Visual Disassembler for the Atmel © AVR Microprocessor Family

October 26, 2016

Table of Contents

Introduction

Visual Disassembler for the Atmel © AVR is an interactive disassembler for the AVR microprocessor family. The user simply opens an Intel hex file and the file is disassembled and placed in a text view. The user does not edit the text directly. Comments and program/ IO labels are added by address through dialog boxes. Each entry shows an immediate result in the text view. Program / IO labels are resolved throughout the entire file. Intermediate work can be saved and resumed later. Upon completion of the disassembly the file can be saved as an assembly listing text file. The assembly listing can be be easily converted to an assembler source file to reconstruct the original hex file. The assembler source file may require editing due to the requirements of the assembler.

New features

- Added AVR part selection
    - resolves memory space, registers, and ISR vectors
- Added memory space control
    - Allows definition of data and code areas in the memory space

The program is a plug-in for the Eclipse IDE and can be sold as an individual component or bundled with the Eclipse CDT IDE.

This can be bundled with the Visual Disassembler:
Latest ECLIPSE CDT
http://www.eclipse.org

Visual Disassembler Jar
com.favorites4u.visualdisassembler.avr_1.0.2.jar
http://www.CreativeRealtimeHeuristic.com

Questions?
Contact us: rhurst24811@gmail.com

Requirements

Visual Disassembler

The Visual Disassembler requires Eclipse version 3.4.x. and will run on both Windows and Linux platforms.

It must be installed in the eclipse/plugins directory.

Eclipse

Eclipse 3.4.x requires the Java Runtime Environment (JRE)6.

It can be obtained from http://java.sun.com.

Installation

Unzip the file in the CD into a new folder.

Find the eclipse folder and locate the file eclipse.exe.

Create a shortcut to your desktop.

Note:

You can create a clean directory by unzipping the eclipse CDT file into a new folder.

Next copy the visual disassembler jar file into the eclipse\plugins directory.

Type eclipse -clean and verify that the visual disassembler plugin is installed by looking in the Help\About Eclipse\Plug-in Details dialog box.

Quick Start

1. Open hex file
2. Select Device
3. Modify Control Table
4. Add comments
5. Add labels
6. Save as vdavr file
7. Repeat steps 3 through 6 until satisfied with disassembly
8. Save listing to text file

Typical View

A typical view of the Visual Disassembler is shown below in figure 1.

File Types Opened by the Application:

The default file extension is - .vdavr

An alternative extension is an Intel hex file - .hex

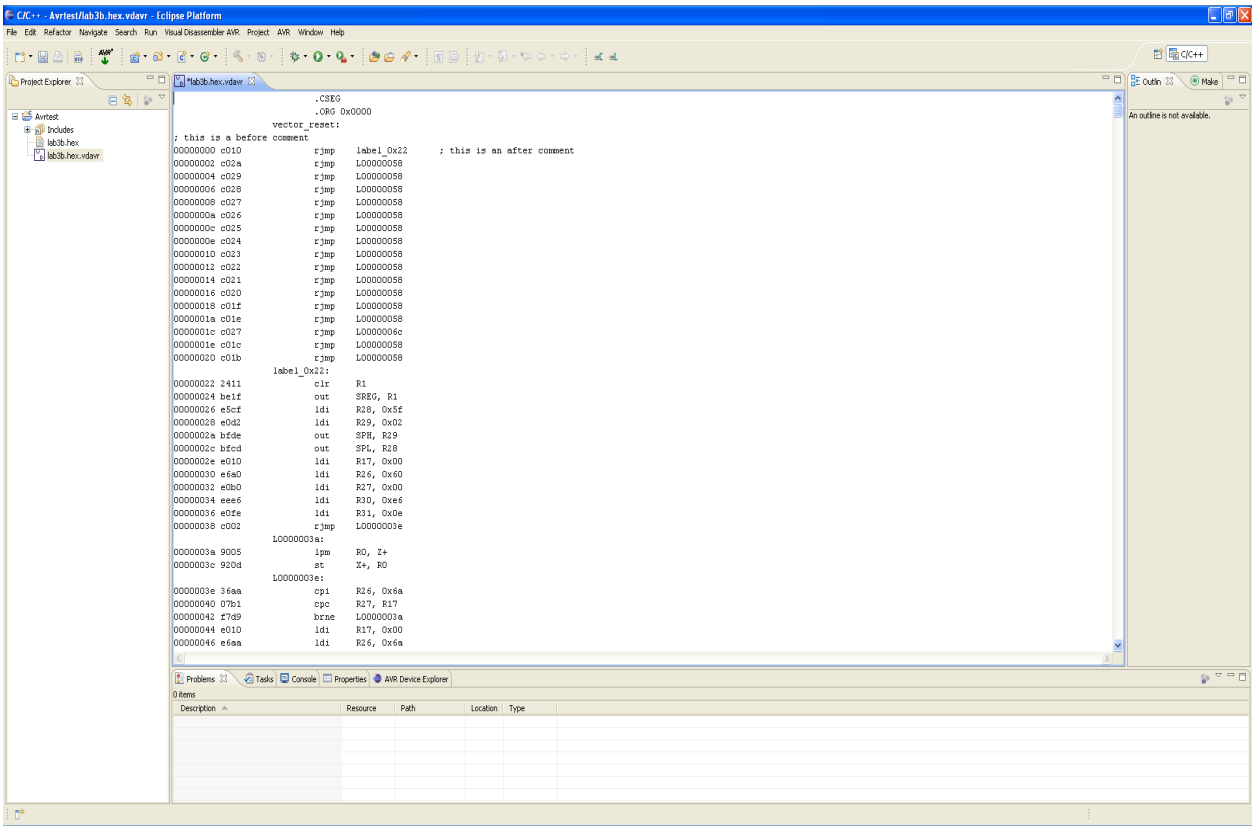


Figure 1

Open a File

Intel Hex File

Figure 2 shows Open / Other... dialog box.

