HARDWARE/SOFTWARE CO-DESIGN Project Assignment 3

GETTING STARTED WITH BEAGLEBONE BLACK AND IMU SENSOR INTERFACING

Rakhat Yskak, Amantay Kapyshev, Ualikhan Tukenov

TASK 1. CONNECTING TO AND COMMUNICATING WITH BBB

We used Rakhat's laptop for this assignment since he uses Arch linux as the primary OS.

```
› ssh debian@192.168.7.2
Debian GNU/Linux 11

BeagleBoard.org Debian Bullseye IoT Image 2023-09-02
Support: https://bbb.io/debian
default username:password is [debian:temppwd]

debian@192.168.7.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Sep  2 18:05:38 2023 from 192.168.7.1
debian@BeagleBone:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
```

Checked if internet is connected to beagle board:

```
debian@BeagleBone:~$ ping -c 3 google.com
ping: google.com: Temporary failure in name resolution
```

NO, so configured internet-over-usb by running 'ip a' and identifying our main and internet over usb adapters:

```
› sudo iptables --table nat --append POSTROUTING --out-interface enp2s0 -j MASQUERADE
› sudo ip6tables --append FORWARD --in-interface enp2s0 -j ACCEPT
› sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

In beagle board:

```
debian@BeagleBone:~$ sudo route add default gw 192.168.7.1
[sudo] password for debian:
debian@BeagleBone:~$ sudo vim /etc/resolv.conf
```

```
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

# No DNS servers known.
# search .
nameserver 10.1.1.50
~
```

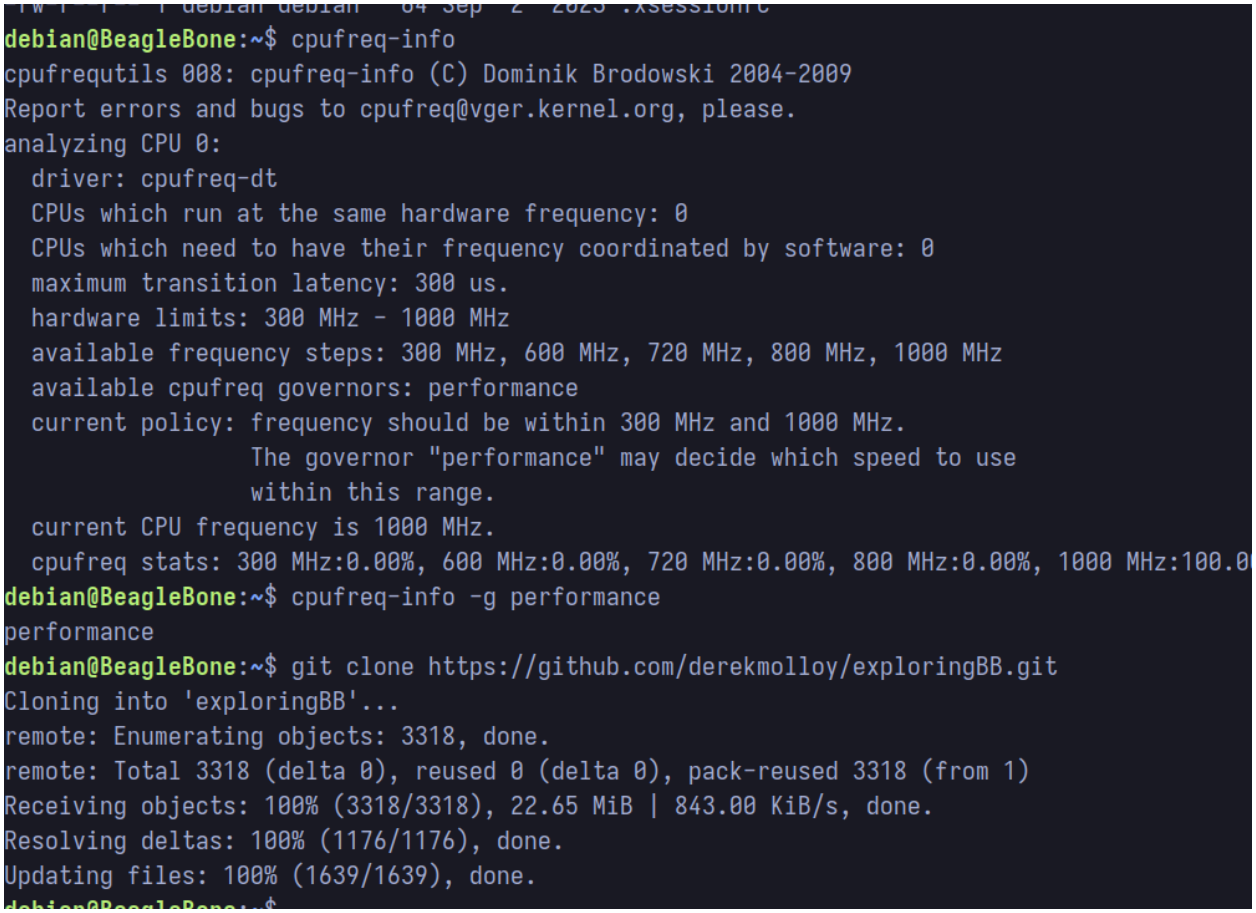
After that rechecked internet connection:

