

TUTORIAL & DEMO

IMAGE AND AUDIO TRANSMISSION ON WIRELESS SENSOR NETWORKS

CARI 2014 TUTORIAL - PART III
GASTON BERGER UNIVERSITY
OCTOBER, 17TH, 2014
SAINT-LOUIS, SENEGAL



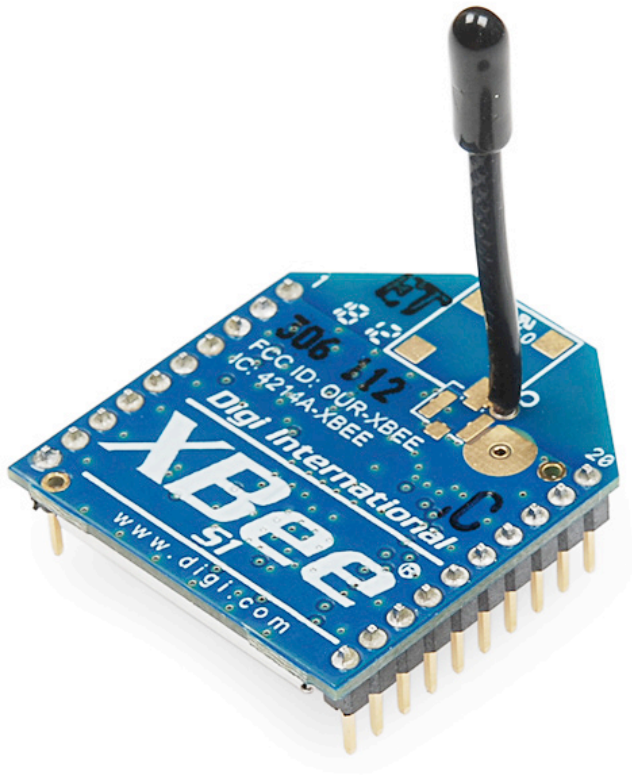
PROF. CONGDUC PHAM
[HTTP://WWW.UNIV-PAU.FR/~CPHAM](http://www.univ-pau.fr/~cpham)
UNIVERSITÉ DE PAU, FRANCE



DEVELOPMENT ENVIRONMENTS

- **LINUX-BASED SYSTEMS FOR HIGHER FLEXIBILITY AND BETTER INTEROPERABILITY**
 - **MOST OF SOFTWARE TOOLS ARE TARGETED FOR UNIX**
 - **MOST OF GATEWAYS DEVICES ARE LINUX-BASED (MESHLIUM, BEAGLE, RASPBERRY, ...)**
- **WHEN POSSIBLE, AVOID JAVA DEVELOPMENT AND PRIVILEGED C, OR C++ AND/OR SCRIPTS (SHELL, PYTHON)**

XBEE RADIO MODULE FROM DIGI



Implements IEEE 802.15.4 standard

64-bit hardware MAC address

0x0013A200409C0343



RADIO (802.15.4) GATEWAYS TO GET DATA IN COMPUTER

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/ 14	variable	2
Frame Control CC61	Sequence Number 58	Destination PAN 3332	Destination Address 0013A200 40922078	Source PAN 3332	Source Address 0013A200 4086D834	Auxiliary Security Header	Frame Payload HELLO	FCS 2B32
MHR							MAC Payload	MFR



HELLO

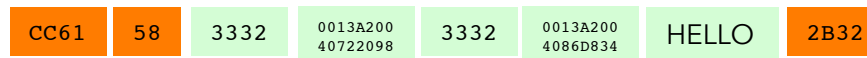
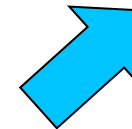
64-bit 0x0013A2004086D834
16-bit 0x0010
PANID 0x3332

Seen as a
serial port
/dev/ttyUSB0

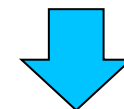


USB-serial converter

Some hardware give access to
Link-layer information



Transparent mode
or Serial line
replacement mode
or **printf-like** function

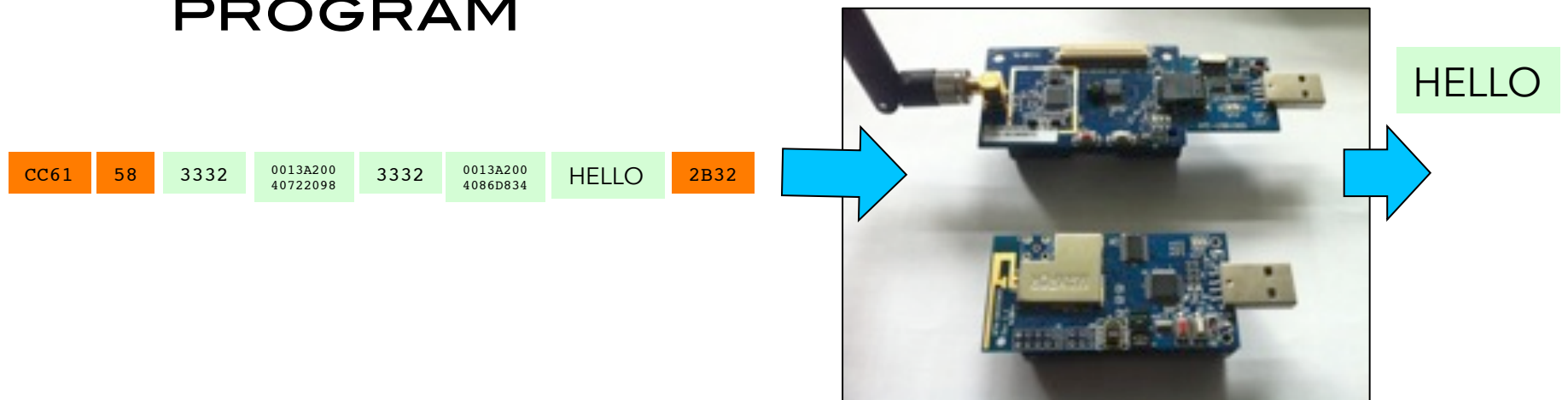


Read from /dev/ttyUSB0

HELLO

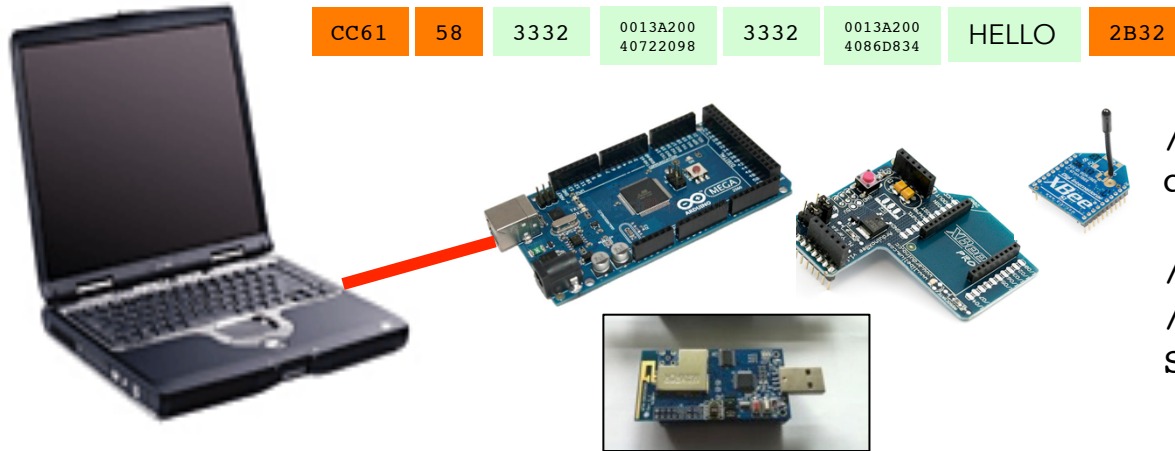
SENSOR MOTE AS GATEWAY

- ❑ PROGRAM A SENSOR MOTE TO WRITE TO SERIAL PORT ALL RECEIVED DATA FROM RADIO
- ❑ EX: ADVANTICSYS TELOS B WITH BASESTATION-LIKE TINYOS PROGRAM



Read from /dev/ttyUSB0

FROM serial TO APPLICATIONS/USERS



```
// get data from radio
data=xbee.getData();

// data = 'hello'
// print data to serial port
Serial.print(data);
```