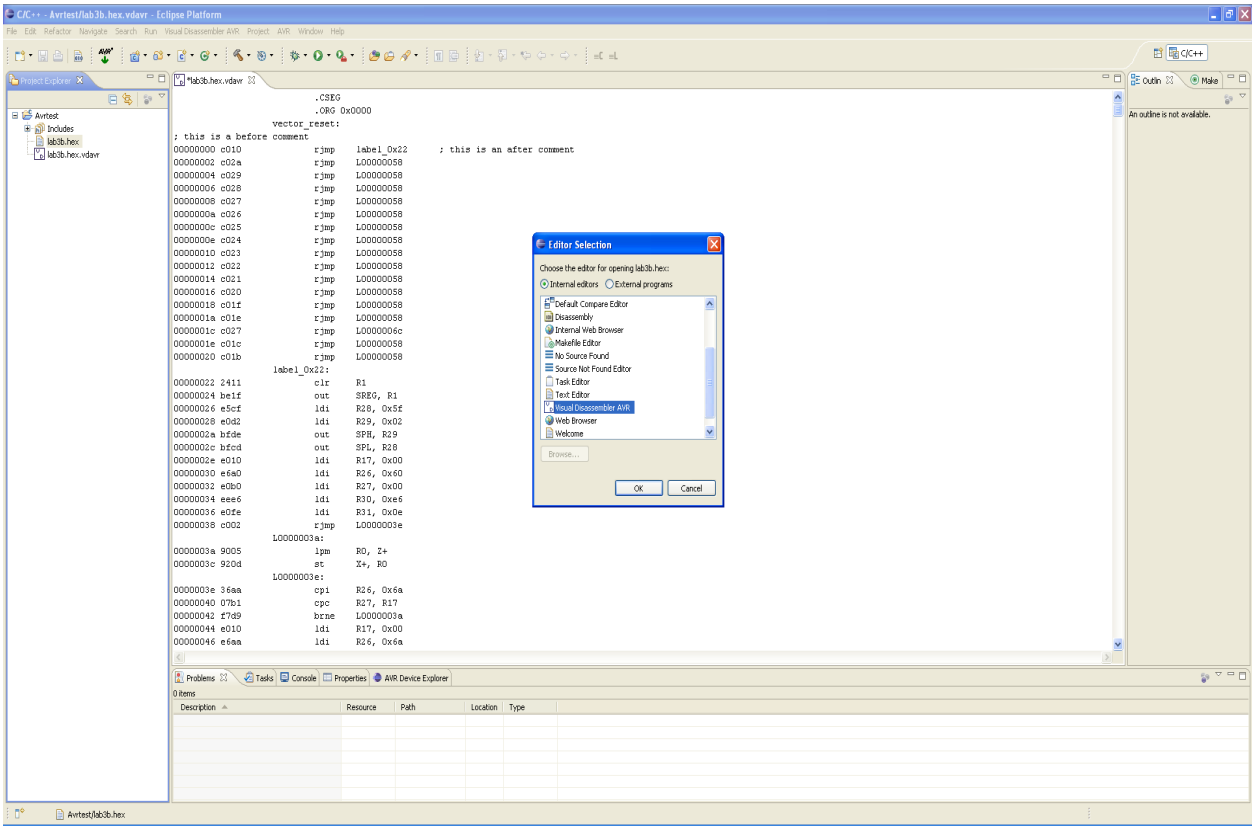


Figure 2

VDAVR File – Default file format

The Open With dialog box is shown in figure 3.

