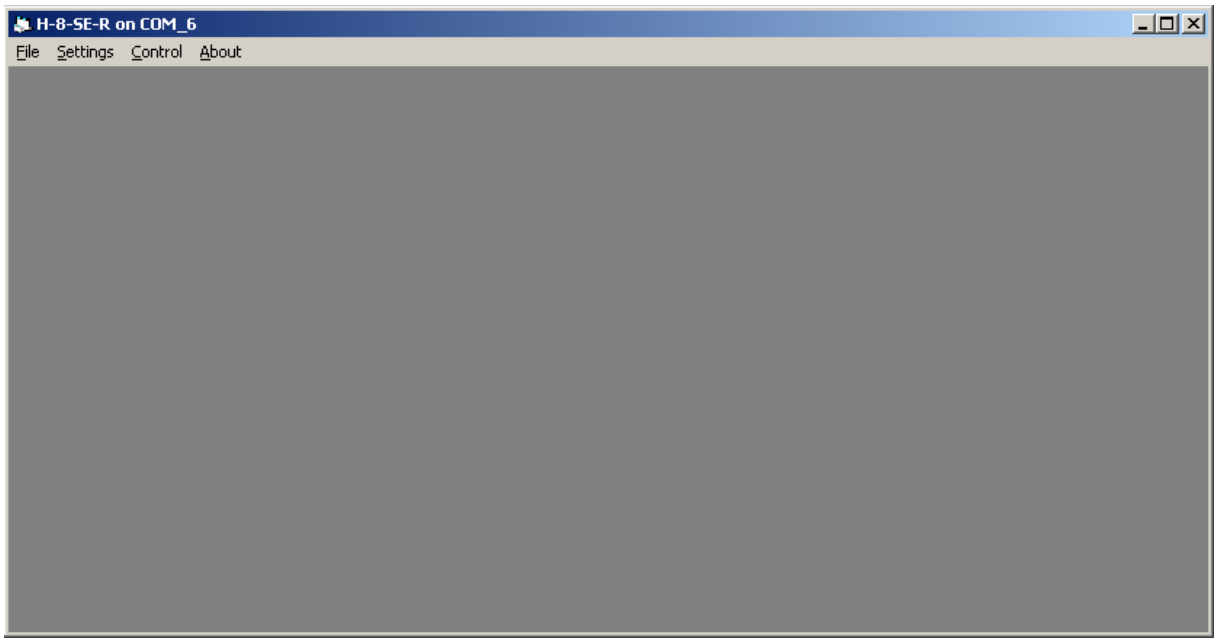
Synthesizer
H-8-SE-VLS

V. 01.00

I. Commands

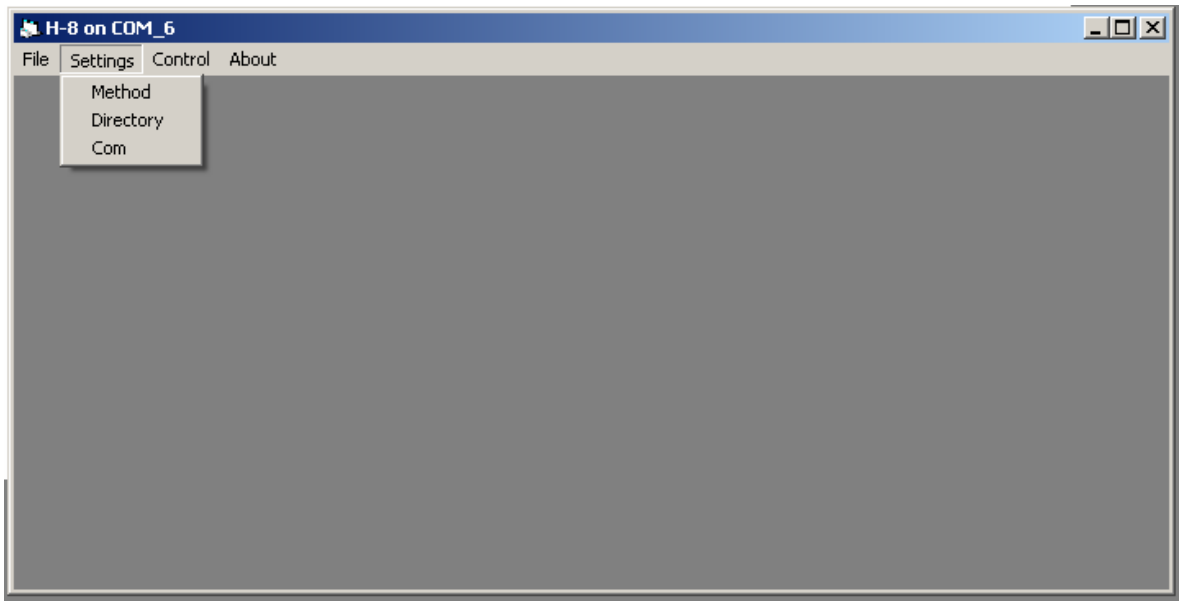
Following Commands are allowed :

- File
- Settings
- Control
- About



The Commands File, About and Controll has been discussed in H-8-SE-VLS-Owners Guide. The Software Manual will describe the programming of methods and subroutines and their influence on synthesis execution.

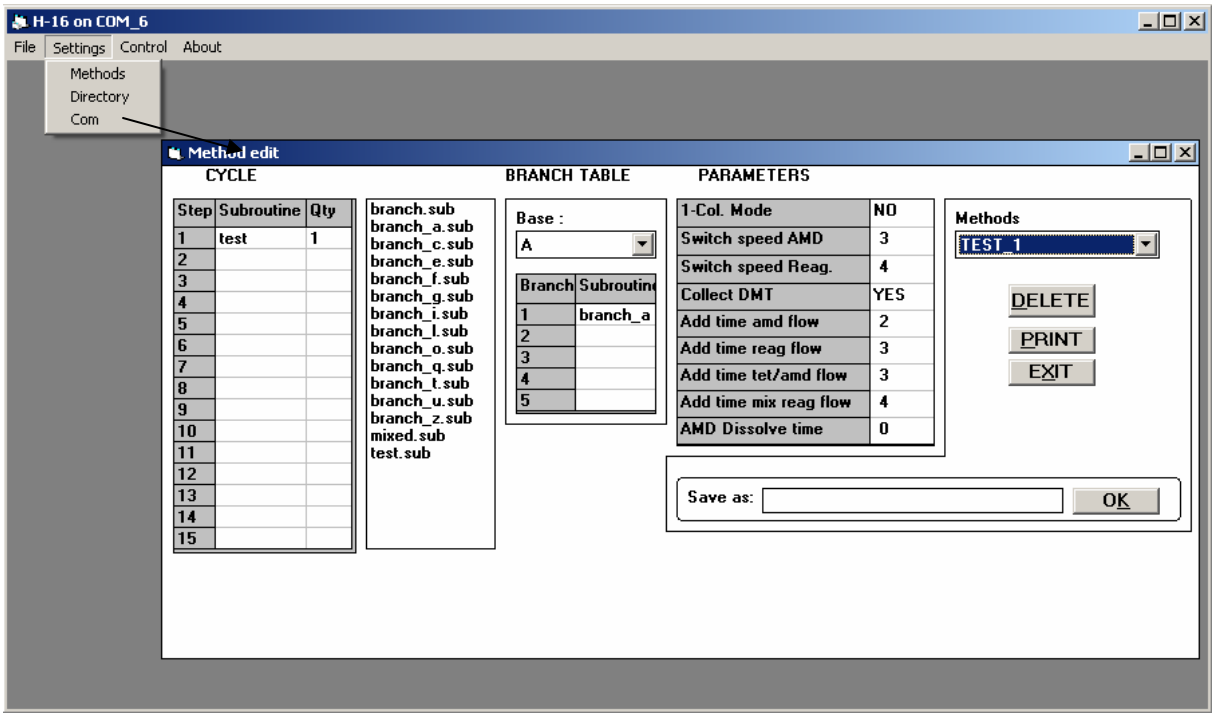
II. Settings



Menu ‘Settings’ contains 3 submenus which enable entering different settings and parameters which are necessary for instrument control.

- Method
- Directory
- Com

II-1 Settings - Method



This menu enables creating the method used for the synthesis. One method consists of three components: **CYCLE, BRANCH TABLE** and **PARAMETERS**.
The synthesis cycle consists of the subroutines which will be executed in the order programmed in the **CYCLE** table.

The Cycle consists of maximum 15 subroutines executed step by step for every synthesis coupling step.
All defined and accessible subroutines are listed in the window on the right side of the CYCLE. A subroutine can be chosen by clicking on the name of the desired subroutine.

