

# EECE416 :Microcomputer Fundamentals and Design

Easy68K

– Editor, Assembler, and Simulator

**TUTORIAL**

Charles Kim

# Easy68K and Download

EASy68K Home, Free 68000 Assembler, 68000 Simulator, 68000 Assembly Language - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.easy68k.com/

Most Visited Free Hotmail Customize Links Windows Marketplace Windows Media Windows

[Prof Kelly](#) [EASy68K](#) [Examples](#) [Forum](#)

## EASy68K Editor/Assembler/Simulator for the 68000



Welcome to the EASy68K home page. EASy68K is a 68000 Structured Assembly Language edit, assemble and run 68000 programs on a Windows PC. No additional hardware is required. project distributed under the GNU general public use license.

EASy68K, the #1 [68000 Assembler](#) and [68000 Simulator](#) according to Google.

### Download

Check the [Forum](#) for latest version information.

[SetupEASy68K.exe](#) Executable with installer

Ads by Google

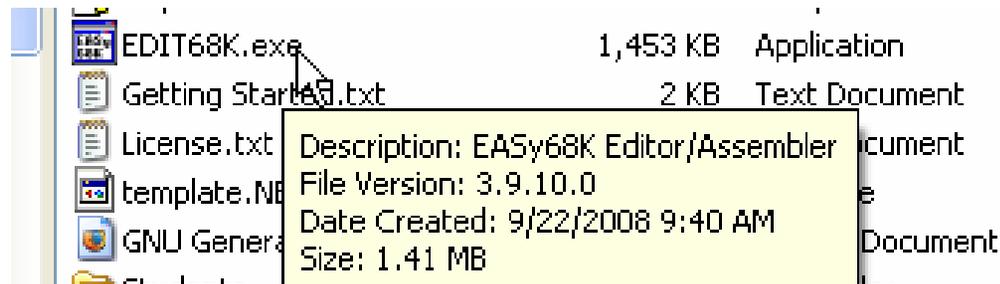
**Prototype PCB Assembly**  
24 hour turn, Full-service, RoHS New customer discount  
[www.screamingcircuits.com](http://www.screamingcircuits.com)

Download and Install this!

# EDIT68K

## ⌘ EDIT68K.EXE

📁 For Coding and Assembling



## ⌘ Coding

## ⌘ Save

```
EASy68K Editor/Assembler v3.9.10
File Edit Project Options Window Help
TrapExample1.X68
-----
* Program   : TrapExample.X68
* Written by : CK
* Date      : 01OCT08
* Description: Trap examples under Easy68k Environment
-----
;DATA PART
BS      EQU  $08   Backspace
HT      EQU  $09   Tab (horizontal 5 characters)
LF      EQU  $0A   New line (line feed)
VT      EQU  $0B   Vertical tab (4 lines)
FF      EQU  $0C   Form Feed (Always end printing with a Form Feed.)
CR      EQU  $0D   Carriage Return

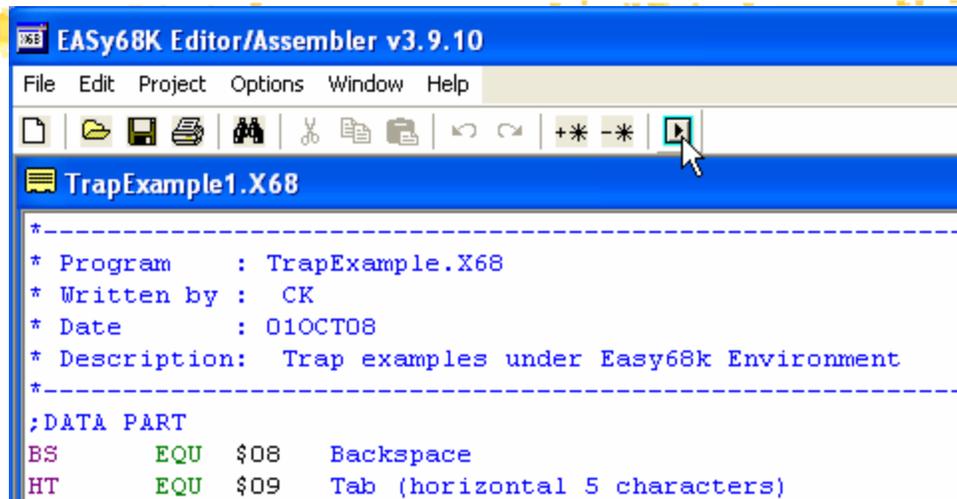
                ORG  $100
msg      DC.B   CR,LF,'good guess!',CR,LF
                DC.B   0                                ;string must be ended with 0
wmsg     DC.B   CR,LF,'guess again',CR,LF
                DC.B   0                                ;string ended with 0
inqr     DC.B   'Guess a character [a - z]',CR,LF
                DC.B   0                                ;ended with 0

                ORG  $200
Store    DS.B   30                                ;Allocate 4 Bytes starting @Store. MEMORY

;TRAP # in EASY68K
; (Put Task number into DO)
; then TRAP #15
```