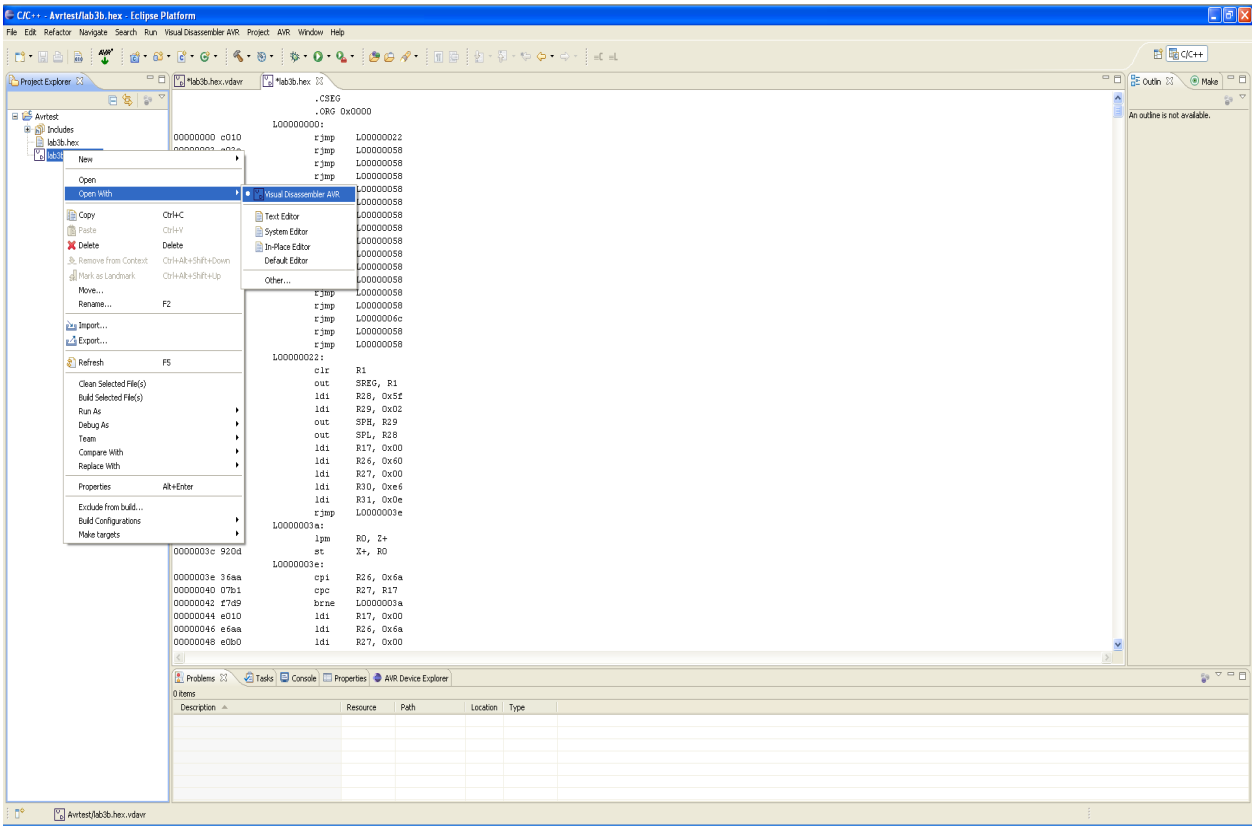


Figure 3

Menu View

Figure 4 shows the Visual Disassembler AVR menu items.

Select Device...	Allows the user to select an AVR device file (*resolves registers, vectors and memory space*)
Edit Comment...	Allows the user to add comments at any location in the source file