```
debian@BeagleBone:~$ ping google.com
PING google.com (216.58.209.14) 56(84) bytes of data.
From 10.27.10.1 (10.27.10.1) icmp_seq=4 Packet filtered
From 10.27.10.1 (10.27.10.1) icmp_seq=5 Packet filtered
From 10.27.10.1 (10.27.10.1) icmp_seq=6 Packet filtered
From 10.27.10.1 (10.27.10.1) icmp_seq=10 Packet filtered
^C
--- google.com ping statistics ---
10 packets transmitted, 0 received, +4 errors, 100% packet loss, time 9153ms
```

Internet is connected

TASK 2. BBB PROGRAMMING PRACTICE

We decided to just download git to beagleboard as textbook suggests in introduction section and we set cpu governor to performance which will auto decide which frequence to use.



```
debian@BeagleBone:~$ lla
-bash: lla: command not found
debian@BeagleBone:~$ ls -la
total 36
drwxr-xr-x  4 debian debian 4096 Feb 19 13:41 .
drwxr-xr-x  3 root   root   4096 Sep  2  2023 ..
-rw-------  1 debian debian  604 Feb 18 15:23 .bash_history
-rw-r--r--  1 debian debian  220 Mar 27  2022 .bash_logout
-rw-r--r--  1 debian debian 3526 Mar 27  2022 .bashrc
drwxr-xr-x  3 debian debian 4096 Sep  2  2023 .config
-rw-r--r--  1 debian debian    0 Sep  2  2023 .gitconfig
-rw-r--r--  1 debian debian  807 Mar 27  2022 .profile
-rw-r--r--  1 debian debian   64 Sep  2  2023 .xsessionrc
drwxr-xr-x 20 debian debian 4096 Feb 19 13:42 exploringBB
debian@BeagleBone:~$ cd exploringBB/chp05/overview
debian@BeagleBone:~/exploringBB/chp05/overview$ _
```

Running helloworld.c

```
debian@BeagleBone:~/exploringBB/chp05/overview$ gcc helloworld.c -o helloworldc
debian@BeagleBone:~/exploringBB/chp05/overview$ ./helloworldc
Hello World!
debian@BeagleBone:~/exploringBB/chp05/overview$ _
```

Running helloworld.cpp

```
debian@BeagleBone:~/exploringBB/chp05/overview$ g++ helloworld.cpp -o helloworldcpp
debian@BeagleBone:~/exploringBB/chp05/overview$ ./helloworldcpp
Hello World!
debian@BeagleBone:~/exploringBB/chp05/overview$ _
```

Running pointers.c

contents:

zsh in rakhat

```c
#include<stdio.h>

int main(){
   int y = 1000;
   int *p;
   p = &y;
   printf("The variable has value %d and the address %p.\n", y, &y);
   printf("The pointer stores %p and points at value %d.\n", p, *p);
   printf("The pointer itself has address %p and size %d.\n", &p, sizeof(p));
   return 0;
}
```

