QEMU A9 可以跑的软件包

田雄兵
2022.07-2022.08

引言：本文档主要是使用 QEMU A9 环境下，测试 RT-thread 的软件包，并把能够使用此环境运行的软件包进行整理，方便在不具备硬件的条件下开发调试。（软件包网址参考：https://github.com/supperthomas/rtthread_software_package_list_show/blob/main/rtthread_softlist.md 在软件包详情页中，都有相关介绍和具体使用、添加方法，本文主要进行实战，列出能够运行的，提供运行截图）

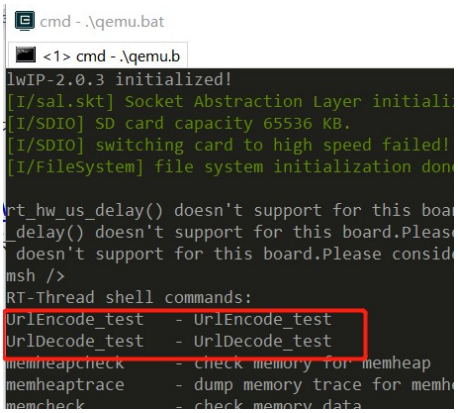
步骤：1.配置 ENV 环境（env1.2.4）。2.下载 rt-thread 源码，进入 BSP 目录，进入 QEMU A9，右键运行 ENV 环境。3.使用 menuconfig 选择软件包，保存并退出，pkgs --update 命令更新包到 BSP 中。4.使用 scons -j4 编译。5.运行.\qemu.bat，调试命令。

tools

1. UrlEncode，一个简单易用的 Url 编解码工具。

在进行网络请求时,经常需要对参数进行 UrlEncode 编码,本软件包可以比较方便的对参数进行编码以及解码.

URL 编码(URL encoding)，也称作百分号编码(Percent-encoding)， 是特定上下文的统一资源定位符 (URL)的编码机制。用于统一资源标识符(URI)的编码，也用于为"application/x-www-form-urlencoded" MIME 准备数据， 因为它用于通过 HTTP 的请求操作(request)提交 HTML 表单数据。



```
msh />U
UrlEncode_test
UrlDecode_test
msh />UrlrlD
msh />UrlrlD
msh />UrlD
UrlDecode_test
msh />UrlDecode_test
src: RT-Thread%2C%E5%B0%8F%E8%80%8C%E7%BE%8E%E7%9A%84%E7%89%A9%E8%81%94%E7%BD%91%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F
src length: 111
result: RT-Thread,σ▌ÂΦÇîτ┘ÄτÜätë←ΦüötÌæµôìΣÌ£τ│╗τ╗ƒ
result length: 43
msh />U
UrlEncode_test
UrlDecode_test
msh />UrlE
UrlEncode_test
msh />UrlEncode_test
src: RT-Thread,σ▌ÂΦÇîτ┘ÄτÜätë←ΦüötÌæµôìΣÌ£τ│╗τ╗ƒ
src length: 43
result: RT-Thread%2C%E5%B0%8F%E8%80%8C%E7%BE%8E%E7%9A%84%E7%89%A9%E8%81%94%E7%BD%91%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9
F
result length: 111
msh />
qemu-system-arm.exe*[32]:15316                    « 180206[64]  1/1  [+] NUM  PRI↕  118x31  (7,32766) 25V  9548  100%
```

2.Memory Performance Testing 这是一个运行在 RT-Thread 上的内存性能测试软件包，用于对 ARM CPU 的 内存性能评。

评估不同类型内存的读写性能

对于带有 cache 的 CPU，可用于评估 cache 性能

使用方法:

在 msh 中运行 <memory_perf 0x10100000 0x100000> 命令

注意事项:

