

MoviEclipse features

with a focus on debugging



About

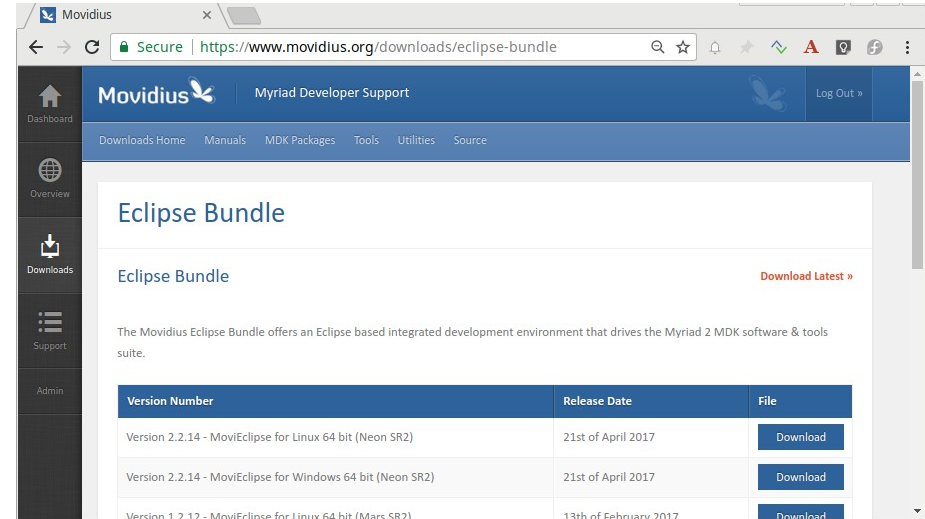
Based on Eclipse Neon SR3. Includes CDT, LinuxTools and other Eclipse frameworks



Distribution

MoviEclipse is distributed as two archives, one for Linux (64 bit), one for Windows (64 bit) which also includes Java Virtual Machine.

Also distributed as an update site. For new version you will not need to download the entire bundle, just use the integrated update mechanism.



The screenshot shows the Movidius Eclipse Bundle download page. The page title is "Eclipse Bundle". Below the title, there is a "Download Latest" link. A description states: "The Movidius Eclipse Bundle offers an Eclipse based integrated development environment that drives the Myriad 2 MDK software & tools suite." Below this, there is a table with three columns: "Version Number", "Release Date", and "File".

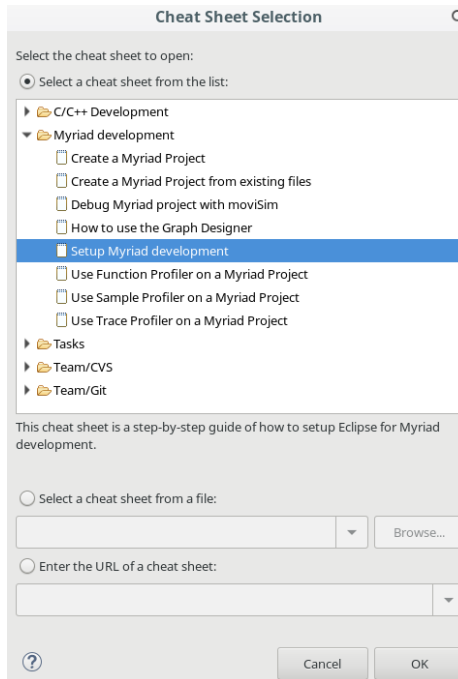
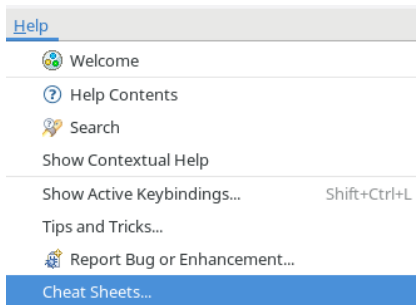
Version Number	Release Date	File
Version 2.2.14 - MoviEclipse for Linux 64 bit (Neon SR2)	21st of April 2017	Download
Version 2.2.14 - MoviEclipse for Windows 64 bit (Neon SR2)	21st of April 2017	Download
Version 1.2.12 - MoviEclipse for Linux 64 bit (Mars SR2)	13th of February 2017	Download

URL: <https://www.movidius.org/downloads/eclipse-bundle>

Cheat sheets

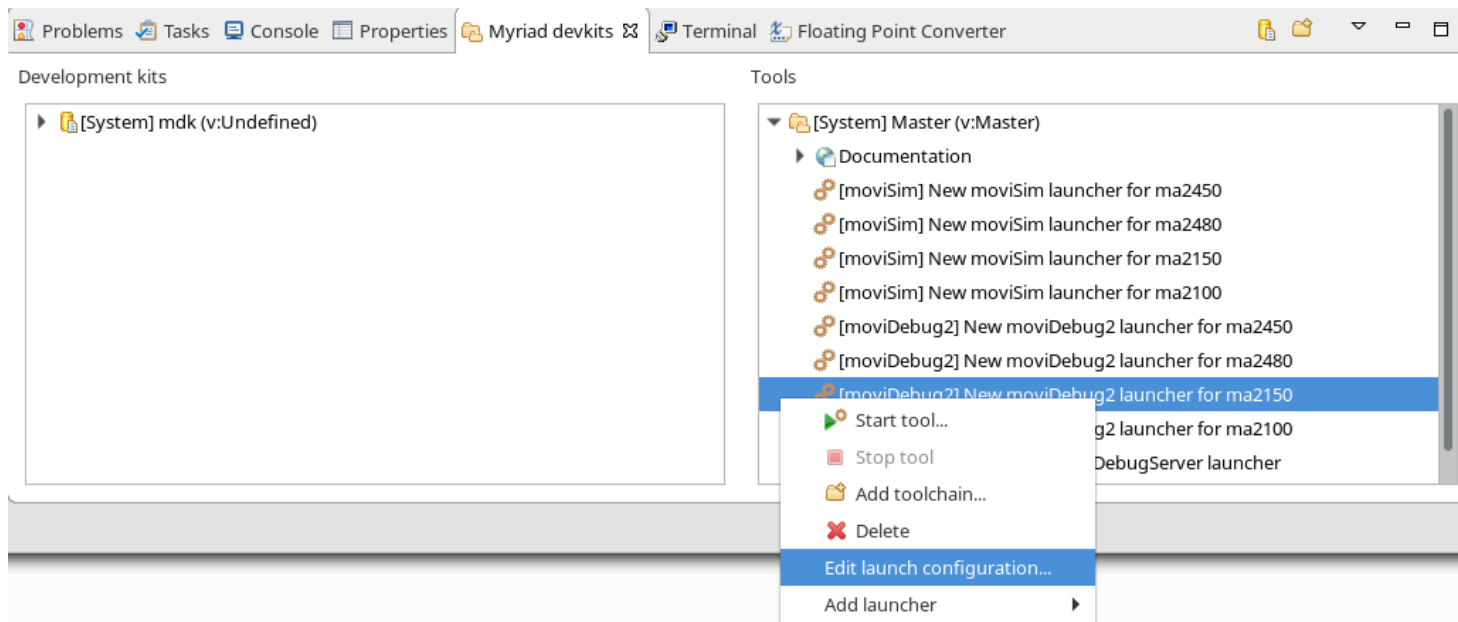
Step-by-step guidance to help user to complete myriad specific tasks.

Accessible from Help menu:

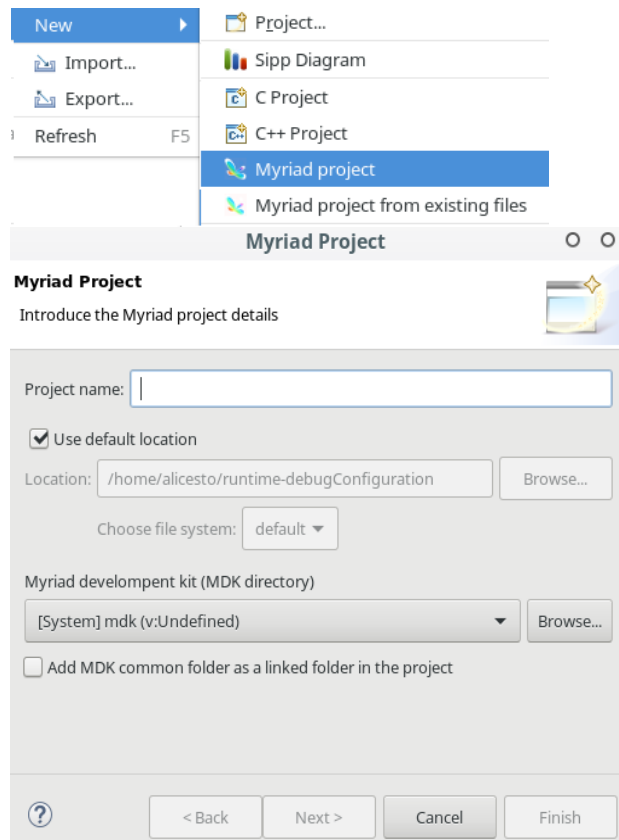


Devkit view

Central point to manage MDK's and toolchains:



Dedicated project wizards



Myriad Project

Introduce the Myriad project details

Project name:

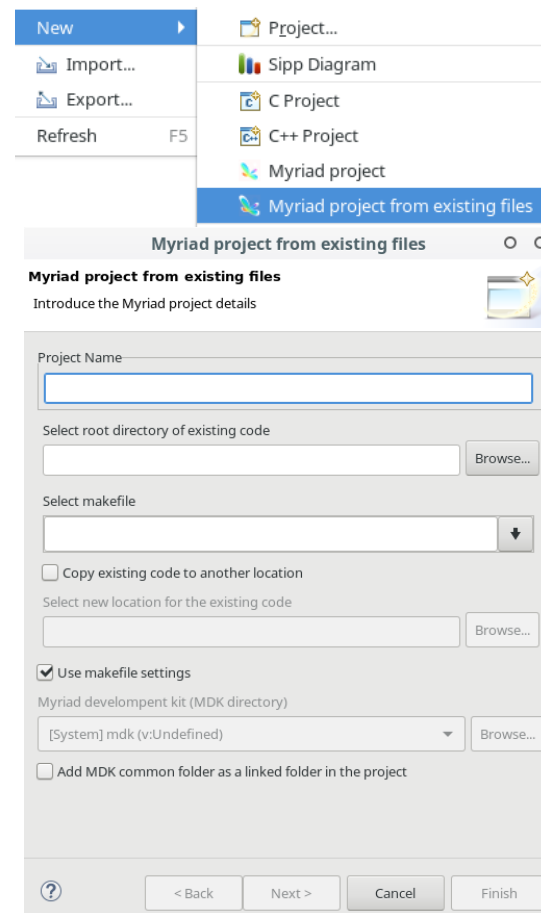
☒ Use default location

Location:

Choose file system:

Myriad developpment kit (MDK directory)

☐ Add MDK common folder as a linked folder in the project



Myriad project from existing files

Introduce the Myriad project details

Project Name:

Select root directory of existing code

Select makefile

☐ Copy existing code to another location

Select new location for the existing code

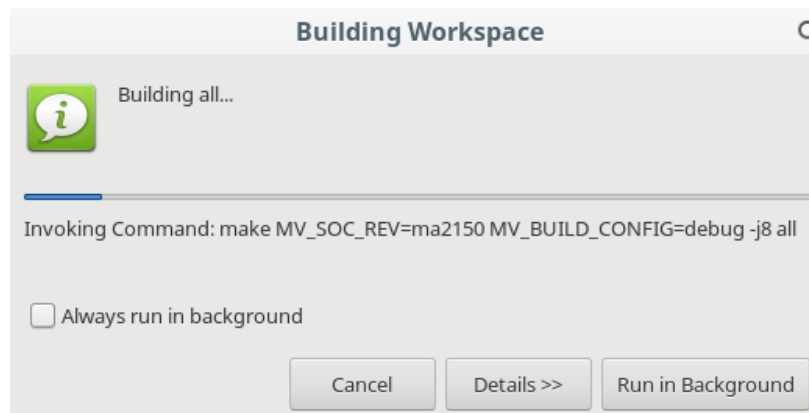
☒ Use makefile settings

Myriad developpment kit (MDK directory)

☐ Add MDK common folder as a linked folder in the project

Building from Eclipse

When a project is build or cleaned from Eclipse “make all” or “make clean” are executed. The same build mechanism that is used from command line is used also from Eclipse



Editing code from Eclipse

CDT parse the make output console and build the project symbols index. This information are then used to provide code-sensing features:

```

126 {
127
128 // Configure the system
129 OsDrvCprInit();
130 OsDrvCprOpen();
131 OsDrvCprAuxClockArrayConfig(auxClk);
132
133 /// Configure the auxiliary clocks using a null terminated array of
134 /// Auxiliary clock configurations
135 ///
136 /// Normally this is handled by DrvCprConfigureSystemClk for most users
137 /// @param[in] pAuxClkCfg[] - null terminated array of tyAuxClkDividerCfg
138 /// @param[out] 0 on success, non-zero otherwise
139 /// @return OS_MYR_DRV_SUCCESS on access to the resource, non-zero otherwise
140 int OsDrvCprAuxClockArrayConfig(const tyAuxClkDividerCfg pAuxClkCfg[]);

```

Problems

Impulse

Press 'F2' for focus

```

126 {
127
128 // Configure the system
129 OsDrvCprInit();
130 OsDrvCprOpen();
131 OsDrvCprAuxClockArrayConfig(auxClk);
132 OsDr
133
134 OsDr

```

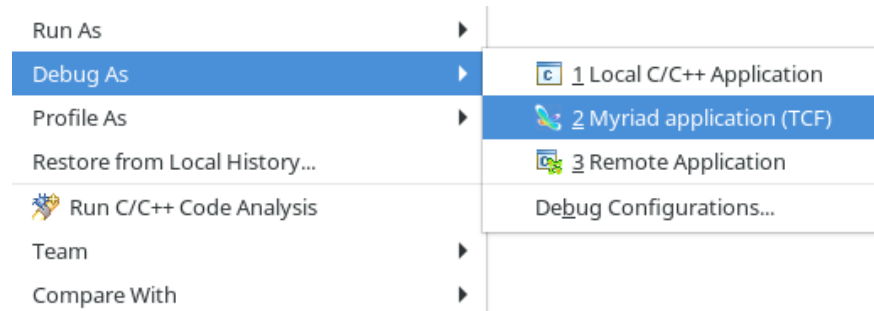
- OsDrvCommonInterruptSendEvent(rtems_id id, rtems_event_set e
- OsDrvCprAuxClockArrayConfig(const tyAuxClkDividerCfg * pAuxC
- OsDrvCprAuxClockEnable(u32 auxClksMask, tyCprClockSrc clkSou
- OsDrvCprConfigureSystemClock(u32 plltargetfreqkhz, u32 clockdi
- OsDrvCprGetClockFreqKhZ(tyClockType clockType, tyClockConfig
- OsDrvCprGetCprVersionId(u32 * cprVersionId) : int
- OsDrvCprGetSysClockKhZ(u32 * result) : int
- OsDrvCprGetSysClockPerUs(u32 * result) : int
- OsDrvCprInit(void) : int
- OsDrvCprOpen(void) : int

Problems

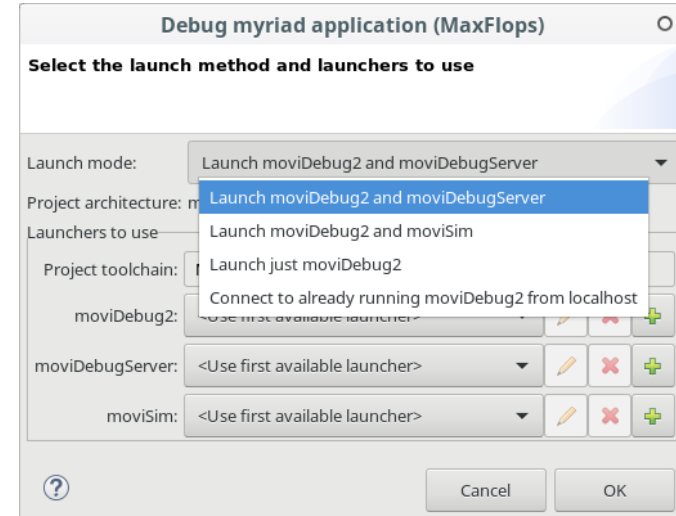
Press 'Ctrl+Space' to show Template Proposals

Debug with moviDebug2

Dedicated launcher, just right click on a project and select “Debug As / Myriad application (TCF)”