# Assembling

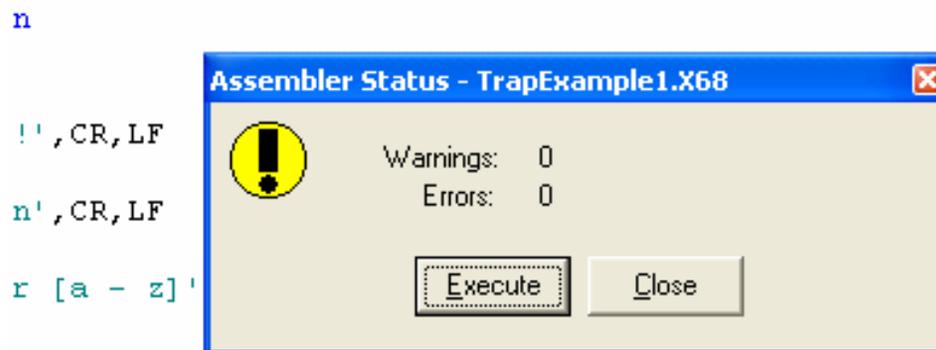
- ⌘ Click the "Assemble Source Code" button OR F9 Key



The screenshot shows the EASy68K Editor/Assembler v3.9.10 interface. The menu bar includes File, Edit, Project, Options, Window, and Help. The toolbar contains various icons, including a button with a right-pointing arrow and a lightning bolt, which is the 'Assemble Source Code' button. A mouse cursor is clicking this button. The main window displays the source code for 'TrapExample1.X68'.

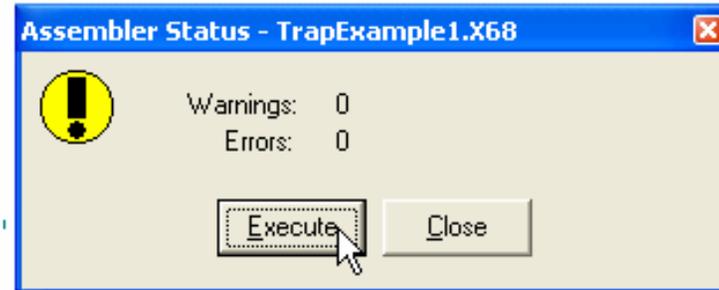
```
*  
* Program      : TrapExample.X68  
* Written by   : CK  
* Date        : 01OCT08  
* Description:  Trap examples under Easy68k Environment  
*  
;DATA PART  
BS      EQU  $08   Backspace  
HT      EQU  $09   Tab (horizontal 5 characters)
```

- ⌘ When there is no Error, the following window pops up.



# Simulation

⌘ Click, "Execute" button.