- ❑ WRITE CUSTOM APPLICATIONS THAT READ THE SERIAL PORT: `/dev/ttyUSBx`, `/dev/ttySx`,...
- ❑ USE TOOLS TO READ SERIAL PORT AND PRINT ON STANDARD OUTPUT, E.G. UNIX `stdout`
 - ❑ MUCH MORE FLEXIBLE, AND RELIABLE
 - ❑ CAN USE UNIX REDIRECTION TO FEED DATA INTO OTHER PROGRAMS

```
• > unix_tool_1 | myapp | unix_tool_2 | unix_tool_3 ...
```

SERIALTOSTDOUT.PY

- SIMPLE PYTHON SCRIPT TO READ SERIAL PORT WHEN NO TRANSLATION IS NEEDED
- CHANGE PORT AND BAUD RATE AS NEEDED

```
import serial
import sys

ser = serial.Serial('/dev/ttyUSB0', 38400, timeout=0)

# flush everything that may have been received on the port to make sure
# that we start with a clean serial input
ser.flushInput()

while True:
    out = ''
    sys.stdout.write(ser.read(1024))
    sys.stdout.flush()
```

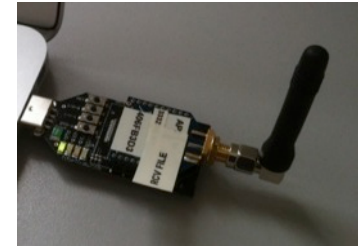
DISPLAY_IMAGE

- ❑ READ STANDARD INPUT FOR IMAGE PACKET
- ❑ WRITE PACKET TO FILE
- ❑ READ FROM FILE AND DISPLAY IMAGE

```
> python SerialToStdout | display_image
```




XBEE RECEIVE (OBSOLETE)



- ❑ MAIN TARGET IS XBEE-BASED GATEWAY
- ❑ TRANSLATES XBEE API FRAME
- ❑ READ FROM THE SERIAL PORT
 - /dev/ttyUSB0, /dev/ttyS0, ...
- ❑ CAN WRITE TO UNIX STDOUT & CAN ACT AS A TRANSPARENT SERIAL REPLACEMENT
- ❑ DISPLAY IMAGES IN IMAGE MODE
- ❑ RECONSTRUCTS FILE IN BINARY MODE
- ❑ CAN ACT IN A DATA STREAM FASHION: NO HEADER FOR PACKETS
- ❑ SerialToStdout.py CAN BE USE INSTEAD OF XBeeReceive WITH AN XBEE IN TRANSPARENT MODE

XBEE RECEIVE CMD LINE (OBSOLETE)

USAGE: ./XBeeReceive -baud b -p dev -onlydisplay img_file.dat -pktd -pktf -B/-I -ap0 -v val
 -stdout -stream -Q 40 file_name

USAGE: -baud, set baud rate, default is 38400

USAGE: -p /dev/ttyUSB1

USAGE: -onlydisplay img_file.dat, display the .dat file only

USAGE: -pktd, display received XBee frames

USAGE: -pktf, generate pkt list file

USAGE: -B/-I, -B for binary mode, -I for image mode, default is image mode

USAGE: -framing, expects 0xFF0x55 for binary mode, 0xFFx50 for image mode, default is no framing

USAGE: -ap0, indicates an Xbee in AP mode 0 (transparent mode) so do not decode frame structure

USAGE: -v 77, use 0x77 to fill in missing value in binary mode

USAGE: -stdout, write to stdout for pipe mode, don't work with image mode

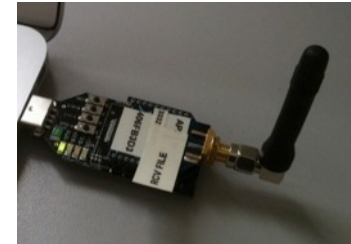
USAGE: -stream, assumes no header & write to stdout for pipe mode in binary mode

USAGE: -Q 40, use 40 as Quality Factor, default is 50

USAGE: file_name, for images: give the original bmp file. for binary: give any file name



XBEESENDCMD



❑ XBEESENDCMD

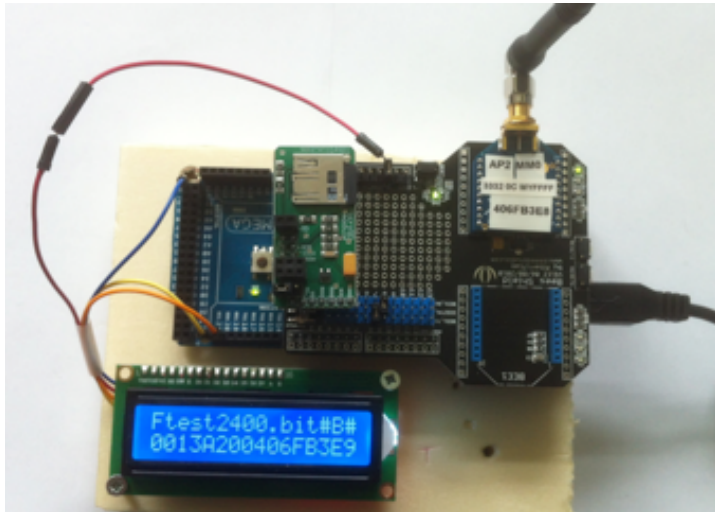
- ❑ USES AN XBEE GATEWAY TO SEND ASCII STRING COMMAND, E.G. « **/@D0030#** »

```
USAGE: ./XBeeSendCmd -p dev [-L][-DM][-at] -tinyos -tinyos_amid id_hex -mac|-net|-addr|-b message
USAGE: -p /dev/ttyUSB1
USAGE: -mac 0013a2004069165d HELLO
USAGE: -net 5678 HELLO
USAGE: -addr 64_or_16_bit_addr HELLO
USAGE: -b HELLO
USAGE: -at to send remote AT command: -at -mac 0013a2004069165d ATMM
USAGE: -L insert Libelium API header
USAGE: -DM to specify DigiMesh firmware
USAGE: -tinyos to forge a TinyOS ActiveMessage compatible packet (0x3F0x05 are inserted)
USAGE: -tinyos_amid 6F, to set the ActiveMessage identifier to 0x6F (0x05 is the default)
```

❑ EXAMPLE:

- ❑ `XBeeSendCmd -addr 0013A2004086D835 hello`
- ❑ `XBeeSendCmd -addr 0013A2004086D835 /@z50#`

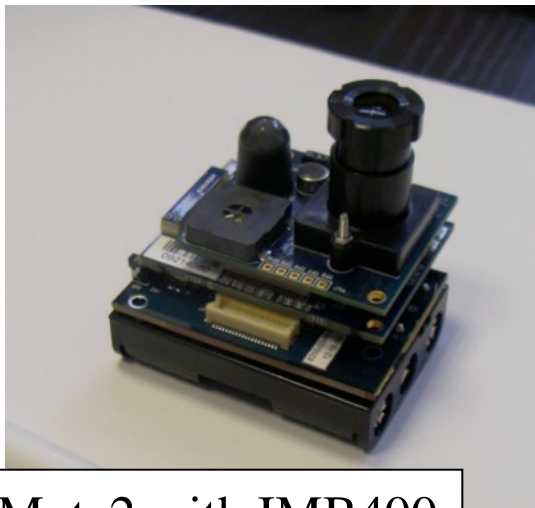
IMAGE DEMO



PREVIOUS IMAGE SENSOR MOTES



iMote2



iMote2 with IMB400 multimedia board



MORE GENERIC SOLUTION: FILE SENDER NODE

Fully configurable:

File to send
Size of packet chunk
Inter-packet delay
Image/Binary mode
Destination node
Clock synchronization

