MobileDeluge User Guide

1. Introduction

MobileDeluge is an over-the-air programming tool developed based on Deluge.

1.1 Directory structure

```
   /Deluge:

      Code from the original Deluge, with very little modification on
“ObjectTransferP.nc” to add led blinking when transmitting the code image.

   /SimpleDrip:

      A replacement of the original Drip.

   /SimpleGateway:

      A general gateway to control the reprogramming process

   /tests:

      Some test applications for SimpleDrip (use it with Deluge).

   /Reprogrammer

      MobileDeluge control layer, on top of Deluge.

   .../DelugeBase

         The MobileBase for MobileDeluge.
```

1.2 Reprogrammer

[TODO]

1.3 Gateway

[TODO]

1.4 SimpleDrip

[TODO]

2. Prerequisite

Before Deluge (and hence MobileDeluge) can be used, a few tools must be set up.

Note: the following commands are for Linux based systems. If you use Mac, the commands can be slightly different. But the tools to be installed are the same.

2.1 Install a few python scripts

Run the following commands in the terminal. Suppose TinyOS root directory is `TOSROOT` (i.e., `/……/tinyos-2_1_2`).

```
% cd $TOSROOT/tools
% ./Bootstrap
  ...
% ./configure
  ...
% cd tinyos/misc
% make; make install
```

If you get any errors about "`aclocal`" or similar, you will need to install "`autoconf`" and "`automake`" tools by running the following command on terminal.

```
% sudo apt-get install automake autoconf
```

2.2 Install Python serial tool

For Linux (e.g. Xubuntu, Ubuntu) you should already have this tool installed. If not, run this command on terminal.

```
% sudo apt-get install python-serial
```

For Windows + Cygwin, download the package from this link:

```
http://pyserial.sourceforge.net/
```

and install.

2.3 Compile Bootloader for different mote platforms

In Deluge, Bootloader is running on each mote to reprogramming itself. It loads the new image from the flash memory to the memory, and reboots the mote. After this process, the mote is running the new application. Run the following commands in terminal to compile the Bootloader for each mote platform (`$TOSDIR` is: `/……/tinyos-2_1_2/tos`):

```
% cd $TOSDIR/lib/tosboot

% make micaz           # for micaz platform

% make telosb

% make iris
```

After all these are done, Deluge (and MobileDeluge) should work properly. If interested, please refer to this link for more information:

```
http://tinyos.stanford.edu/tinyos-wiki/index.php/Deluge_T2
```

3. Quick Start

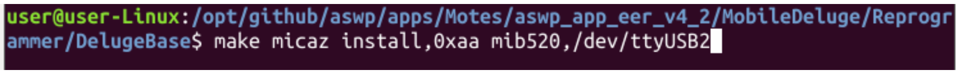
3.1 Preparation

- Use **MicaZ** mote as the DelugeBase.
- Make sure the target motes have voltage higher than **2.7** Volts.
- A TelosB mote as the sniffer (not necessary but preferred).
- Start MobileGateway.

3.1.1 Install DelugeBase

Suppose the serial port of Mib520 board is /dev/ttyUSB2 and /dev/ttyUSB3.

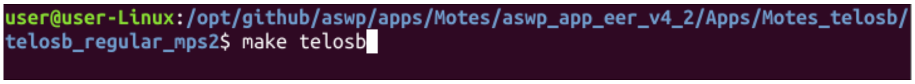
Go to /MobileDeluge/Reprogrammer/DelugeBase and install it to the MicaZ mote.



```
user@user-Linux:/opt/github/aswp/apps/Motes/aswp_app_eer_v4_2/MobileDeluge/Reprogr
ammer/DelugeBase$ make micaz install,0xaa mib520,/dev/ttyUSB2
```

