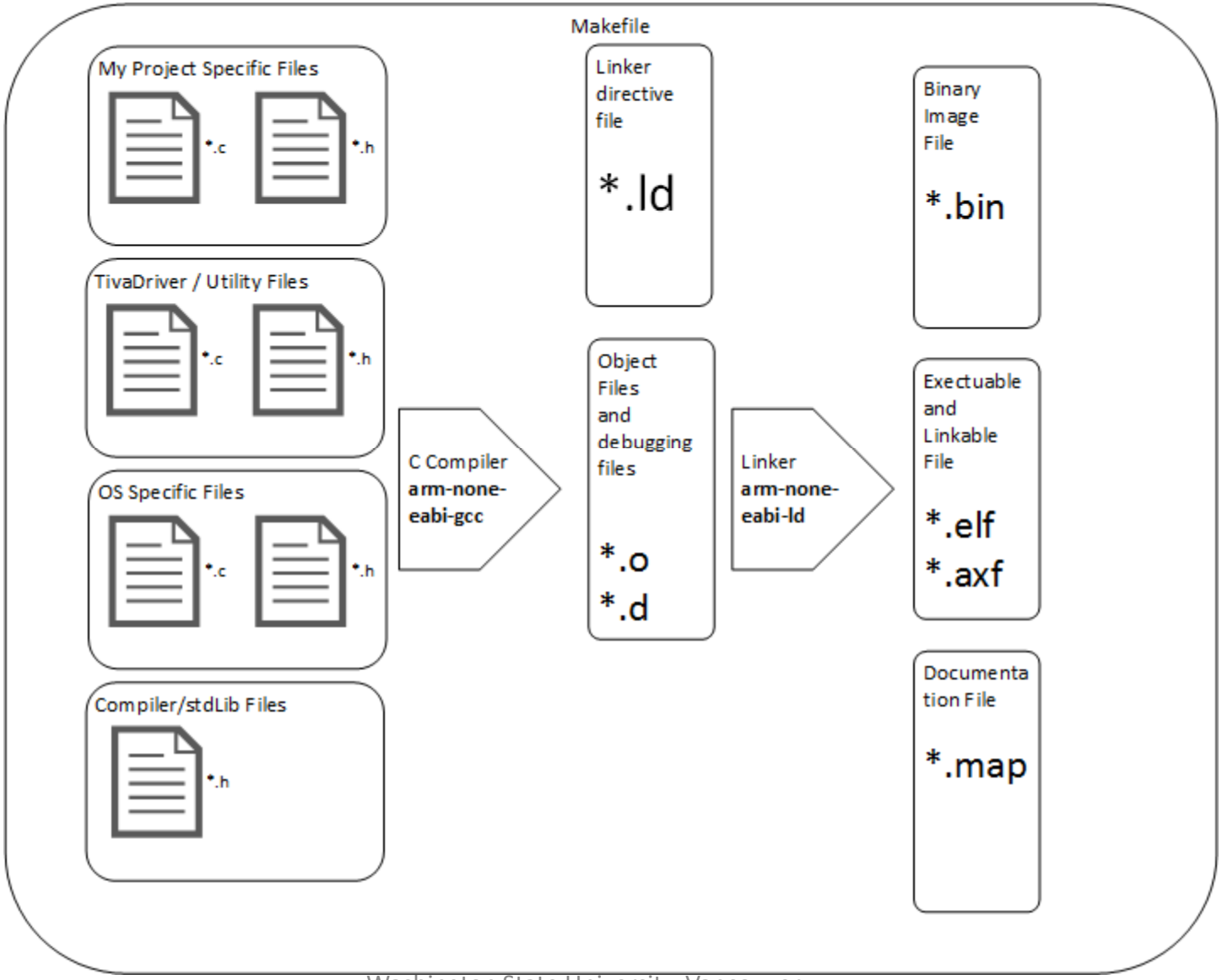