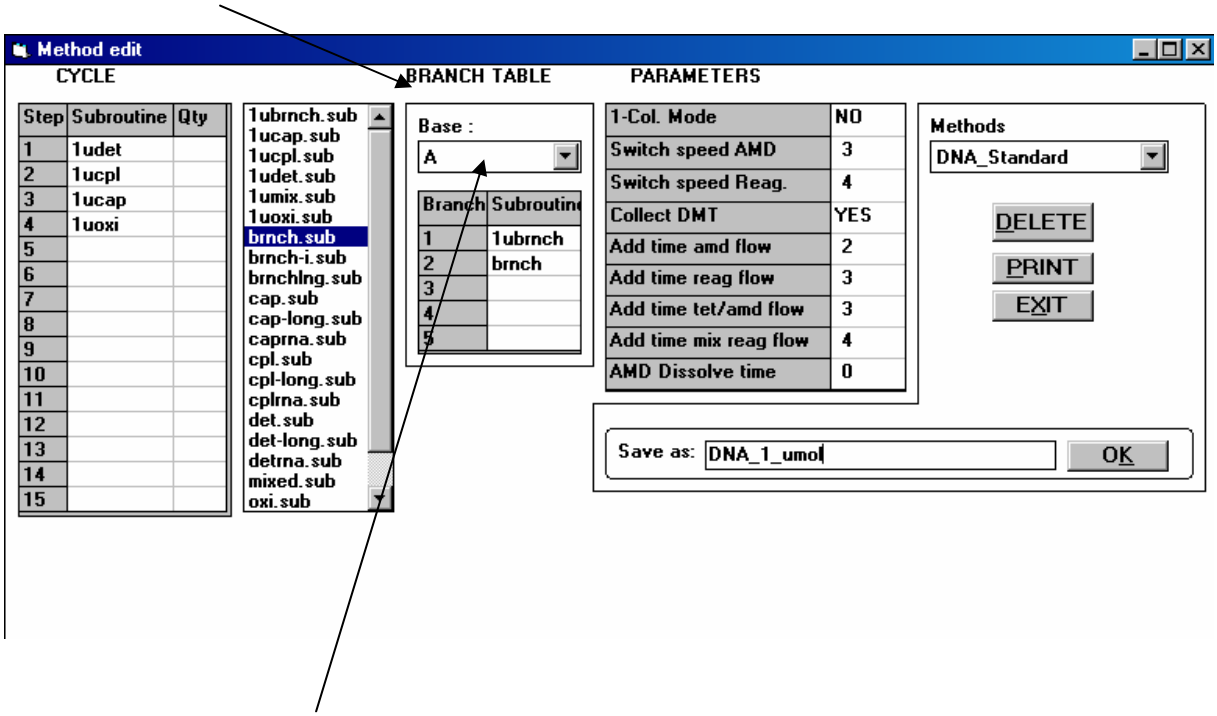
The synthesis execution starts always from step 1 of the CYCLE and continues until the last step of the Cycle. The Cycle will be repeated until the end of the oligo synthesis.

Each subroutine can be repeated 1 to 5 times, according to the entry in the Qty field.

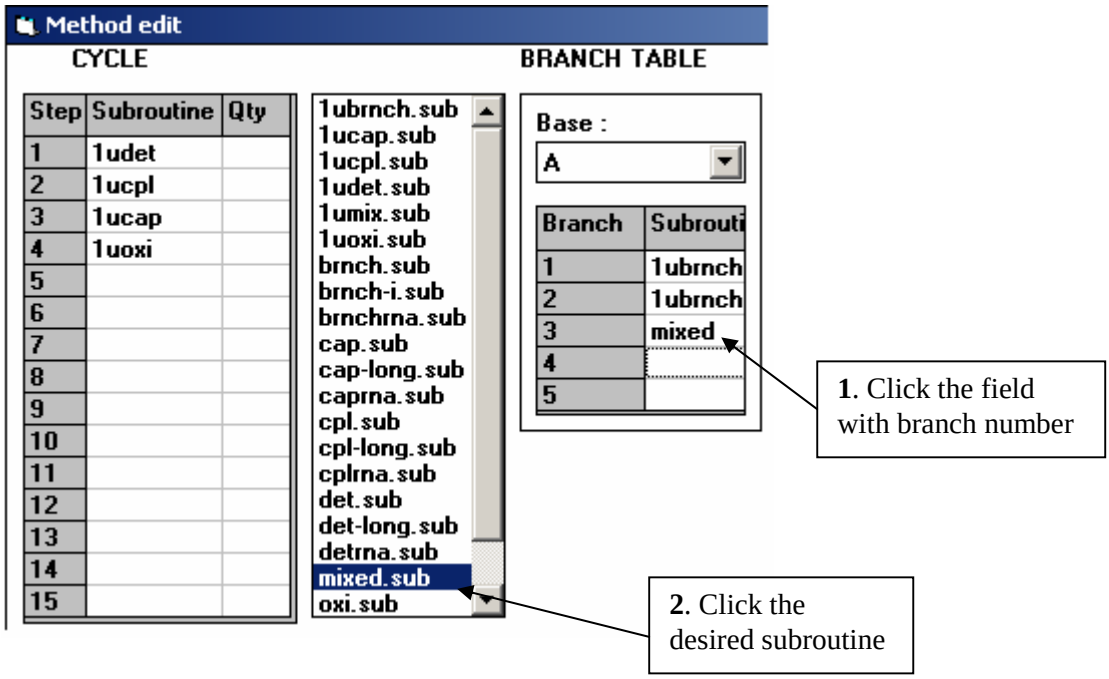
In the example above the subroutine’s order is: **1udet 1ucpl 1ucap 1uoxi.**

The **BRANCH** TABLE enables creating the so-called ‘Branch’. ‘Branch‘ is the subroutine which can be executed dedicated to the amidite position. The implementing of the ‘Branch’ will be discussed in chapter **III-1-4. Subroutine programming-examples** of this manual.

5 different branches can be defined for every amidite position and every wobble combination.

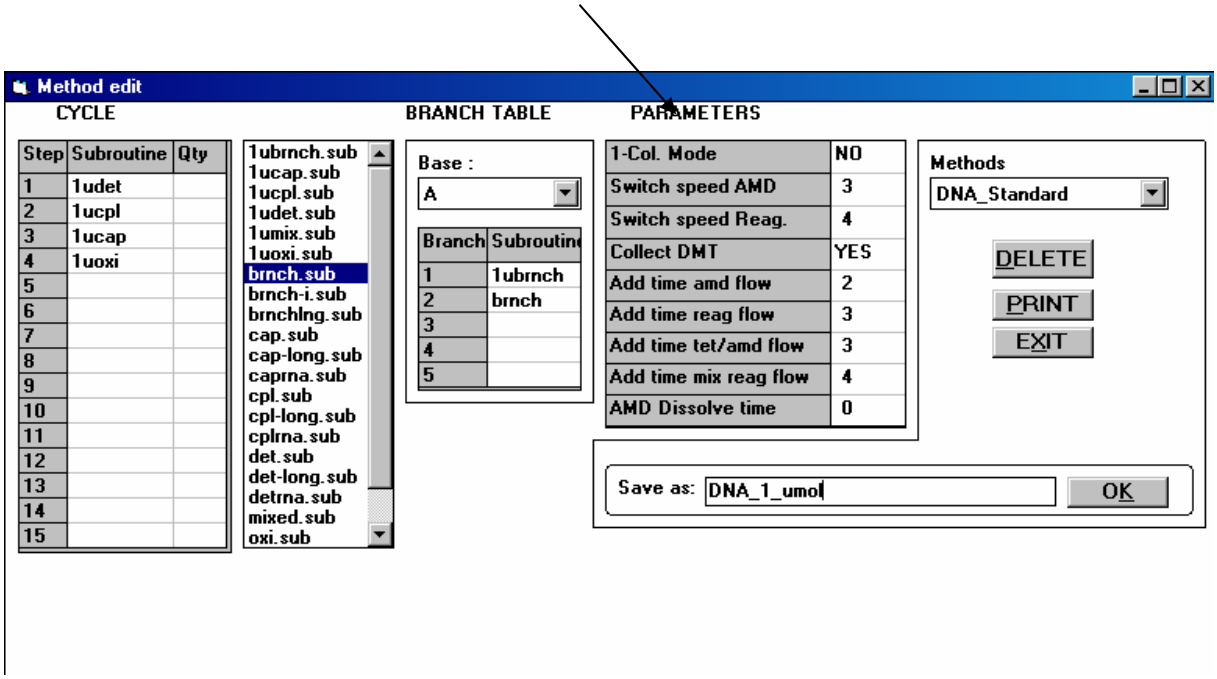


The choice of the amidite position, for which the branch will be programmed, can be made in the choice line on the left top position of the menu.



The choice of the subroutine can be made by clicking on the name of the desired subroutine. All defined and accessible subroutines are listed in the window on the left side of the branch table.