3.1.2 Prepare the new application image

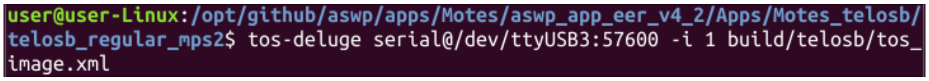
Go to the folder of the new application. For example, suppose the folder is: aswp_app_eer_v4_2/Apps/Motes_telosb/telosb_regular_mps2, compile the application and inject it to the DelugeBase.



```
user@user-Linux:/opt/github/aswp/apps/Motes/aswp_app_eer_v4_2/Apps/Motes_telosb/
telosb_regular_mps2$ make telosb
```

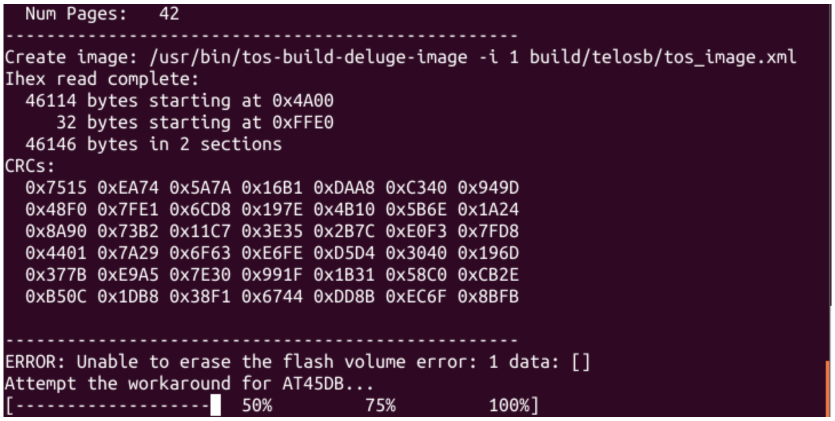
Then, use the following command to inject the new application.

```
% tos-deluge serial@/dev/ttyUSB3:57600 –i 1 build/telosb/tos_image.xml
```



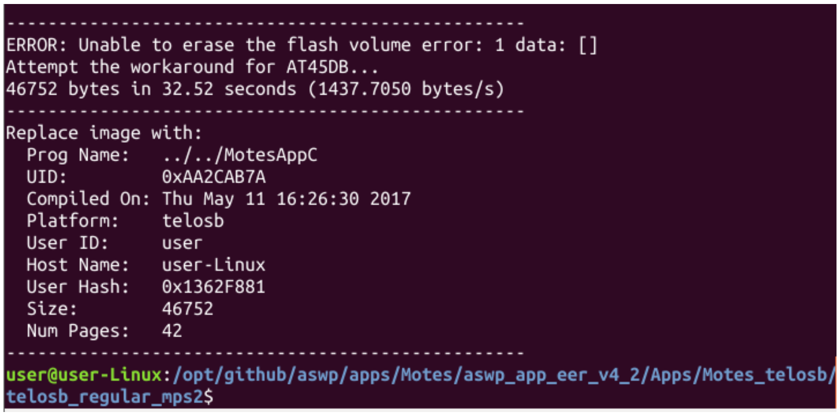
```
user@user-Linux:/opt/github/aswp/apps/Motes/aswp_app_eer_v4_2/Apps/Motes_telosb/
telosb_regular_mps2$ tos-deluge serial@/dev/ttyUSB3:57600 -i 1 build/telosb/tos_
image.xml
```

You can see the process of the injection as following:



```
  Num Pages:   42
--------------------------------------------------
Create image: /usr/bin/tos-build-deluge-image -i 1 build/telosb/tos_image.xml
Ihex read complete:
  46114 bytes starting at 0x4A00
     32 bytes starting at 0xFFE0
  46146 bytes in 2 sections
CRCs:
  0x7515 0xEA74 0x5A7A 0x16B1 0xDAA8 0xC340 0x949D
  0x48F0 0x7FE1 0x6CD8 0x197E 0x4B10 0x5B6E 0x1A24
  0x8A90 0x73B2 0x11C7 0x3E35 0x2B7C 0xE0F3 0x7FD8
  0x4401 0x7A29 0x6F63 0xE6FE 0xD5D4 0x3040 0x196D
  0x377B 0xE9A5 0x7E30 0x991F 0x1B31 0x58C0 0xCB2E
  0xB50C 0x1DB8 0x38F1 0x6744 0xDD8B 0xEC6F 0x8BFB

--------------------------------------------------
ERROR: Unable to erase the flash volume error: 1 data: []
Attempt the workaround for AT45DB...
[-------------------█  50%         75%         100%]
```