COST:
~132€

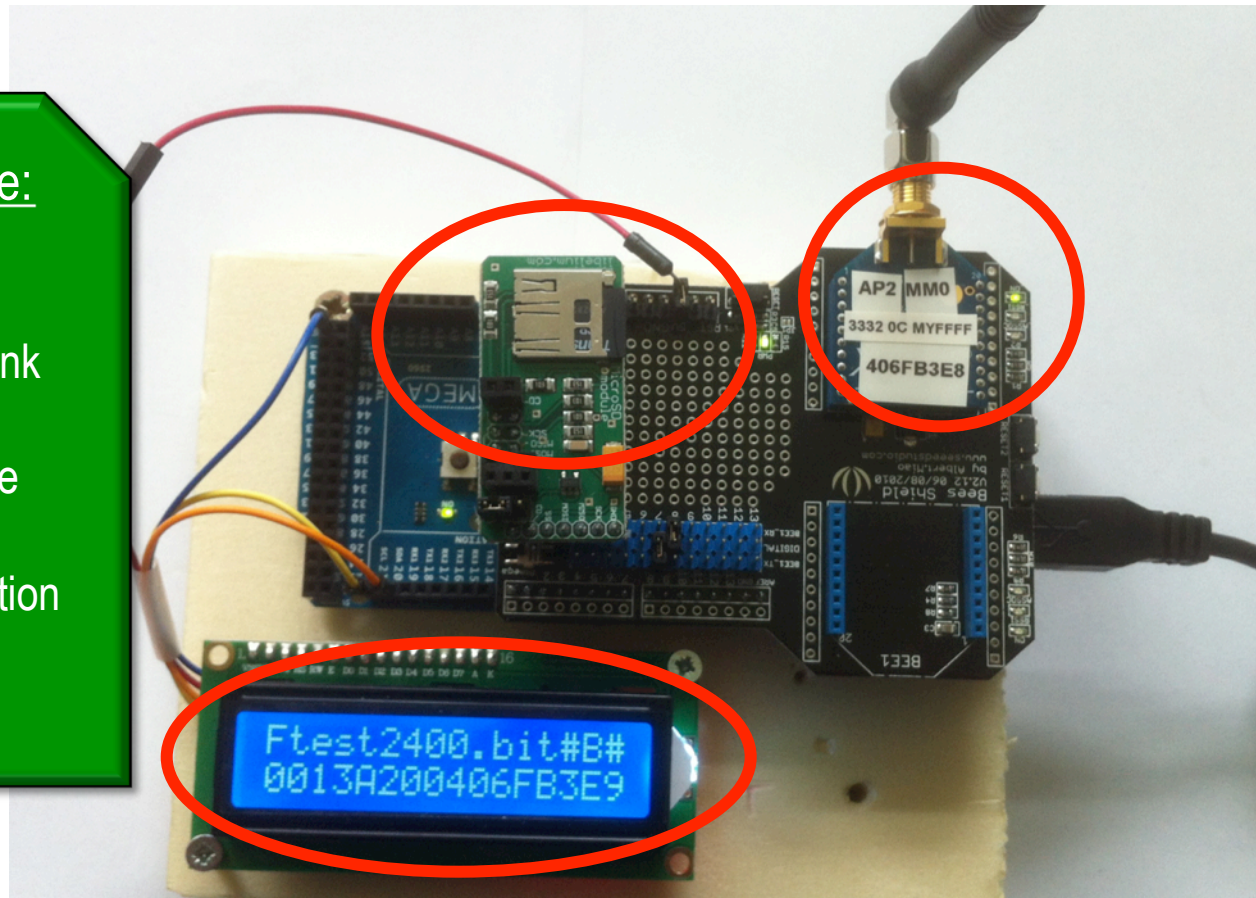


IMAGE ENCODING

❑ ENCODE ANY .BMP IMAGE FILE

❑ `JPEGencoding -MSS 90 -Q 5 original_image_file.bmp`

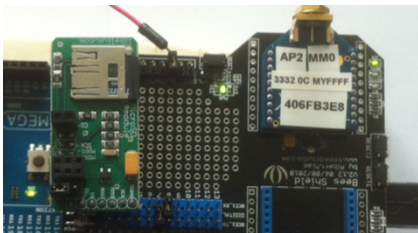
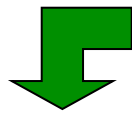
❑ SPECIFY A QUALITY FACTOR

12 128 128 5

Original BMP 128x128



```
000C 0080 0080 0005 0059 00 00 CD E5 43 C2 3A C8 1B D0 A0 E7 2A C7 D1 6C EE 60 FD 5A
A8 AB 53 87 C0 29 D4 85 01 43 B0 95 C5 93 7C A8 65 9C 1C 13 F4 16 CA 02 27 08 EB 4A
38 AB 06 06 22 8E 74 61 EA 18 4A BD EB 77 0D 46 D9 8F 8B 05 92 E1 EE D9 9A 98 D4 A3
3F 83 16 E0 D0 6B 87 14 48 58 4E 13 7F 0056 00 20 D5 BE 0B 70 DF 3C 0F 1B 48 13 93 9E
73 45 03 94 1E 9B A5 0B BB CF 52 04 4D 3E 88 36 B6 02 77 CE 9F F8 32 42 18 A8 F8 AA
79 9E 35 20 A0 A5 98 FE 32 16 1E 97 90 47 87 EC 69 61 60 7D AB 96 B9 AC CD B4 B8 5D
B9 88 23 60 20 32 44 30 8A DA 5B A4 00 46 36 15 0057 00 32 E8 24 33 13 48 69 28 66 73
BD 03 3E 2F 37 CC CF E4 9E A9 1E CD EB 45 36 B6 12 18 FA 6A BA A9 94 1A 95 EB 49 50
CD A1 77 C2 8E 73 21 CB DC 5D 06 E4 29 A3 B6 70 C5 E0 F9 C4 62 16 9F 94 98 CD 1C AE
0D AF 7E DE 24 44 33 FB 54 6A 08 A8 1A 70 9E A2 DA 50 AC 27 0056 00 45 E8 45 44 23 09
42 33 69 5E 96 2E 7E 7F EB 8D 59 88 9F 23 9E 4F B2 77 62 30 8B BA AC 73 89 2F 3D 18
91 73 FC CF BF 36 DC B1 86 22 98 02 93 C7 86 01 06 36 85 89 BE 2F 01 6E 57 B0 51 82
AE 87 7B 43 80 42 5B E9 4B 78 58 CF 59 F8 56 29 C6 A8 1C 0C 72 09 97 0058 00 5B D9 2E
88 09 71 6D 35 C4 7A EA 22 DA DE 38 EF 8D 9C B2 A8 B4 F0 9C CA 1A 0E F5 A7 E6 2B 39
20 E6 07 67 34 CB 1F B1 BA C4 28 F0 69 EC 45 37 AF AF DE E0 04 9A A8 CE 95 73 9A 75
B5 8E A1 F0 1C 2E DF 16 23 38 A1 2E DA C6 72 3D 20 E2 C7
0058 00 6E E8 50 B3 86 C5 77 B9 97 57 8C AC CE 20 E2 C7
63 9A 03 D5 2B 23 15 5C 1B 26 0E 0F 3C 82 4E 61 20 E2 C7 DC 91 EA
71 66 60 46 F5 16 22 5D ED 7D 79 E8 46 FE 3C 6E 8D 6D
BE FC 14 C6 3F 0056 00 84 D3 27 D6 C1 A4 52 F8 3D 94
09 3E D6 F6 72 AA 63 0C C1 DC 48 E2 8F DD 55 53 BF D7 9B
F3 3F 55 8A 7F 47 BA 67 FA CB 17 16 EE 24 A2 72 C2 80
FA 7A FC 82 66 33 49 43 0058 .. .. 36 69
```



Collaboration with CRAN
laboratory, Nancy, France.
Very robust image
encoding techniques
against packet losses