The last component of the Method are **PARAMETERS**

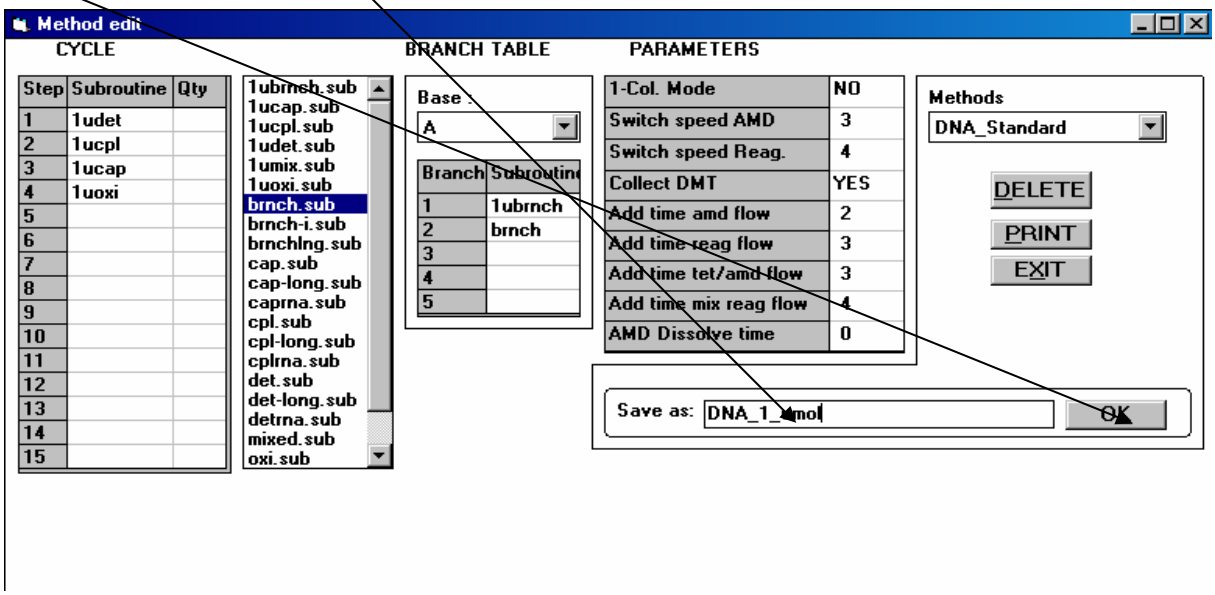


Following parameters are implemented and can be entered:

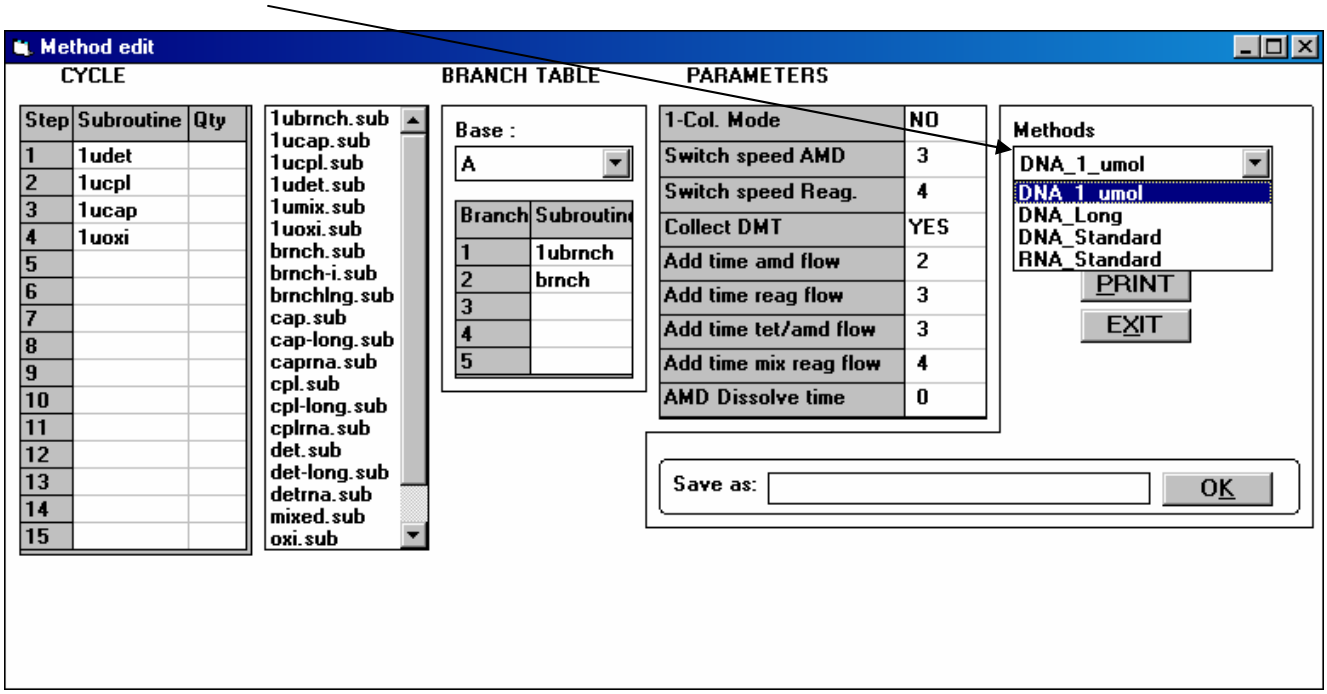
- **1-Col. Mode**: defines the fluid delivery to the synthesis columns. When YES, the columns will be filled in serial mode (closer explanation in the chapter **III-1-1. Subroutine programming-examples**). Status can be changed by clicking at the field.
- **Switch speed AMD**: defines the switching speed of the valves while mixed amidite and other reagent's flow during the synthesis (closer explanation in the chapter **III-1-2. Subroutine programming-examples**). Lower the number-higher the switch speed of the valves. When 0, all involved valves will be opened and no switching will be executed.
- **Switch speed Reag**: defines the switching speed of the valves while two mixed reagents flow during the synthesis (closer explanation in the chapter **III-1-2. Subroutine programming-examples**) Lower the number-higher the switch speed of the valves. When 0, all involved valves will be opened and no switching will be executed.
- **Collect DMT**: defines the collecting and evaluation of the trityl monitor during the synthesis. Can be changed to NO by click on the field. Status NO. Recommended only when trityl monitor is not working properly.
- **Add time amd flow**: defines the correction of the flow time when amidite must be delivered to more than one column at the same time (closer explanation in the chapter **III-1-3. Subroutine programming-examples**)
- **Add time reag flow**: defines the correction of the flow time when reagent must be delivered to more than one column at the same time (closer explanation in the chapter **III-1-3. Subroutine programming-examples**)
- **Add time tet/amd flow**: defines the correction of the flow time when the mixture of amd and other reagent must be delivered to more than one column at the same time (closer explanation in the chapter **III-1-3. Subroutine programming-examples**)

- **Add time mix reag flow flow**: defines the correction of the flow time when the mixture of two reagents must be delivered to more than one columns at the same time (closer explanation in the chapter **III-1-3. Subroutine programming-examples**)
- **AMD Dissolve time**: this parameter is not applied in this software version.

When CYCLE, BRANCH TABLE and PARAMETERS are entered, the METHOD can be save now. For that, the name of the Method must be entered in the field: Save as and the key OK must be clicked.



The click on the Methods list enables the selecting of the existing method:

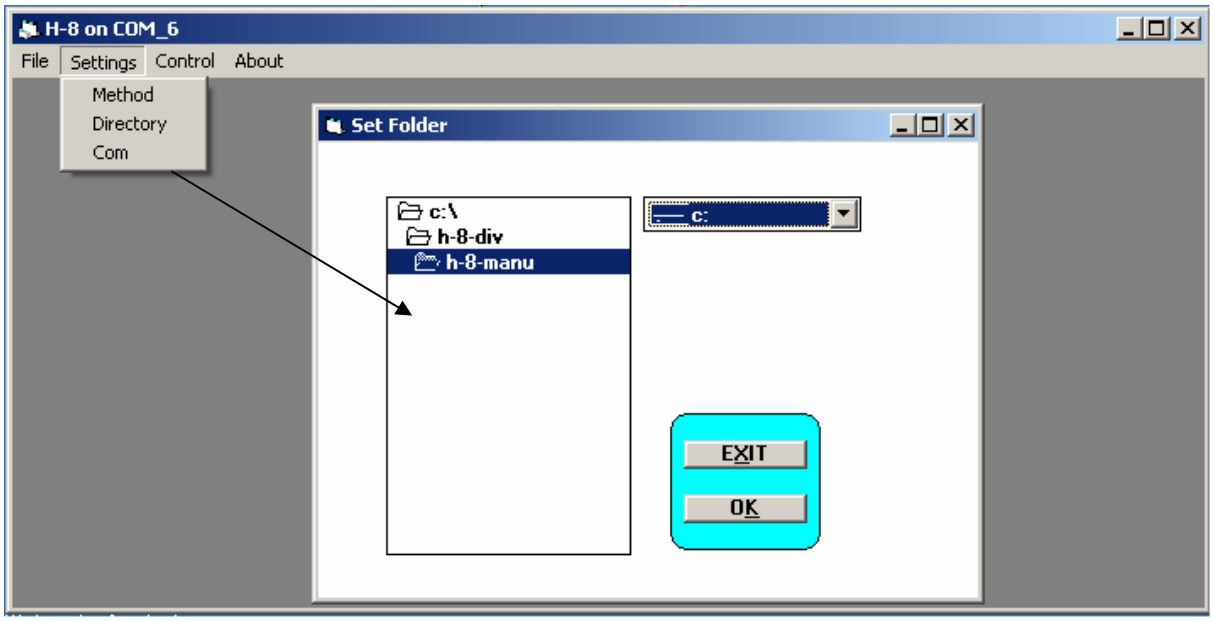


Ø The click on the button PRINT causes printing of the Cycle, branch table and parameters for selected method.

Ø The click on the button DELETE causes deleting of the selected method.