Edit I/O / Program Label...	Allows the user to add labels any program location *(These labels are resolved on branch and call instructions.)* Allows the user to add I/O port labels *(These labels are resolved in the in / out instructions.)*
Edit Control Table...	Allows the user to define the memory space boundaries

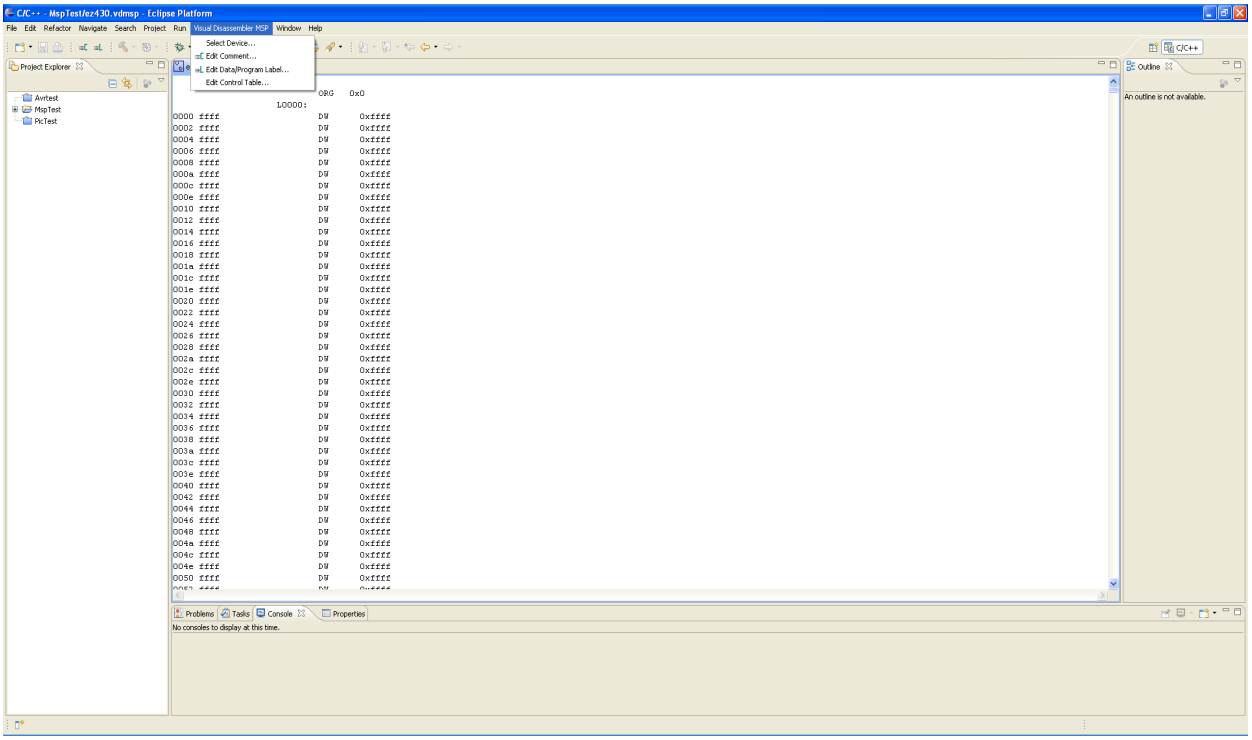
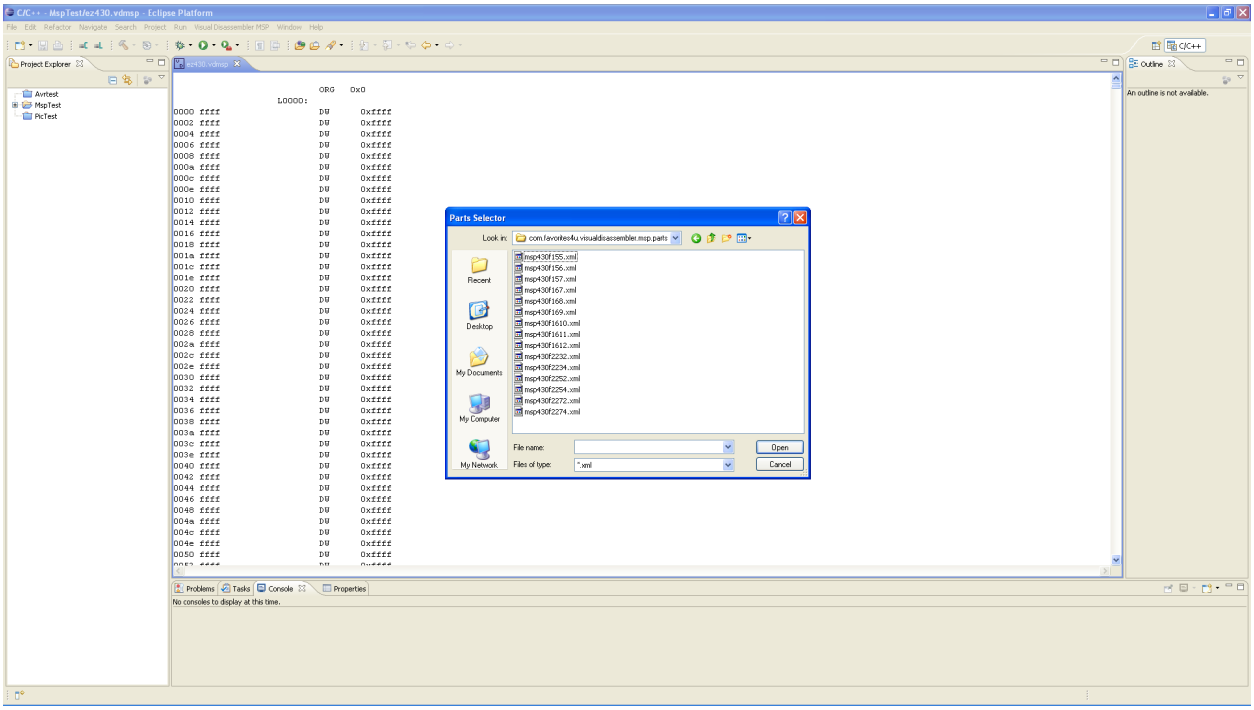