ADJUSTABLE QUALITY FACTOR

200X200, SUITABLE FOR SITUATION-AWARENESS

Original BMP 40000b



Q=50 S=11045b 142pkts



PSNR=25.1661

Q=40 S=9701b 123pkts



PSNR=24.2231

Q=30 S=8100b 101pkts



PSNR=23.2264

Q=20 S=6236b 76pkts



PSNR=22.1293

Q=15 S=5188b 63pkts



PSNR=21.4475

Q=10 S=3868b 47pkts



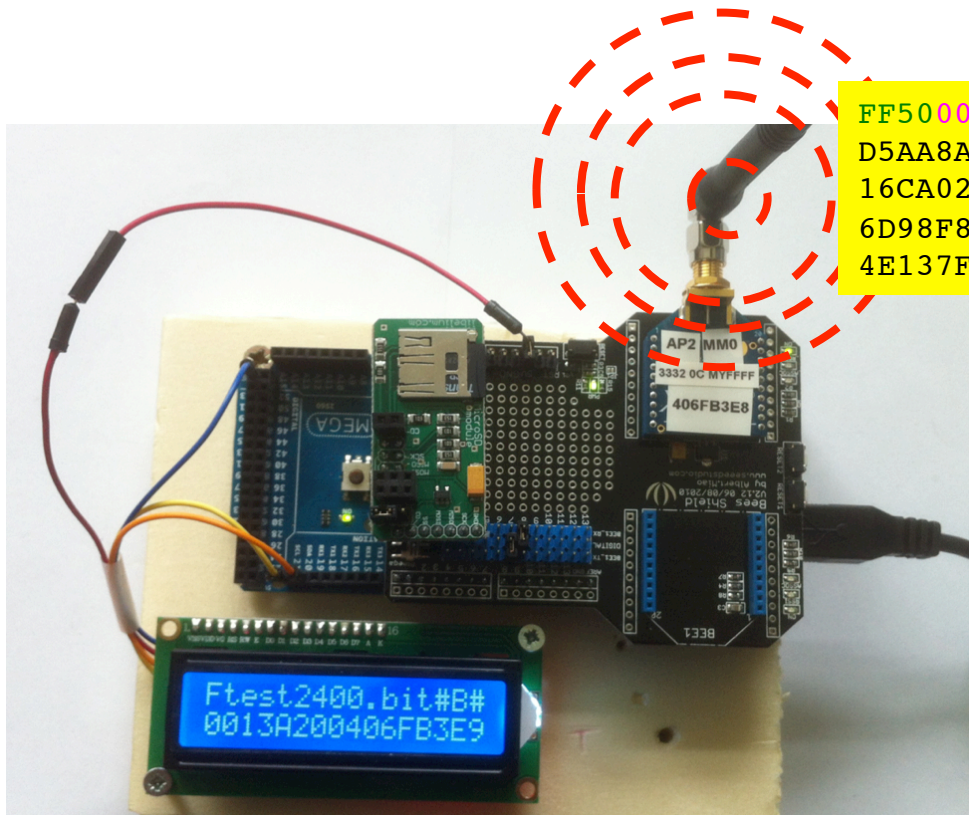
PSNR=20.5255

Q=5 S=2053b 24pkts



PSNR=18.937

TRANSMIT IMAGE PACKET



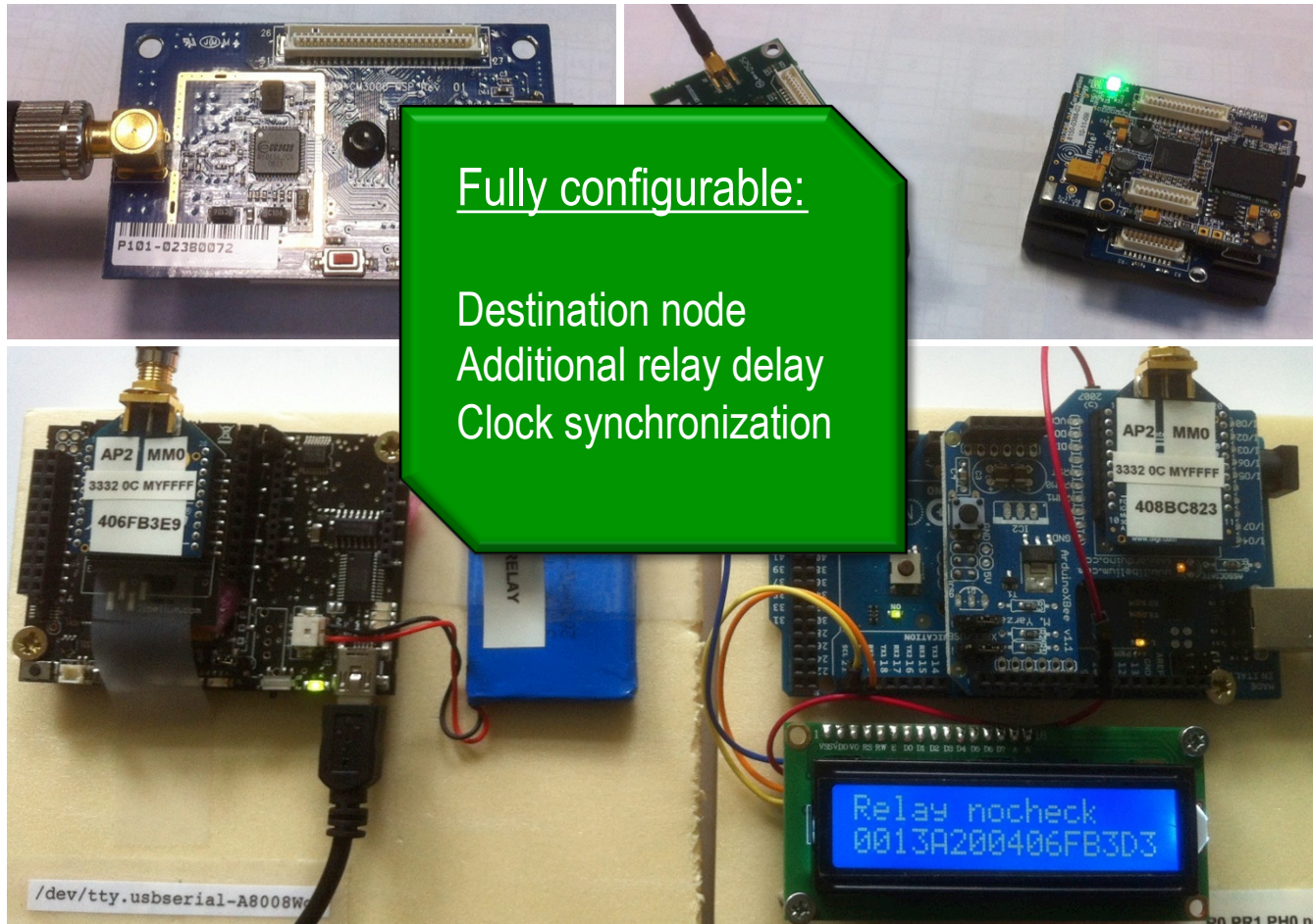
```
FF5000590000CDE543C23AC81BD0A0E72AC7D16CEE60F
D5AA8AB5387C029D4850143B095C5937CA8659C1C13F4
16CA022708EB4A38AB0606228E7461EA184ABDEB770D4
6D98F8B0592E1EED99A98D4A33F8316E0D06B87144858
4E137F
```

```
000C 0080 0080 0005 0059 00 00 CD E5 43 C2 3A C8 1B D0 A0 E7 2A C7 D1 6C EE 60 FD 5A
A8 AB 53 87 C0 29 D4 85 01 43 B0 95 C5 93 7C A8 65 9C 1C 13 F4 16 CA 02 27 08 EB 4A
38 AB 06 06 22 8E 74 61 EA 18 4A BD EB 77 0D 46 D9 8F 8B 05 92 E1 EE D9 9A 98 D4 A3
3F 83 16 E0 D0 6B 87 14 48 58 4E 13 7F 0056 00 20 D5 BE 0B 70 DF 3C 0F 1B 48 13 93 9E
73 45 03 94 1E 9B A5 0B BB CF 52 04 4D 3E 88 36 B6 02 77 CE 9F F8 32 42 18 A8 F8 AA
79 9E 35 20 A0 A5 98 FE 32 16 1E 97 90 47 87 EC 69 61 60 7D AB 96 B9 AC CD B4 B8 5D
B9 88 23 60 20 32 44 30 8A DA 5B A4 00 46 36 15 0057 .. ..
```