When finished, you can see something like this:



```
--------------------------------------------------
ERROR: Unable to erase the flash volume error: 1 data: []
Attempt the workaround for AT45DB...
46752 bytes in 32.52 seconds (1437.7050 bytes/s)
--------------------------------------------------
Replace image with:
  Prog Name:   ../../MotesAppC
  UID:         0xAA2CAB7A
  Compiled On: Thu May 11 16:26:30 2017
  Platform:    telosb
  User ID:     user
  Host Name:   user-Linux
  User Hash:   0x1362F881
  Size:        46752
  Num Pages:   42
--------------------------------------------------
user@user-Linux:/opt/github/aswp/apps/Motes/aswp_app_eer_v4_2/Apps/Motes_telosb/
telosb_regular_mps2$
```

3.1.3 Prepare the sniffer

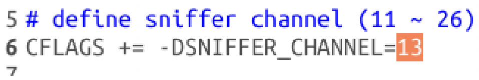
(Suppose the serial port of the TelosB mote is `/dev/ttyUSB1`)

Sniffer can hear all the radio transmission in the environment. It can be used to monitor the dissemination process of the application from DelugeBase and the target mote.

First, build the C serial forwarder in TinyOS (i.e., TinyOS serial library `libmote.a`):

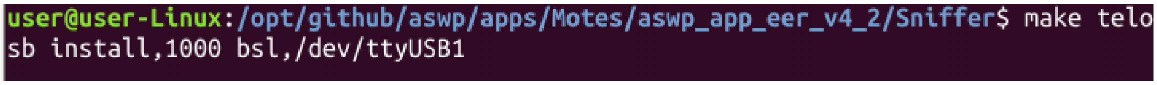
```
% cd $TOSROOT/support/sdk/c/sf
% ./bootstrap
% ./configure
% make
```

Go to `aswp_app_eer_v4_2/Sniffer`, edit the value of `SNIFFER_CHANNEL` in `Makefile` to select a channel to be monitored. In the following we monitor channel 13.



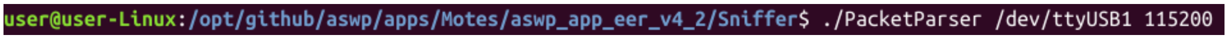
```
# define sniffer channel (11 ~ 26)
CFLAGS += -DSNIFFER_CHANNEL=13
```

Install this application to a TelosB mote:



```
user@user-Linux:/opt/github/aswp/apps/Motes/aswp_app_eer_v4_2/Sniffer$ make telo
sb install,1000 bsl,/dev/ttyUSB1
```

Then, run the following command to start the parser of sniffer application:



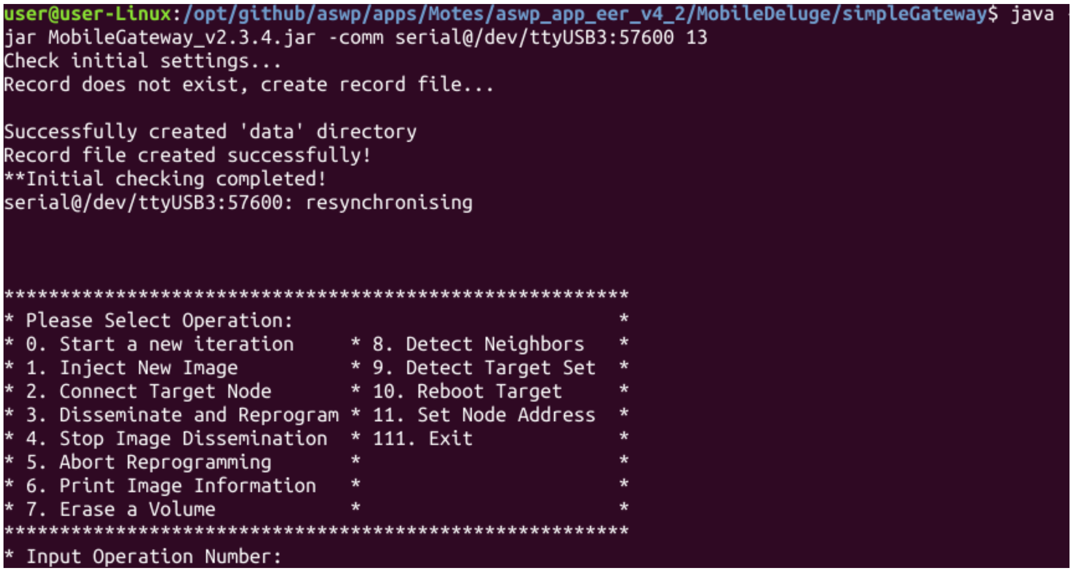
```
user@user-Linux:/opt/github/aswp/apps/Motes/aswp_app_eer_v4_2/Sniffer$ ./PacketParser /dev/ttyUSB1 115200
```

3.1.4 Start MobileGateway

Go to `/MobileDeluge/simpleGateway/` and run:

```
%  java -jar MobileGateway_v2.3.4.jar -comm serial@/dev/ttyUSB3:57600 13


   ## usage: java -jar gatewayjar.jar -comm [serial source] [new channel]
```

When it starts successfully, you will see the following:

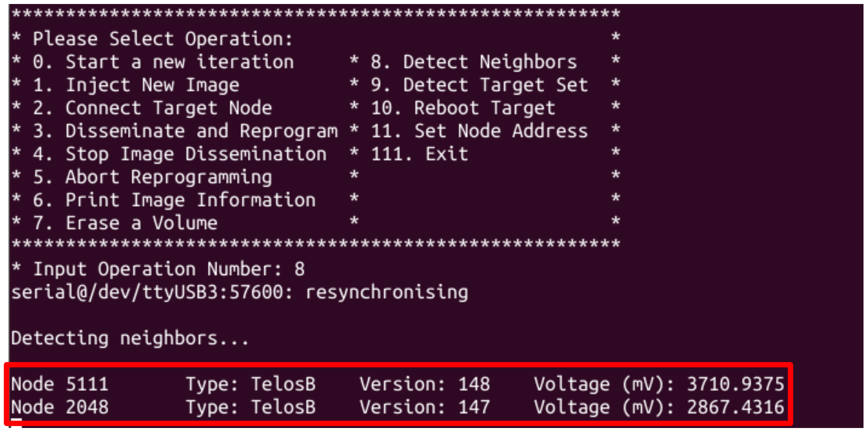


```
user@user-Linux:/opt/github/aswp/apps/Motes/aswp_app_eer_v4_2/MobileDeluge/simpleGateway$ java
jar MobileGateway_v2.3.4.jar -comm serial@/dev/ttyUSB3:57600 13
Check initial settings...
Record does not exist, create record file...

Successfully created 'data' directory
Record file created successfully!
**Initial checking completed!
serial@/dev/ttyUSB3:57600: resynchronising



********************************************************
* Please Select Operation:                             *
* 0. Start a new iteration     * 8. Detect Neighbors   *
* 1. Inject New Image          * 9. Detect Target Set  *
* 2. Connect Target Node       * 10. Reboot Target     *
* 3. Disseminate and Reprogram * 11. Set Node Address  *
* 4. Stop Image Dissemination  * 111. Exit             *
* 5. Abort Reprogramming       *                       *
* 6. Print Image Information   *                       *
* 7. Erase a Volume            *                       *
********************************************************
* Input Operation Number:
```

Now we can start to use the MobileDeluge tool to reprogram the motes and discover the motes nearby.