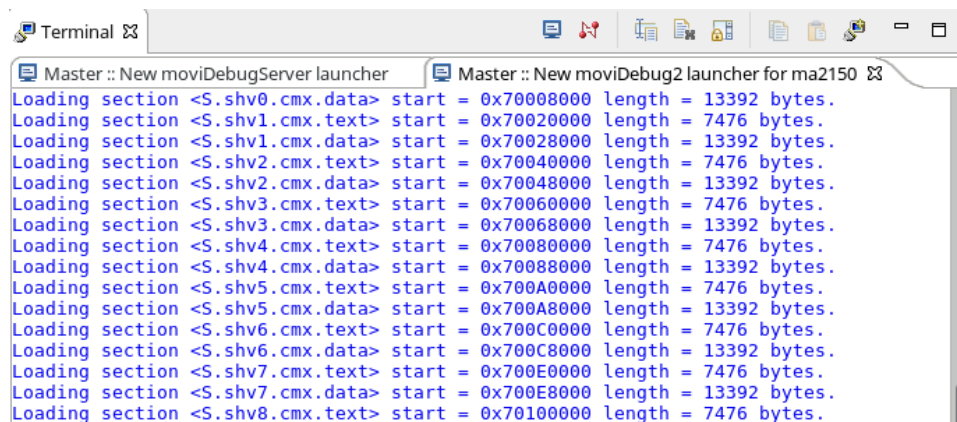


Multiple scenarios supported



Debug with moviDebug2

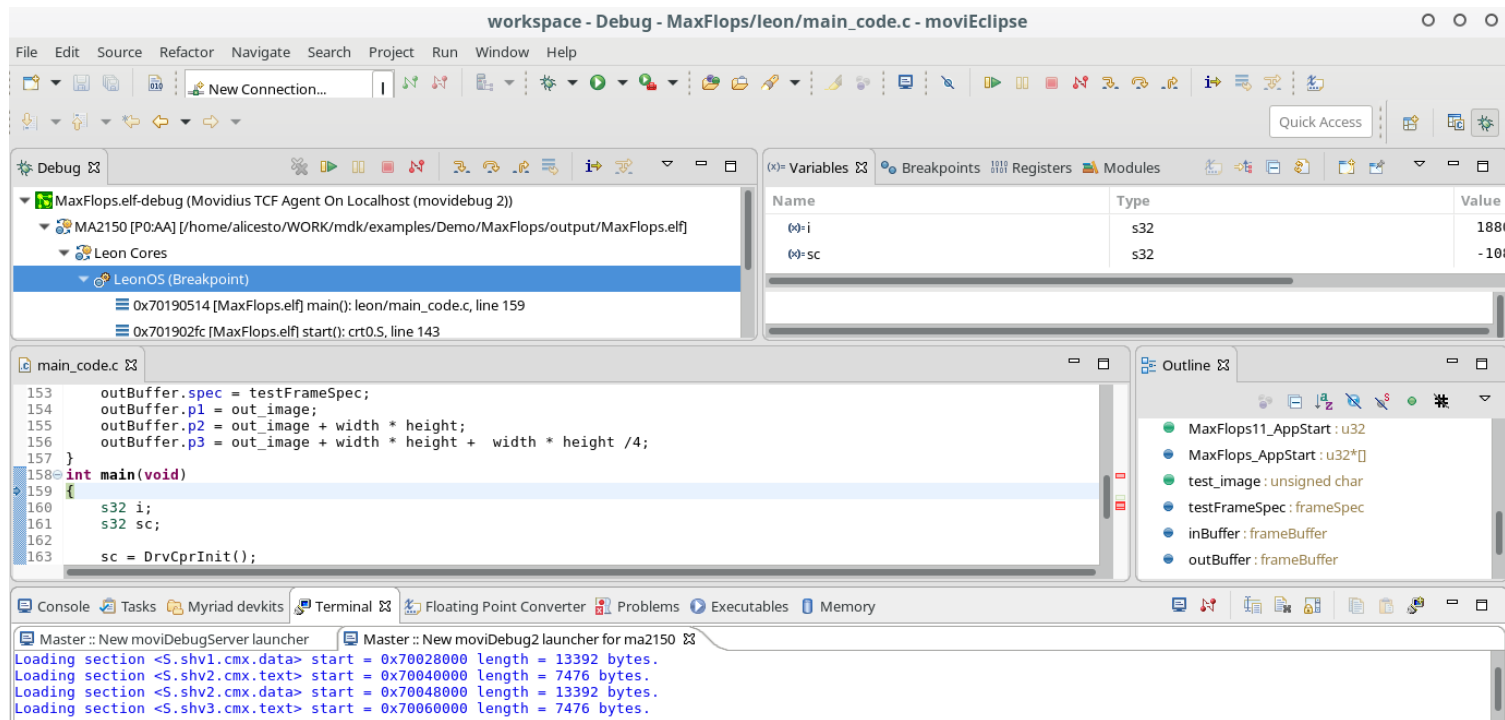
moviDebug2, moviSim and moviDebugServer when executed from moviEclipse have an interactive ANSI enabled terminal



```
Master :: New moviDebugServer launcher
Master :: New moviDebug2 launcher for ma2150
Loading section <S.shv0.cmx.data> start = 0x70008000 length = 13392 bytes.
Loading section <S.shv1.cmx.text> start = 0x70020000 length = 7476 bytes.
Loading section <S.shv1.cmx.data> start = 0x70028000 length = 13392 bytes.
Loading section <S.shv2.cmx.text> start = 0x70040000 length = 7476 bytes.
Loading section <S.shv2.cmx.data> start = 0x70048000 length = 13392 bytes.
Loading section <S.shv3.cmx.text> start = 0x70060000 length = 7476 bytes.
Loading section <S.shv3.cmx.data> start = 0x70068000 length = 13392 bytes.
Loading section <S.shv4.cmx.text> start = 0x70080000 length = 7476 bytes.
Loading section <S.shv4.cmx.data> start = 0x70088000 length = 13392 bytes.
Loading section <S.shv5.cmx.text> start = 0x700A0000 length = 7476 bytes.
Loading section <S.shv5.cmx.data> start = 0x700A8000 length = 13392 bytes.
Loading section <S.shv6.cmx.text> start = 0x700C0000 length = 7476 bytes.
Loading section <S.shv6.cmx.data> start = 0x700C8000 length = 13392 bytes.
Loading section <S.shv7.cmx.text> start = 0x700E0000 length = 7476 bytes.
Loading section <S.shv7.cmx.data> start = 0x700E8000 length = 13392 bytes.
Loading section <S.shv8.cmx.text> start = 0x70100000 length = 7476 bytes.
```

Debug with moviDebug2

Dedicated perspective for debugging:



The screenshot displays the Eclipse IDE in the 'workspace - Debug - MaxFlops/leon/main_code.c - moviEclipse' perspective. The interface is divided into several panels:

- Debug Console:** Shows the connection to 'MaxFlops.elf-debug (Movidius TCF Agent On Localhost (movidebug2))' and the selected core 'LeonOS (Breakpoint)'. It lists two breakpoints: '0x70190514 [MaxFlops.elf] main(): leon/main_code.c, line 159' and '0x701902fc [MaxFlops.elf] start(): crt0.S, line 143'.
- Variables:** A table showing the current state of variables:

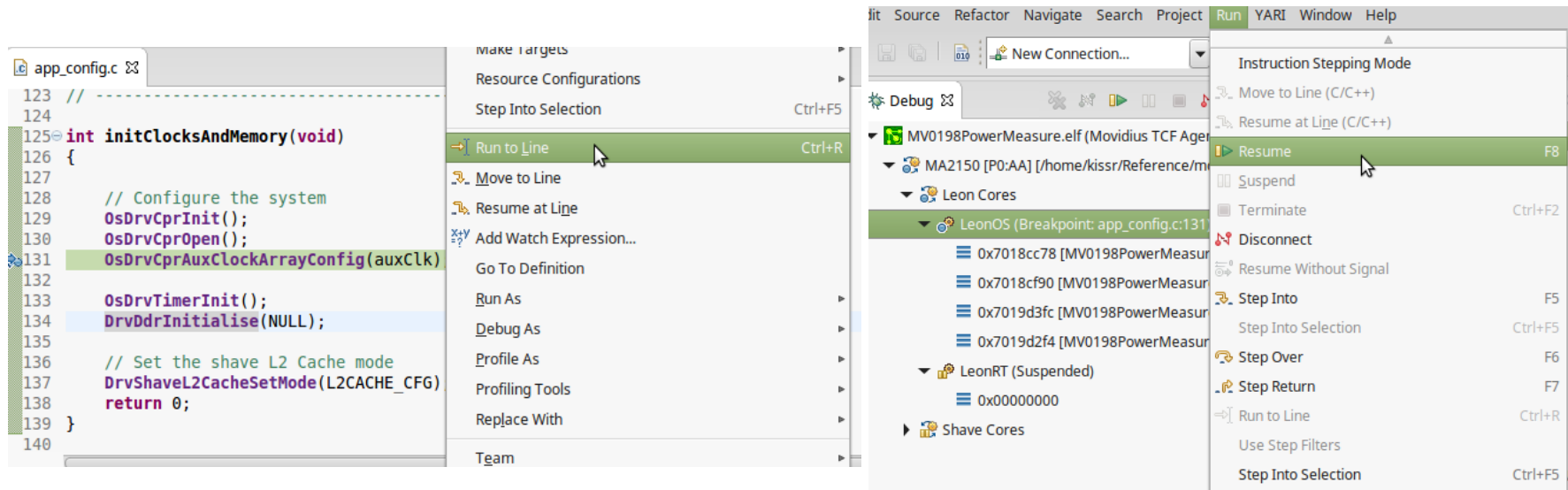
Name	Type	Value
(*) i	s32	1886
(*) sc	s32	-106
- Breakpoints, Registers, Modules:** Panels for managing breakpoints, viewing registers, and listing loaded modules.
- Outline:** A list of symbols in the loaded image:
 - MaxFlops11_AppStart : u32
 - MaxFlops_AppStart : u32*
 - test_image : unsigned char
 - testFrameSpec : frameSpec
 - inBuffer : frameBuffer
 - outBuffer : frameBuffer
- Console:** Displays the output of the 'New moviDebug2 launcher for ma2150', showing the loading of sections from the .shv1.cmx.data and .shv2.cmx.text segments.
- Main Editor:** Shows the source code of 'main_code.c' with the following content:


```

153     outBuffer.spec = testFrameSpec;
154     outBuffer.p1 = out_image;
155     outBuffer.p2 = out_image + width * height;
156     outBuffer.p3 = out_image + width * height + width * height / 4;
157 }
158 int main(void)
159 {
160     s32 i;
161     s32 sc;
162
163     sc = DrvCprInit();
      
```

Debug with moviDebug2

Many operations possible from shortcuts or from context menu for all cores and code types: LOS/LRT/SHAVE or C/CPP/ASM





Debug with moviDebug2


Various views available:


- Debug view
- Local variables
- Breakpoints
- Disassembly view
- Registers view
- Memory view


Debug view


 Debug



 MV0198PowerMeasure.elf (Movidius TCF Agent On Localhost (movidebug 2))



 MA2150 [P0:AA] [/home/kissr/Reference/mdk/examples/HowTo/MV0198PowerMeasure/output/MV0198PowerMeasure]



 Leon Cores



 LeonOS (Breakpoint: app_config.c:131)



 0x7018cc78 [MV0198PowerMeasure.elf] initClocksAndMemory(): leon/app_config.c, line 131



 0x7018cf90 [MV0198PowerMeasure.elf] POSIX_Init(): leon/main.c, line 60



 0x7019d3fc [MV0198PowerMeasure.elf] _Thread_Handler(): threadhandler.c, line 199



 0x7019d2f4 [MV0198PowerMeasure.elf] _Thread_Handler(): threadhandler.c, line 96



 LeonRT (Suspended)


 0x00000000


 Shave Cores


 Shave 0 (Suspended)


 Shave 1 (Suspended)


 Shave 2 (Suspended)

Variables view

Dedicated view or inline

```

102 // get first 8 lines in the internal cmx buffer
103 for(idx = 0; idx < kernel_size; idx++)
104 {
105     inAddr = inFrBuff + (clamp(crtInLn, 0, height) * width);
106     // ...
107     // ...
108     // ...
109     // ...
110     // ...
111     // ...
112     // ...
113     // ...
114     if(k
115     else
116     // ...
117     // ...
118     // ...
119     //ro
120     rota
121     // ...
122     // ...
123     // ...