Figure 4

Menu Item – Select Device...

This menu item allows the user to select a specific device to use for the disassembly. It provides memory space, register, and ISR vector information to the disassembler.



ToolBar Button View – Edit Comment Tooltip

The Visual Disassembler AVR Edit Comment Button shown in figure 5 duplicates the menu item in the Visual Disassembler AVR / Edit Comment... menu.

Edit Comment - Allows the user to add comments at any location in the source file

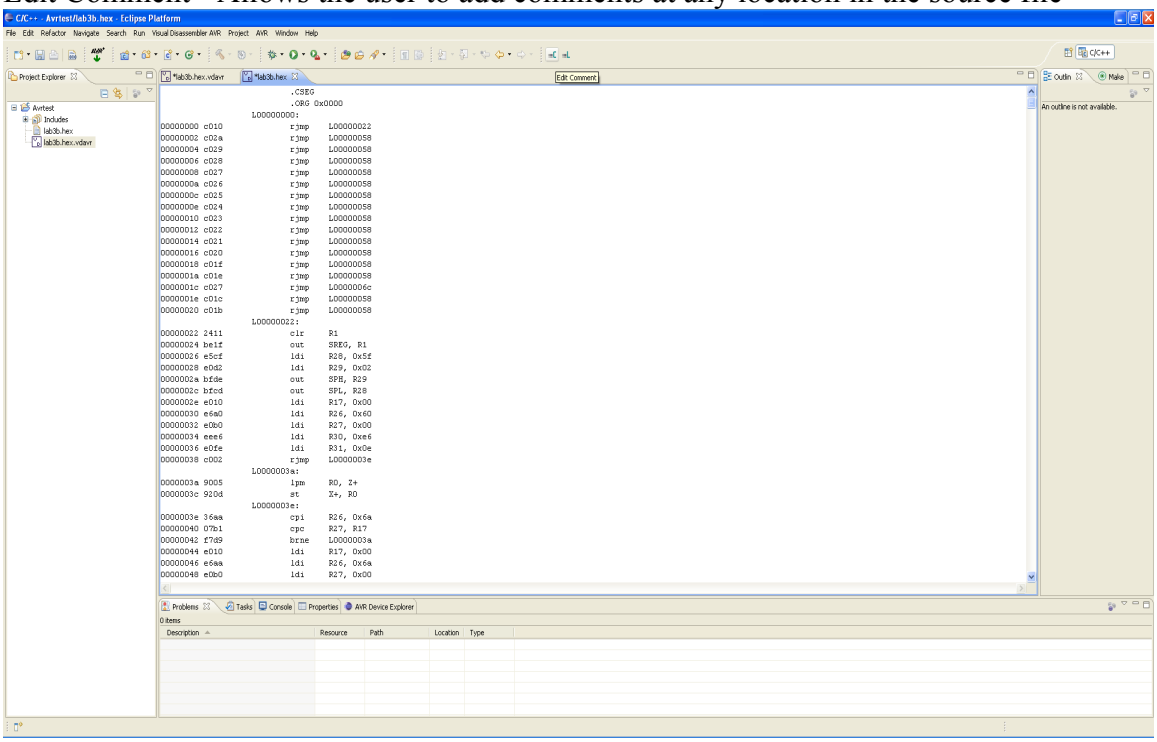


Figure 5

Menu Item – Edit Comment

Figures 6 and 7 show the Visual Disassembler AVR Edit Comment dialog box.
This dialog box allows entering a comment before or after any line in the source file