3.2 Detect neighbors

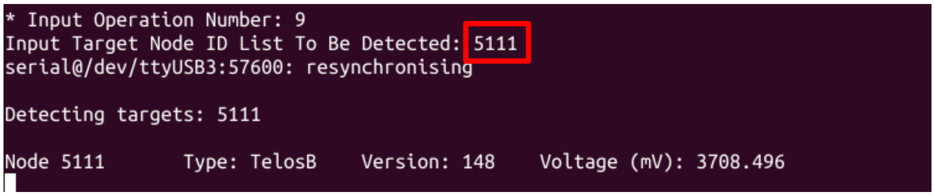
In the MobileGateway terminal, enter **“8”** and you will see the node nearby:



```
**********************************************************
* Please Select Operation:                               *
* 0. Start a new iteration     * 8. Detect Neighbors     *
* 1. Inject New Image          * 9. Detect Target Set    *
* 2. Connect Target Node       * 10. Reboot Target       *
* 3. Disseminate and Reprogram * 11. Set Node Address    *
* 4. Stop Image Dissemination  * 111. Exit               *
* 5. Abort Reprogramming       *                         *
* 6. Print Image Information   *                         *
* 7. Erase a Volume            *                         *
**********************************************************
* Input Operation Number: 8
serial@/dev/ttyUSB3:57600: resynchronising

Detecting neighbors...

Node 5111       Type: TelosB    Version: 148    Voltage (mV): 3710.9375
Node 2048       Type: TelosB    Version: 147    Voltage (mV): 2867.4316
```

Wait about 10 seconds to let all nearby motes reply. Press `Enter` to show the menu again.

You can also use option **“9”** to detect specific target motes (use “**space**” to separate mote IDs):

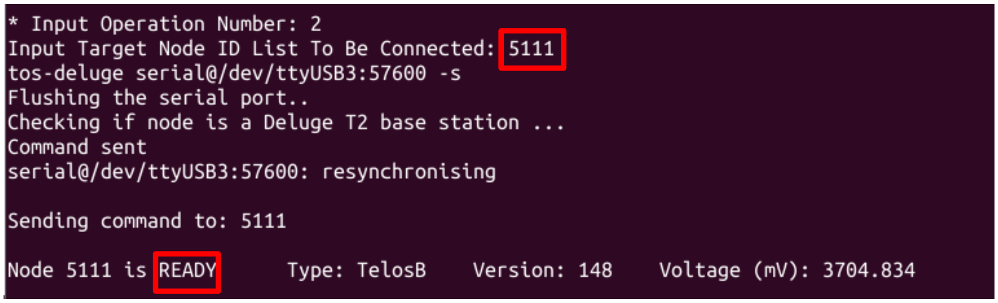


```
* Input Operation Number: 9
Input Target Node ID List To Be Detected: 5111
serial@/dev/ttyUSB3:57600: resynchronising

Detecting targets: 5111

Node 5111       Type: TelosB    Version: 148    Voltage (mV): 3708.496
```

If you see nothing, then you can reboot the DelugeBase and the MobileGateway, and try again.

3.4 Connect to the target motes

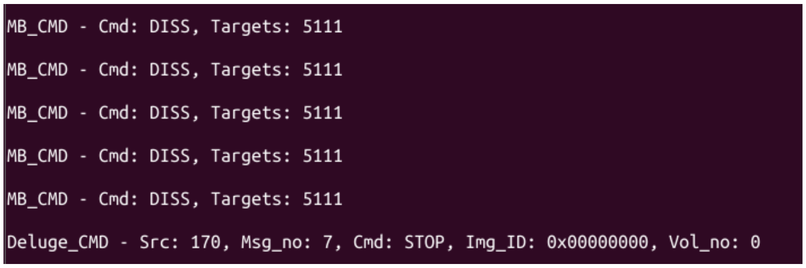
Input option **“2”** to connect to the target motes for programming. Similar to option **“9”**, use “**space**” to separate multiple target IDs:



```
* Input Operation Number: 2
Input Target Node ID List To Be Connected: 5111
tos-deluge serial@/dev/ttyUSB3:57600 -s
Flushing the serial port..
Checking if node is a Deluge T2 base station ...
Command sent
serial@/dev/ttyUSB3:57600: resynchronising

Sending command to: 5111

Node 5111 is READY      Type: TelosB    Version: 148    Voltage (mV): 3704.834
```