```

Name	Type	Value
inAddr	u8 *	"TUVVUTUVWSJGDEC8124*#!\034\
0x *	u8	'T'

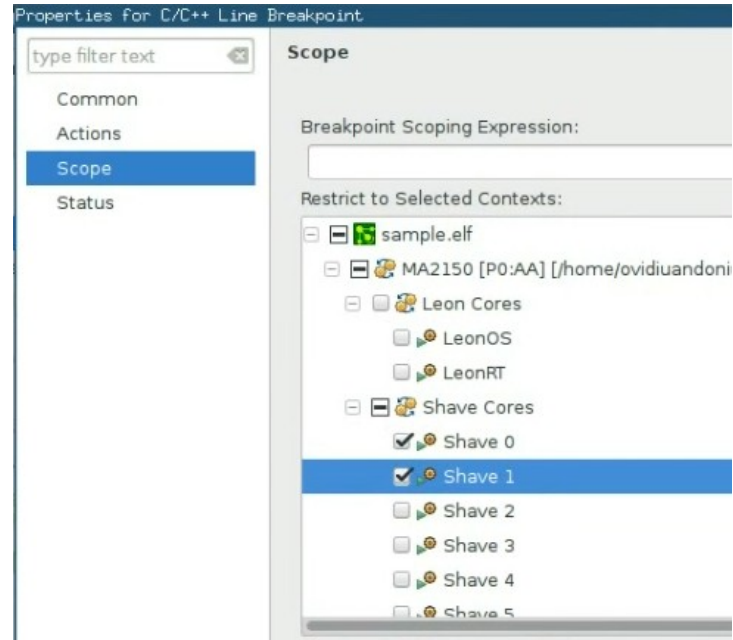
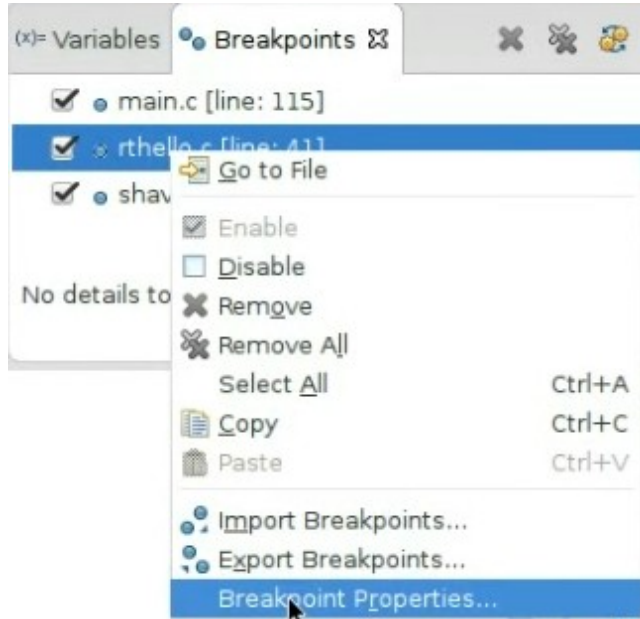
"TUVVUTUVWSJGDEC8124*#!\034\033\033\035 ""*#666),,*)*"+167543/101,-.-021010.-,*(''
Hex: 80000000, **Dec:** 2147483648, **Oct:** 020000000000, **At:** lrt_inputFrame
Bin: 1000,0000,0000,0000,0000,0000,0000,0000
Size: 4 bytes, **Type:** u8 *
Address: 0x700c8008

(*)= Variables
 Breakpoints
 Expressions
 Registers
 Modules

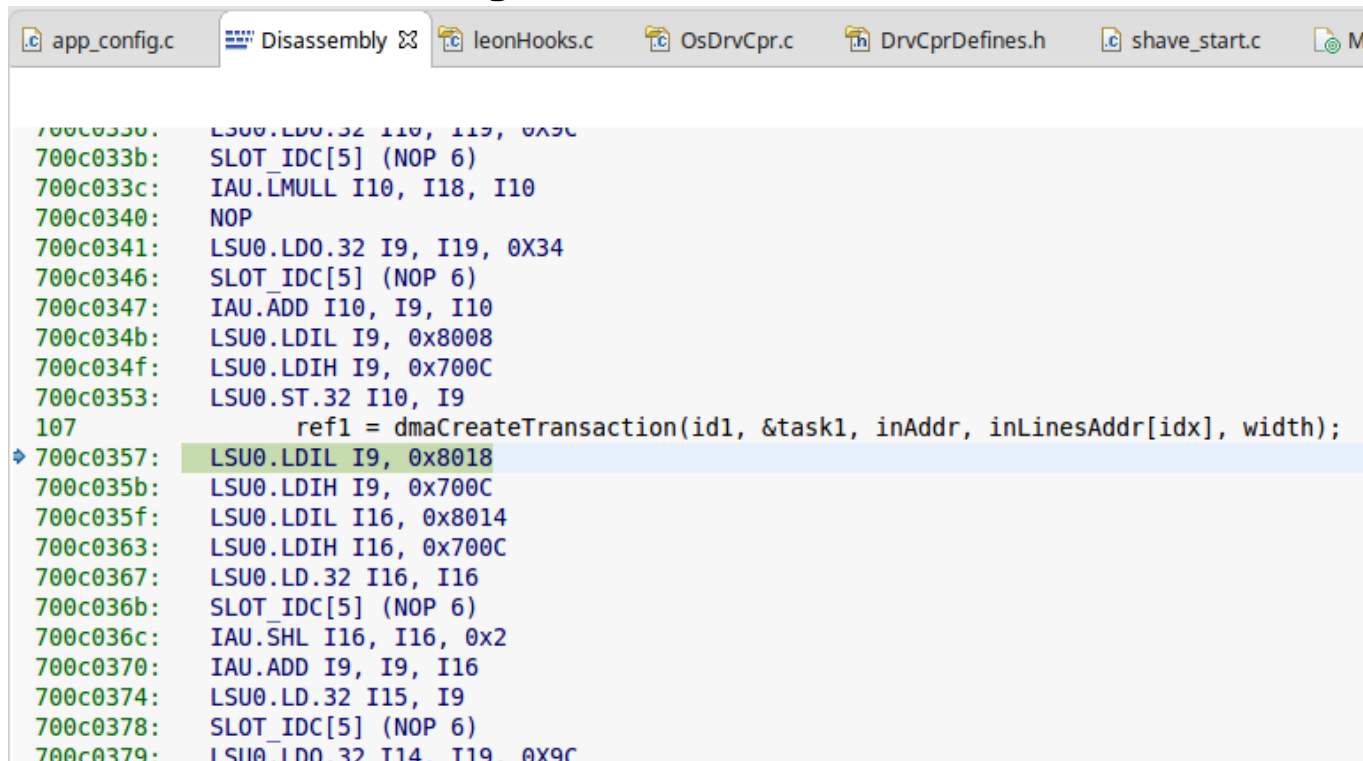
Name	Type	Value
(*)= ReqId	dmaRequesterId	26
NewTransaction	dmaTransactionList *	{linkAddress=0x00000000, cfgLink={cfgBit
linkAddress	void *	0x00000000
cfgLink	<Structure>	{cfgBits={type=0x00000000, priority=0x00
cfgBits	configBits	{type=0x00000000, priority=0x00000002, t
type	u32	0x00000000
priority	u32	0x00000002
brstLength	u32	0x0000000f
id	u32	0x00000000

0x00000000
Hex: 00000000, **Dec:** 0, **Oct:** 0
Bin: 0000,0000,0000,0000,0000,0000,0000,0000
Size: 4 bytes, **Type:** u32
Pieces: Address: 0x781804f4, Bit Offset: 0, Bit Count: 2