检查内存测试地址与长度是否设置正确

测试内存性能时推荐对比打开 cache 与 关闭 cache 的测试结果

cmd - .\qemu.bat

<1> cmd - .\qemu.b

```
msh />
msh />
msh />
msh />memo
memory_perf
msh />memory_perf 0x10100000 0x100000

MemoryPerf version 1.0.0
Copyright (c) 2022 SummerGift (summergift2019@gmail.com)
Licensed under the MIT License version.

Memory performance testing start...
address: 0x600bf124, length: 0x100000, iterations: 200
Data length : 209 MB.
-------------------------------------
8-bit write speed test begin.
Spend time : 0.980000 s.
8-bit write speed: 213.995102 M/s.
8-bit read speed test begin.
Spend time : 0.920000 s.
8-bit Read speed: 227.951294 M/s.
-------------------------------------
16-bit write speed test begin.
Spend time : 0.520000 s.
16-bit write speed: 403.298462 M/s.
16-bit read speed test begin.
Spend time : 1.080000 s.
16-bit Read speed: 194.180740 M/s.
-------------------------------------
32-bit write speed test begin.
Spend time : 0.290000 s.
32-bit Write speed: 723.155884 M/s.
32-bit read speed test begin.
Spend time : 0.570000 s.
32-bit Read speed: 367.921417 M/s.
64bit not implemented
Memory performance completed.
msh />
msh />
```

3785 chars {1,32734}-{9,32766}:{9,32766} stream selection

3.mbedtls_bench 是 mbedtls 加密算法的性能测试工具，mbedtls 性能测试。分数表示可以处理的块数据量，分数越高意味着性能越好。

```
msh />tl
tls_test
msh />tls_test
Msh />  test sample!
memoryMemory usage before the handshake connection is established:
total    : 7084800
used     : 23564
maximum  : 23564
available: 7061236
Start handshake tick:1080
Finish handshake tick:1212
MbedTLS connect success...
Memory usage after the handshake connection is established:
total    : 7084800
used     : 69196
maximum  : 74792
available: 7015604
Writing HTTP request success...
Getting HTTP response...
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Tue, 02 Aug 2022 04:10:00 GMT
Content-Type: text/plain
Content-Length: 267
Last-Modified: Sat, 04 Aug 2018 02:14:51 GMT
Connection: keep-alive
Vary: Accept-Encoding
ETag: "5b650c1b-10b"
Strict-Transport-Security: max-age=1800; includeSubdomains; preload
Accept-Ranges: bytes

RT-Thread is an open source IoT operating system from China, which has strong scalability: from a tiny kernel running
on a tiny core, for example ARM Cortex-M0, or Cortex-M3/4/7, to a rich feature system running on MIPS32, ARM Cortex-A8
, ARM Cortex-A9 DualCore etc.

MbedTLS connection close success.

msh />
```

4.lwlog:单文件日志打印库。

```
msh />lw
lwlog_demo
msh />lwlog_demo
[EMERG]   lwlog_demo (packages\lwlog-latest\lwlog_demo.c:20) This a emerge log. errno: None
[ALERT]   lwlog_demo (packages\lwlog-latest\lwlog_demo.c:21) This a alert log. errno: None
[CRIT]    lwlog_demo (packages\lwlog-latest\lwlog_demo.c:22) This a crit log. errno: None
[ERR]     lwlog_demo (packages\lwlog-latest\lwlog_demo.c:23) This a err log. errno: None
[WARNING] lwlog_demo (packages\lwlog-latest\lwlog_demo.c:24) This a warning log. errno: None
[NOTICE]  lwlog_demo (packages\lwlog-latest\lwlog_demo.c:25) This a notice log. errno: None
[INFO]    lwlog_demo (packages\lwlog-latest\lwlog_demo.c:26) This a info log.
[DEBUG]   lwlog_demo (packages\lwlog-latest\lwlog_demo.c:27) This a debug log.
msh />
qemu-system-arm.exe*[32]:21832                    « 180206[64]  1/1  [+] NUM  PRI↕  118x39
```

5. logmgr: 日志管理系统功能支持。

该软件包主要用于配置和管理系统中日志相关功能，实现功能如下：

- 支持 ulog 文件后端功能启动；
- 重定向系统 hardfault 和 assert 异常错误回调，添加更多系统异常相关日志输出，包括
  - 函数调用栈日志
  - 内核运行日志
  - 系统负荷监视器日志
  - 当前系统 IPC 状态、内存状态、JS 堆等日志信息