```
debian@BeagleBone:~/exploringBB/chp05/overview$ gcc pointers.c -o pointersc
debian@BeagleBone:~/exploringBB/chp05/overview$ ./pointersc
The variable has value 1000 and the address 0xbead553c.
The pointer stores 0xbead553c and points at value 1000.
The pointer itself has address 0xbead5538 and size 4.
debian@BeagleBone:~/exploringBB/chp05/overview$ _
```

Running cppstrings.cpp

contents:

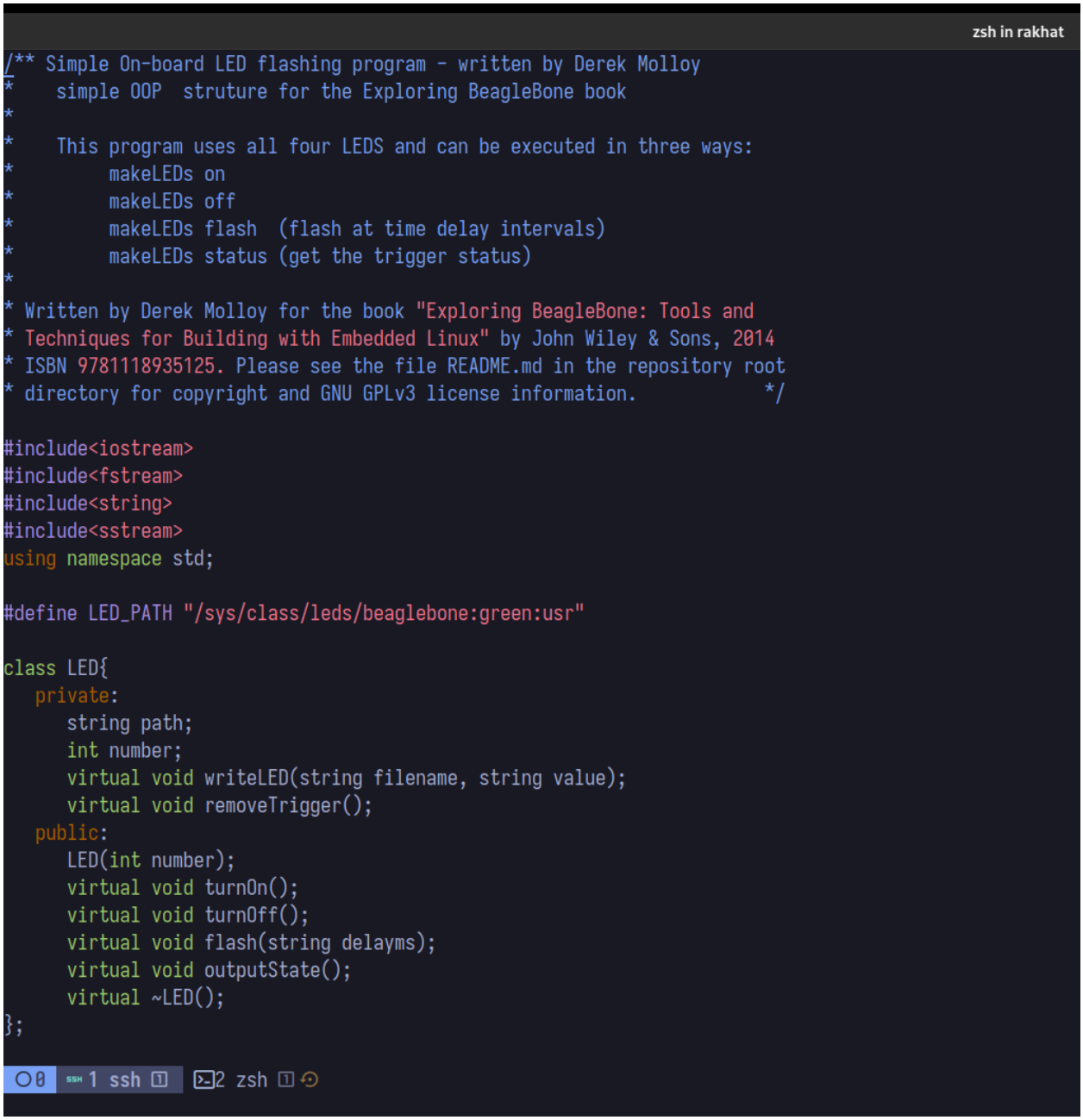
zsh in rakhat

```
#include<iostream>
#include<sstream>    // to tokenize the string
//#include<cstring>  // how to include a c header if needed
using namespace std;

int main(){
   string a = "hello ";
   char temp[] = {'w','o','r','l','d','!','\0'};  //the \0 is important!
   string b(temp);

   a[0]='H';
   string c = a + b;
   cout << "The string c is: " << c << endl;
   cout << "The length of c is: " << c.length() << endl;

   int loc = c.find_first_of('w');
   c.replace(loc,1,1,'W');
   cout << "The string c is now: " << c << endl;

   if (string("cat")< string("dog")){
      cout << "cat comes before dog (lexiographically)\n";
   }
   c.insert(5," to the");
   cout << "The c string is now: " << c << endl;

   //tokenize string using spaces - could use Boost.Tokenizer
   // or C++11 to improve syntax. Using stringstream this time.
   stringstream ss;
   ss << c;  // put the c string on the stringstream
   string token;
   while(getline(ss, token, ' ')){
      cout << "Token: " << token << endl;
   }
   return 0;
}
```


debian@BeagleBone:~/exploringBB/chp05/overview$ g++ cppstrings.cpp -o cppstrings
./cppstringsdebian@BeagleBone:~/exploringBB/chp05/overview$ ./cppstrings
The string c is: Hello world!
The length of c is: 12
The string c is now: Hello World!
cat comes before dog (lexiographically)
The c string is now: Hello to the World!
Token: Hello
Token: to
Token: the
Token: World!
debian@BeagleBone:~/exploringBB/chp05/overview$ _
1 ssh
2 zsh

Running makeLEDs.cpp

contents:



building:

```
debian@BeagleBone:~/exploringBB/chp05/makeLEDOOP$ ls -la
total 40
drwxr-xr-x  2 debian debian  4096 Feb 19 13:55 .
drwxr-xr-x 19 debian debian  4096 Feb 19 13:42 ..
-rwxr-xr-x  1 debian debian   121 Feb 19 13:42 build
-rwxr-xr-x  1 debian debian 20996 Feb 19 13:55 makeLEDs
-rw-r--r--  1 debian debian  3061 Feb 19 13:42 makeLEDs.cpp
debian@BeagleBone:~/exploringBB/chp05/makeLEDOOP$ ./build
Exploring BeagleBone - Building the makeLEDs application
Finished
debian@BeagleBone:~/exploringBB/chp05/makeLEDOOP$ ./makeLEDs status
Starting the makeLEDs program
none usb-gadget usb-host rfkill-any rfkill-none kbd-scrolllock kbd-numloc
llock kbd-ctrlrlock timer oneshot disk-activity disk-read disk-write ide-
 4a101000.mdio:00:100Mbps 4a101000.mdio:00:10Mbps
none usb-gadget usb-host rfkill-any rfkill-none kbd-scrolllock kbd-numloc
llock kbd-ctrlrlock timer oneshot disk-activity disk-read disk-write ide-
 4a101000.mdio:00:100Mbps 4a101000.mdio:00:10Mbps
none usb-gadget usb-host rfkill-any rfkill-none kbd-scrolllock kbd-numloc
llock kbd-ctrlrlock timer oneshot disk-activity disk-read disk-write ide-
 4a101000.mdio:00:100Mbps 4a101000.mdio:00:10Mbps
none usb-gadget usb-host rfkill-any rfkill-none kbd-scrolllock kbd-numloc
llock kbd-ctrlrlock timer oneshot disk-activity disk-read disk-write ide-
 4a101000.mdio:00:100Mbps 4a101000.mdio:00:10Mbps
Finished the makeLEDs program
destroying the LED with path: /sys/class/leds/beaglebone:green:usr3
destroying the LED with path: /sys/class/leds/beaglebone:green:usr2
destroying the LED with path: /sys/class/leds/beaglebone:green:usr1
destroying the LED with path: /sys/class/leds/beaglebone:green:usr0
```

running:

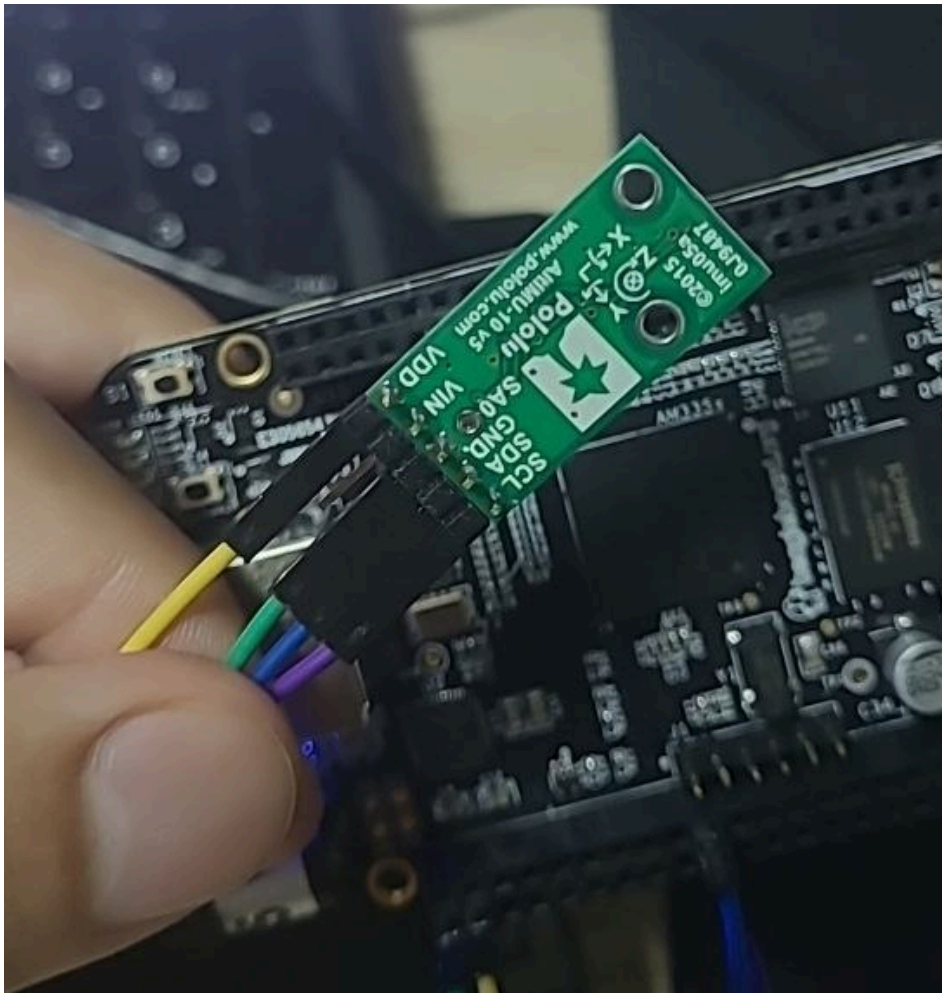
```
destroying the LED with path: /sys/class/leds/beaglebone:green:usr3
destroying the LED with path: /sys/class/leds/beaglebone:green:usr2
destroying the LED with path: /sys/class/leds/beaglebone:green:usr1
destroying the LED with path: /sys/class/leds/beaglebone:green:usr0
debian@BeagleBone:~/exploringBB/chp05/makeLEDOOP$ sudo ./makeLEDs flash
[sudo] password for debian:
Starting the makeLEDs program
Making LED0 flash.
Making LED1 flash.
Making LED2 flash.
Making LED3 flash.
Finished the makeLEDs program
destroying the LED with path: /sys/class/leds/beaglebone:green:usr3
destroying the LED with path: /sys/class/leds/beaglebone:green:usr2
destroying the LED with path: /sys/class/leds/beaglebone:green:usr1
destroying the LED with path: /sys/class/leds/beaglebone:green:usr0
debian@BeagleBone:~/exploringBB/chp05/makeLEDOOP$ sudo ./makeLEDs off
Starting the makeLEDs program
Turning LED0 off.
Turning LED1 off.
Turning LED2 off.
Turning LED3 off.
Finished the makeLEDs program
destroying the LED with path: /sys/class/leds/beaglebone:green:usr3
destroying the LED with path: /sys/class/leds/beaglebone:green:usr2
destroying the LED with path: /sys/class/leds/beaglebone:green:usr1
destroying the LED with path: /sys/class/leds/beaglebone:green:usr0
debian@BeagleBone:~/exploringBB/chp05/makeLEDOOP$ _
 0  1 ssh   2 zsh
```

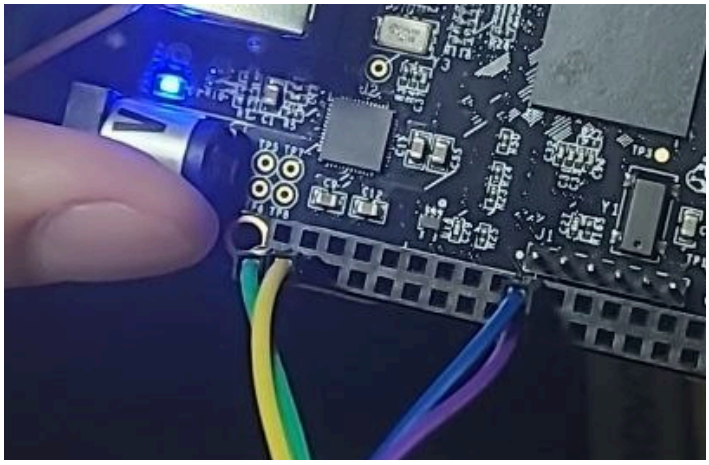
Restoring leds:

Video

TASK 4. IMU INTERFACING

2. connect the IMU to the BBB
SCL – P9_19
SDA – P9_20
GND – P9_1
VIN – disconnected
VDD – P9_3





3. Follow the textbook (pp. 280-283) and test i2c-tools

