

4357. Embedded Firmware Essentials

Homework #4

Jae Yang Park (jaeyangp@gmail.com)

Debug with CMSIS-DAP using pyOCD

Python and pyOCD installation

Python

```
$ sudo apt-get install python libusb-1.0.0-dev libncurses5:i386
```

Update USB setting to get non-root access to DAP

```
$ sudo sh -c 'echo SUBSYSTEM=="usb", ATTR{idVendor}=="0d28",  
ATTR{idProduct}=="0204", MODE:="666" > /etc/udev/rules.d/mbed.rules'  
$ sudo /etc/init.d/udev restart
```

Download pyusb and pyOCD

```
$ git clone https://github.com/walac/pyusb  
$ cd pyusb  
$ sudo python setup.py install  
$ cd ..  
$ git clone https://github.com/mbedmicro/pyOCD  
$ sudo python setup.py install
```

pyOCD test

```
$ python test/basic_test.py
```

Debugging homework 4

Target source code (from quiz 4)

```
/*  
 * Enter a hex number [0-9a-fA-F]; Decode it in 4-bit binary format and display them on 4 on board  
 * leds.  
 */  
  
#include "mbed.h"  
  
Serial pc(USBTX, USBRX); // tx, rx  
DigitalOut ledArr[4] = {DigitalOut(LED1), DigitalOut(LED2), DigitalOut(LED3), DigitalOut(LED4)};  
  
void DisplayLed(int ch)  
{  
    int i=0;  
  
    if (ch>='0' && ch<='9')  
        ch-='0';  
    else if (ch>='A' && ch<='F') {  
        ch-='A';  
        ch+=10;  
    } else if (ch>='a' && ch<='f') {  
        ch-='a';  
        ch+=10;  
    } else  
        ch=0;  
  
    for (i=0; i<4; i++) {  
        if(ch& (1<<i))  
            ledArr[i] = 1;  
        else  
            ledArr[i] = 0;  
    }  
}
```

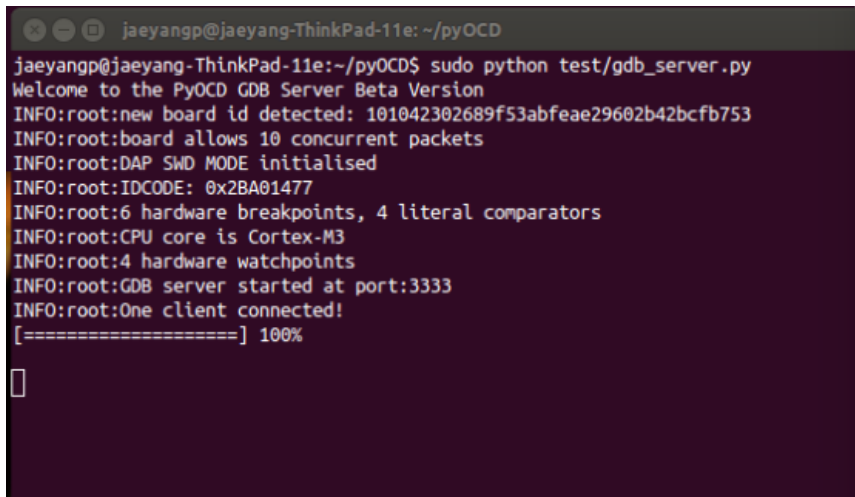
```

int main(void)
{
    int ch;
    pc.baud(9600);
    pc.printf("\r\nHello World!");
    while(1) {
        pc.printf("\r\nEnter:");
        ch = pc.getc();
        pc.putc(ch);
        DisplayLed(ch);
    }
}

```

Terminal 1:

```
$ sudo python test/gdb_server.py
```



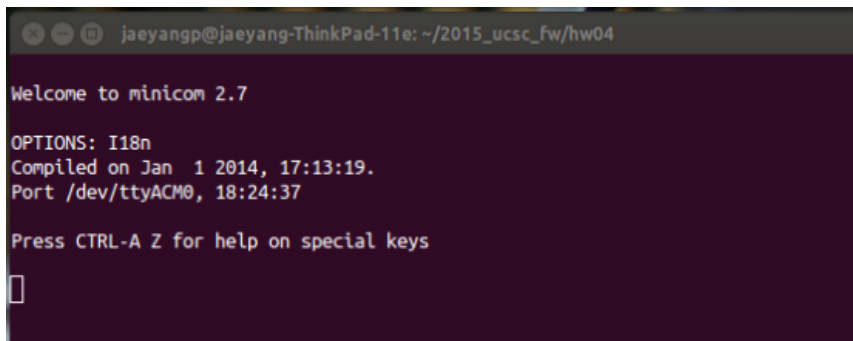
```

jaeyangp@jaeyang-ThinkPad-11e: ~/pyOCD
jaeyangp@jaeyang-ThinkPad-11e:~/pyOCD$ sudo python test/gdb_server.py
Welcome to the PyOCD GDB Server Beta Version
INFO:root:new board id detected: 101042302689f53abfeae29602b42bcfb753
INFO:root:board allows 10 concurrent packets
INFO:root:DAP SWD MODE initialised
INFO:root:IDCODE: 0x2BA01477
INFO:root:6 hardware breakpoints, 4 literal comparators
INFO:root:CPU core is Cortex-M3
INFO:root:4 hardware watchpoints
INFO:root:GDB server started at port:3333
INFO:root:One client connected!
[=====] 100%