RELAY NODES

ADVANTICSYS TELOS B

MICAZ AND IMOTE2

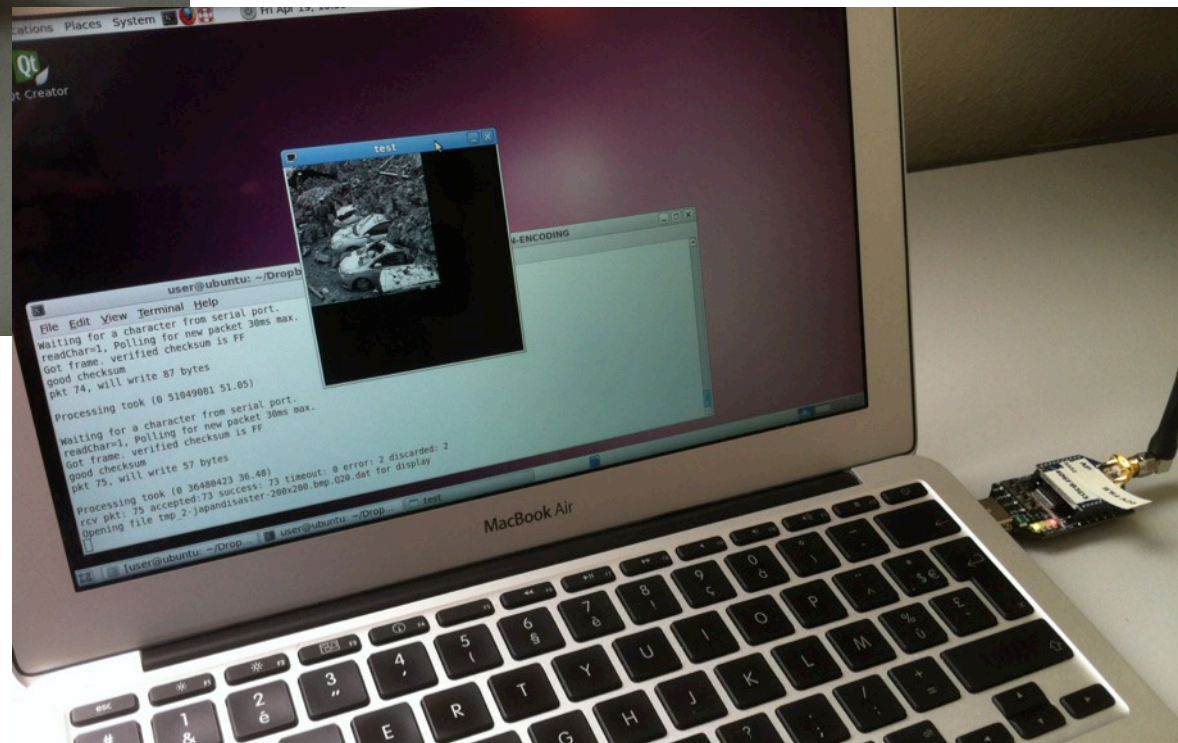
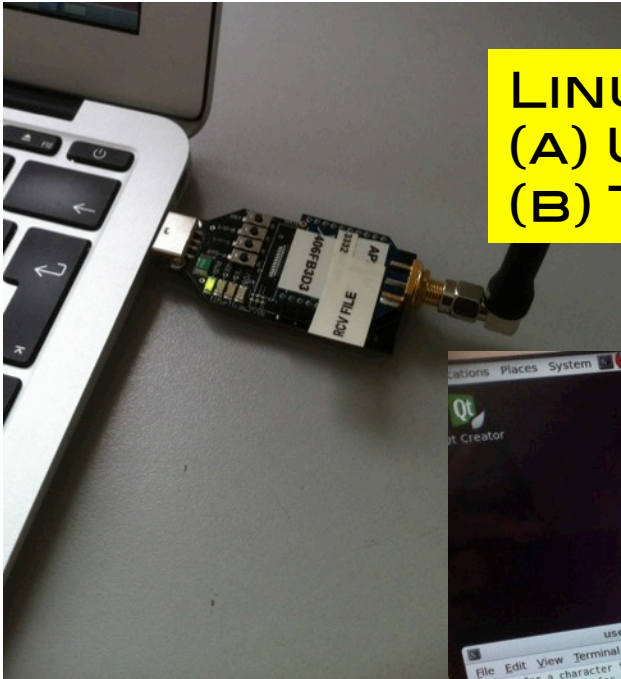


LIBELIUM WASPMOTE

ARDUINO MEGA2560

SINK NODE

LINUX PC/LAPTOP WITH
(A) USB/SERIAL GATEWAY XBEE RADIO
(B) TELOS6 AS BASESTATION



TRANSMISSION & RELAY TIME

Original BMP 40000b

250kbps: 1.28s

400pkt of 100bytes:
400*0.0115=4.6s

Relay overhead:
400*0.0157=6.28s

Q=50 S=11045b 142pkts



PSNR=25.1661

Q=20 S=6236b 76pkts

250kbps: 0.199s

76pkt of 95bytes:
76*0.0113=0.858s

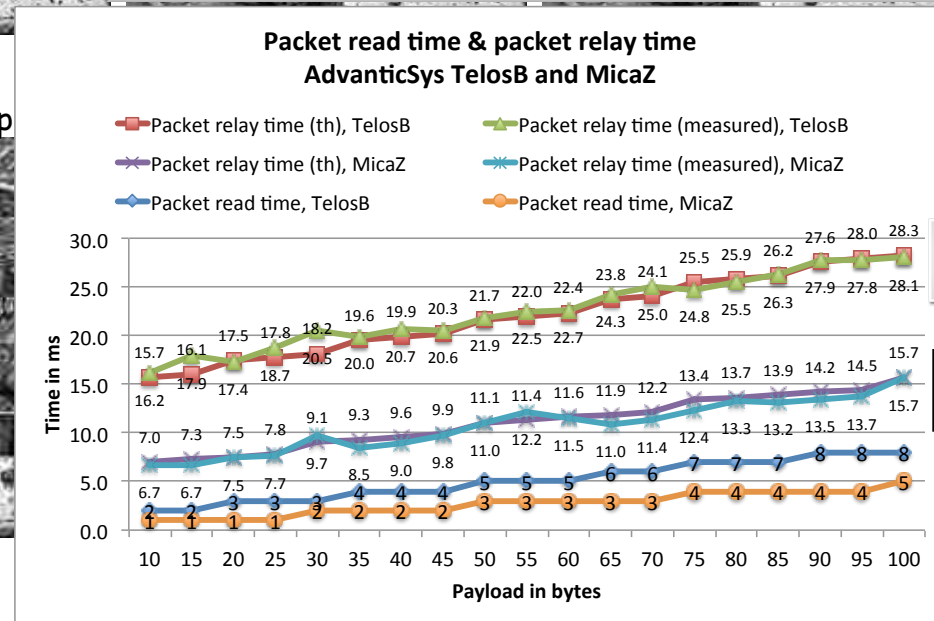
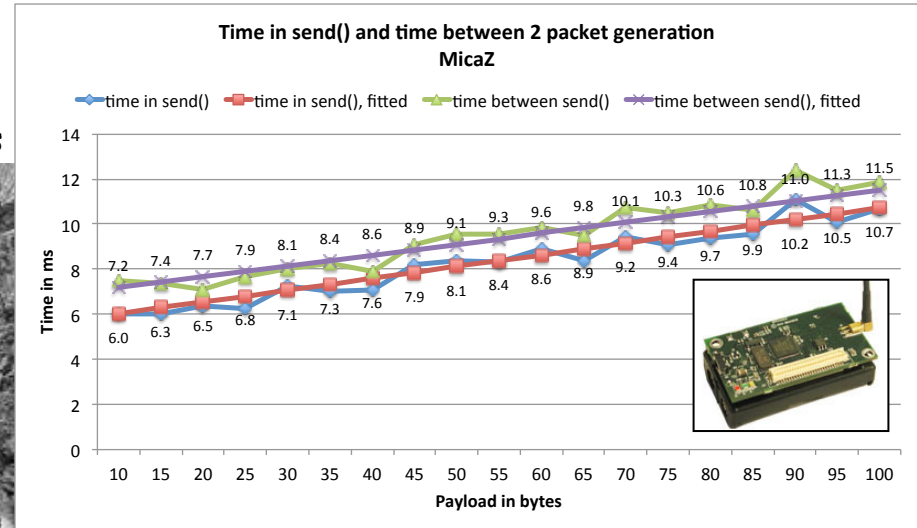
Relay overhead:
76*0.0145=1.102s