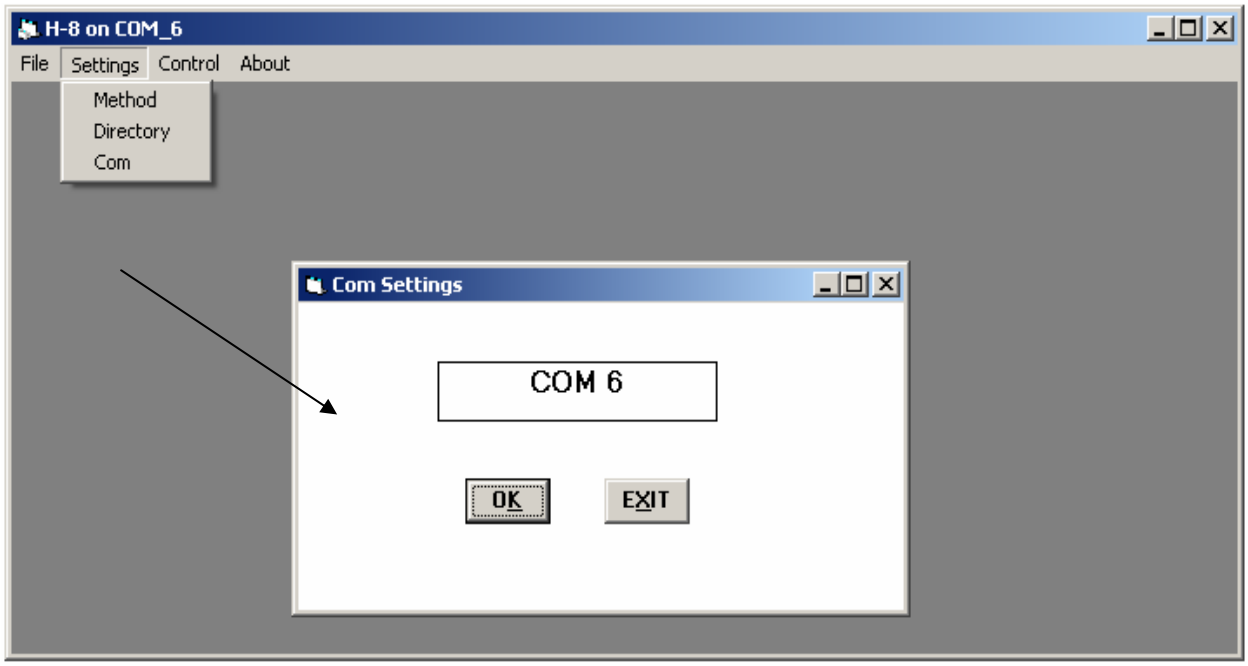
Ø The click on the button EXIT causes exit from the 'METHOD' mask without saving.

II-2 Settings Directory

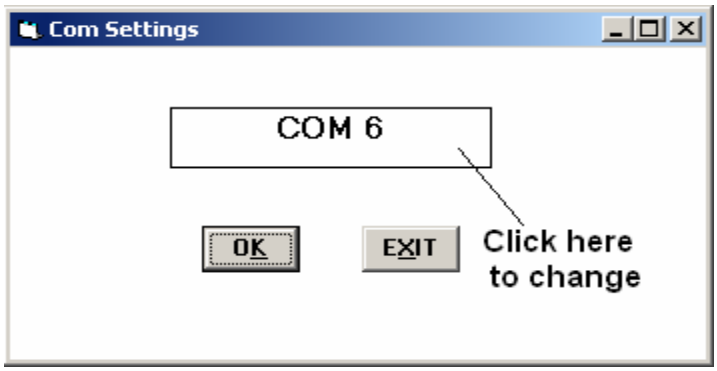


This menu enables the choice of the folder in which all synthesis data will be managed. The choice can be made by double click on the proper folder and confirming by clicking on OK button.

II-3 Settings Com

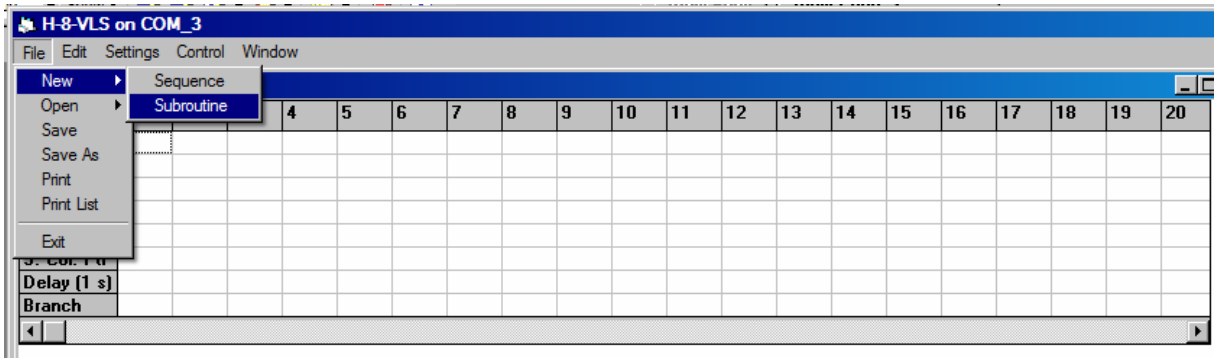


This menu enables the choice of COM number of the serial interface on the PC for the communication with the synthesizer. The COM number can be checked in the system settings of the PC.



The COM number can be changed by clicking on the text field and confirmed by clicking on the OK button.

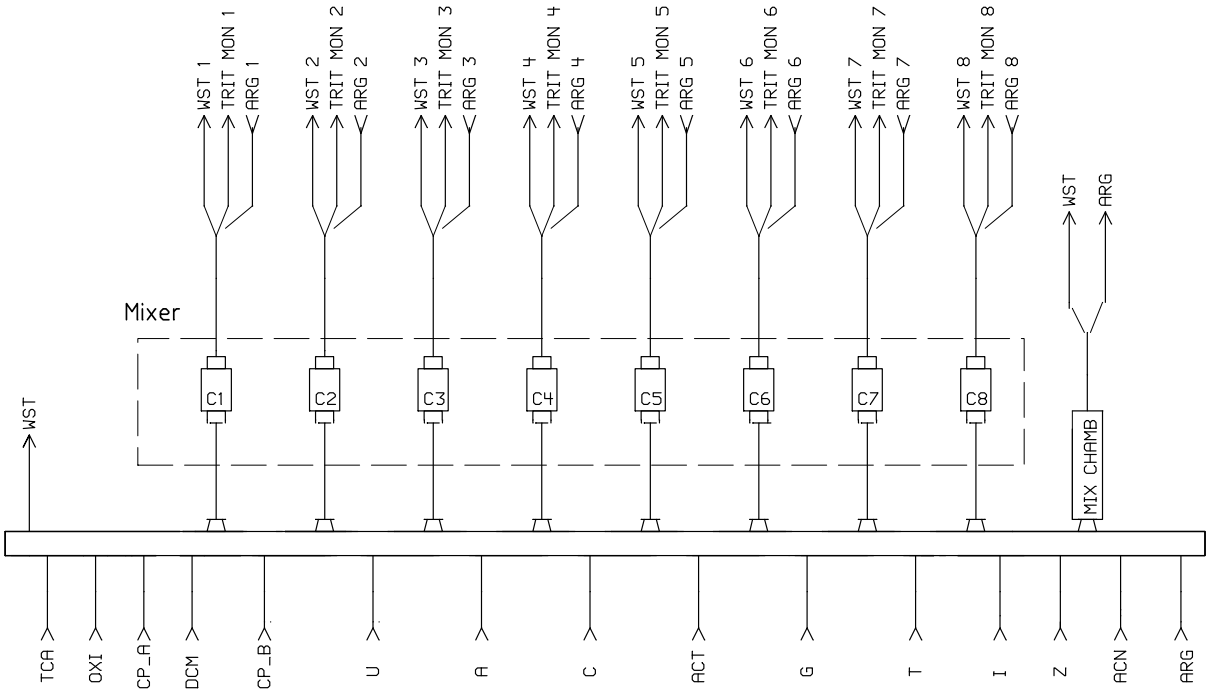
III. Subroutine programming



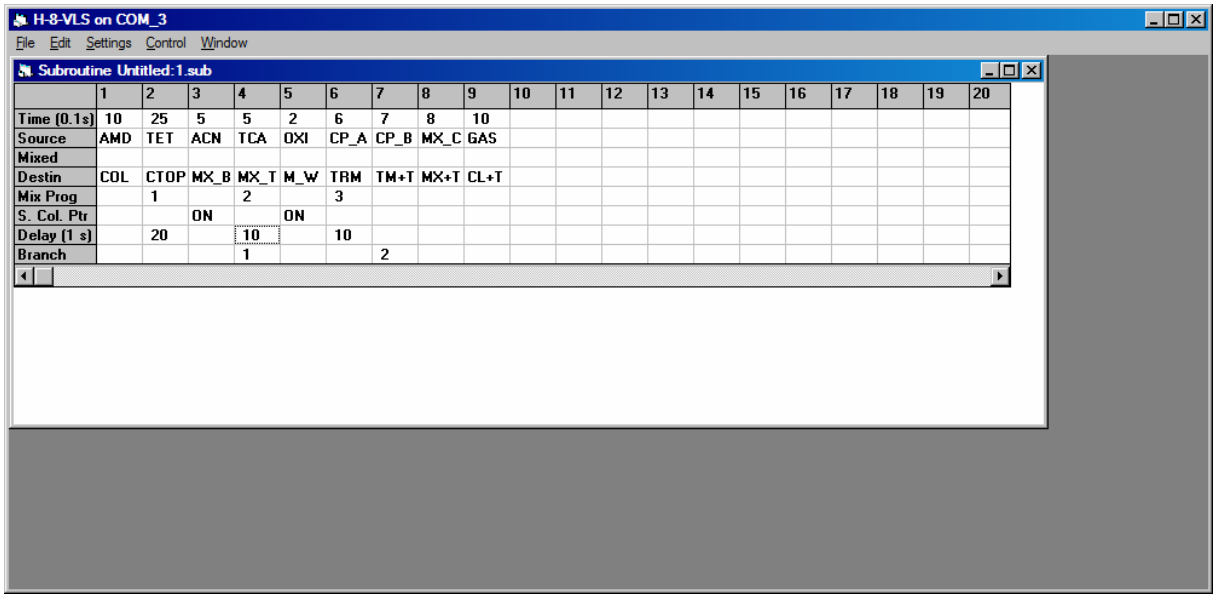
The subroutine is the most important tool for detailed controlling of the synthesis. Every instrument's activity will be defined in subroutines.