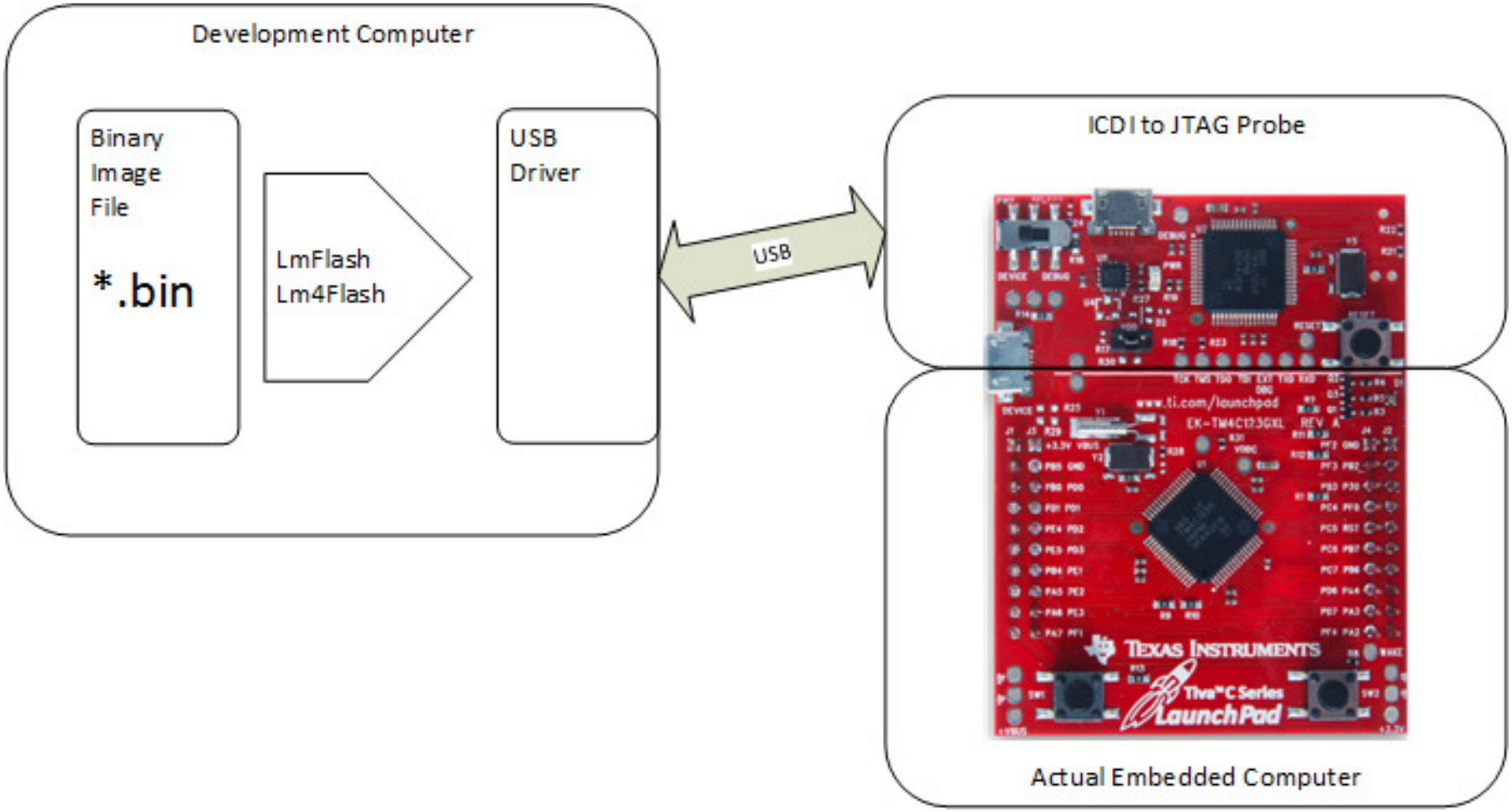
CS466 Lab03 Supplement