- 支持系统异常时日志输出到 Flash，并在重启后导出到文件功能；

```
msh />lo
logmgr_test
msh />logmgr_test INIT
[2651] I/logmgr: logmgr (v1.0.0) initialized success.
msh />lo
logmgr_test
msh />logmgr_test DIVBYZERO
z:2147483647
msh />lo
logmgr_test
msh />logmgr_test
msh />logmgr_test UNALIGNED
data abort:Execption:
r00:0x00000000 r01:0x60101118 r02:0x00000008 r03:0x00000000
r04:0xdeadbeef r05:0xdeadbeef r06:0xdeadbeef r07:0xdeadbeef
r08:0xdeadbeef r09:0xdeadbeef r10:0xdeadbeef
fp :0x60143084 ip :0xffffffff
sp :0x60143058 lr :0x60012790 pc :0x600127c4
cpsr:0x60080013
thread   cpu bind pri  status      sp     stack size max used left tick  error
-------- --- ---- ---  ------- ---------- ----------  ------  ---------- ---
tshell     0   2   20  running 0x00000144 0x00001000    33%   0x00000002 OK
aio      N/A   2   16  suspend 0x0000008c 0x00000800    06%   0x0000000a OK
mmcsd_de N/A   2   22  suspend 0x000000b8 0x00000400    57%   0x00000013 OK
sys work N/A   2   23  suspend 0x00000088 0x00000800    52%   0x0000000a OK
tcpip    N/A   2   10  suspend 0x000000f4 0x00000400    88%   0x00000013 OK
etx      N/A   2   12  suspend 0x000000b4 0x00000400    26%   0x0000000f OK
erx      N/A   2   12  suspend 0x000000b8 0x00000400    60%   0x00000010 OK
tsystem  N/A   2   30  suspend 0x000000a0 0x00000400    19%   0x00000020 OK
tidle1     1   1   31  running 0x00000064 0x00000400    20%   0x00000015 OK
tidle0   N/A   0   31  ready   0x00000068 0x00000400    21%   0x0000000f OK
timer    N/A   2    4  suspend 0x00000084 0x00000400    19%   0x00000009 OK
shutdown...
(0) assertion failed at function:rt_hw_cpu_shutdown, line number:74
qemu-system-arm.exe*[32]:23260                    « 180206[64]  1/1  [+] NUM
```

6. Dhrystone 单片机性能测试小工具。

RT-Thread 上的 MCU/CPU 性能测试小工具，在 menuconfig 里选中软件包后，在 msh 中输入：

msh> dhrystone_test

就可以看到跑分结果了，例如：

cmd - .\qemu.bat

<1> cmd - .\qemu.b

```
dhrystone_test
msh />dhrystone_test

Dhrystone Benchmark, Version 2.1 (Language: C)

Program compiled without 'register' attribute

Execution starts, 320000 runs through Dhrystone
Execution ends

Final values of the variables used in the benchmark:

Int_Glob:            5
        should be:   5
Bool_Glob:           1
        should be:   1
Ch_1_Glob:           A
        should be:   A
Ch_2_Glob:           B
        should be:   B
Arr_1_Glob[8]:       7
        should be:   7
Arr_2_Glob[8][7]:    320010
        should be:   Number_Of_Runs + 10
Ptr_Glob->
  Ptr_Comp:          1611943884
        should be:   (implementation-dependent)
  Discr:             0
        should be:   0
  Enum_Comp:         2
        should be:   2
  Int_Comp:          17
        should be:   17
  Str_Comp:          DHRYSTONE PROGRAM, SOME STRING
        should be:   DHRYSTONE PROGRAM, SOME STRING
Next_Ptr_Glob->
  Ptr_Comp:          1611943884
        should be:   (implementation-dependent), same as above
  Discr:             0
        should be:   0
  Enum_Comp:         1
        should be:   1
  Int_Comp:          18
```