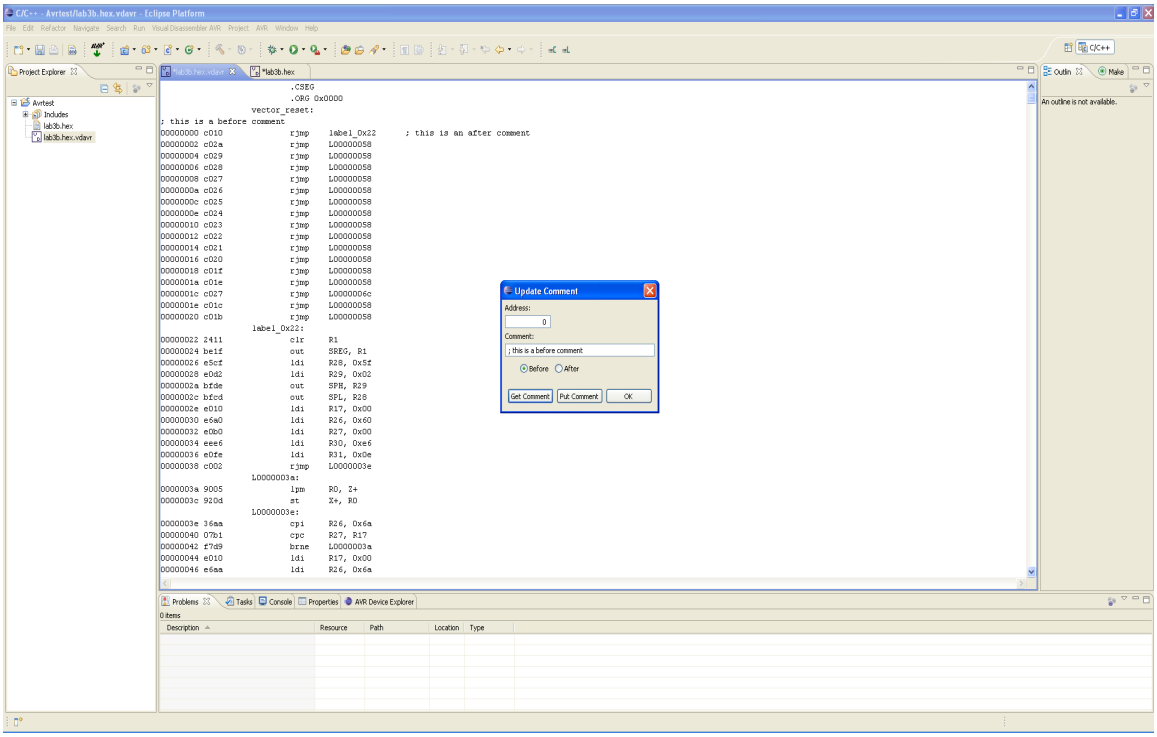


Figure 6

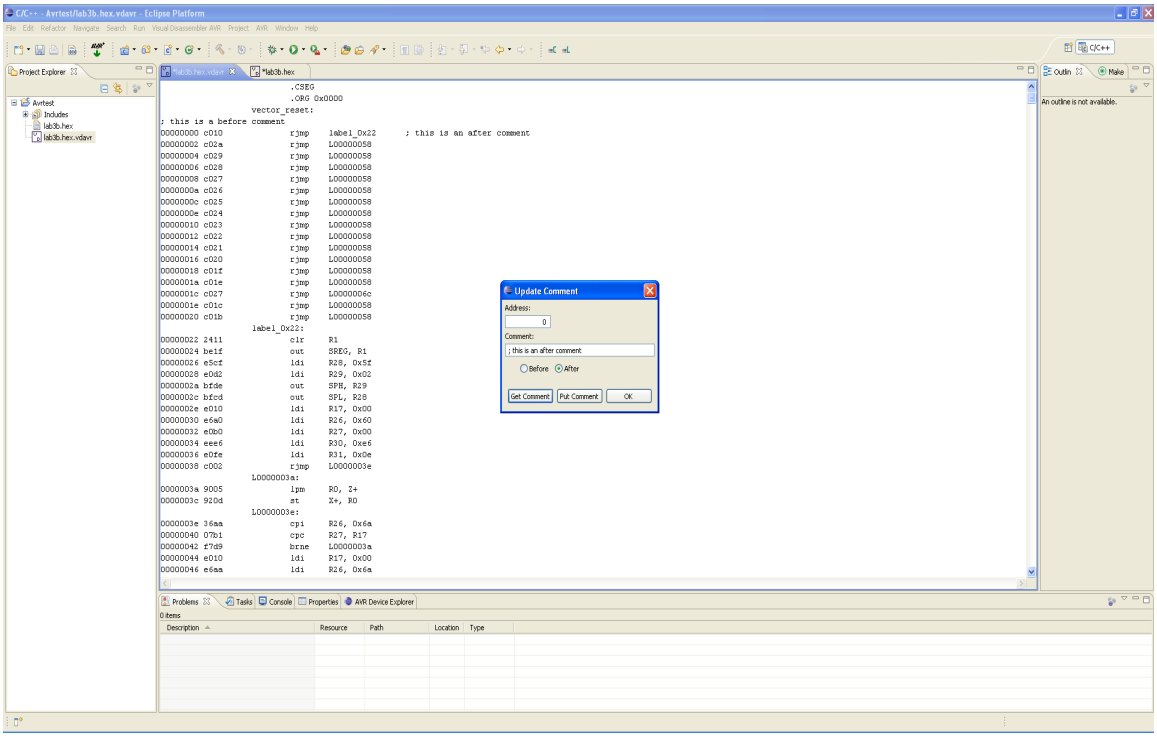

C/C++ - Avrtest/lab3b.hex.vdavr - Eclipse Platform
Project Explorer
Avrtest
Includes
lab3b.hex
lab3b.hex.vdavr
*lab3b.hex
.CSEG
.ORG 0x0000
vector_reset:
; this is a before comment
00000000 c010 rjmp label_0x22 ; this is an after comment
00000002 c02a rjmp L00000058
00000004 c029 rjmp L00000058
00000006 c028 rjmp L00000058
00000008 c027 rjmp L00000058
0000000a c026 rjmp L00000058
0000000c c025 rjmp L00000058
0000000e c024 rjmp L00000058
00000010 c023 rjmp L00000058
00000012 c022 rjmp L00000058
00000014 c021 rjmp L00000058
00000016 c020 rjmp L00000058
00000018 c01f rjmp L00000058
0000001a c01e rjmp L00000058
0000001c c027 rjmp L0000006c
0000001e c01c rjmp L00000058
00000020 c01b rjmp L00000058
label_0x22:
00000022 2411 clr R1
00000024 be1f out SREG, R1
00000026 e5cf ldi R28, 0x5f
00000028 e0d2 ldi R29, 0x02
0000002a bfde out SPH, R29
0000002c bfcd out SPL, R28
0000002e e010 ldi R17, 0x00
00000030 e6a0 ldi R26, 0x60
00000032 e0b0 ldi R27, 0x00
00000034 eee6 ldi R30, 0xe6
00000036 e0fe ldi R31, 0x0e
00000038 c002 rjmp L0000003e
L0000003a:
0000003a 9005 lpm R0, Z+
0000003c 920d st X+, R0
L0000003e:
0000003e 36aa cpi R26, 0x6a
00000040 07b1 cpc R27, R17
00000042 f7d9 brne L0000003a
00000044 e010 ldi R17, 0x00
00000046 e6aa ldi R26, 0x6a
Update Comment
Address:
0
Comment:
; this is an after comment
Before
After
Get Comment
Put Comment
OK
Outlin
Make
An outline is not available.
Problems
Tasks
Console
Properties
AVR Device Explorer
0 items
Description
Resource
Path
Location
Type

Figure 7

ToolBar Button View – Edit I/O / Program Label Tooltip

Figure 8 shows the Visual Disassembler AVR button item – Edit I/O / Program Label
This button duplicates the menu item Visual Disassembler AVR / Edit I/O / Program Label... menu item

Edit I/O / Program Label - Allows the user to add labels any program location
(These labels are resolved on branch and call instructions.)
Allows the user to add I/O port labels
(These labels are resolved in the in / out instructions.)