Build and Debug of producerConsumer.c

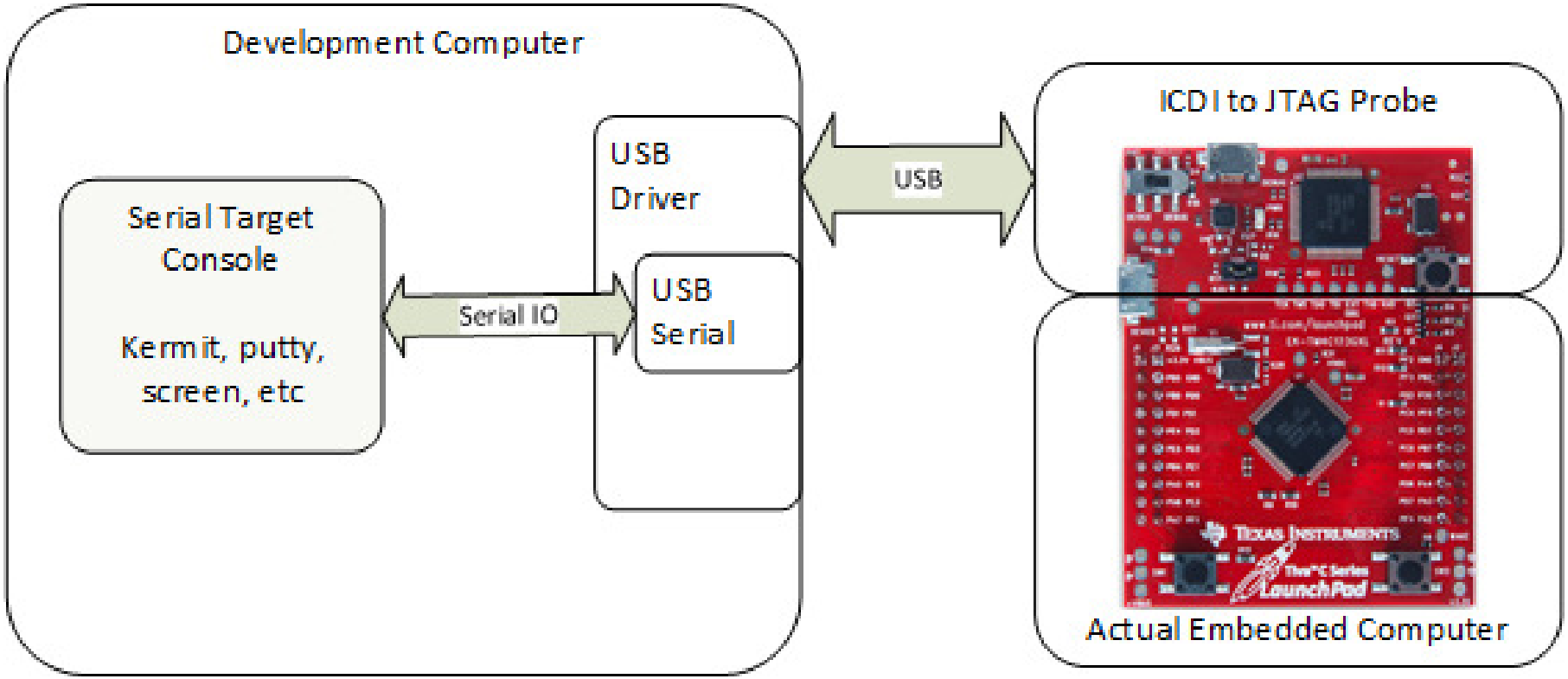
Building


Makefile
My Project Specific Files
*.c
*.h
TivaDriver / Utility Files
*.c
*.h
OS Specific Files
*.c
*.h
Compiler/stdLib Files
*.h
C Compiler
arm-none-eabi-gcc
Linker directive file
*.ld
Object Files and debugging files
*.o
*.d
Linker
arm-none-eabi-ld
Binary Image File
*.bin
Exectuable and Linkable File
*.elf
*.axf
Documentation File
*.map

