

EASy68K Editor/Assembler/Simulator Tutorial

Written by Charles Norona

Special thanks to Robert Templeton and Prof. Chuck Kelly of Monroe County Community College, Michigan.

12th of January, 2009

Click on the link below to download or run the installer for EASy68k.

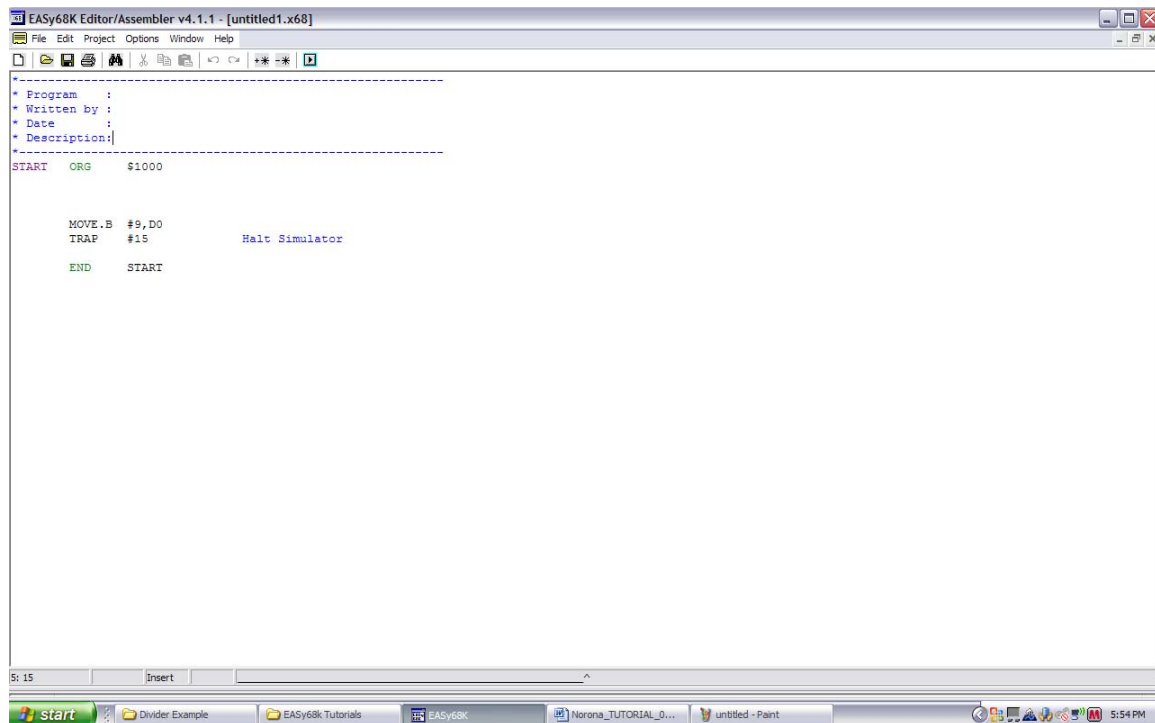
<http://www.monroecc.edu/ckelly/Files/SetupEASy68K.exe>

Follow the installation instructions. Once complete, the EASy68k directory and executable should be located in **Start -> My Programs -> EASy68k**.

For more information on EASy68k visit: www.easy68k.com

Using EASy68k

Initializing EASy68k brings up a window like this which is a skeleton of an assembly code. **NOTE:** You might find it useful to delete the line `MOVE.B #9,D0` if you use the D0 register for anything.