The single subroutine consists of 25 steps (columns in the subroutine's table) and every step is defined through the lines: **Time, Source, Mixed, Destin, Mix Prog., S. Col. Ptr., Delay, Branch.** The subroutine's columns define, step by step, from which bottle to which point of the synthesizer for how long the reagent must be delivered.

See the flow diagram of the H-SE-VLS synthesizer below:



H-8SE-VLS - Flow chart



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Time (0.1s)	10	25	5	5	2	6	7	8	10											
Source	AMD	TET	ACN	TCA	OXI	CP_A	CP_B	MX_C	GAS											
Mixed																				
Destin	COL	CTOP	MX_B	MX_T	M_W	TRM	TM+T	MX+T	CL+T											
Mix Prog		1		2		3														
S. Col. Ptr			ON		ON															
Delay (1 s)		20		10		10														
Branch				1			2													

- **Time**: the number in that field defines the duration time of the operation like flow time or wait time. The unit is 0.1 sec. (10 = 1 sec).

- **Source**: the code in the Source field defines from which point the reagent has to be delivered.

  Source code table:

Pressed key	Text in Source field	Description
'1'	AMD	Flow from amidite bottle
'2'	TET	Flow from TET bottle

‘3’	ACN	Flow from ACN bottle
‘4’	TCA	Flow from TCA bottle
‘5’	OXI	Flow from OXI bottle
‘6’	CP_A	Flow from CP_A bottle
‘7’	CP_B	Flow from CP_B bottle
‘8’	MX_C	Flow from Mix chamber
‘9’	GAS	Flow from ARG connection
‘D’	DCM	Flow from DCM bottle
‘A’	_A_	Flow from the bottle A
‘C’	_C_	Flow from the bottle C
‘G’	_G_	Flow from the bottle G
‘T’	_T_	Flow from the bottle T
‘S’	S_1	Flow from S_1 bottle
‘W’	S_2	Flow from S_2 bottle
‘Q’	S_3	Flow from S_3 bottle
‘E’	S_4	Flow from S_4 bottle
‘R’	S_5	Flow from S_5 bottle
‘D’	DCM	Flow from DCM bottle
‘M’	WMO	Wait only if modifikation
‘H’	WTH	Wait only if S-oligo

- **Mixed**: the code in the Mixed field defines from which point the reagent has to be delivered together with the reagent defined in source field. The delivery will be performed by switching the valves for the reagents defined in **both** fields: Source and Mixed. The keys to be pressed and the codes are exactly the same like Source codes.
- **Destin**: the code in the Destin field defines the point to where the reagent has to be delivered.

  Destination code table:

Pressed key	Text in Destin field	Description
‘1’	COL	Flow through the column to the waste
‘2’	CTOP	Flow from the column’s top, only with GAS as Source
‘3’	MX_B	Flow into the mix chamber.
‘4’	MX_T	Flow from the mix chamber’s top, only with GAS as Source
‘5’	M_W	Flow through the valve block to the waste.
‘6’	TRM	Flow through the column to the trityl monitor.
‘7’	TM+T	Flow through the column to the trityl monitor with flow time correction when more than one column to deliver

		simultaneously. For details see III-1-3
‘8’	MX+T	Flow into the mixing chamber with flow time correction when more than one column to deliver simultaneously. For details see III-1-3
‘9’	CL+T	Flow through the column to the waste with flow time correction when more than one column to deliver simultaneously. For details see III-1-3
‘A’	AMD	Flow into amidite bottle. The amidite position according to the sequence

- **Mix Prog**: switch on the mixer with the chosen program. Three programs (1 or 2 or 3 are allowed. The free programs of the mixer C1 or C2 or C3 will be correspondingly executed. **The mixer must be set to program C1 before the synthesis starts** to provide the correct execution.

- **S. Col. Ptr.**: means ‘Single Column Pointer’ which is the pointer for serial execution of the part of the subroutine, which is placed between two ‘ON’ markers in the subroutine. The parts of the subroutine which are beyond the ON markers will be executed simultaneously on all columns. For details see **III-1-1. Subroutine programming-examples**

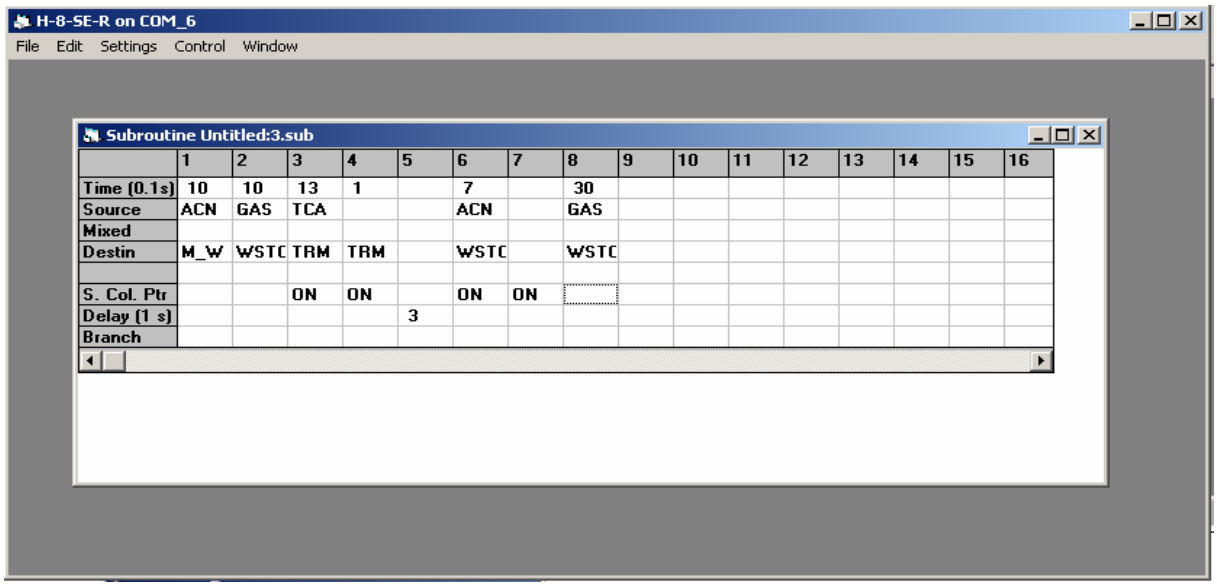
- **Delay**: means simply ‘wait – do nothing’ function. The unit is 1 sec.

- **Branch**: this field contains the branch subroutine’s call which will be executed due to the amidite in the synthesis. For details see **III-1-4. Subroutine programming-examples - Programming branches**

III-1. Subroutine programming-examples

III-1-1. Single Column Pointer programming. 1-Col. Mode Parameter.

Assuming all 8 columns synthesis on the synthesizer are running and following subroutine will be executed:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Time (0.1s)	10	10	13	1		7		30								
Source	ACN	GAS	TCA			ACN		GAS								
Mixed																
Destin	M_W	WSTC	TRM	TRM		WSTC		WSTC								
S. Col. Ptr			ON	ON		ON	ON									
Delay (1 s)					3											
Branch																

When Parameter **1-Col. Mode** set to **NO**

1-Col. Mode	NO
Switch Speed AMD	3
Switch Speed Reag	4
Collect DMT	YES
Add Time AMD Flow	0
Add Time REAG Flow	0
Add T. ACT/AMD Flow	0
Add T. MIX REAG Flow	0
Double Cycle	NO

The subroutine will be executed as follows:

1) Step 1: 1 sec ACN flow through the valve block to the waste
2) Step 2: 1 sec GAS flow through all 8 columns to the waste (column emptying)

Start of the Single Column Pointer.

3) Step 3: 1.3 sec TCA flow through columns 1 & 2 to the trityl monitor.
4) Step 4: 0.1 sec wait idle (only for formal marking at the end of the single column pointer).
5) Step 3a: 1.3 sec TCA flow through columns 3 & 4 to the trityl monitor.
6) Step 4a: 0.1 sec wait idle
7) Step 3b: 1.3 sec TCA flow through columns 5 & 6 to the trityl monitor.
8) Step 4b: 0.1 sec wait idle
9) Step 3c: 1.3 sec TCA flow through columns 7 & 8 to the trityl monitor.
10) Step 4c: 0.1 sec wait idle

End of the Single Column Pointer.

