R.A.I.D.

(R)untime (A)id for (I)nteractive (D)ebugging

A 6809 Debugger

Revised: 30-Jul-90

Dunfield Development Systems

----------------------------

High quality tools for

Embedded Development

at low prices.

http://www.dunfield.com

Copyright 1983-2005 Dave Dunfield

All rights Reserved

RAID

TABLE OF CONTENTS

Page

1. INTRODUCTION 1

2. USING RAID 2

2.1 RAID Commands 2

2.2 Breakpoints 4

3. WARNINGS 5

RAID Page: 1

1. INTRODUCTION

RAID is an interactive debugging facility for the motorola 6809

microprocessor. It contains display/alter memory/register facilities,

as well as a full 6809 disassembler, breakpoint, and software single

step functions. Download and transparent communication modes are also

included.

The format of the RAID command is as follows:

'RAID/[<device>] [<filename>]'

If <filename> is specified, RAID first loads that file, and sets

its saved program counter (PC) to its starting address. If no

<filename> is given, raid goes directly to it's 'RAID>' prompt, and

awaits your commands.

If the '/<device>' qualifier is given, RAID communicates with the

specified serial device (0-7). If not given, RAID assumes the current

console device. Note that once RAID has been invoked, it will

continue to communicate via the device that was console at the time

that it was started, even if the DOS console assignment is

subsequently changed.

RAID operates on each command character as soon as it is typed,

there is no need to press return following command entry. Note that

all hex values must be entered as two or four digits.

RAID Page: 2

2. USING RAID

2.1 RAID Commands

The following commands are implemented in RAID:

B <breakpoint#> <address>

Set breakpoint at specified address. Breakpoint is removed

if address is 0000. There can be up to eight breakpoints,

which are referenced by the numbers 0 to 7.

C <register> <value>

Changes 6809 registers values. Register is a single

character, which may be as follows:

A - Set A accumulator (8 bit value).

B - Set B accumulator (8 bit value).

C - Set condition code register (8 bit value).

D - Set direct page register (8 bit value).

X - Set X register (16 bit value).

Y - Set Y register (16 bit value).

U - Set user stack pointer (16 bit value).

S - Set system stack pointer (16 bit value).

P - Set program counter (16 bit value).

sp - (space) Set D accumulator (16 bit value).

D <start>,<end>

Disassembles memory, starting at indicated address. If

space is entered for <end> address, assumes FFFF.

Disassembler output contains address, opcodes bytes, ASCII

equivalent of opcode bytes, instruction neumonic, and

operands to instruction. Disassembler output can be

aborted by pressing the <ESC> key. The line feed key may

be used to suspend output to the screen, after which each

time line feed is pressed, one additional line will be

output. Pressing the carriage return key will resume

normal output.

E <address>

Edit's memory, Address and contents are displayed, Enter

two hex digits to change value, or a single quote followed

by a character. Entering a space skips to the next

location, Backspace backups to the previous location.

Carriage return terminates the edit command.

G <address>

Begins execution at the indicated address. If a space is

entered instead on an address, begins execution at the

address in the saved 6809 program counter.

RAID Page: 3

M <start>,<end>

Displays memory, in HEX/ASCII dump format, starting at the

indicated address. If a space is entered for <end>

address, assumes FFFF. The output of the memory command

can be aborted by pressing the <ESC> key, line feed and

carriage return may be used to suspend output.

R

Displays the values of the 6809 registers.

S <address>

Begins single-stepping at the indicated address. If a

space is entered for the address, then single-stepping

begins at the address in the saved 6809 program counter.

Disassembled instruction is displayed on the terminal, and

RAID waits for a key to be pressed. If a space is entered,

steps to the next instruction. Carriage return toggles the

automatic display of registers. '?' displays the register

values, and escape terminates the single-step command.

V

Displays the current breakpoint settings.

RAID Page: 4

2.2 Breakpoints

RAID allows breakpoints to be placed in the program under test,

which when encountered during the execution of that program, cause

the program to interrupt, and RAID is entered. Up to 8 such

breakpoints (numbered from 0 to 7) may be placed in the program at

any one time.

When entered from a breakpoint, RAID will first display a

message indicating the number of the breakpoint which was

encountered, and then enter command mode.

Once RAID has been entered from a breakpoint, the registers of

the program under test are saved. Note that the address of the

breakpoint is in the saved 6809 program counter, and therefore

using a space as the operand to a 'G' or 'S' command will allow

execution or single stepping to continue from the breakpoint.

Breakpoints are entirely transparent, and will remain in the

program until explicitly removed.

RAID Page: 5

3. WARNINGS

When executing breakpointed instruction, or when single stepping,

RAID copies the instruction into it's own RAM locations before

executing it. Because PC relative instruction depend on the actual

6809 program counter being at their address in the code, PC relative

instructions will not do proper addressing if they are breakpointed,

or single stepped.

When RAID is invoked it attaches itself to the operating system in

such a way that it becomes an essential part of the operating system.

If for any reason RAID is terminated abnormally, it is advisable to

reboot the operating system.