```
debian@BeagleBone:~$ sudo apt-get install i2c-tools

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
i2c-tools is already the newest version (4.2-1+b1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
debian@BeagleBone:~$ i2cdetect -l
i2c-1   i2c             OMAP I2C adapter                        I2C adapter
i2c-2   i2c             OMAP I2C adapter                        I2C adapter
i2c-0   i2c             OMAP I2C adapter                        I2C adapter
```

By probing we can see that IMU is on 3rd:

The LSM6DS33 and LIS3MDL each have separate slave addresses on the I²C bus. The following table shows the slave addresses of the sensors:

Sensor	Slave Address (default)
LSM6DS33 (gyro and accelerometer)	1101011b (0x6B)
LIS3MDL (magnetometer)	0011110b (0x1E)

```
debian@BeagleBone:~$ i2cdetect -y -r 2
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- 1e --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- 5d -- --
60: -- -- -- -- -- -- -- -- -- -- -- 6b -- -- -- --
70: -- -- -- -- -- -- -- --
debian@BeagleBone:~$
```

4. Value 2 is used because of P9_19 and P9_20 which correspond to 2nd (indexing from 0) bus

```
debian@BeagleBone:~$ i2cdump -y 2 0x6b
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 69    ...............i
10: 00 00 04 00 00 00 00 00 38 38 00 00 00 00 00 bb    ..?.....88.....?
20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
30: 00 00 00 00 00 00 00 00 00 00 00 10 00 00 ff ff    ............?....
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
60: 00 00 00 00 00 ff 00 00 00 00 00 00 00 00 00 00    ................
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
debian@BeagleBone:~$ _
```

5. Configuration registers must be set to the normal mode of operation.

LSM6DS33		LIS3MDL	
Register (address)	Control word	Register	Control word
CTRL1_XL (0x10)	0x30	CTRL_REG1 (0x20)	0x0C
CTRL2_G (0x11)	0x30	CTRL_REG2 (0x21)	0x00
CTRL7_G (0x16)	0x00		
CTRL8_XL (0x17)	0x00		
CTRL9_XL (0x18)	0x38		

CTRL1_XL (0x10):

is responsible for setting the frequency at which the accelerometer samples data