11) Step 5: 5 sec wait idle.

Start of the Single Column Pointer.

12) Step 6: 0.7 sec ACN flow through columns 1 & 2 to the waste.
13) Step 7: formal marking at the end of the single column pointer.
14) Step 6: 0.7 sec ACN flow through column 3 & 4 to the waste.
15) Step 7: formal marking at the end of the single column pointer.
16) Step 6: 0.7 sec ACN flow through column 5 & 6 to the waste.
17) Step 7: formal marking at the end of the single column pointer.
18) Step 6: 0.7 sec ACN flow through column 7 & 8 to the waste.
19) Step 7: formal marking at the end of the single column pointer.

End of the Single Column Pointer.

20) Step 8: 3 sec GAS flow through all 8 columns to the waste (column emptying)
21) Step 8a: 2 sec wait idle.

When Parameter **1-Col. Mode** set to **YES**

1-Col. Mode	YES
Switch Speed AMD	3
Switch Speed Reag	4
Collect DMT	YES
Add Time AMD Flow	0
Add Time REAG Flow	0
Add T. ACT/AMD Flow	0
Add T. MIX REAG Flow	0
Double Cycle	NO

1) Step 1: 1 sec ACN flow through the valve block to the waste
2) Step 2: 1 sec GAS flow through all 8 columns to the waste (column emptying)

Start of the Single Column Pointer.

3) Step 3: 1.3 sec TCA flow through column 1 to the trityl monitor.
4) Step 4: 0.1 sec wait idle (only for formal marking of the end of the single column pointer).
5) Step 3a: 1.3 sec TCA flow through column 2 the trityl monitor.
6) Step 4a: 0.1 sec wait idle
7) Step 3b: 1.3 sec TCA flow through column 3 the trityl monitor.
8) Step 4b: 0.1 sec wait idle
9) Step 3c: 1.3 sec TCA flow through column 4 the trityl monitor.
10) Step 4c: 0.1 sec wait idle
11) Step 3d: 1.3 sec TCA flow through column 5 the trityl monitor.
12) Step 4d: 0.1 sec wait idle
13) Step 3e: 1.3 sec TCA flow through column 6 the trityl monitor.
14) Step 4e: 0.1 sec wait idle
15) Step 3f: 1.3 sec TCA flow through column 7 the trityl monitor.

16) Step 4f: 0.1 sec wait idle
17) Step 3g: 1.3 sec TCA flow through column 8 the trityl monitor.
18) Step 4g: 0.1 sec wait idle

End of the Single Column Pointer.

19) Step 5: 5 sec wait idle.

Start of the Single Column Pointer.

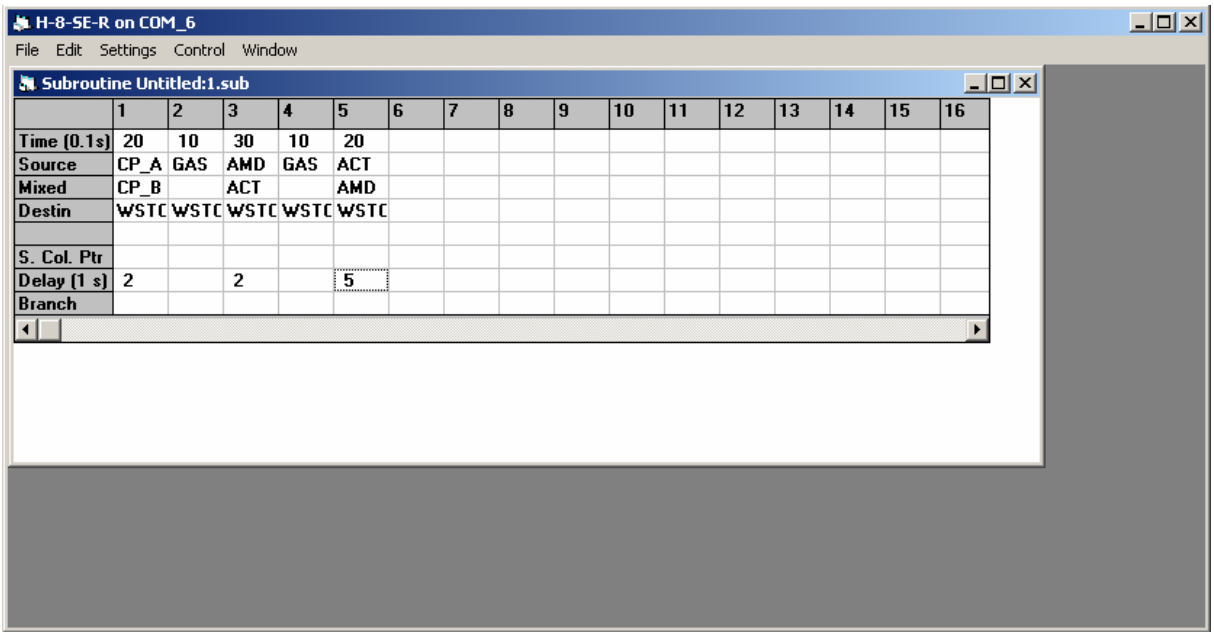
20) Step 6: 0.7 sec ACN flow through column 1 to the waste.
21) Step 7: formal marking at the end of the single column pointer.
22) Step 6a: 0.7 sec ACN flow through column 2 to the waste.
23) Step 7a: formal marking at the end of the single column pointer.
24) Step 6b: 0.7 sec ACN flow through column 3 to the waste.
25) Step 7b: formal marking at the end of the single column pointer.
26) Step 6c: 0.7 sec ACN flow through column 4 to the waste.
27) Step 7c: formal marking at the end of the single column pointer.
28) Step 6d: 0.7 sec ACN flow through column 5 to the waste.
29) Step 7d: formal marking at the end of the single column pointer.
30) Step 6e: 0.7 sec ACN flow through column 6 to the waste.
31) Step 7e: formal marking at the end of the single column pointer.
32) Step 6f: 0.7 sec ACN flow through column 7 to the waste.
33) Step 7f: formal marking at the end of the single column pointer.
34) Step 6g: 0.7 sec ACN flow through column 8 to the waste.
35) Step 7g: formal marking at the end of the single column pointer..

End of the Single Column Pointer.

36) Step 8: 3 sec GAS flow through all 8 columns to the waste (column emptying)
37) Step 8a: 2 sec wait idle.

The Parameter **1-Col. Mode** set to **YES** means flow only through one column at the same time. This can lead to an increase of cycle times.

III-1-2. Switch speed AMD-parameter
Switch speed Reag.-parameter



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Time (0.1s)	20	10	30	10	20											
Source	CP_A	GAS	AMD	GAS	ACT											
Mixed	CP_B		ACT		AMD											
Destin	WSTC	WSTC	WSTC	WSTC	WSTC											
S. Col. Ptr																
Delay (1 s)	2		2		5											
Branch																

1-Col. Mode	YES
Switch Speed AMD	3
Switch Speed Reag	4
Collect DMT	YES
Add Time AMD Flow	2
Add Time REAG Flow	4
Add T. ACT/AMD Flow	2
Add T. MIX REAG Flow	3
Double Cycle	NO

Used when AMD used as Source or Mixed

Used when Source and Mixed any other reagent

The subroutine will be executed as follows:

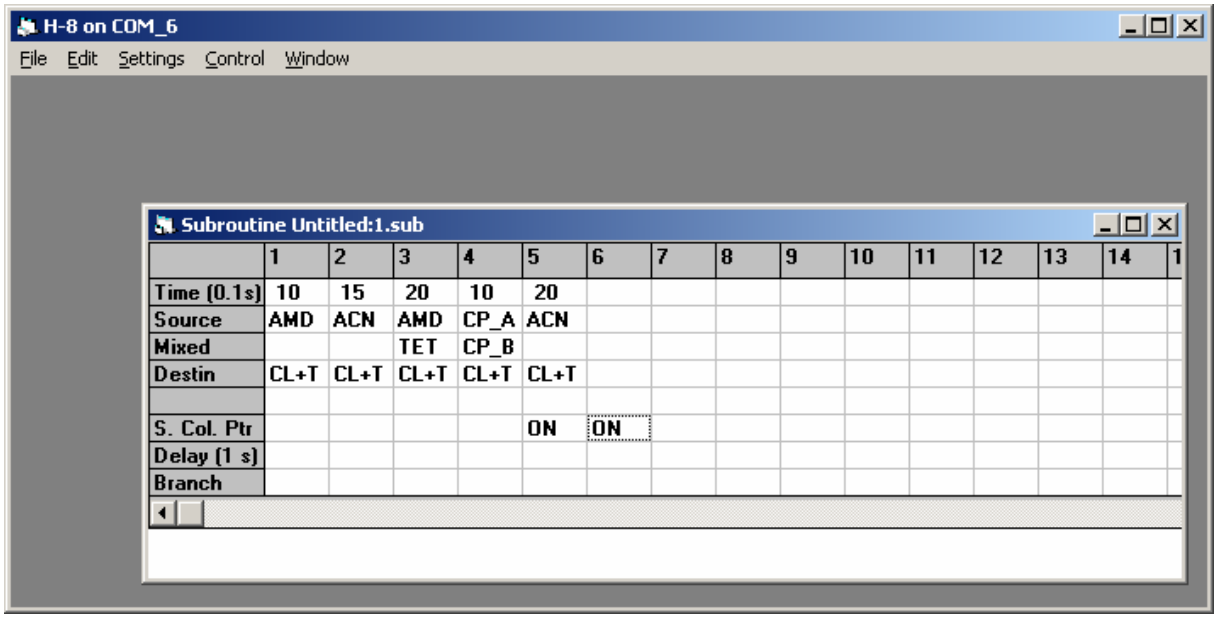
1) Step 1a: flow through all columns to the waste in switching mode: valve CP_A **on**, valve CP_B **off** for 5 units (Switch speed Reag. Parameter)
   Step 1b: flow through all columns to the waste in switching mode: valve CP_A **off**, valve CP_B **on** for 5 units.
   Repeat Step 1a and 1b until 2 seconds from the field 'Time' are over.
   Step 1c: Wait 2 sec. idle.
2) Step 2: GAS flow through all columns (emptying of the columns).
3) Step 3a: flow through all columns to the waste in switching mode: valve AMD **on**, valve ACT **off** for 3 units (Switch speed AMD Parameter)
   Step 3b: flow through all columns to the waste in switching mode: valve AMD **off**, valve ACT **on** for 3 units.
   Repeat Step 3a and 3b until 3 seconds from the field 'Time' are over.
   Step 3c: Wait 2 sec. idle.
4) Step 4: GAS flow through all columns (emptying of the columns).