```

Terminal 2: (minicom, 9600, n, 8, 1, /dev/ttyACM0)

```
$ sudo minicom
```



```

jaeyangp@jaeyang-ThinkPad-11e: ~/2015_ucsc_fw/hw04
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan  1 2014, 17:13:19.
Port /dev/ttyACM0, 18:24:37

Press CTRL-A Z for help on special keys

```

Terminal 3:

```
$ arm-none-eabi-gdb hw04.elf
```

```
jaeyangp@jaeyang-ThinkPad-11e: ~/2015_ucsc_fw/hw04
jaeyangp@jaeyang-ThinkPad-11e:~/2015_ucsc_fw/hw04$ arm-none-eabi-gdb hw04.elf
GNU gdb (GNU Tools for ARM Embedded Processors) 7.8.0.20150304-cvs
Copyright (C) 2014 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=x86_64-unknown-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 hw04.elf...done.
(gdb) target remote localhost:3333
Remote debugging using localhost:3333
0x00000e06 in serial_getc ()
(gdb) load
Loading section .text, size 0x8258 lma 0x0
Loading section .ARM.exidx, size 0x8 lma 0x8258
Loading section .data, size 0xb8 lma 0x8260
Start address 0x87c, load size 33560
Transfer rate: 7 KB/sec, 1766 bytes/write.
(gdb) b main
Breakpoint 1 at 0x248: file main.cpp, line 34.
(gdb) c
Continuing.
Note: automatically using hardware breakpoints for read-only addresses.

Breakpoint 1, main () at main.cpp:34
34          pc.baud(9600);
(gdb) █
```

```
jaeyangp@jaeyang-ThinkPad-11e: ~/2015_ucsc_fw/hw04
Welcome to minicon 2.7
OPTIONS: I18n
Compiled on Jan 1 2014, 17:13:19.
Port /dev/ttyACM0, 18:24:37
Press CTRL-A Z for help on special keys

Hello World!
Enter:a
Enter:█

INFO:root:IDCODE: 0x2BA01477
INFO:root:6 hardware breakpoints, 4 literal comparators
INFO:root:CPU core is Cortex-M3
INFO:root:4 hardware watchpoints
INFO:root:GDB server started at port:3333
INFO:root:One client connected!
[=====] 100%
█

jaeyangp@jaeyang-ThinkPad-11e: ~/2015_ucsc_fw/hw04
Transfer rate: 7 KB/sec, 1766 bytes/write.
(gdb) b main
Breakpoint 1 at 0x248: file main.cpp, line 34.
(gdb) c
Continuing.
Note: automatically using hardware breakpoints for read-only addresses.

Breakpoint 1, main () at main.cpp:34
34          pc.baud(9600);
(gdb) l
29          }
30      }
31
32      int main(void) {
33          int ch;
34          pc.baud(9600);
35          pc.printf("\r\nHello World!");
36          while(1) {
37              pc.printf("\r\nEnter:");
38              ch = pc.getc();
(gdb) n
32      int main(void) {
(gdb) n
34          pc.baud(9600);
(gdb) n
35          pc.printf("\r\nHello World!");
(gdb) n
37          pc.printf("\r\nEnter:");
(gdb) n
38              ch = pc.getc();
(gdb) n
39              pc.putc(ch);
(gdb) n
40              DisplayLed(ch);
(gdb) n
41          }
(gdb) n
37          pc.printf("\r\nEnter:");
(gdb) n
38              ch = pc.getc();
(gdb) █
```