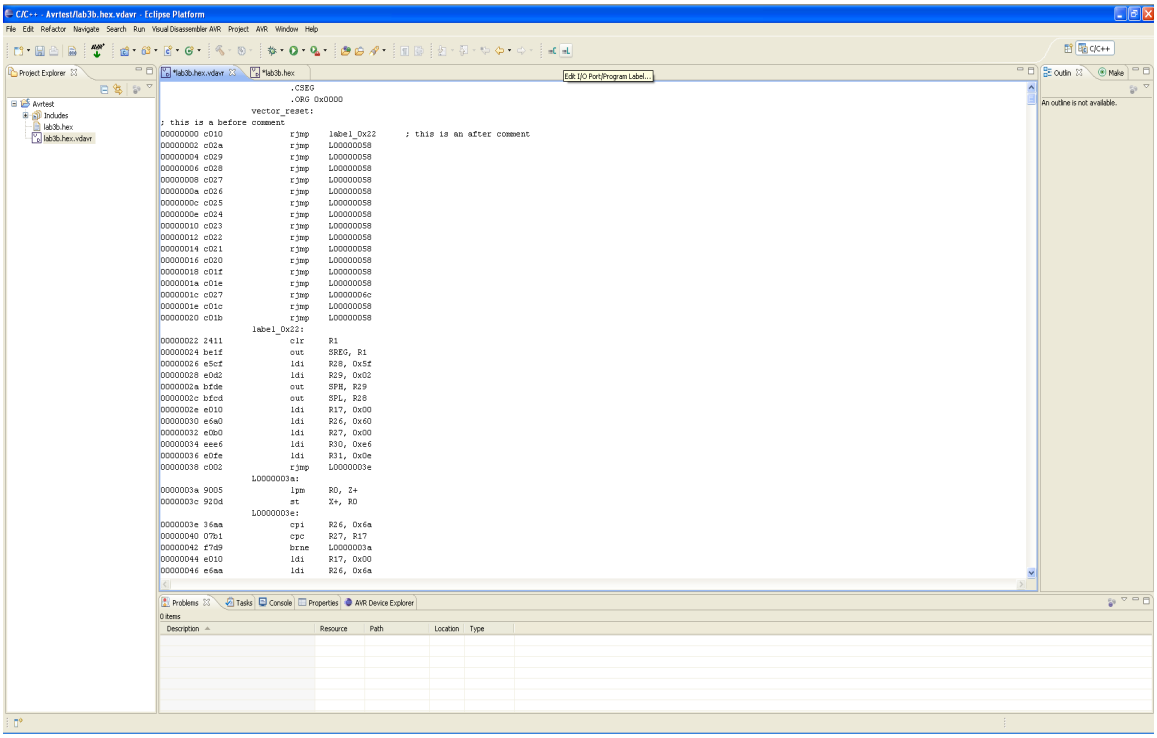


Figure 8

Dialog Box – Edit Program Label

Figures 9 and 10 display the Visual Disassembler AVR Edit I/O / Program Label dialog box.

Edit I/O / Program Label - Allows addition of an I/O Port or Program label at any line in the source file