5) Step 5a: flow through all columns to the waste in switching mode: valve ACT **on**, valve AMD **off** for 3 units (Switch speed AMD Parameter)
Step 5b: flow through all columns to the waste in switching mode: valve ACT **off**, valve AMD **on** for 3 units.
Repeat Step 5a and 5b until 2 seconds from the field 'Time' are over.
Step 5c: Wait 5 sec. idle.

III-1-3. Add time amd flow.
Add time reag flow
Add time tet/amd flow
Add time mix reag flow

1-Col. Mode	YES
Switch Speed AMD	3
Switch Speed Reag	4
Collect DMT	YES
Add Time AMD Flow	2
Add Time REAG Flow	4
Add T. ACT/AMD Flow	2
Add T. MIX REAG Flow	3
Double Cycle	NO

Assuming all 8 columns synthesis on the synthesizer and following subroutine:

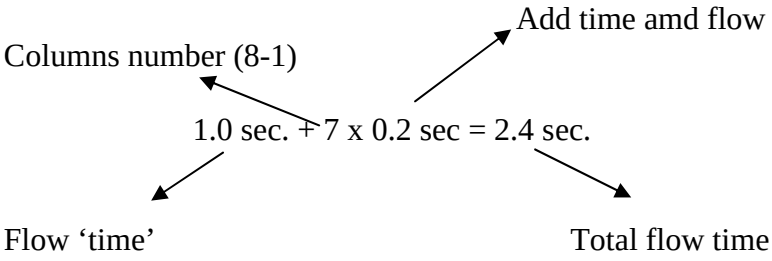


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1
Time (0.1s)	10	15	20	10	20										
Source	AMD	ACN	AMD	CP_A	ACN										
Mixed			TET	CP_B											
Destin	CL+T	CL+T	CL+T	CL+T	CL+T										
S. Col. Ptr					ON	ON									
Delay (1 s)															
Branch															

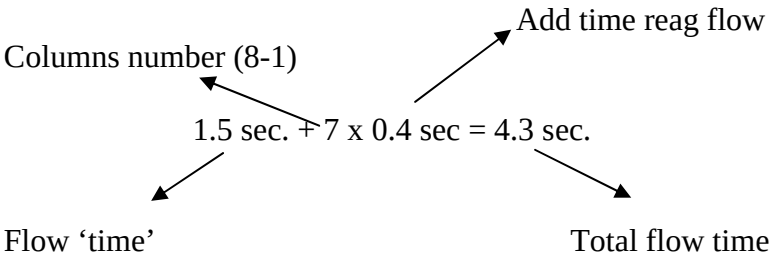
The time adding (time correction) will be executed only when **more than 1** columns have to be delivered simultaneously.

The subroutine will be executed as follows:

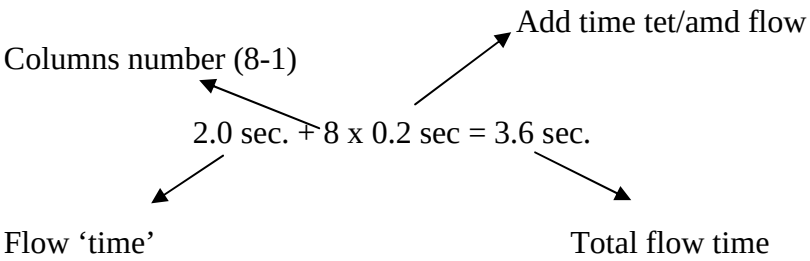
1) **Step 1**: flow through all columns to the waste for 1 sec + time correction ‘add time amd flow’. The time correction will be calculated as follows:



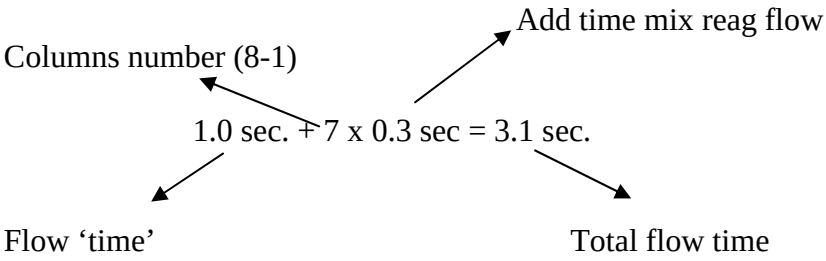
2) **Step 2**: flow through all columns to the waste for 1.5 sec + time correction ‘add time reag flow’. The time correction will be calculated as follows:



3) **Step 3**: flow through all columns to the waste for 1 sec + time correction ‘add time tet/amd reag flow’. The time correction will be calculated as follows:



4) **Step 4**: flow through all columns to the waste for 1 sec + time correction ‘add time mix reag flow’. The time correction will be calculated as follows:



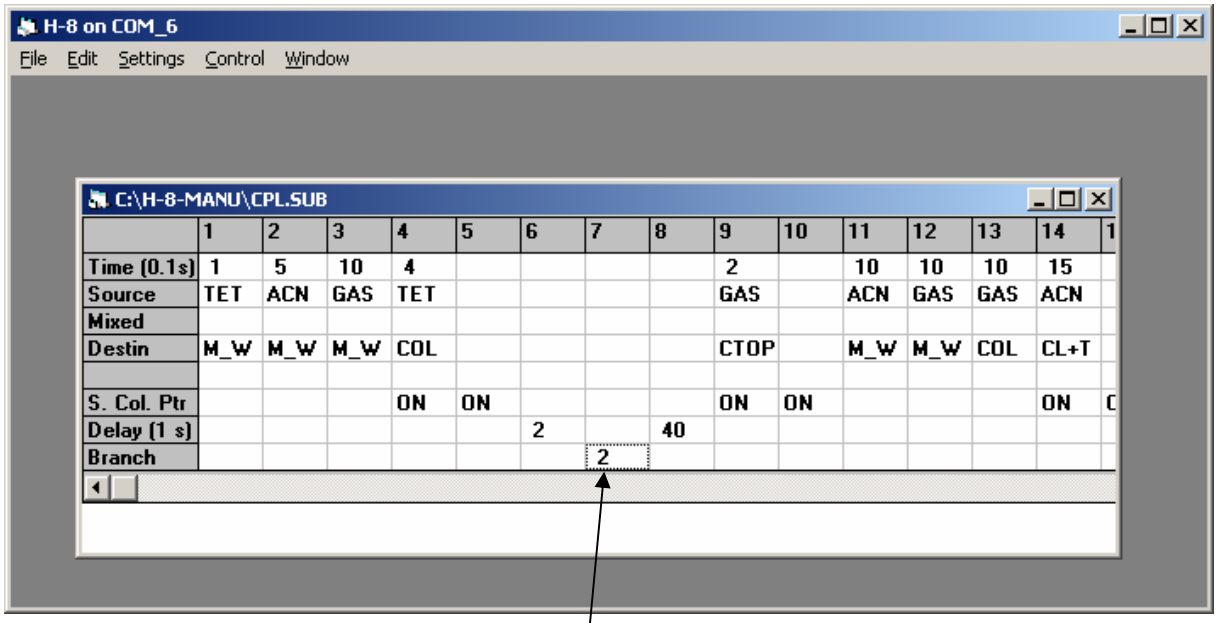
Step 5 & 6: flow to the waste in 'single column mode'. The time correction will be 0.4 sec. 2 columns will be delivered in this mode. See **III-1-1. Single Column Pointer programming.**

III-1-4. Branch Programming

'Branch' allows the execution of the subroutine due to the amidite in the sequence and is very useful for coupling programming.
Assuming following sequences in step 6 of the synthesis and subroutine

5' 3'