The image shows two terminal windows. The left window is a minicom 2.7 session connected to a device. It displays 'Welcome to minicom 2.7', options like 'I18n', and a 'Hello World!' message. The user enters 'a', 'f', and 'f'. The right window is a GDB session for an ARM target. It shows the loading of 'HellWorld2.elf' and the execution of a program that prints 'Hello World!' and then enters an infinite loop.

```

jaeyangsp@jaeyang-ThinkPad-11e: ~/2015_ucsc_fw/hw04
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan  1 2014, 17:13:19.
Port /dev/ttyACM0, 18:24:37

Press CTRL-A Z for help on special keys

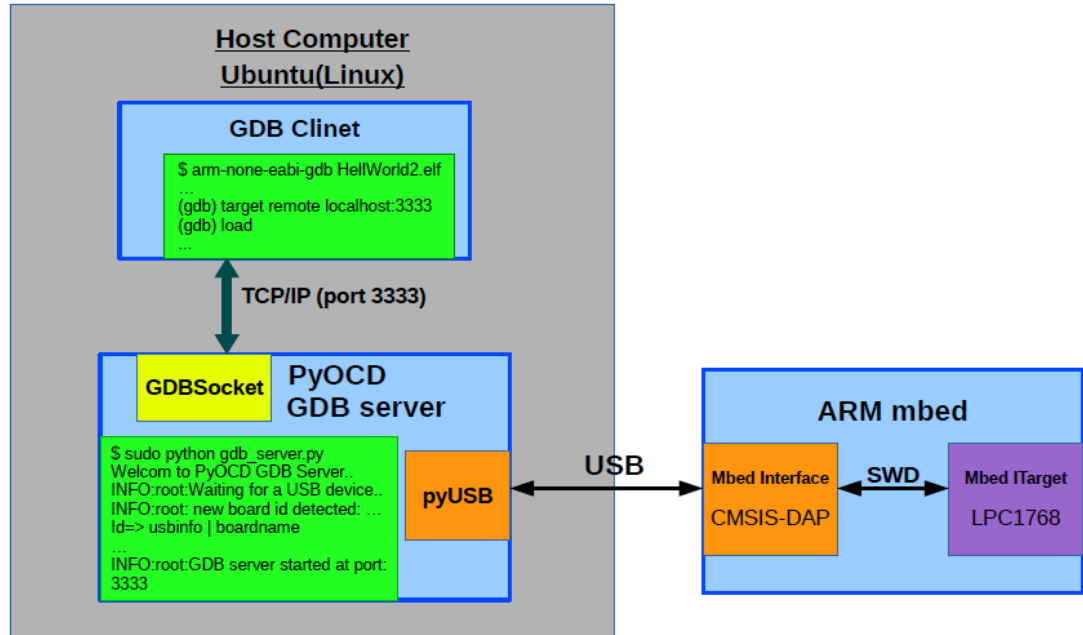
Hello World!
Enter: a
Enter:
Enter: f
Enter: f
Enter:

INFO:root:IDCODE: 0x2BA01477
INFO:root:6 hardware breakpoints, 4 literal comparators
INFO:root:CPU core is Cortex-M3
INFO:root:4 hardware watchpoints
INFO:root:GDB server started at port:3333
INFO:root:One client connected!
[=====] 100%
[]

jaeyangsp@jaeyang-ThinkPad-11e: ~/2015_ucsc_fw/hw04
13 if (ch>='0' && ch<='9')
(gdb) p/x ch
$3 = 0xd
(gdb) n
10 {
(gdb) n
13 if (ch>='0' && ch<='9')
(gdb) n
15 else if (ch>='A' && ch<='F') {
(gdb) n
18 } else if (ch>='a' && ch<='f') {
(gdb) n
20     ch+=10;
(gdb) n
22     ch=0;
(gdb) n
24     for (i=0; i<4; i++) {
(gdb) n
25         if(ch& (1<<i))
(gdb) n
26             ledArr[i] = 1;
(gdb) n
28             ledArr[i] = 0;
(gdb) n
24     for (i=0; i<4; i++) {
(gdb) n
28         ledArr[i] = 0;
(gdb) n
24     for (i=0; i<4; i++) {
(gdb) p/a ledArr[0]
$4 = {gpio = {pin = 0x2009c032, mask = 0x40000, reg_dir = 0x2009c020,
reg_set = 0x2009c038, reg_clr = 0x2009c03c, reg_in = 0x2009c034}}
(gdb) p/x ch
$5 = 0xd
(gdb) n
25 if(ch& (1<<i))
(gdb) n
26     ledArr[i] = 1;
(gdb) p/x ch
$6 = 0xd
(gdb)

```

What I learned from this:
 CMSIS-DAP, SWD, pyOCD, How to debugging Cortex-M



- Total spent hours: 3.5 hours**
- Python and pyOCD installation and test: 0.5
 - Debugging code with pyOCD: 1
 - CMSIS-DAP, pyOCD and GDB study: 1
 - Report: 1