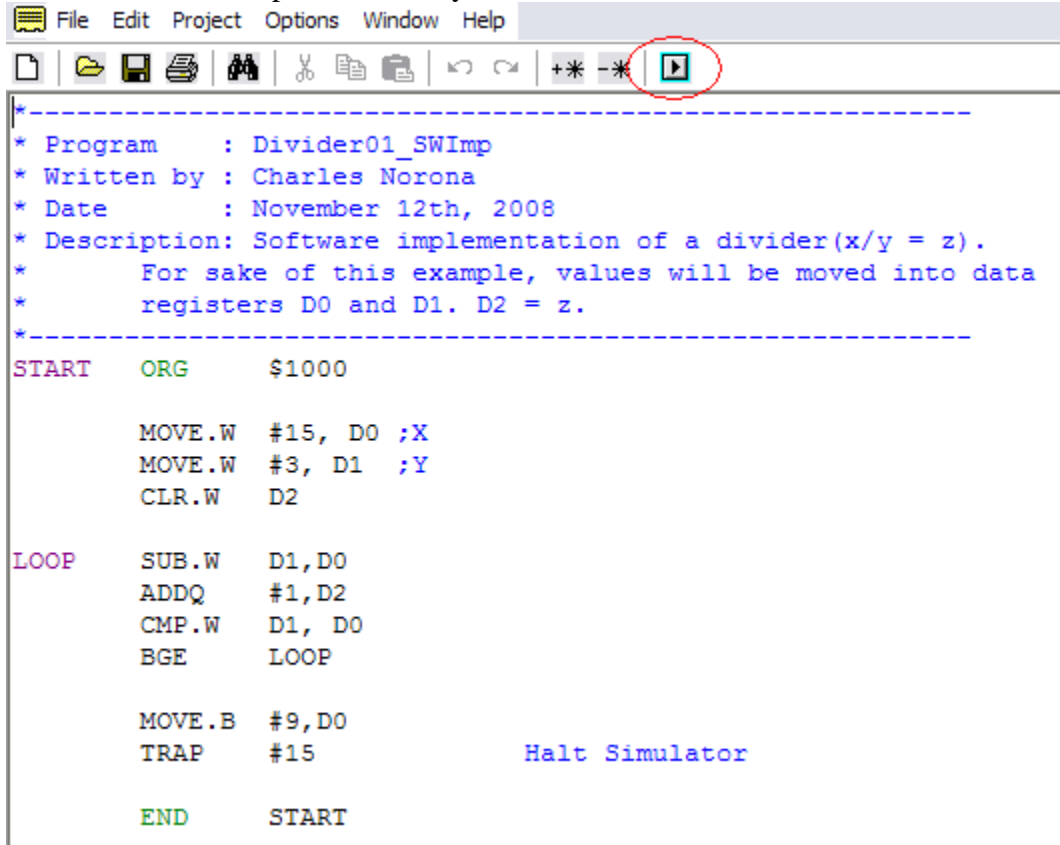


The screenshot shows the EASy68K Editor/Assembler v4.1.1 window with the following assembly code:

```
-----  
* Program :  
* Written by :  
* Date :  
* Description :  
-----  
START ORG $1000  
  
MOVE.B #9,D0  
TRAP #15 Halc Simulator  
  
END START
```

Using this code as an example, one can see how the last MOVE instruction will cause some confusion when D0 is supposed to be of interest.

When the code is complete and ready to execute, click on the “assemble source” icon.