Breakpoints view



Disassembly view



```
app_config.c | Disassembly | leonHooks.c | OsDrvCpr.c | DrvCprDefines.h | shave_start.c | M
700c0330: LSU0.LD0.32 I10, I19, 0x9C
700c033b: SLOT_IDC[5] (NOP 6)
700c033c: IAU.LMULL I10, I18, I10
700c0340: NOP
700c0341: LSU0.LD0.32 I9, I19, 0x34
700c0346: SLOT_IDC[5] (NOP 6)
700c0347: IAU.ADD I10, I9, I10
700c034b: LSU0.LDIL I9, 0x8008
700c034f: LSU0.LDIH I9, 0x700C
700c0353: LSU0.ST.32 I10, I9
107      ref1 = dmaCreateTransaction(id1, &task1, inAddr, inLinesAddr[idx], width);
➔ 700c0357: LSU0.LDIL I9, 0x8018
700c035b: LSU0.LDIH I9, 0x700C
700c035f: LSU0.LDIL I16, 0x8014
700c0363: LSU0.LDIH I16, 0x700C
700c0367: LSU0.LD.32 I16, I16
700c036b: SLOT_IDC[5] (NOP 6)
700c036c: IAU.SHL I16, I16, 0x2
700c0370: IAU.ADD I9, I9, I16
700c0374: LSU0.LD.32 I15, I9
700c0378: SLOT_IDC[5] (NOP 6)
700c0379: LSU0.LD.32 I14, I19, 0x9C
```

Registers view

Leon Registers

Shave Registers

(x)= Variables Breakpoints Expressions Registers Modules			
Name	Hex	Decimal	Description
PC	701cb170	1880928624	
NPC	701cb174	1880928628	
g0	00000000	0	
g1	00000000	0	
g2	20f00000	552599552	
g3	00000030	48	
g4	00000003	3	
g5	00000080	128	
g6	00000000	0	

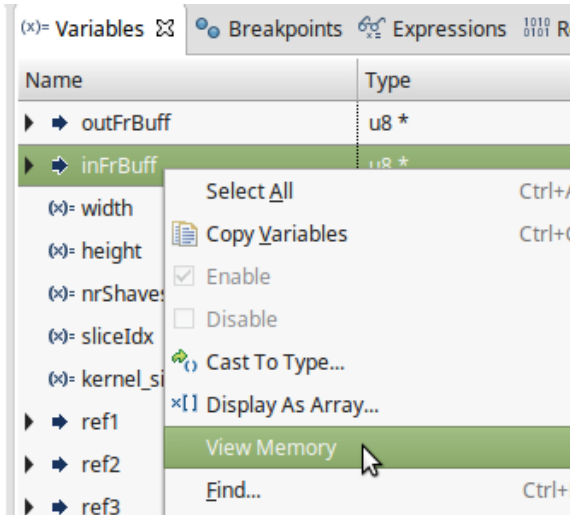
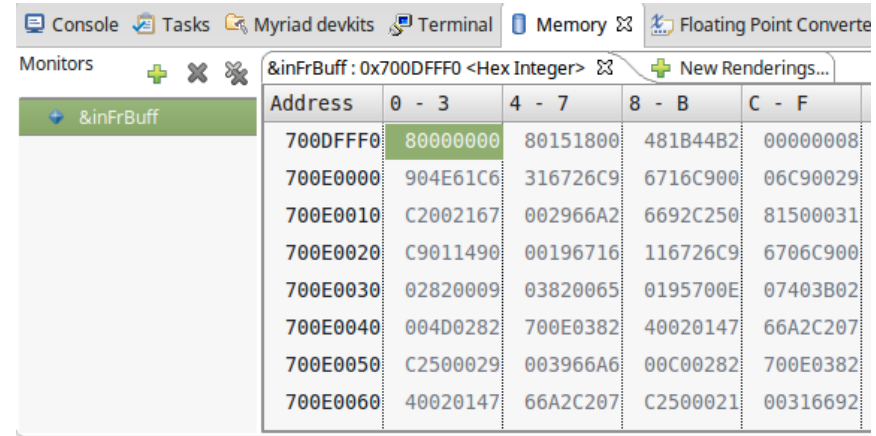
Hex: 701cb170, **Dec:** 1880928624, **Oct:** 016007130560, **At:** DrvSvuSwtHalted() + 0x3c
Bin: 0111,0000,0001,1100,1011,0001,0111,0000
Size: 4 bytes, readable, writable

(x)= Variables Breakpoints Expressions Registers Modules		
Name	Hex	Decimal
IP	700c0357	1879835479
INEXT	700c035b	1879835483
i0	700c00c0	1879834816
i1	00080004	524292
i2	00084400	541696
i3	00000208	520
i4	fffffffe	4294967294
i5	00000001	1
i6	78800000	2021654528
i7	78802120	2021663008
i8	78800000	2021654528
i9	700c8008	1879867400

Hex: 700c00c0, **Dec:** 1879834816, **Oct:** 016003000300
Bin: 0111,0000,0000,1100,0000,0000,1100,0000
Size: 4 bytes, readable, writable

Memory View

Can be triggered also from variables view

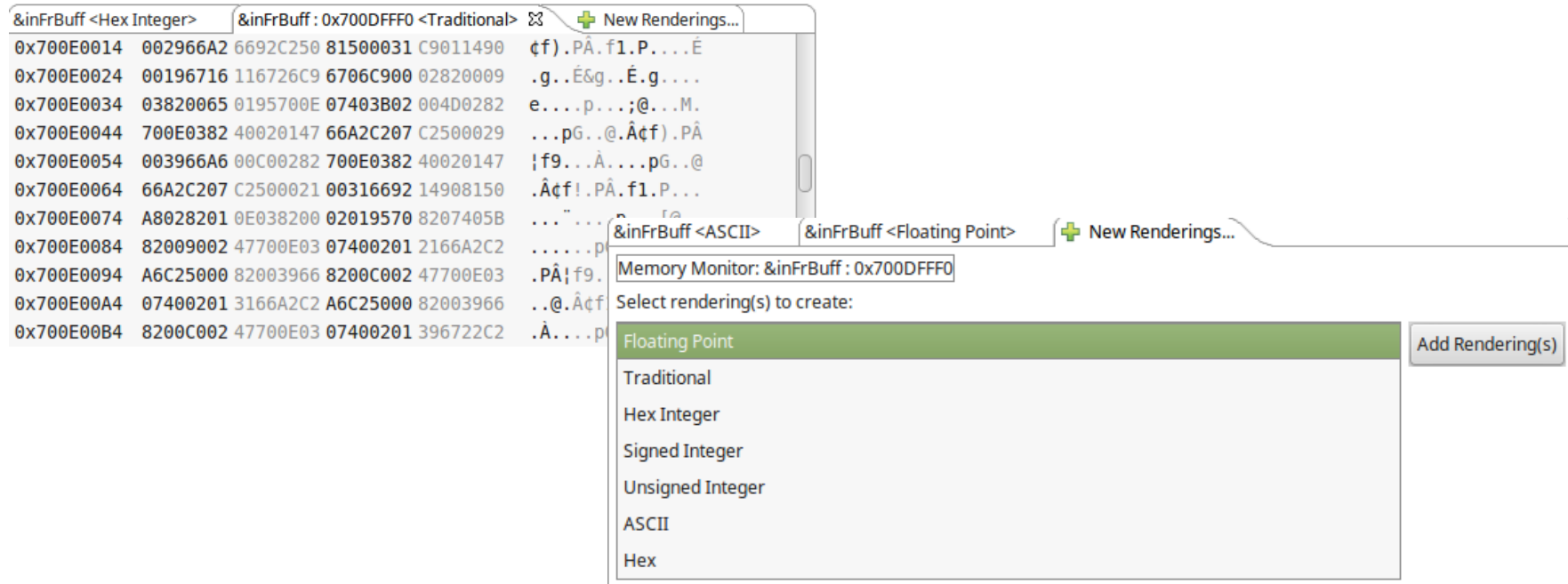



The screenshot shows the 'Memory' pane with a hex dump of the 'inFrBuff' variable. The address range is 700DFFF0 to 700E0060. The data is displayed in a table with columns for Address, 0-3, 4-7, 8-B, and C-F.

Address	0 - 3	4 - 7	8 - B	C - F
700DFFF0	80000000	80151800	481B44B2	00000008
700E0000	904E61C6	316726C9	6716C900	06C90029
700E0010	C2002167	002966A2	6692C250	81500031
700E0020	C9011490	00196716	116726C9	6706C900
700E0030	02820009	03820065	0195700E	07403B02
700E0040	004D0282	700E0382	40020147	66A2C207
700E0050	C2500029	003966A6	00C00282	700E0382
700E0060	40020147	66A2C207	C2500021	00316692

Memory View

Various renderings modes



The screenshot displays a memory viewer interface with a list of memory addresses and their corresponding data. The interface includes tabs for different rendering modes: Hex Integer, Traditional, ASCII, and Floating Point. A dropdown menu is open, showing the selected mode (Floating Point) and other available modes: Traditional, Hex Integer, Signed Integer, Unsigned Integer, ASCII, and Hex. An 'Add Rendering(s)' button is visible next to the dropdown.