qemu-system-arm.exe*[32]:23884 « 180206[64]

```
Int_3_Loc:           7
        should be:   7
Enum_Loc:            1
        should be:   1
Str_1_Loc:           DHRYSTONE PROGRAM, 1'ST STRING
        should be:   DHRYSTONE PROGRAM, 1'ST STRING
Str_2_Loc:           DHRYSTONE PROGRAM, 2'ND STRING
        should be:   DHRYSTONE PROGRAM, 2'ND STRING

Measured time too small to obtain meaningful results
Please increase number of runs

msh />
```

qemu-system-arm.exe*[32]:23884 « 180206[64]

7. devmem 读写内存/寄存器的工具。

在 menuconfig 里选中软件包后，在 msh 上输入 devmem 查看使用说明。

写内存/寄存器

byte 方式写入

devmem 0x600a4000 b 0xa

halfword 方式写入

devmem 0x600a4000 h 0xab

word 方式写入

devmem 0x600a4000 w 0xabcd

读内存/寄存器

byte 方式读取

msh />devmem 0x600a4000 b

0x0a

halfword 方式读取

msh />devmem 0x600a4000 h

0x00ab

word 方式读取

msh />devmem 0x600a4000 w

0x0000abcd

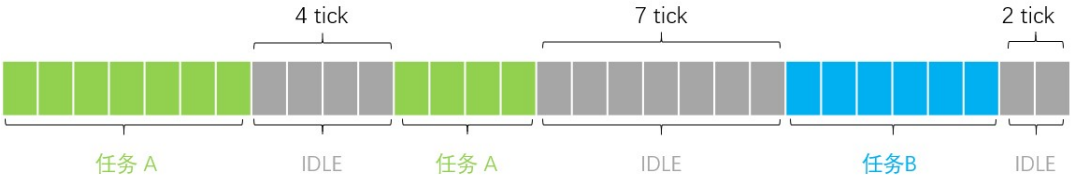
```
msh />devmem

Usage:  devmem address [type [data]]
        address : memory address to act upon
        type    : access operation type : [b]yte, [h]alfword, [w]ord
        data    : data to be written

msh />devmem 0x600a4000 b 0xa
msh />devmem 0x600a4000 b
0x0a
msh />devmem 0x600a4000 w 0xabcd
msh />devmem 0x600a4000 w
0x0000abcd
```

```
msh />devmem

Usage:  devmem address [type [data]]
        address : memory address to act upon
        type    : access operation type : [b]yte, [h]alfword, [w]ord
        data    : data to be written

msh />de
devmem
msh />devmem data
data abort:Execption:
r00:0x000000da r01:0x00000000 r02:0x000000da r03:0x000000da
r04:0xdeadbeef r05:0xdeadbeef r06:0xdeadbeef r07:0xdeadbeef
r08:0xdeadbeef r09:0xdeadbeef r10:0xdeadbeef
fp :0x6014888c ip :0x07ffffff
sp :0x60148860 lr :0xffffffff pc :0x6001284c
cpsr:0x60000013
thread   cpu bind pri  status      sp     stack size max used left tick  error
-------- --- ---- ---  ------- ---------- ---------- ------  ---------- ---
tshell     0    2  20  running 0x00000144 0x00001000    11%   0x00000006 OK
aio      N/A    2  16  suspend 0x0000008c 0x00000800    06%   0x0000000a OK
mmcsd_de N/A    2  22  suspend 0x000000b8 0x00000400    57%   0x00000013 OK
sys work N/A    2  23  suspend 0x00000088 0x00000800    52%   0x00000008 OK
tcpip    N/A    2  10  suspend 0x000000f4 0x00000400    88%   0x00000012 OK
etx      N/A    2  12  suspend 0x000000b0 0x00000400    28%   0x0000000d OK
erx      N/A    2  12  suspend 0x000000bc 0x00000400    60%   0x00000010 OK
tsystem  N/A    2  30  suspend 0x000000a0 0x00000400    19%   0x00000020 OK
tidle1     1    1  31  running 0x00000064 0x00000400    20%   0x00000013 OK
tidle0   N/A    0  31  ready   0x00000068 0x00000400    21%   0x00000015 OK
timer    N/A    2   4  suspend 0x00000080 0x00000400    19%   0x00000009 OK
shutdown...
```