```
debian@BeagleBone:~$ i2cset -y 2 0x6b 0x10 0x30
debian@BeagleBone:~$ i2cget -y 2 0x6b 0x10
0x30
```

We need 50Hz and ±2g which is

ODR_XL 0011 = 52Hz
We can't get exactly 50Hz so best match is 52Hz
FS_XL 00 = ±2g
BW_XL we dont care so we left default 00

Which makes
0011 | 00 | 00 in base 2
30 in base 16

CTRL2_G (0x11)

Is responsible for setting the frequency at which the gyroscope samples data

```
debian@BeagleBone:~$ i2cset -y 2 0x6b 0x11 0x30
debian@BeagleBone:~$ i2cget -y 2 0x6b 0x11
0x30
```

We need 50Hz and 245 deg/sec which is

ODR_G 0011 = 52Hz
We can't get exactly 50Hz so best match is 52Hz
FS_G 00 = 245 dps
FS_125 we don't care so we left default 0

Which makes
0011 | 00 | 0 | 0 in base 2
30 in base 16

CTRL7_G (0x16)

is mainly used for power mode configuration and filtering settings

```
debian@BeagleBone:~$ i2cget -y 2 0x6B 0x16
0x00
debian@BeagleBone:~$
```

Since we are using normal mode, we keep HR = 0, LP_EN = 0.
We do not enable high-pass filtering HP_EN = 0
We leave reserved bits as 0.
So we set to the default value of 0x00 which is standard settings with no special filtering.

CTRL8_XL (0x17)

is responsible for configuring low-pass and high-pass filtering in the accelerometer