If the target ID returns **READY**, then we can start to disseminate the new application.

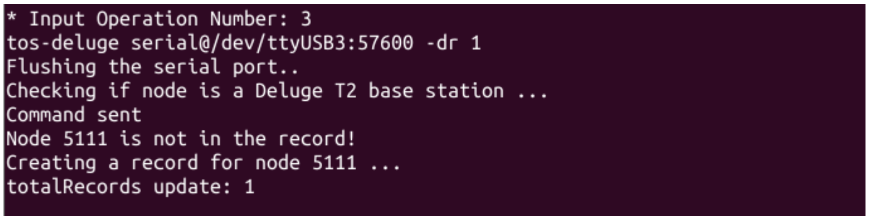
In the sniffer terminal, you can see some packets showing up:



```
MB_CMD - Cmd: DISS, Targets: 5111

MB_CMD - Cmd: DISS, Targets: 5111

MB_CMD - Cmd: DISS, Targets: 5111

MB_CMD - Cmd: DISS, Targets: 5111

MB_CMD - Cmd: DISS, Targets: 5111

Deluge_CMD - Src: 170, Msg_no: 7, Cmd: STOP, Img_ID: 0x00000000, Vol_no: 0
```

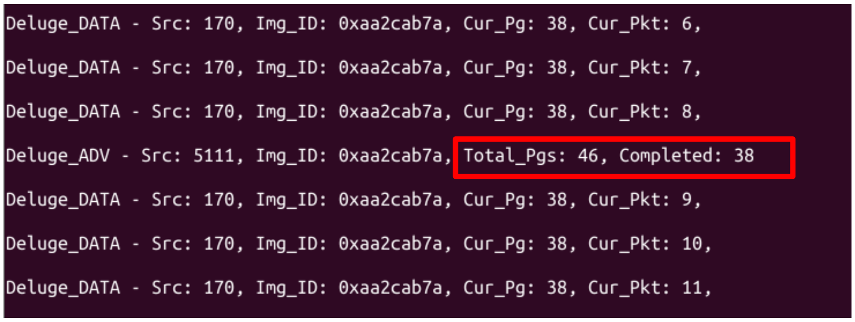
3.5 Dissemination and reprogramming

Input option **"3"** to start disseminate the new application to the target motes.



```
* Input Operation Number: 3
tos-deluge serial@/dev/ttyUSB3:57600 -dr 1
Flushing the serial port..
Checking if node is a Deluge T2 base station ...
Command sent
Node 5111 is not in the record!
Creating a record for node 5111 ...
totalRecords update: 1
```

You can see the packets flushing in the sniffer terminal. In the following, the application has 46 pages, and DelugeBase has transmitted 38 pages.