Address	Hex Integer	Traditional	ASCII	Floating Point
0x700E0014	002966A2 6692C250 81500031 C9011490	çf).PÀ.f1.P...È		
0x700E0024	00196716 116726C9 6706C900 02820009	.g...È&g...È.g...		
0x700E0034	03820065 0195700E 07403B02 004D0282	e...p...;@...M.		
0x700E0044	700E0382 40020147 66A2C207 C2500029	...pG...@.Âçf).PÀ		
0x700E0054	003966A6 00C00282 700E0382 40020147	!f9...Â...pG...@		
0x700E0064	66A2C207 C2500021 00316692 14908150	.Âçf!.PÀ.f1.P...		
0x700E0074	A8028201 0E038200 02019570 8207405B		
0x700E0084	82009002 47700E03 07400201 2166A2C2p		
0x700E0094	A6C25000 82003966 8200C002 47700E03	.PÀ!f9.		
0x700E00A4	07400201 3166A2C2 A6C25000 82003966	..@.Âçf		
0x700E00B4	8200C002 47700E03 07400201 396722C2	.Â...p		

Memory Monitor: &inFrBuff : 0x700DFFF0

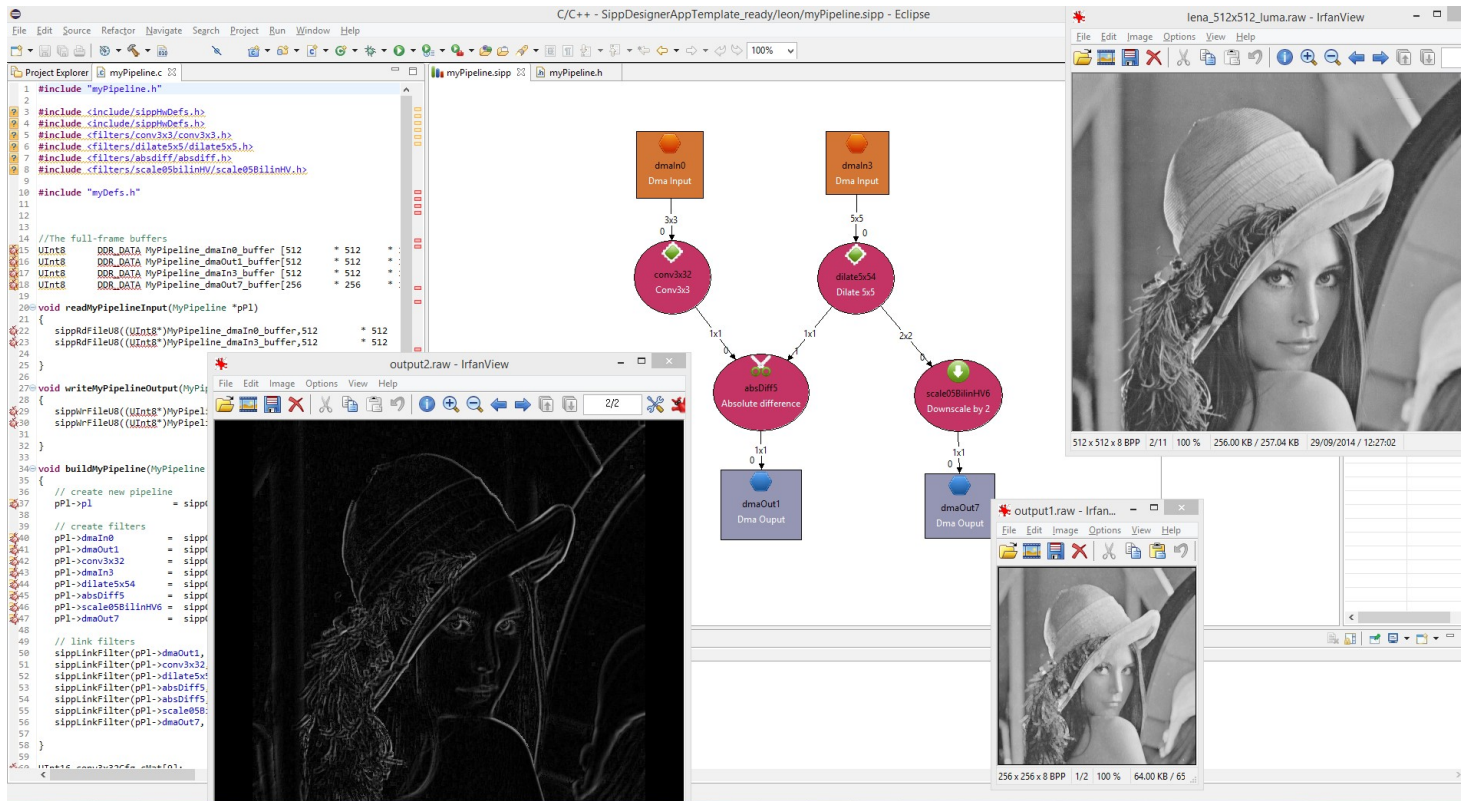
Select rendering(s) to create:

- Floating Point
- Traditional
- Hex Integer
- Signed Integer
- Unsigned Integer
- ASCII
- Hex

Add Rendering(s)

ISP Graph Designer

Pipeline Editor & Code Generator



The screenshot displays the ISP Graph Designer interface, which includes a code editor, a pipeline graph, and image preview windows.

Code Editor (myPipeline.h):

```

1 #include "myPipeline.h"
2
3 #include <include/sippmDefs.h>
4 #include <include/sippmDefs.h>
5 #include <filters/conv3x3/conv3x3.h>
6 #include <filters/dilate5x5/dilate5x5.h>
7 #include <filters/absdiff/absdiff.h>
8 #include <filters/scale08b11inH/scale08b11inH.h>
9
10 #include "myDefs.h"
11
12
13
14 //The full-frame buffers
15 UInt8 DOR_DATA MyPipeline_dmaIn0_buffer[512 * 512 * 3]
16 UInt8 DOR_DATA MyPipeline_dmaOut1_buffer[512 * 512 * 3]
17 UInt8 DOR_DATA MyPipeline_dmaIn3_buffer[512 * 512 * 3]
18 UInt8 DOR_DATA MyPipeline_dmaOut7_buffer[256 * 256 * 3]
19
20 void readMyPipelineInput(MyPipeline *p1)
21 {
22     sippmFileU8((UInt8*)MyPipeline_dmaIn0_buffer, 512 * 512 * 3)
23     sippmFileU8((UInt8*)MyPipeline_dmaIn3_buffer, 512 * 512 * 3)
24 }
25
26
27 void writeMyPipelineOutput(MyPipeline *p1)
28 {
29     sippmFileU8((UInt8*)MyPipeline_dmaOut1_buffer, 512 * 512 * 3)
30     sippmFileU8((UInt8*)MyPipeline_dmaOut7_buffer, 256 * 256 * 3)
31 }
32
33
34 void buildMyPipeline(MyPipeline *p1)
35 {
36     // create new pipeline
37     p1->p1 = sippm
38
39     // create filters
40     p1->dmaIn0 = sippm
41     p1->dmaOut1 = sippm
42     p1->conv3x3 = sippm
43     p1->dmaIn3 = sippm
44     p1->dilate5x5 = sippm
45     p1->absdiff = sippm
46     p1->scale08b11inH = sippm
47     p1->dmaOut7 = sippm
48
49     // link filters
50     sippmLinkFilter(p1->dmaIn0, p1->conv3x3)
51     sippmLinkFilter(p1->conv3x3, p1->dilate5x5)
52     sippmLinkFilter(p1->dilate5x5, p1->absdiff)
53     sippmLinkFilter(p1->absdiff, p1->scale08b11inH)
54     sippmLinkFilter(p1->scale08b11inH, p1->dmaOut7)
55 }
56
57
58 }
59
60
61

```

Pipeline Graph:

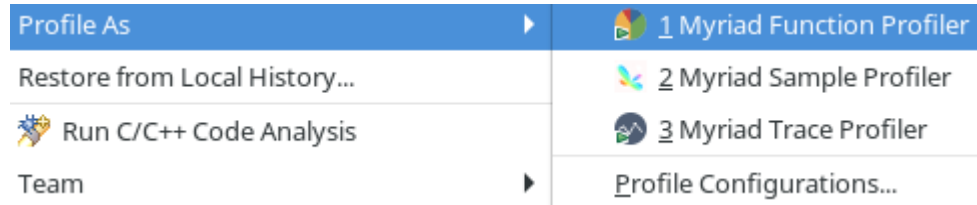
- dmaIn0** (Dma Input) connects to **conv3x32** (Conv3x3).
- conv3x32** connects to **absdiff5** (Absolute difference).
- absdiff5** connects to **dmaOut1** (Dma Output).
- dmaIn3** (Dma Input) connects to **dilate5x54** (Dilate 5x5).
- dilate5x54** connects to **scale08b11inH6** (Downscale by 2).
- scale08b11inH6** connects to **dmaOut7** (Dma Output).

Image Preview Windows:

- lena_512x512_luma.raw - IrfanView:** Shows the original input image (lena_512x512_luma.raw) with dimensions 512 x 512 x 8 BPP, 2/11, 100%, 256.00 KB / 257.04 KB, 29/09/2014 / 12:27:02.
- output2.raw - IrfanView:** Shows the output image (output2.raw) with dimensions 256 x 256 x 8 BPP, 1/2, 100%, 64.00 KB / 65.