⌘ Initial Simulation Screen

⌘

C:\Easy68k\TrapExample1.S68

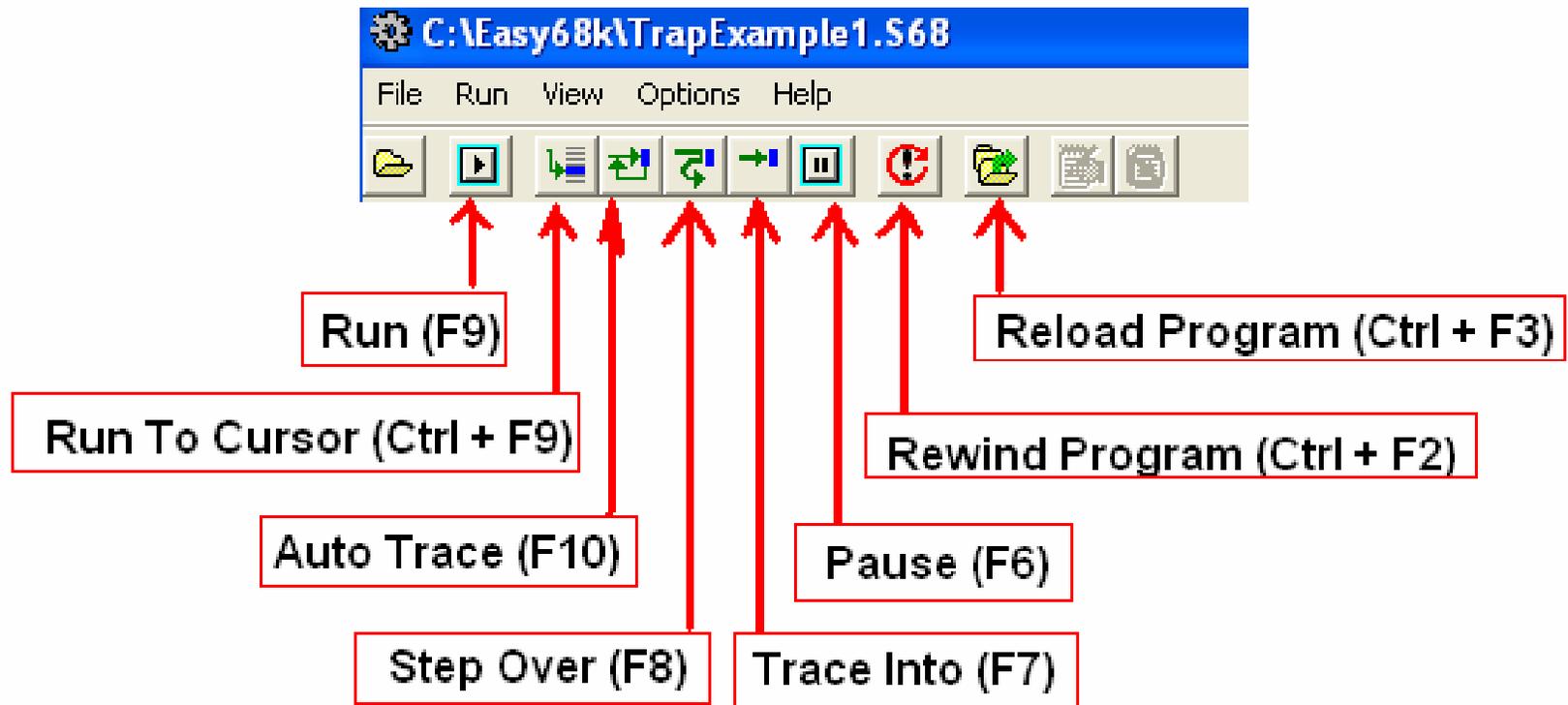
File Run View Options Help

Registers

D0=00000000	D4=00000000	A0=00000000	A4=00000000	T S INT XNZVC	Cycles
D1=00000000	D5=00000000	A1=00000000	A5=00000000	SR=0010000000000000	0
D2=00000000	D6=00000000	A2=00000000	A6=00000000	US=00FF0000	Clear Cycles
D3=00000000	D7=00000000	A3=00000000	A7=01000000	SS=01000000	PC=00000000

Address	Code	Line	Source
0000021E		30	;TASK No. FUNCTIONS
0000021E		31	;=====
0000021E		32	;0 PRINT_MSG (CRLF) (Display Mess
0000021E		33	;1 PRINT_MSG (Same as 0 but w/o C
0000021E		34	;5 READ_CHR Read a single charact
0000021E		35	;6 PRINT_CHR Display a single ch
0000021E		36	;9 HALT halt the simulator
0000021E		37	
0000021E		38	
0000021E		39	;PROGRAM PART
00000000		40	START ORG \$0

# Simulation Buttons

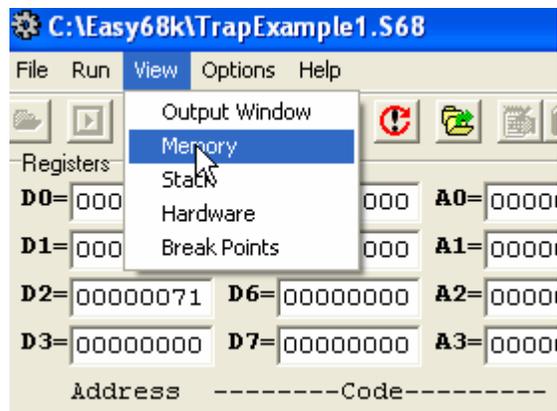


# Memory View

⌘ After Auto Trace or Run (with Cmd screen)

```
Sim68K I/O
Guess a character [a - z]
a
guess again
c
guess again
d
guess again
e
guess again
f
guess again
g
guess again
q
good guess!
```

⌘ Click “View” and Select “Memory”





# TRAPs in EASY68K

## ⌘ TRAP

- ☒ Interruption of Execution
- ☒ Respond Programmer's Task Command

## ⌘ Number of Tasks in Easy68K

- ☒ 22 Tasks
- ☒ Examples
  - ☒ Read Message (from Keyboard)
  - ☒ Print Message (to Computer Monitor)
  - ☒ Read a Character
  - ☒ Print a Character
  - ☒ Read a Number
  - ☒ Print a Number
  - ☒ Etc

## ⌘ Traditional TRAP tasks

- ☒ Print Message (to Computer Monitor) of address at A1
  - --- Task # 0 (with CR.LF)
  - --- Task #1 (without CR.LF)
- ☒ Read a Character and store it at D1.B --- Task #5
- ☒ Print a Character stored at D1.B --- Task #6
- ☒ Key Echo On on or off --- Task #12 (OFF- D1.B==0, ON --- D1.B== Non zero)

## ⌘ How is a TRAP called/executed?

- ☒ Put task number to A1 or D0
- ☒ Put address into A1 (for "print message" case)
- ☒ Then "Trap #15" instruction line

```
MOVE.B #5, D0          Read a single character
TRAP   #15             Store into D1
```

```
└ MOVE.B #0, D0
  MOVEA.L #rmsg, A1
  TRAP   #15
```