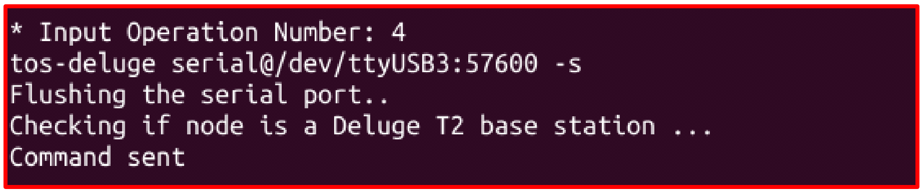


```
Deluge_DATA - Src: 170, Img_ID: 0xaa2cab7a, Cur_Pg: 38, Cur_Pkt: 6,

Deluge_DATA - Src: 170, Img_ID: 0xaa2cab7a, Cur_Pg: 38, Cur_Pkt: 7,

Deluge_DATA - Src: 170, Img_ID: 0xaa2cab7a, Cur_Pg: 38, Cur_Pkt: 8,

Deluge_ADV - Src: 5111, Img_ID: 0xaa2cab7a, Total_Pgs: 46, Completed: 38

Deluge_DATA - Src: 170, Img_ID: 0xaa2cab7a, Cur_Pg: 38, Cur_Pkt: 9,

Deluge_DATA - Src: 170, Img_ID: 0xaa2cab7a, Cur_Pg: 38, Cur_Pkt: 10,

Deluge_DATA - Src: 170, Img_ID: 0xaa2cab7a, Cur_Pg: 38, Cur_Pkt: 11,
```

You will also see the LEDs on the DelugeBase flashing during the dissemination. When the flash stops, the dissemination is completed.

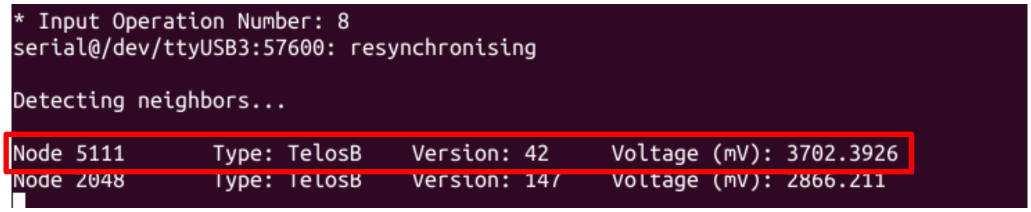
3.6 Stop further dissemination

This is a critical step after every dissemination. When dissemination is finished, the STOP command must be sent. Issue command **"4"**:



```
* Input Operation Number: 4
tos-deluge serial@/dev/ttyUSB3:57600 -s
Flushing the serial port..
Checking if node is a Deluge T2 base station ...
Command sent
```

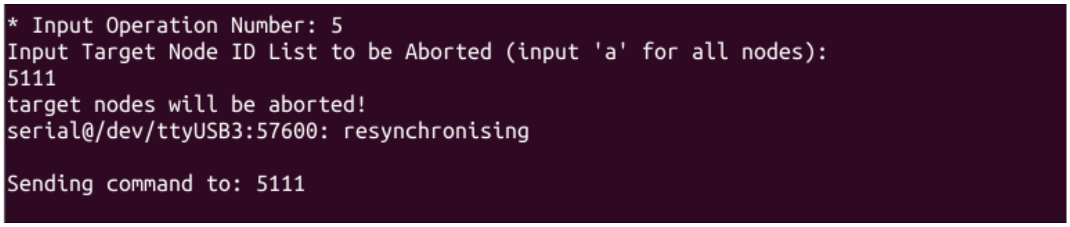
After this, you can use command “**8**” (detect neighbors) again to check the application version in the mote. As we can see below, mote 5111 now is running application version 42:



```
* Input Operation Number: 8
serial@/dev/ttyUSB3:57600: resynchronising

Detecting neighbors...

Node 5111       Type: TelosB    Version: 42     Voltage (mV): 3702.3926
Node 2048       Type: TelosB    Version: 147    Voltage (mV): 2866.211
```

3.7 Abort wrong targets

Input **“5”** to abort one or more target motes if any of them are wrongly selected (after option **“2”**). Use **“space”** to separate multiple IDs. Input “**a**” to abort all motes.



```
* Input Operation Number: 5
Input Target Node ID List to be Aborted (input 'a' for all nodes):
5111
target nodes will be aborted!
serial@/dev/ttyUSB3:57600: resynchronising

Sending command to: 5111
```

There will be no reply from the aborted mote. So wait for a few seconds and press **`Enter`** to return to the main menu.