Profiler launchers

There is a launcher for every type of profiler




Myriad Function Profiler

Execution statistics

10 cpus/threads and 127 functions statistics in the current trace

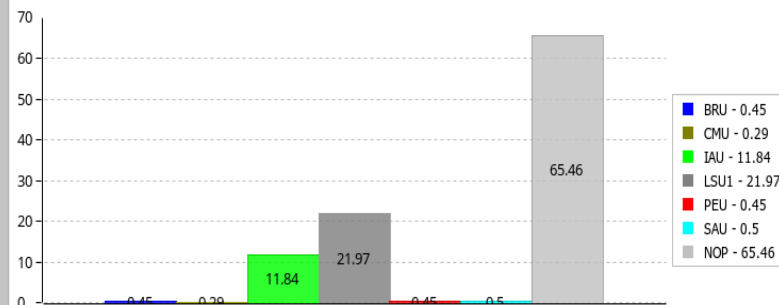
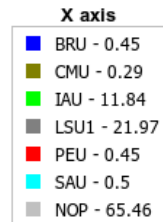
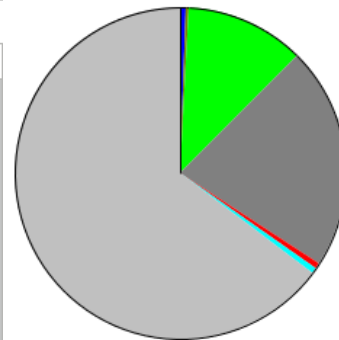
	Thread/Functi... Name	Calls	Code density	Inclusive Net					Usage	Avg	Min
				Avg	Min	Max	Total				
45	unlockCrit...	9	5.91	567.00	536	667	5,109	0.00 %	567.00		
46	▲ LOS-24577										
47	__getreent	72	10.34	1,329.00	1,322	1,414	95,721	0.08 %	1,329.00		
48	_System_s...	2	5.10	428.00	414	443	857	0.00 %	428.00		
49	_Thread_...	72	4.75	418.00	415	475	30,134	0.02 %	418.00		
50	buff_out_c...	75	8.88	2,646.00	2,588	4,345	198,459	0.16 %	2,646.00		
51	convertTo...	75	3.12	425.00	423	524	31,938	0.03 %	425.00		
52	Fatal_exte...	1	1.53	669.00	669	669	669	0.00 %	669.00		
53	mvQueue...	75	4.32	1,722.00	1,674	3,365	129,157	0.11 %	1,722.00		
54	▲ LOS-24578										
55	__getreent	62	10.36	1,333.00	1,322	1,447	82,656	0.07 %	1,333.00		
56	_System_s...	4	5.21	438.00	414	507	1,752	0.00 %	438.00		
57	_Thread_...	62	4.77	420.00	415	478	26,077	0.02 %	420.00		
58	buff_out_c...	62	8.85	2,621.00	2,596	3,090	162,561	0.13 %	2,621.00		
59	convertTo...	62	3.14	427.00	423	522	26,491	0.02 %	427.00		

Myriad Sample Profiler

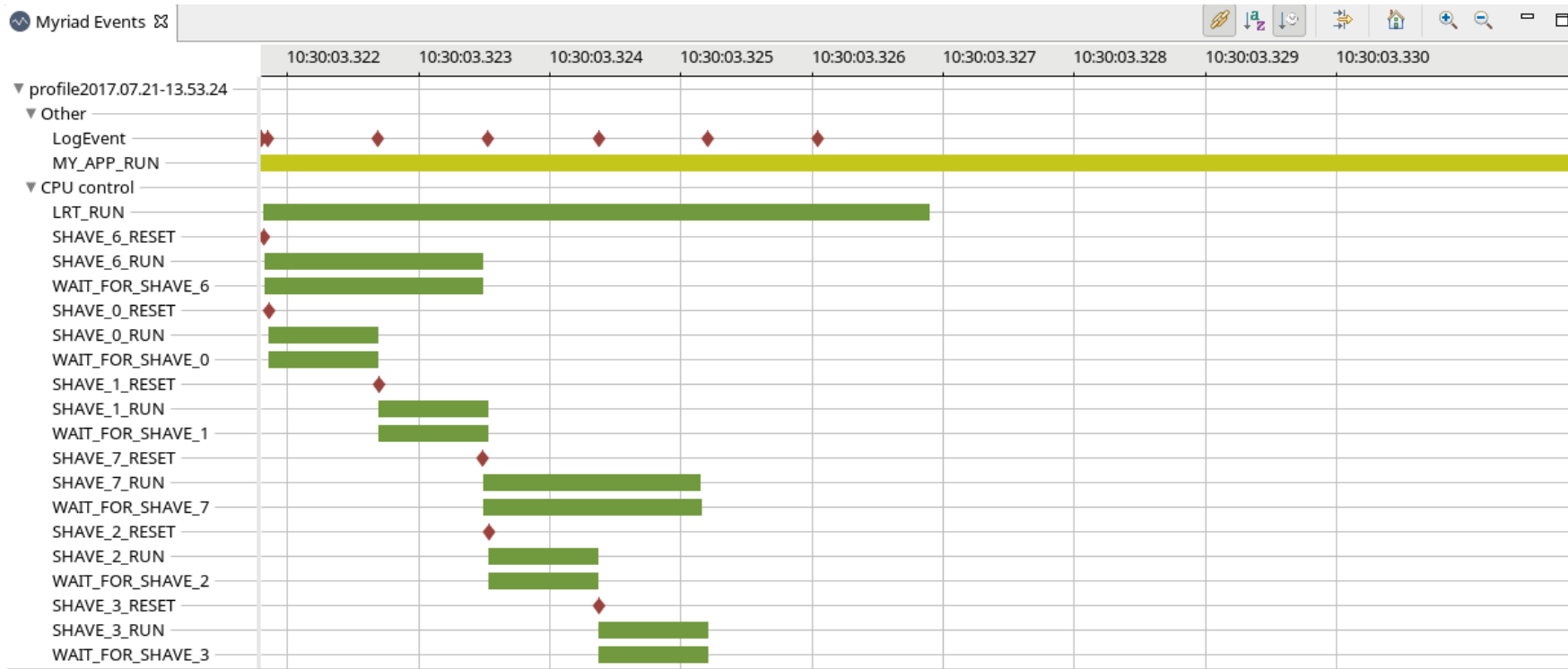
Myriad Samples 

Loaded sample

Name/Location	↓	S	ILP	BRU	CMU	IAU	LSU	LSU1	PEU	SAU	VAU	NOP	Dev	% T	% Sai
Summary		1059											10.2%	100.0%	
Shave 4		1324											3.63%	12.5%	100.0%
...eProfilerExample/shave/in		1324											3.63%	12.5%	100.0%
spExample4_func1		8050	0.35	0.48	0.35	11.6%	0.00	21.6%	0.48	0.50	0.00	65.8%	0.89%	0.76	6.08
lrt_spExample4_func1		8049	0.35	0.45	0.29	11.8%	0.00	21.9%	0.45	0.50	0.00	65.4%	0.89%	0.76	6.08
spExample4_func2		7530	0.35	0.40	0.42	11.5%	0.00	21.8%	0.40	0.42	0.00	65.8%	0.86%	0.71	5.69
lrt_spExample4_func2		7530	0.35	0.50	0.52	11.5%	0.00	21.7%	0.50	0.42	0.00	65.7%	0.86%	0.71	5.69
lrt_spExample4_func3		7011	0.35	0.39	0.24	11.7%	0.00	21.8%	0.39	0.46	0.00	65.7%	0.83%	0.66	5.29
spExample4_func3		7010	0.35	0.44	0.39	11.5%	0.00	21.8%	0.44	0.43	0.00	65.7%	0.83%	0.66	5.29
spExample4_func4		6492	0.36	0.92	0.00	11.9%	0.00	21.4%	0.92	0.82	0.00	65.6%	0.80%	0.61	4.90
lrt_spExample4_func4		6491	0.36	0.86	0.00	12.1%	0.00	21.5%	0.86	0.89	0.00	65.5%	0.80%	0.61	4.90
spExample4_func5		5972	0.36	0.90	0.00	11.8%	0.00	21.3%	0.90	0.82	0.00	65.8%	0.77%	0.56	4.51
lrt_spExample4_func5		5972	0.36	0.90	0.00	11.8%	0.00	21.5%	0.90	0.67	0.00	65.6%	0.77%	0.56	4.51
lrt_spExample4_func6		5453	0.35	0.37	0.26	11.4%	0.00	21.7%	0.37	0.48	0.00	66.1%	0.73%	0.51	4.12
spExample4_func6		5452	0.35	0.40	0.37	11.4%	0.00	21.9%	0.40	0.35	0.00	65.8%	0.73%	0.51	4.12
lrt_spExample4_func7		4934	0.34	0.00	0.83	11.0%	0.00	22.3%	0.00	0.00	0.00	65.8%	0.70%	0.47	3.73
spExample4_func7		4933	0.34	0.00	0.85	11.1%	0.00	22.2%	0.00	0.00	0.00	65.7%	0.70%	0.47	3.73
spExample4_func8		4415	0.34	0.00	0.84	11.1%	0.00	22.1%	0.00	0.00	0.00	65.8%	0.66%	0.42	3.33