8. CPUU: CPU 使用率统计小工具，目前不支持多核。

每个时间片侦测一次当前线程，如果当前正在运行 idle 线程，空闲计数器自增。一个周期后，计算 IDEL 线程运行时间的占比。

例如：



上图展示一个周期内，某个 CPU 上线程时间片信息。假设一个方格代表一个时间片。

一个周期总时间片数 30 tick

idle 总共运行 13 tick

CPU 使用率 = 13 / 30 * 100

msh 命令行输入 usage -l 50，调整 CPU 使用率，使其不低于 50%

```
msh />us
usage
msh />usage
cpu: 0
msh />us
usage
msh />usage -l 50
CPU usage adjusted to 50%
msh />us
usage
msh />usage
cpu: 2
```

9.CoreMark：EEMBC 的单片机性能测试小工具。

```
msh />co
core mark
msh />core mark
Benchmark started, please make sure it runs for at least 10s.

2K performance run parameters for coremark.
CoreMark Size    : 666
Total ticks      : 624
Total time (secs): 6
Iterations/Sec   : 600
ERROR! Must execute for at least 10 secs for a valid result!
Iterations       : 3600
Compiler version : GCC10.2.1 20201103 (release)
Compiler flags   :
Memory location  : STACK
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x5275
Errors detected
```

10.anv_trace 帮助跟踪代码执行过程。

```
msh />anv
anv_trace_example
msh />anv_trace_example
== [MESSAGE] [                FILENAME: LINE | FUNCTION           ] Begin Trace (2018/01/01 - 08:00:53)

-- [Debug  ] [packages\anv_trace-latest\anv_trace_example.c:   11 | anv_trace_example   ] << entering "anv_trace_example"
-- [Debug  ] [packages\anv_trace-latest\anv_trace_example.c:   12 | anv_trace_example   ] Hello Debug!
-- [Info   ] [packages\anv_trace-latest\anv_trace_example.c:   13 | anv_trace_example   ] Hello Info!
-- [Warning] [packages\anv_trace-latest\anv_trace_example.c:   14 | anv_trace_example   ] Hello Warning!
-- [Error  ] [packages\anv_trace-latest\anv_trace_example.c:   15 | anv_trace_example   ] Hello Error!
-- [Fatal  ] [packages\anv_trace-latest\anv_trace_example.c:   16 | anv_trace_example   ] Hello Fatal!
-- [Debug  ] [packages\anv_trace-latest\anv_trace_example.c:   17 | anv_trace_example   ] >> leaving  "anv_trace_example"

== [MESSAGE] [                FILENAME: LINE | FUNCTION           ] End Trace   (2018/01/01 - 08:00:53)
msh />
qemu-system-arm.exe*[32]:13340                                   « 180206[64]  1/1  [+] NUM  PRI:  135x43   (7,32766) 25V
```

11.anv_testsuit 单元测试框架。

```
msh />anv_te
anv_testsuit_example
msh />anv_testsuit_example
==== BEGIN RUNNING TESTS ====
---- BEGIN TEST SUITE: test_group_1 ----

[0] test_group_1_fixture_1         => FAILURE(10): 'a == b'
[1] test_group_1_fixture_2         => SUCCESS
[2] test_group_1_fixture_3         => FAILURE(20): '0' (rip)

SUCCESS: 1/3

---- END TEST SUITE: test_group_1 ----
---- BEGIN TEST SUITE: test_group_2 ----

[0] test_group_2_fixture_1         => SUCCESS

SUCCESS: 1/1

---- END TEST SUITE: test_group_2 ----
==== END RUNNING TESTS ====

msh />
qemu-system-arm.exe*[32]:25804
```