Normal Flashing



Debug Log (serial I/O)



Debug Log (serial I/O using kermit)

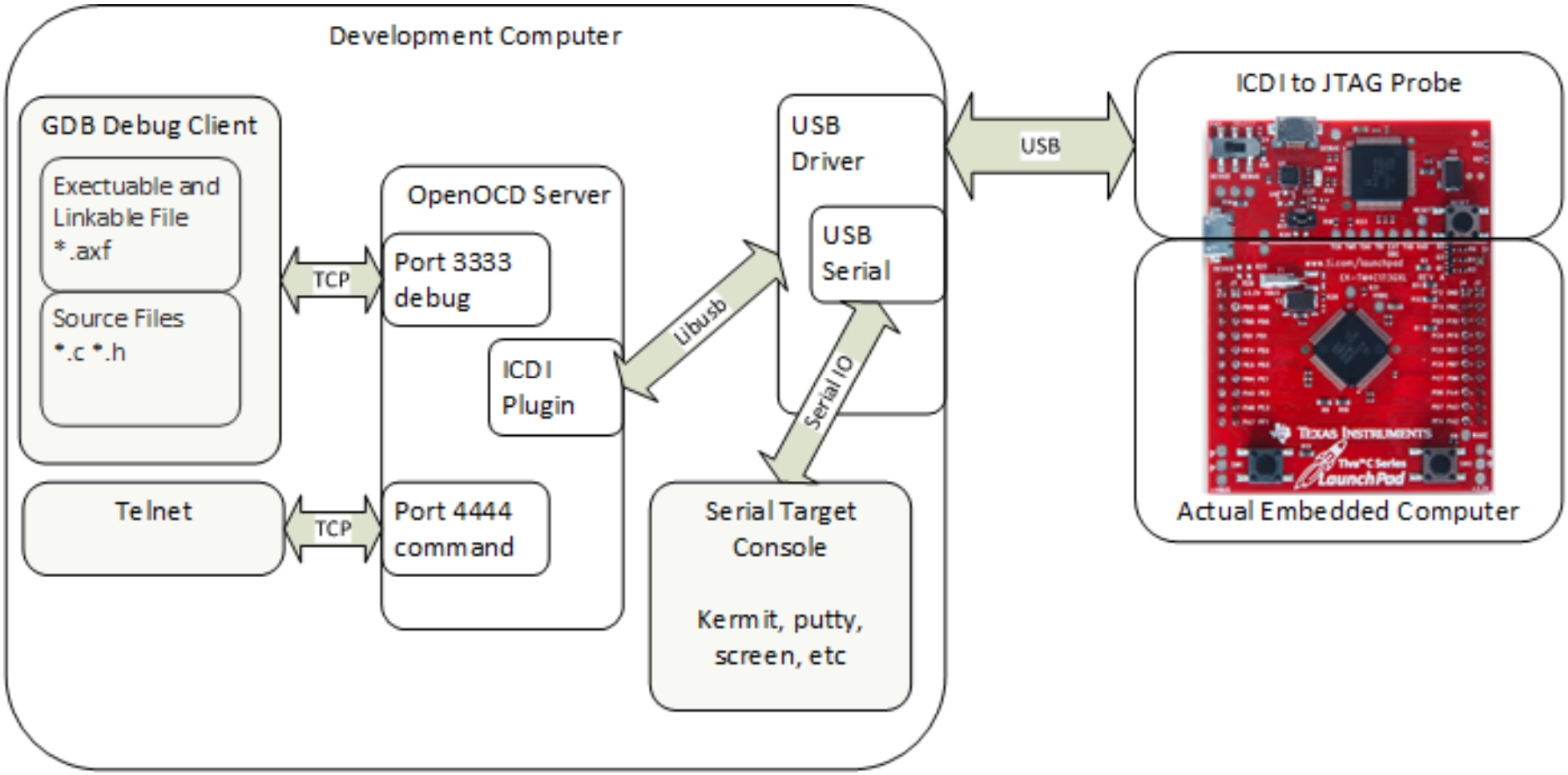
```
$ cat ~/kermACM0
set line /dev/ttyACM0
set speed 115200
set carrier off
set parity none
set handshake none
set stop-bits 1
set flow xon/xof

c
q
```

```
$ kermit ~/kermACM0
Connecting to /dev/ttyACM0, speed 115200
 Escape character: Ctrl-\ (ASCII 28, FS): enabled
Type the escape character followed by C to get back,
or followed by ? to see other options.
----------------------------------------------------
C-Receive
C-Produce prod1
C-Receive
C-Produce prod2
C-Receive
```

```
        "       "
#ifdef USB_SERIAL_OUTPUT
    UARTprintf("C-Receive\n");
#endif
        "       "
```