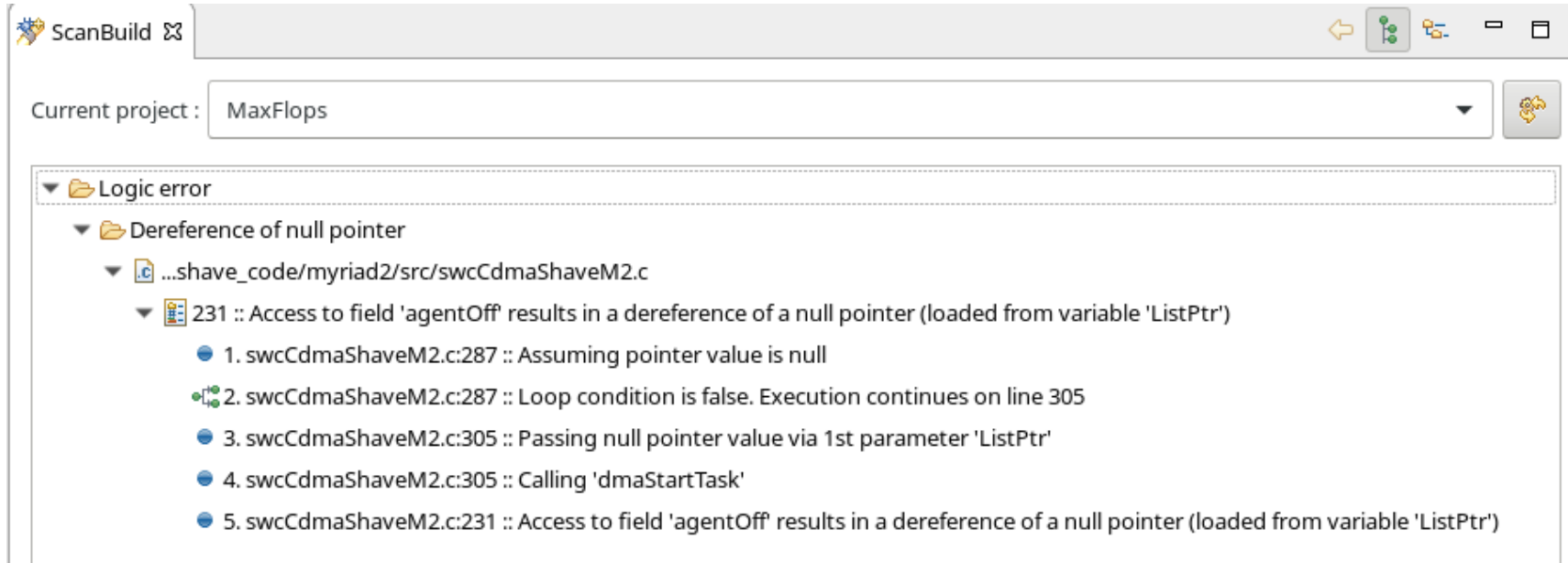


Myriad Trace Profiler



ScanBuild view

Displays a static code analysis of the C code



The screenshot shows the ScanBuild tool interface. At the top, there is a tab labeled "ScanBuild" with a gear icon. Below the tab, a "Current project:" dropdown menu is set to "MaxFlops". The main area displays a tree view of errors under the heading "Logic error". The tree is expanded to show a "Dereference of null pointer" error. This error is further expanded to show a specific instance in the file "...shave_code/myriad2/src/swcCdmaShaveM2.c" at line 231. The error message states: "Access to field 'agentOff' results in a dereference of a null pointer (loaded from variable 'ListPtr')". Below this message, a list of five steps describes the execution flow leading to the error:

- 1. swcCdmaShaveM2.c:287 :: Assuming pointer value is null
- 2. swcCdmaShaveM2.c:287 :: Loop condition is false. Execution continues on line 305
- 3. swcCdmaShaveM2.c:305 :: Passing null pointer value via 1st parameter 'ListPtr'
- 4. swcCdmaShaveM2.c:305 :: Calling 'dmaStartTask'
- 5. swcCdmaShaveM2.c:231 :: Access to field 'agentOff' results in a dereference of a null pointer (loaded from variable 'ListPtr')

Floating Point Converter View

Allow conversions between floating point 16, floating point 32 and floating point 64

Console

Myriad devkits

Floating Point Converter

f16

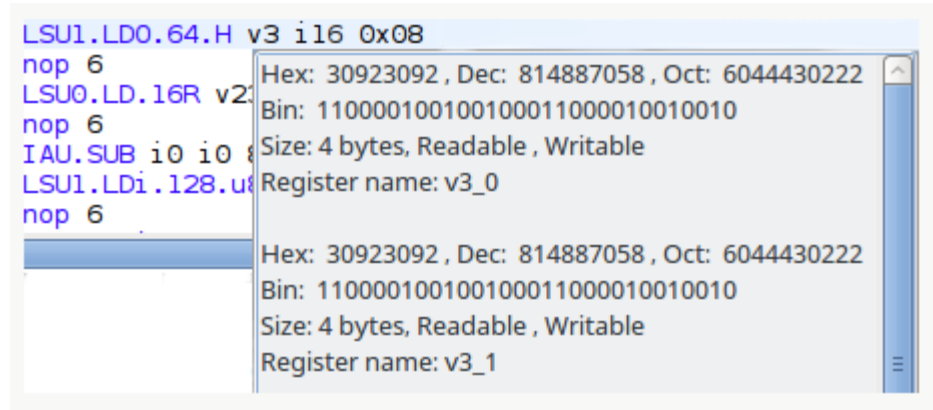
f32

f64

	Sign	Exponent	Mantissa
Value	-1	2 ⁴	1.14453125
Encoded as:	1	19	148
Binary	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Decimal Representation	<input type="text" value="-18.3125"/>		
Binary Representation	<input type="text" value="1100110010010100"/>		
Hexadecimal Representation	<input type="text" value="0xCC94"/>		
Casted to double precision	<input type="text" value="-18.3125"/>		

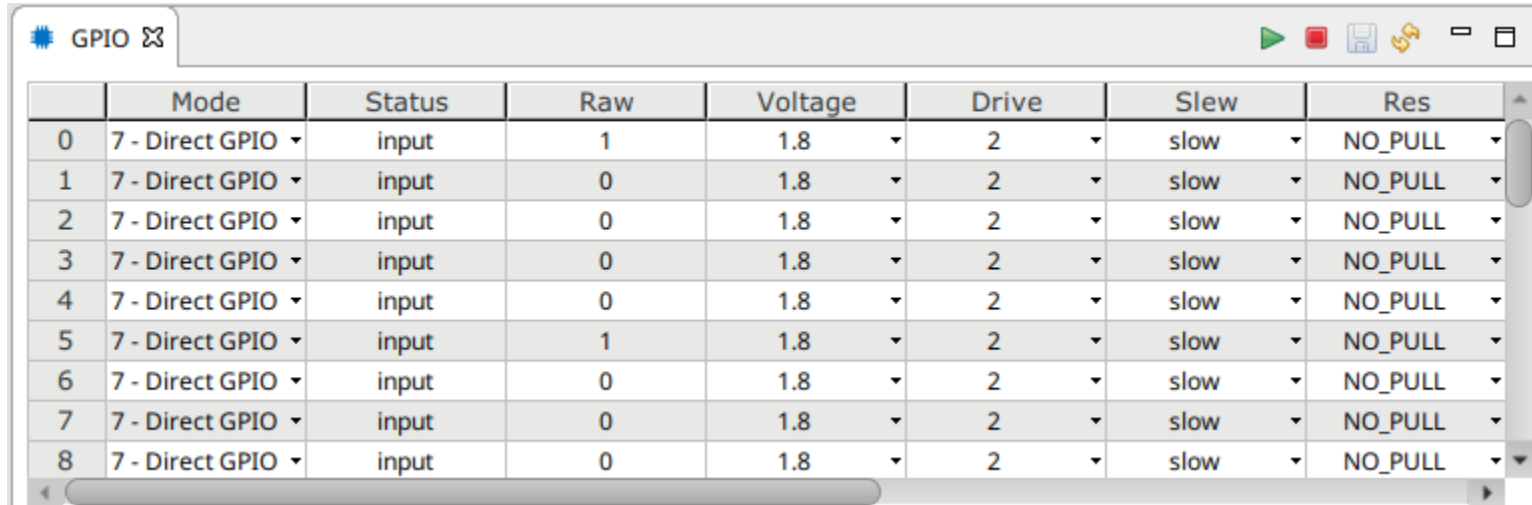
ASM editor

Displays the register values when hovering the mouse over the used registers during the debugging process.



GPIO editor

Allows you to visualize and modify the pin status of General Purpose Input Output (GPIO)



The screenshot shows a software window titled "GPIO" with a table of pin configurations. The table has 8 columns: Pin, Mode, Status, Raw, Voltage, Drive, Slew, and Res. The rows represent pins 0 through 8. All pins are configured as "7 - Direct GPIO" in "input" mode. The "Raw" values are 1, 0, 0, 0, 0, 1, 0, and 0 respectively. The "Voltage" is 1.8V for all pins. The "Drive" is 2 for all pins. The "Slew" is "slow" for all pins. The "Res" is "NO_PULL" for all pins.

	Mode	Status	Raw	Voltage	Drive	Slew	Res
0	7 - Direct GPIO	input	1	1.8	2	slow	NO_PULL
1	7 - Direct GPIO	input	0	1.8	2	slow	NO_PULL
2	7 - Direct GPIO	input	0	1.8	2	slow	NO_PULL
3	7 - Direct GPIO	input	0	1.8	2	slow	NO_PULL
4	7 - Direct GPIO	input	0	1.8	2	slow	NO_PULL
5	7 - Direct GPIO	input	1	1.8	2	slow	NO_PULL
6	7 - Direct GPIO	input	0	1.8	2	slow	NO_PULL
7	7 - Direct GPIO	input	0	1.8	2	slow	NO_PULL
8	7 - Direct GPIO	input	0	1.8	2	slow	NO_PULL