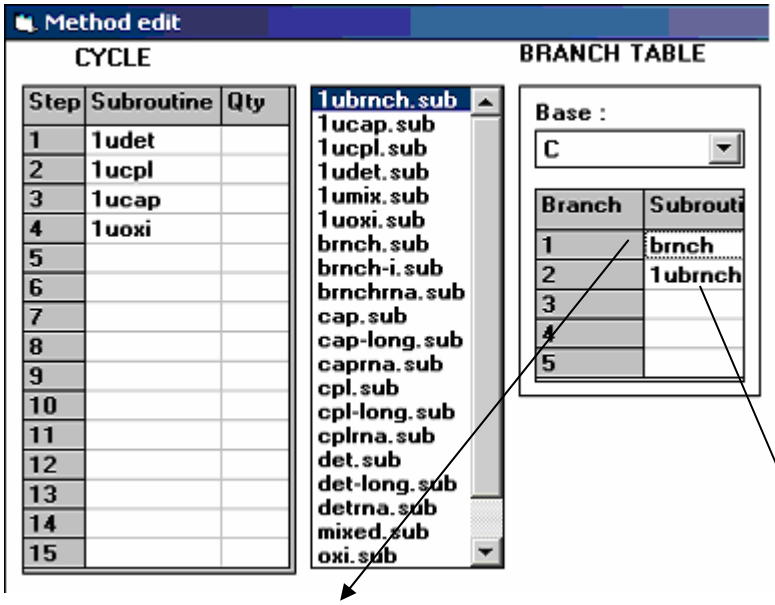
COL 1: ACC AGT GTA CCA CGT **C**CG ACT
COL 2: CCA GTG TAC CAC GTA **C**GA CTG
COL 3: CAG TGT ACC ACG TAC **G**AC TGG
COL 4: AGT GTA CCA CGT ACG **A**CT GGA
COL 5: GTG TAC CAC GTA CGA **C**TG GAC
COL 6: TGT ACC ACG TAC GAC **T**GG ACA
COL 7: GTA CCA CGT ACG ACT **G**GA CAA
COL 8: TAC CAC GTA CGA CTG **A**CA AGG



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1
Time (0.1s)	1	5	10	4					2		10	10	10	15	
Source	TET	ACN	GAS	TET					GAS		ACN	GAS	GAS	ACN	
Mixed															
Destin	M_W	M_W	M_W	COL					CTOP		M_W	M_W	COL	CL+T	
S. Col. Ptr				ON	ON				ON	ON				ON	C
Delay (1 s)						2		40							
Branch							2								

When the subroutine execution reaches step 7 of above subroutine:

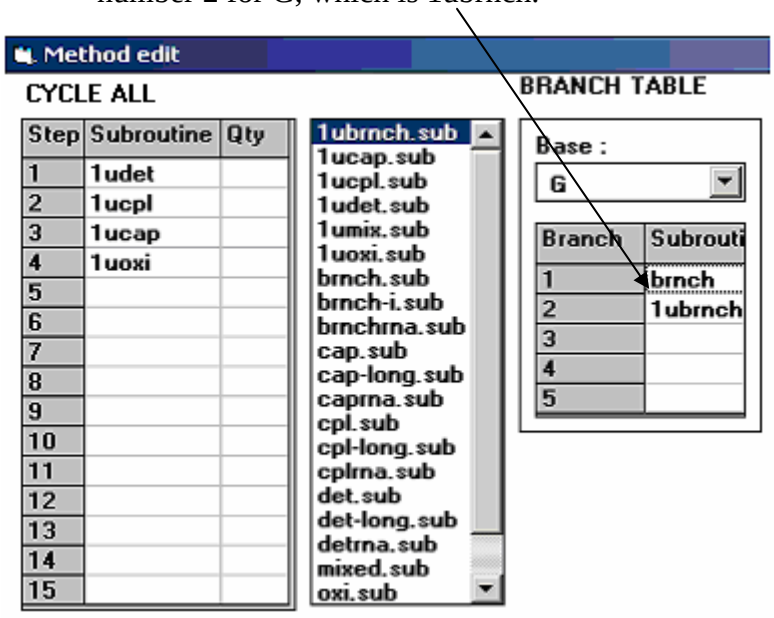
1) The program checks which amidite occurs in the sequence on the column 1. It is **C** in the example.
2) The programs check in the Table (**III-2 Tables**) the subroutine's name from the table on the place pointed by the number from the field 'Branch' in the subroutine.



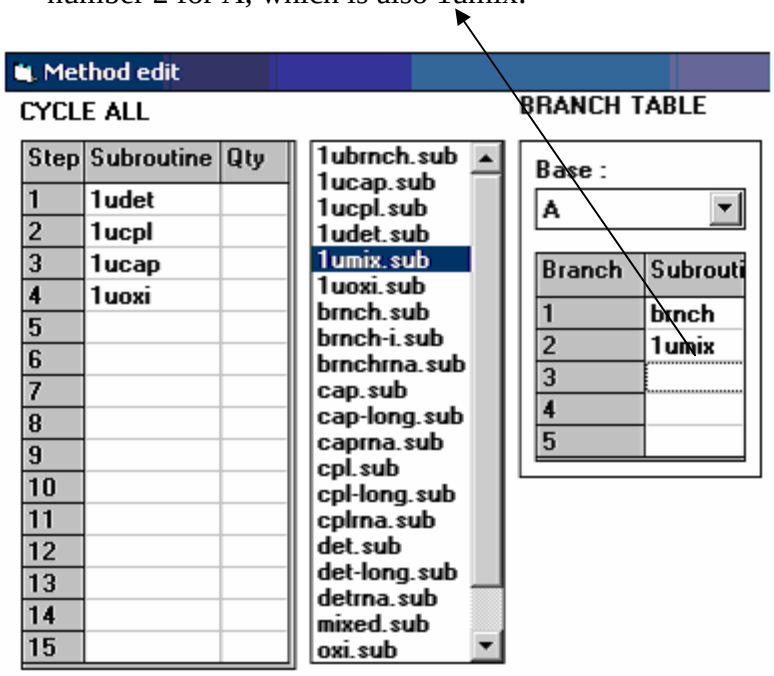
The number for the branch
From the field 'Branch' in the subroutine

Subroutine's name which will be executed now

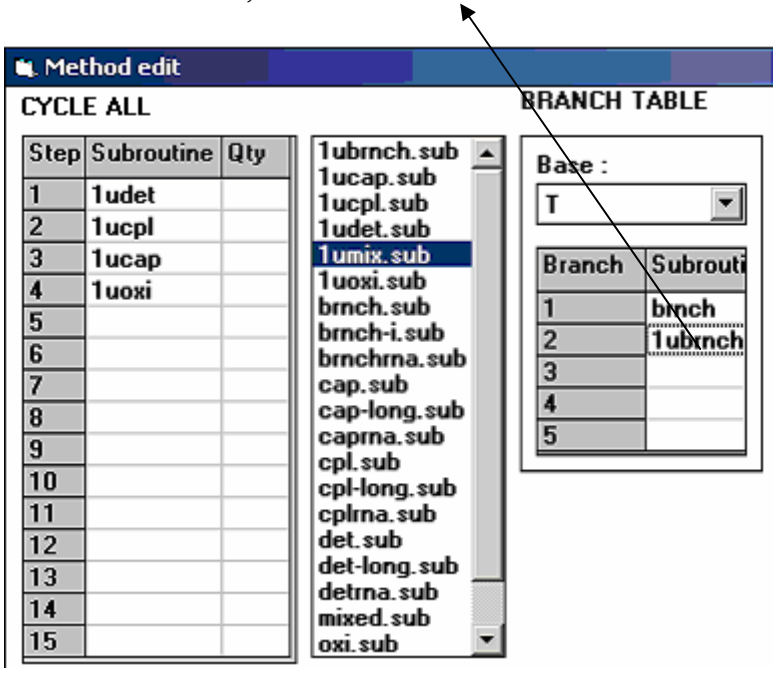
3) The program executes completely the subroutine '1ubrnch' which is pointed by the number form the field 'Branch' in the subroutine. The branch subroutine, in this case 1ubrnch, will be executed for **all** sequences in which **C** occurs, in the example for the columns **1 2 5.**
4) When branch subroutine is finished, the execution returns to step 7 of the subroutine and checks the next amidite which was not involved yet; **G** on column 3 in the example.
5) The programs check in the Table the subroutine's name from the table on the place number 2 for G, which is 1ubrnch.



6) The program executes completely the subroutine ‘1ubrnch’ **all** sequences in which **G** occurs, in the example for the columns **3 7.**
7) When branch subroutine is finished, the execution returns to step 7 of the subroutine and checks the next amidite which was not involved yet; **A** on column 4 in the example.
8) The programs check in the Table the subroutine’s name from the table on the place number 2 for A, which is also 1umix.



9) The program executes completely the subroutine ‘1umix’ **all** sequences in which **A** occurs, in the example for the columns **4 8.**
10) When branch subroutine is finished, the execution returns to step 7 of the subroutine and checks the next amidite which was not involved yet; **T** on column 6 in the example.
11) The programs check in the Table the subroutine’s name from the table on the place number 2 for T, which is also 1ubrnch.



12) When branch subroutine is finished, the execution returns to step 7 of the subroutine and checks the next amidite which was not involved yet. The execution resumes at the step 8 of the subroutines when all amidites had been already involved.

It is not allowed to use branch call within branch subroutine.