PSNR=22.1293

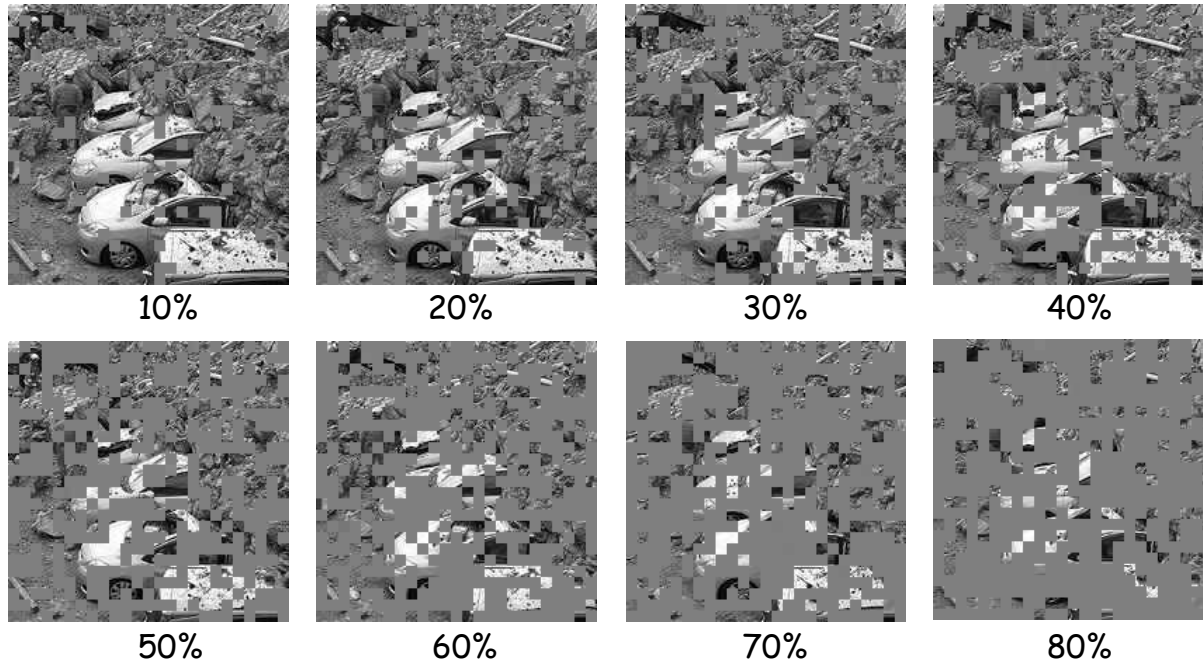
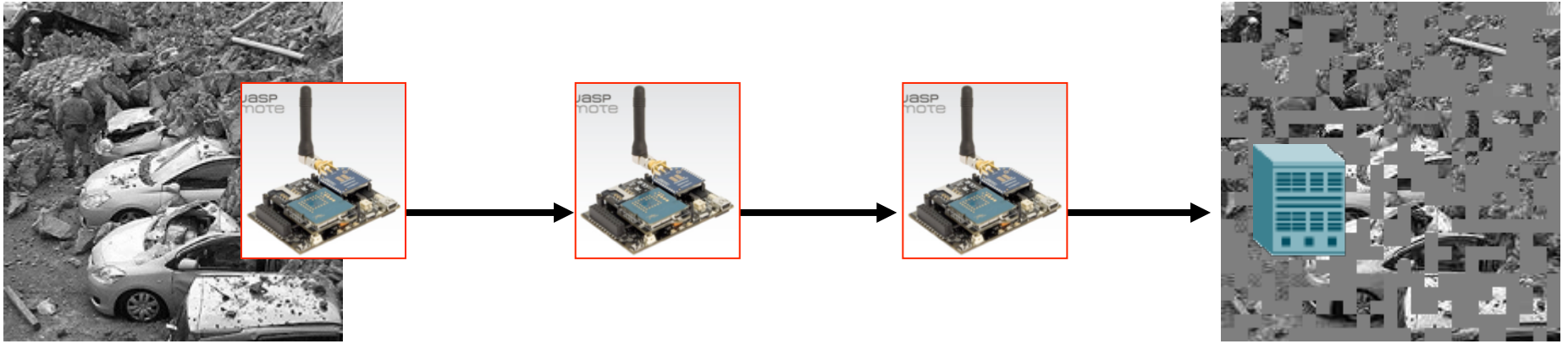
Q=15 S=5188b 63p



PSNR=21.4475

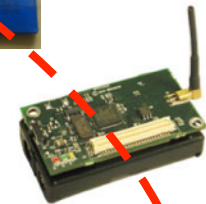
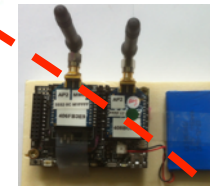
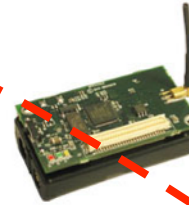
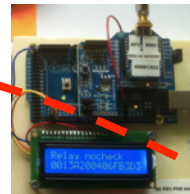
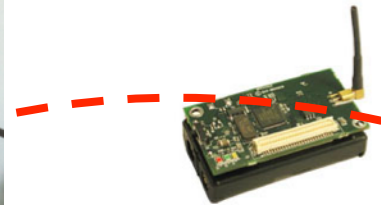
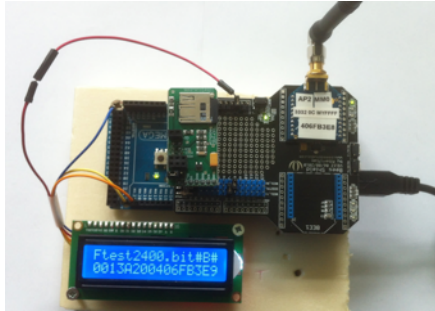


MULTI-HOP FORWARDING INCREASES PKT LOSS RATE



TEST BED

D0013A2004086D828 set the 64-bit dest. mac addr
D0080 set the 16-bit dest. mac addr



T130 transmit with inter pkt time of 130ms
Z50 set the pkt size for binary mode
FdesQ5.dat set the file name to desQ5.dat
D0013A2004086D828 set the 64-bit dest. mac addr
D0080 set the 16-bit dest. mac addr
I or B set to image mode/set to binary mode

All commands must be prefixed by « /@ »
and ended/separated by « # »

Examples:

/@T130#, /@FjapanQ20.dat#I#

serial Unix tool

IMAGE DEMO

0x0013A200408BC81B

- > XBeeSendCmd -addr 0013A200408BC81B /@FjapanQ20.dat#I#
- XBeeSendCmd -addr 0013A200408BC81B /@D0013A20040762191#
- XBeeSendCmd -addr 0013A200408BC81B /@T90#



Q=20 S=6236b 76pkts



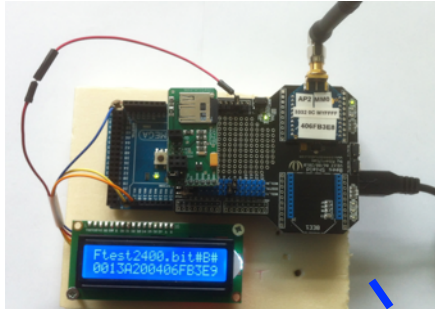
0x0013A20040762191



➤ python SerialToStdout | display_image -framing -Q 20 japandisaster-200x200.bmp serial Unix tool

IMAGE DEMO

0x0013A200408BC81B



Q=20 S=6236b 76pkts



```
> XBeeSendCmd -addr 0013A200408BC81B /@FjapanQ20.dat#I#  
➤ XBeeSendCmd -addr 0013A200408BC81B /@D0020#  
➤ XBeeSendCmd -addr 0020 /@D0013A20040762191#  
➤ XBeeSendCmd -addr 0013A200408BC81B /@T90#
```

```
> XBeeSendCmd -addr 0013A200408BC81B /@FjapanQ20.dat#I#  
➤ XBeeSendCmd -addr 0013A200408BC81B /@D0020#  
➤ XBeeSendCmd -addr 0020 /@D0010#  
➤ XBeeSendCmd -addr 0010 /@D0013A20040762191#  
➤ XBeeSendCmd -addr 0013A200408BC81B /@T90#
```

0x0020

0x0010

0x0013A20040762191

0x0080

```
> XBeeSendCmd -addr 0013A200408BC81B /@D0080#  
➤ XBeeSendCmd -addr 0080 /@D0013A20040762191#  
➤ XBeeSendCmd -addr 0013A200408BC81B /@T90#
```

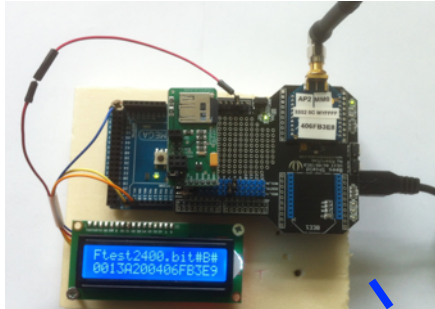


```
➤ python SerialToStdout | display_image -framing -Q 20 japandisaster-200x200.bmp
```

serial Unix tool

IMAGE DEMO

0x0013A200408BC81B



Q=20 S=6236b 76pkts



0x0020



0x0010



0x0080

0x0013A20040762191



- XBeeSendCmd -addr 0013A200408BC81B /@D0020#
- XBeeSendCmd -addr 0010 /@D0080#
- XBeeSendCmd -addr 0013A20040762191 /@T90#