```
debian@BeagleBone:~$ i2cget -y 2 0x6B 0x17
0x00
debian@BeagleBone:~$
```

The default value is 0x00 (0b00000000) meaning

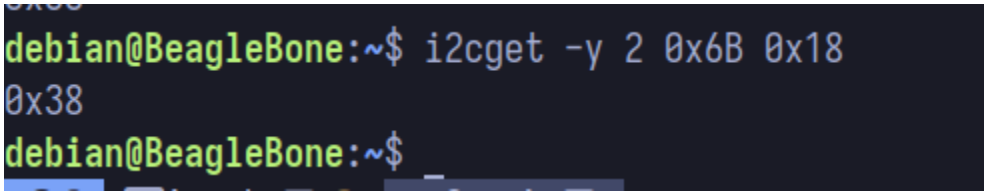
- no high-pass filtering HP_SLOPE_XL_EN = 0

- no low-pass filtering LPF2_XL_EN = 0
- no fast-settling mode FASTSETTL_MODE = 0

We want to keep things simple for normal operation, so we'll leave it at default (0x00), unless filtering is needed.

CTRL9_XL (18h)

is used to enable/disable individual axes of the accelerometer and to control advanced features
We need to enable all - X, Y, Z - axes of accelerometer
I.e set
Xen_XL = 1
Yen_XL = 1
Zen_XL = 1
Which is
00111000 in binary
And 38 in hex
It is already set, so we skip.



CTRL_REG1 (0x20)

configures the magnetometer data rate (ODR), operating mode, and temperature compensation.
We need:

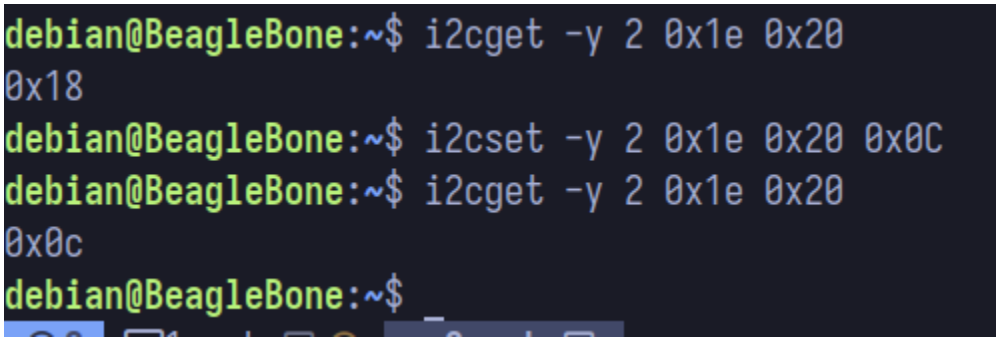
ODR = 6.25Hz
But there is no such configuration, the best match is 5hz
So DO2, DO1, DO0 = 011

X and Y axes operating mode selection:
OM1, OM0 = 00
Low-power mode

Temperature sensor disabled TEMP_EN = 0

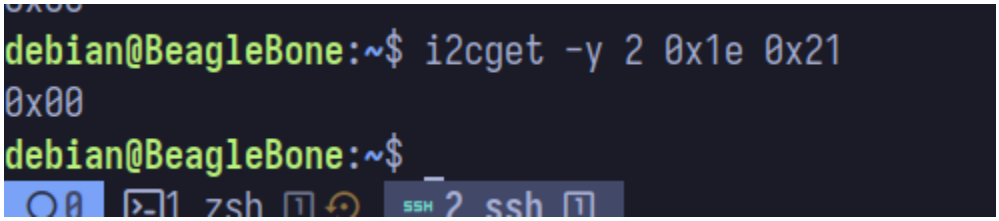
Self-test disabled ST = 0
Fast ODR disabled Fast_ODR = 0

So binary is 00001100 which is 0x0C

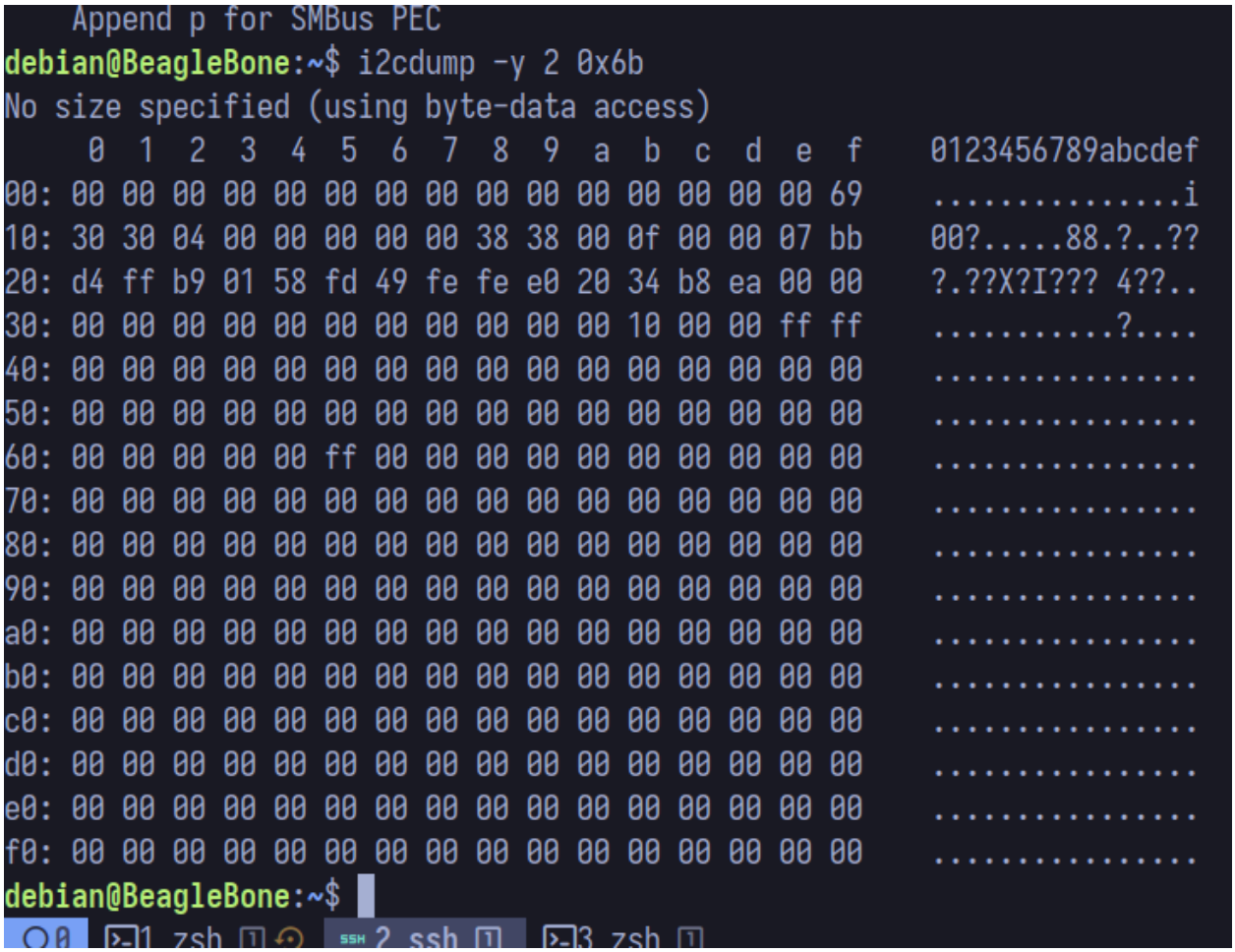


CTRL_REG2 (0x21)

configures measurement range of the magnetometer
We need ±4 *gauss* so FS = 00, and all remaining values to 0
So 00000000 is 0x00



Results


Append p for SMBus PEC
debian@BeagleBone:~$ i2cdump -y 2 0x6b
No size specified (using byte-data access)
0 1 2 3 4 5 6 7 8 9 a b c d e f 0123456789abcdef
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 69 i
10: 30 30 04 00 00 00 00 00 38 38 00 0f 00 00 07 bb 00?.....88.?..??
20: d4 ff b9 01 58 fd 49 fe fe e0 20 34 b8 ea 00 00 ?.??X?I??? 4??..
30: 00 00 00 00 00 00 00 00 00 00 00 10 00 00 ff ff .....?....
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
60: 00 00 00 00 00 ff 00 00 00 00 00 00 00 00 00 00 ....
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ....
debian@BeagleBone:~$
0 1 zsh 2 ssh 3 zsh