The screenshot shows a software interface for editing assembly code. At the top is a menu bar with 'File', 'Edit', 'Project', 'Options', 'Window', and 'Help'. Below the menu bar is a toolbar with various icons. The 'assemble source' icon, which is a square with a right-pointing arrow, is circled in red. The main area of the window displays assembly code with several comments and instructions. The code is as follows:

```
*-----*
* Program      : Divider01_SWImp
* Written by   : Charles Norona
* Date        : November 12th, 2008
* Description: Software implementation of a divider(x/y = z).
*             For sake of this example, values will be moved into data
*             registers D0 and D1. D2 = z.
*-----*
START  ORG      $1000

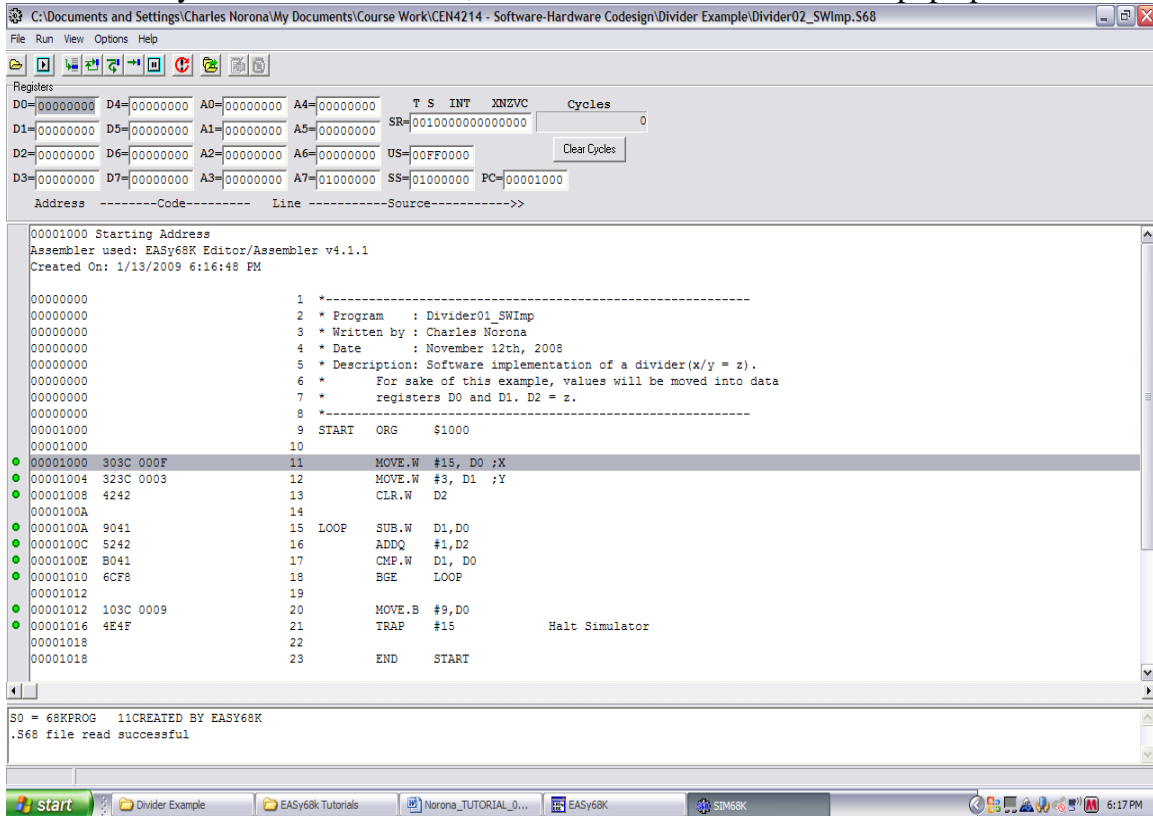
      MOVE.W   #15, D0 ;X
      MOVE.W   #3, D1  ;Y
      CLR.W    D2

LOOP   SUB.W   D1,D0
      ADDQ    #1,D2
      CMP.W   D1, D0
      BGE     LOOP

      MOVE.B   #9,D0
      TRAP    #15           Halt Simulator

      END     START
```

Once you assemble the code, the simulator user interface will pop up.



The green buttons to the left of the code are break points that turn red when activated. To activate a breakpoint, just click on it.

The icons at the top are as follows with their descriptions.

Open – To open a .S68 file.

Run – Executes the program up until it hits the TRAP or an activated breakpoint.

Run To Cursor – Runs the program but stops at the line highlighted in grey.

Auto Trace – Runs the program whilst highlighting each line the simulator executes.

Step Over – Executes the code and highlights one line at a time for the execution of subroutines.

Trace Into – Executes the code one line at a time, highlighting each line of code that it executes including subroutines.

Pause – Halts execution of simulation.

Rewind Program – Clears all registers and cycles then resets program counter.

Reload Program – Load the last program simulated.