➤ python SerialToStdout | display_image -framing -Q 20 japandisaster-200x200.bmp

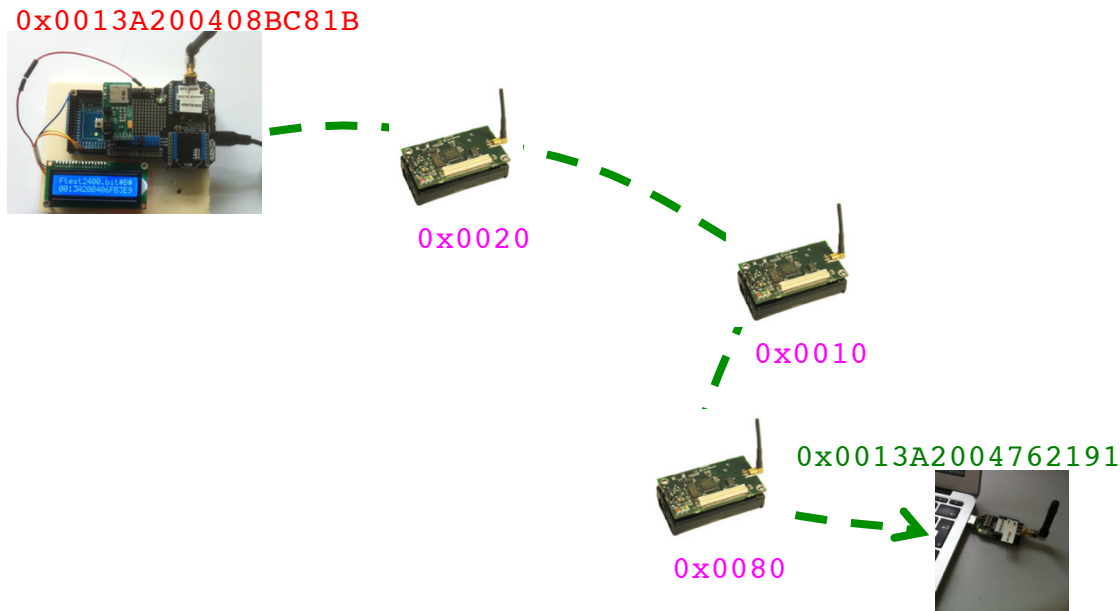
serial Unix tool

AUTOMATIZED RELAY CONFIGURATION

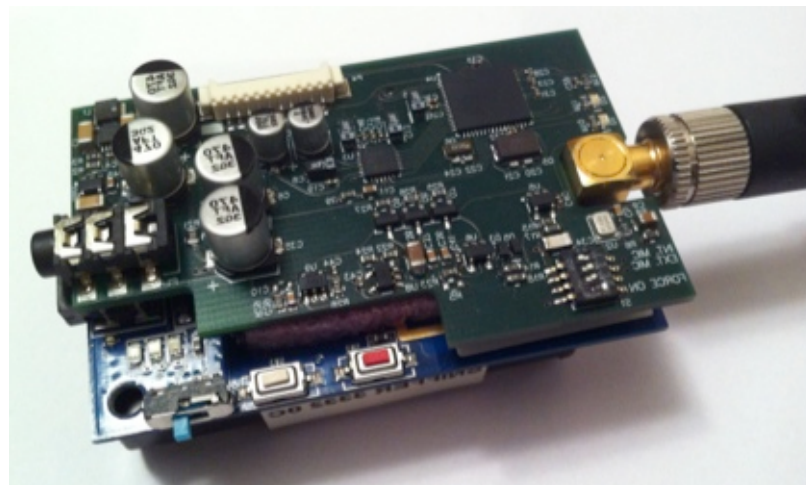
❑ SHELL SCRIPT TO XbeeSendCmd FOR CONFIGURING ROUTE

❑ relay_conf.sh

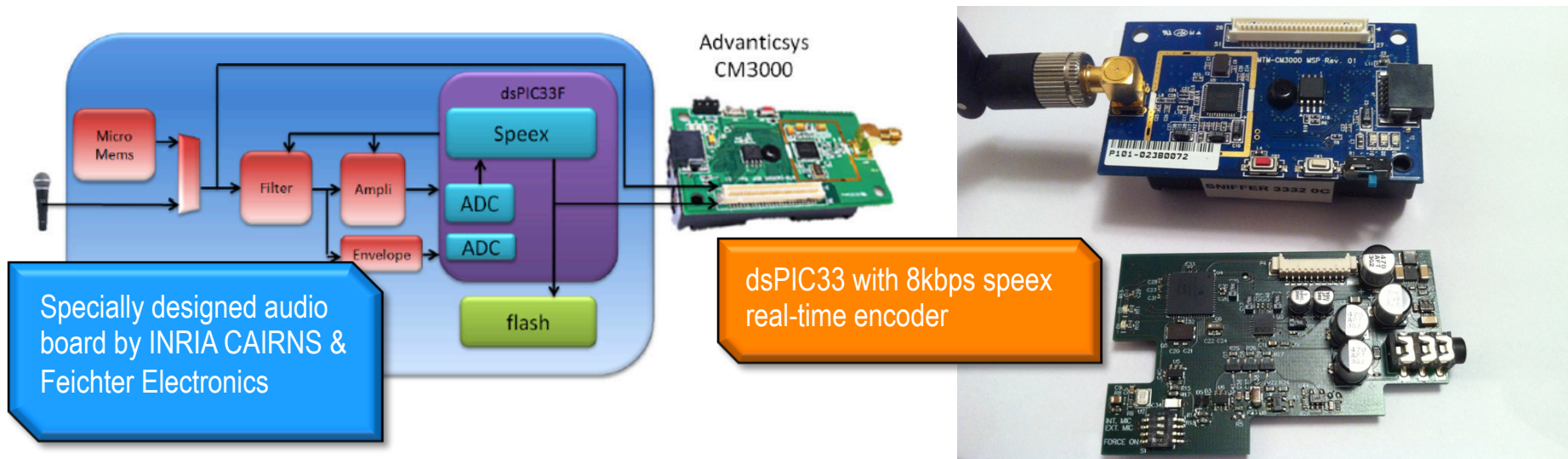
```
> relay_conf.sh /dev/ttyUSB0 /@FjapanQ20.dat#I# 0013A200408BC81B 0020 0010 0080 0013A20040762191
```