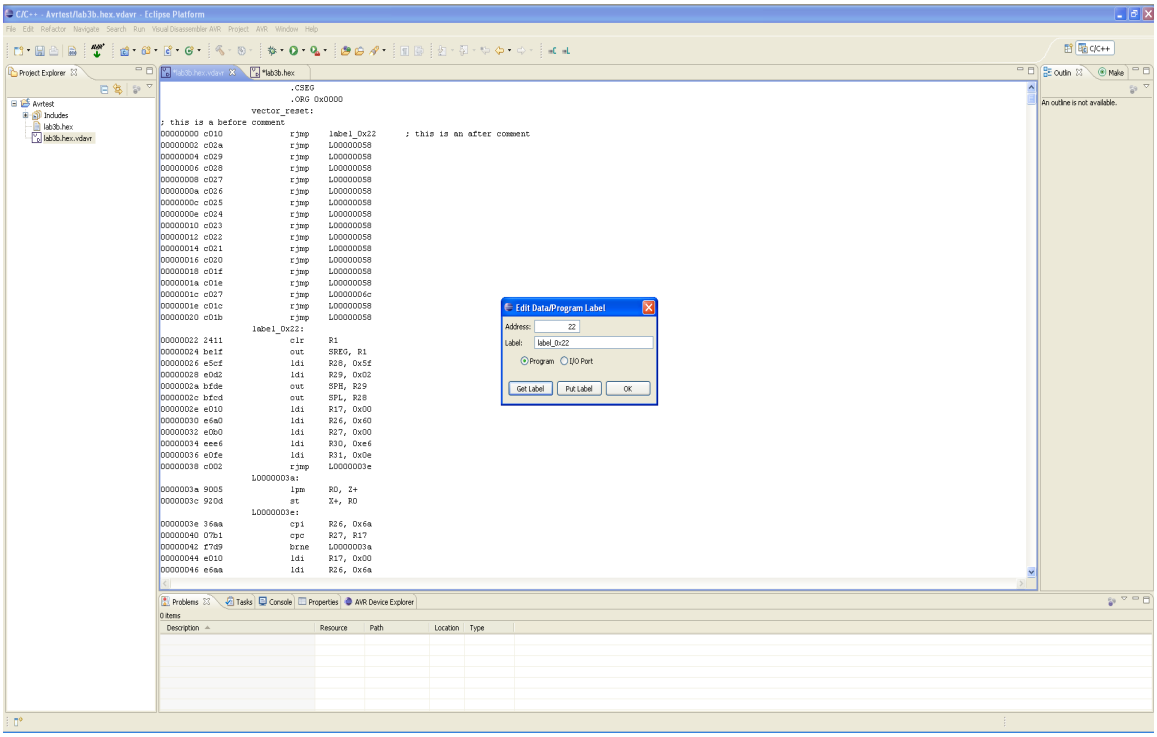


Figure 9

Dialog Box – Edit I/O Label

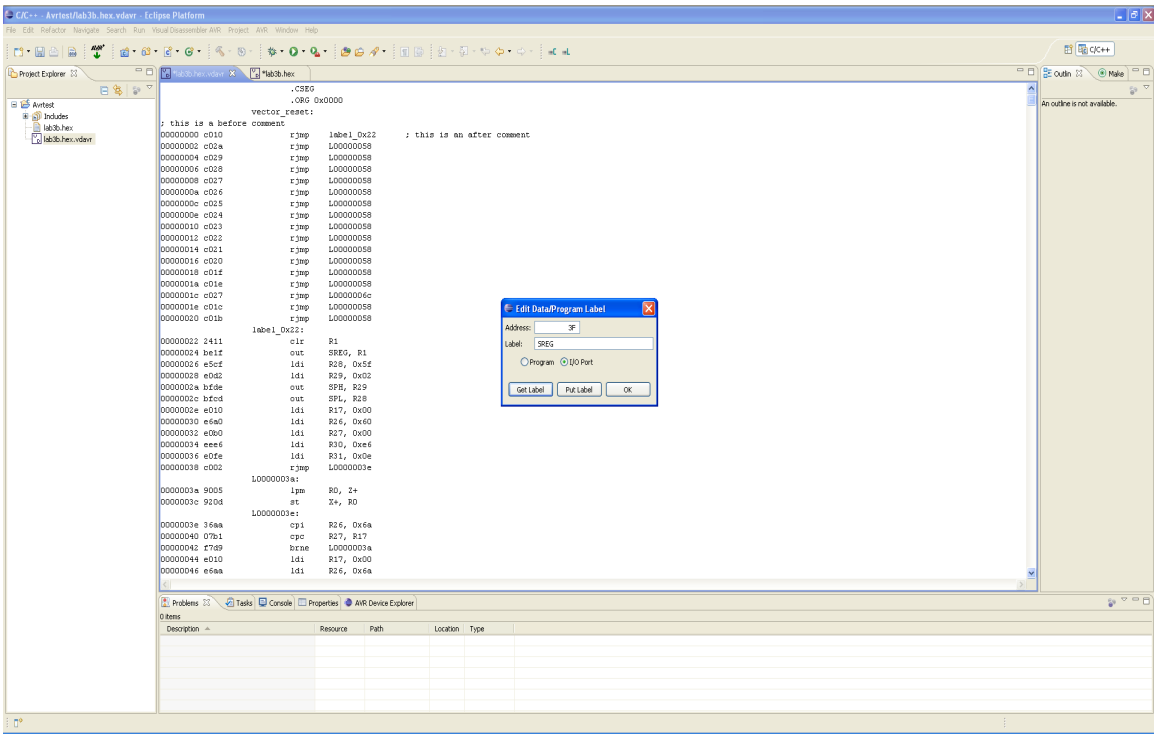
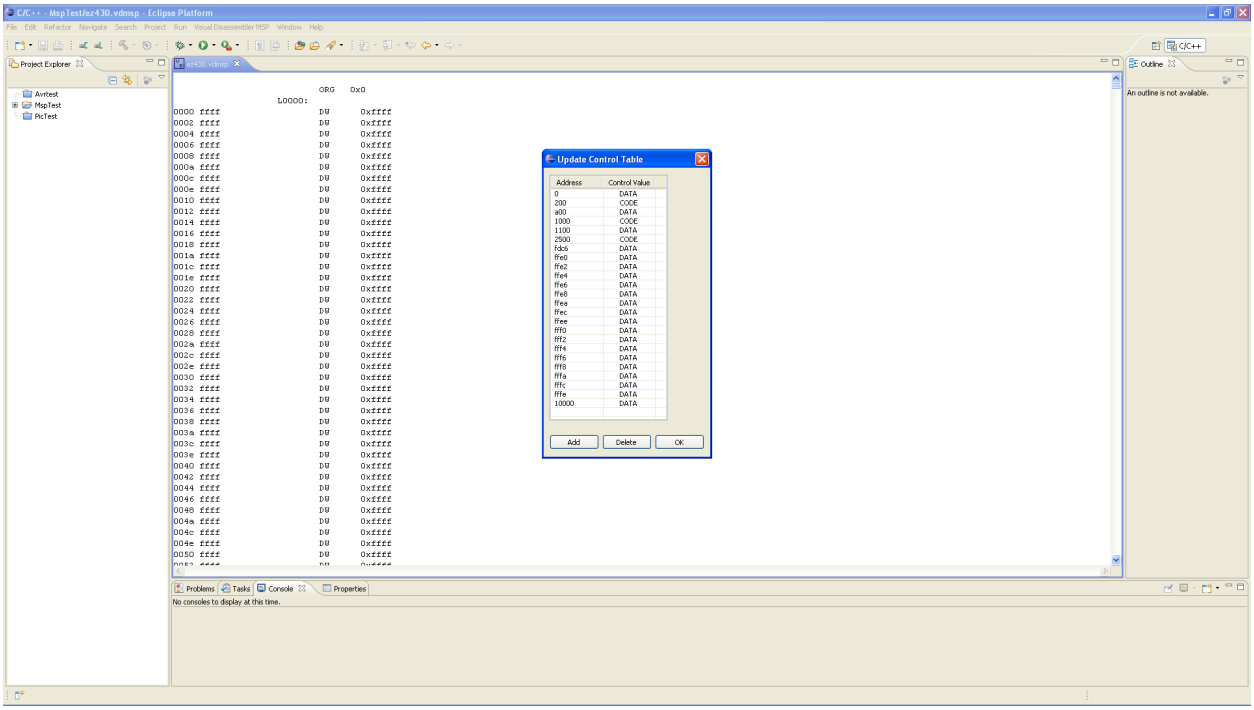


Figure 10

Menu Item – Edit Control Table...

This menu item allows the user to modify the memory space parameters. The user can select between DATA and CODE space. A memory region specified as DATA space forces the disassembler to disassemble all opcodes as defined word (DW). The disassembler will disassemble all MSP opcodes in regions defined as CODE.



Saving the Assembly Listing

Select all of the text in the Visual Disassembler text window (ctrl-A)
Copy all of the text in the Visual Disassembler text window (ctrl-C)
Open a text editor window
Paste the text from the Visual Disassembler text window into the text editor window.
Perform a Save-As and save the contents.
Editing the Assembly Listing may be required for the desired assembler.

Note: The address and opcode fields must be removed before it can be assembled.