```
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
debian@BeagleBone:~$ i2cdump -y 2 0x1e
No size specified (using byte-data access)
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f    0123456789abcdef
00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3d    ...............=
10: 14 05 40 77 55 3a 01 44 92 1a 16 a0 00 00 00 00    ??@wU:?D????....
20: 0c 00 03 00 00 00 00 00 22 00 d8 f2 44 ed 00 00    ?.?.....".??D?..
30: e8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ?...............
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3d    ...............=
90: 14 05 40 77 55 3a 01 44 92 1a 16 a0 00 00 00 00    ??@wU:?D????....
a0: 0c 00 03 00 00 00 00 00 22 00 d8 f2 44 ed 00 00    ?.?.....".??D?..
b0: e8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ?...............
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
debian@BeagleBone:~$
 0  1 zsh   ssh 2 ssh   3 zsh
```

6. Writing program

```
debian@BeagleBone:~$ gcc ./testAltIMU.c -o testAltIMU
debian@BeagleBone:~$ ./testAltIMU
Starting the AltuIMU sensor application
The Device ID is: 6B (LSM6DS33)
X=440 Y=-671 Z=-435 (Gyroscope)
X=-8008 Y=13304 Z=-5356 (Accelerometer)
X=34 Y=-3368 Z=-4796 (Magnetometer)
debian@BeagleBone:~$
 0  1 zsh   ssh 2 ssh   3 zsh
```

7. Register Map

https://www.pololu.com/file/0J1087/LSM6DS33.pdf

ADDR	Name	Type	Description
0x1E		—	I2C device address
0x28	OUTX_L_M	R	Low byte of X-axis magnetometer data
0x29	OUTX_H_M	R	High byte of X-axis magnetometer data
0x2A	OUTY_L_M	R	Low byte of Y-axis magnetometer data
0x2B	OUTY_H_M	R	High byte of Y-axis magnetometer data
0x2C	OUTZ_L_M	R	Low byte of Z-axis magnetometer data
0x2D	OUTZ_H_M	R	High byte of Z-axis magnetometer data
0x20	CTRL_REG1	R/W	Control register 1
0x21	CTRL_REG2	R/W	Control register 2

https://www.pololu.com/file/0J1089/LIS3MDL.pdf

ADDR	Name	Type	Description

0x6B		—	I2C device address (not an internal register)
0x0F	WHO_AM_I	R	Device identification register
0x10	CTRL1_XL	R/W	Accelerometer control register
0x11	CTRL2_G	R/W	Gyroscope control register
0x16	CTRL7_G	R/W	Additional gyroscope control register
0x17	CTRL8_XL	R/W	Additional accelerometer control register
0x18	CTRL9_XL	R/W	Additional accelerometer control register
0x22	OUTX_L_G	R	Low byte of X-axis gyroscope data
0x23	OUTX_H_G	R	High byte of X-axis gyroscope data
0x24	OUTY_L_G	R	Low byte of Y-axis gyroscope data
0x25	OUTY_H_G	R	High byte of Y-axis gyroscope data
0x26	OUTZ_L_G	R	Low byte of Z-axis gyroscope data
0x27	OUTZ_H_G	R	High byte of Z-axis gyroscope

			data
0x28	OUTX_L_XL	R	Low byte of X-axis accelerometer data
0x29	OUTX_H_XL	R	High byte of X-axis accelerometer data
0x2A	OUTY_L_XL	R	Low byte of Y-axis accelerometer data
0x2B	OUTY_H_XL	R	High byte of Y-axis accelerometer data
0x2C	OUTZ_L_XL	R	Low byte of Z-axis accelerometer data
0x2D	OUTZ_H_XL	R	High byte of Z-axis accelerometer data

TASK 5. IMU SIGNAL PROCESSING

We need to convert the raw sensor data into physical units before feeding them into the filter.

Acceleration is measured in m/s^2, we set full-scale range is set to ±2g, the sensitivity is LA_So = 0.061 mg/LSB from https://www.pololu.com/file/0J1087/LSM6DS33.pdf, page 15, 1 mg = 9.81 * 10^(-3). So, the scaling factor for acceleration 0.061 * 9.81 * 10^(-3);

Gyroscope is set to the full-scale range of ±245 dps, the sensitivity is G_So = 8.75 mdps/LSB from same page. Then, the scaling factor is 8.75 mdps/LSB -> 8.75 * 10^(−3) dps/LSB

Magnetometer is set to the full-scale range of ±4 gauss, and the sensitivity is 6842 LSB/gauss. Therefore, the scaling factor is 1 / 6842 gauss/LSB = 0.0001461 gauss/LSB

The code can be found here: https://github.com/4ry1337/embedded-imu/blob/main/task5/quaternions.c

TASK 6. IMU GUI DEVELOPMENT

Demo here: https://drive.google.com/file/d/1uk7avvDPi2LYZR5nVTGpJwIZ3MtXKgtJ/view?usp=sharing
The code can be found here: https://github.com/4ry1337/embedded-imu