Debug Connection



Starting OpenOCD Server

```
$ # run the openocd command below in it's own terminal
$
$ openocd --file board/ek-tm4c123gxl.cfg -c init -c halt
Open On-Chip Debugger 0.10.0+dev-00173-g496fcfd (2017-09-25-11:01)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.org/doc/doxygen/bugs.html
Info : The selected transport took over low-level target control. The
results might differ compared to plain JTAG/SWD
adapter speed: 500 kHz
Info : clock speed 32767 kHz
Info : ICDI Firmware version: 9270
Info : tm4c123gh6pm.cpu: hardware has 6 breakpoints, 4 watchpoints
target halted due to debug-request, current mode: Thread
xPSR: 0x61000000 pc: 0x00001bc0 psp: 0x20001078

// lmflash will not work when this server is running
// Ctrl-C will kill the openocd server instance
```

Setting up gdb

```
$ # GDB requires a global setup file in your $HOME dir
$ cat ~/.gdbinit
set auto-load safe-path /
$
```

```
$ # This file goes in the directory you build in
$ cat .gdbinit
define reload
    monitor reset halt
    load
    monitor reset init
End
define mr
    make
    reload
End
target extended-remote :3333
Reload
b main
B _assert_failed
$
```

Starting gdb

```
$ arm-none-eabi-gdb gcc/producerConsumer.axf
GNU gdb (GNU Tools for ARM Embedded Processors) 7.10.1.20160616-cvs
Copyright (C) 2015 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "--host=i686-linux-gnu --target=arm-none-eabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from gcc/producerConsumer.axf...done.
ResetISR () at startup_gcc.c:257
257       {
adapter speed: RCLK - adaptive
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000271c msp: 0x20006b40
Loading section .isr_vector, size 0x26c lma 0x0
Loading section .text, size 0x2a54 lma 0x270
Loading section .data, size 0x434 lma 0x2cc4
Start address 0x271c, load size 12532
Transfer rate: 9 KB/sec, 4177 bytes/write.
adapter speed: RCLK - adaptive
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x0000271c msp: 0x20006b40
(gdb)
```

gdb break at _heartbeat entry

```
(gdb) b _heartbeat
Breakpoint 3 at 0x233c: file producerConsumer.c, line 136.
(gdb) c
Continuing.

Breakpoint 3, _heartbeat (notUsed=0x0 <g_pfnVectors>) at producerConsumer.c:136
136           uint32_t greenMs = 500 / portTICK_RATE_MS;
(gdb) l
131       }
132
133       static void
134       _heartbeat( void *notUsed )
135       {
136           uint32_t greenMs = 500 / portTICK_RATE_MS;
137           uint32_t ledOn = 0;
138
139           while(1)
140           {
(gdb) b 139
Breakpoint 4 at 0x2346: file producerConsumer.c, line 139.
(gdb) c
Continuing.

Breakpoint 4, _heartbeat (notUsed=0x0 <g_pfnVectors>) at producerConsumer.c:141
141               ledOn = !ledOn;
(gdb) p ledOn
$1 = 0
```