AUDIO DEMO

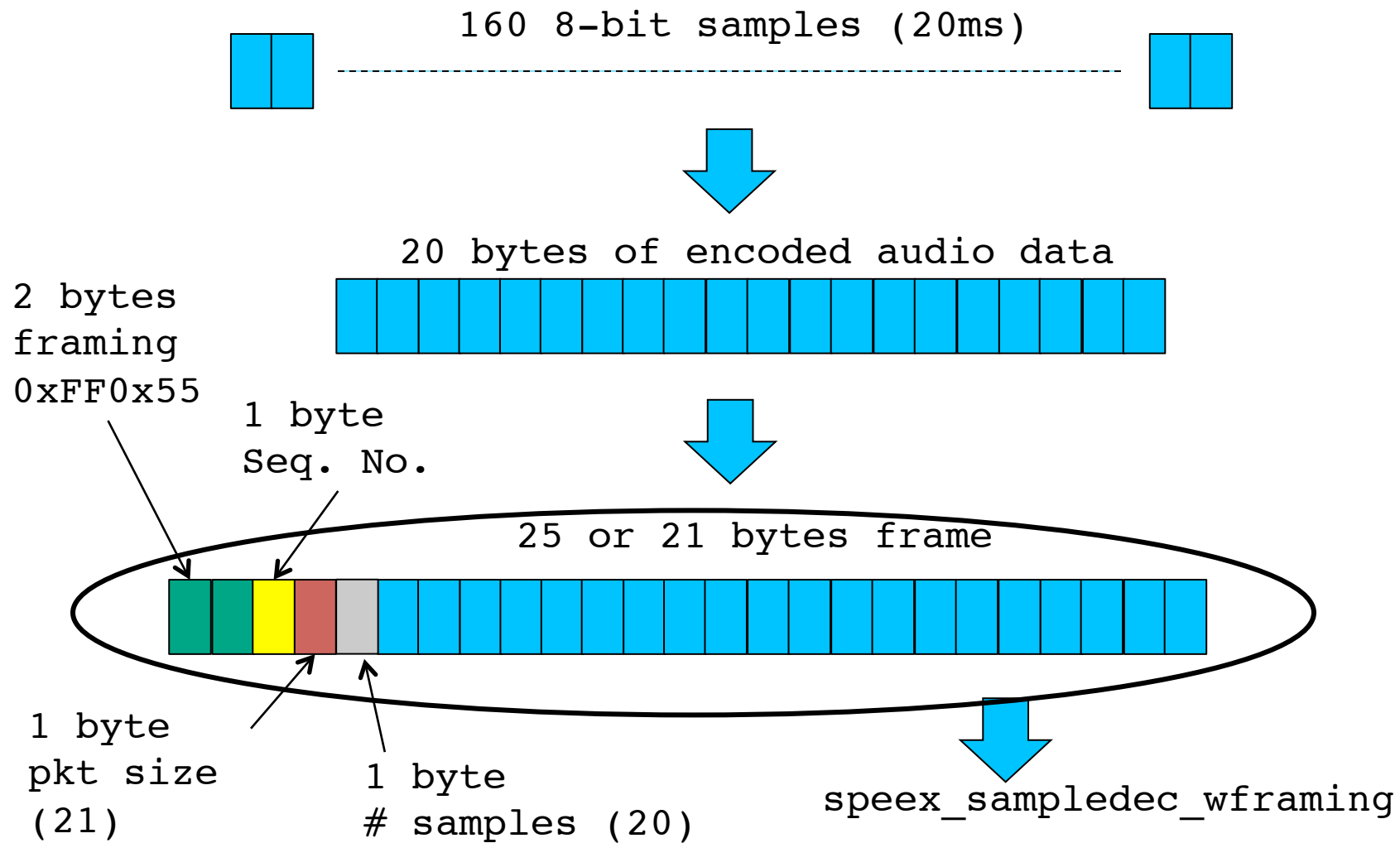


- Use dedicated audio board for sampling/storing/encoding at 8kbps



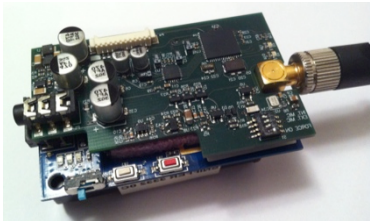
- Allows for multi-hop, encoded audio streaming scenarios

speex at 8kbps



1-hop test-bed w/audio board

0x0090



A1/2/3/4 aggregate audio frames
D0100 set the 16-bit dest. mac addr
C0/1 power off/on the audio board

SPEEX AUDIO ENCODING
8KBPS

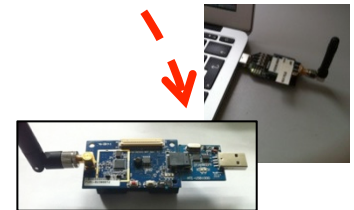
python script,
standard Unix scripting tool

speex decoding tool from
open-source speex distrib

play tool from sox
open-source distrib

DECODE & PLAY
RECEIVED AUDIO

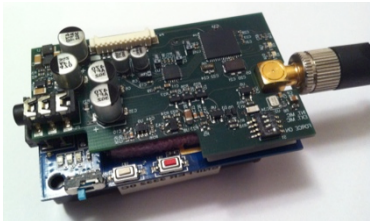
0x0100



```
python 115200SerialToStdout.py | speex_sampledec_wframing | play --buffer 100 -t raw -r 8000 -s -2 -
```

2-hop test-bed w/audio board

0x0090



SPEEX AUDIO ENCODING
8KBPS

A1/2/3/4/6 aggregate audio frames
D0020 set the 16-bit dest. mac addr
C0/1 power off/on the audio board

python script,
standard Unix scripting tool

speex decoding tool from
open-source speex distrib

0x0020



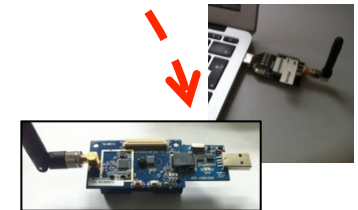
D0100 set the 16-bit
dest. mac addr

RELAY

DECODE & PLAY
RECEIVED AUDIO

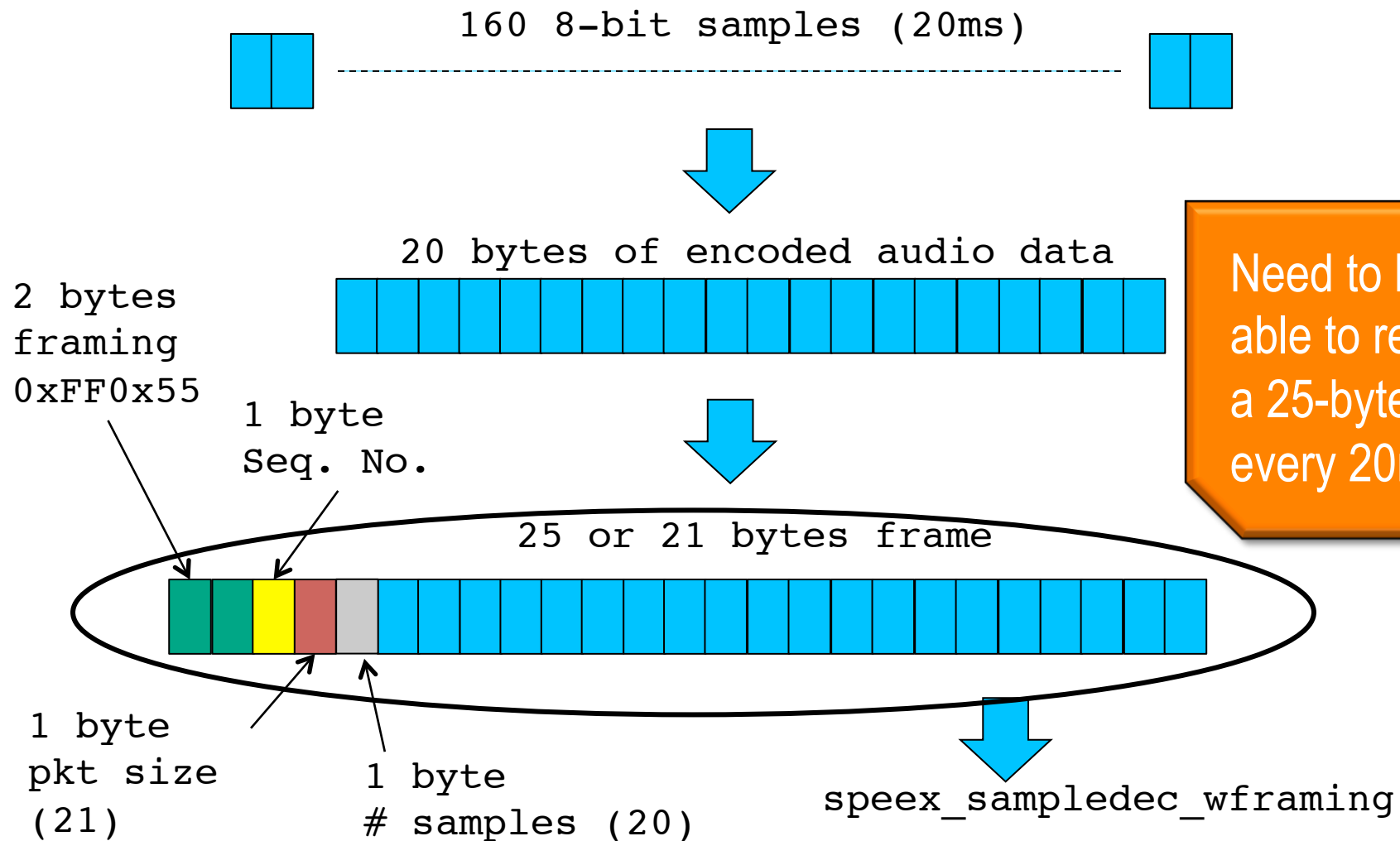
play tool from sox
open-source distrib

0x0100

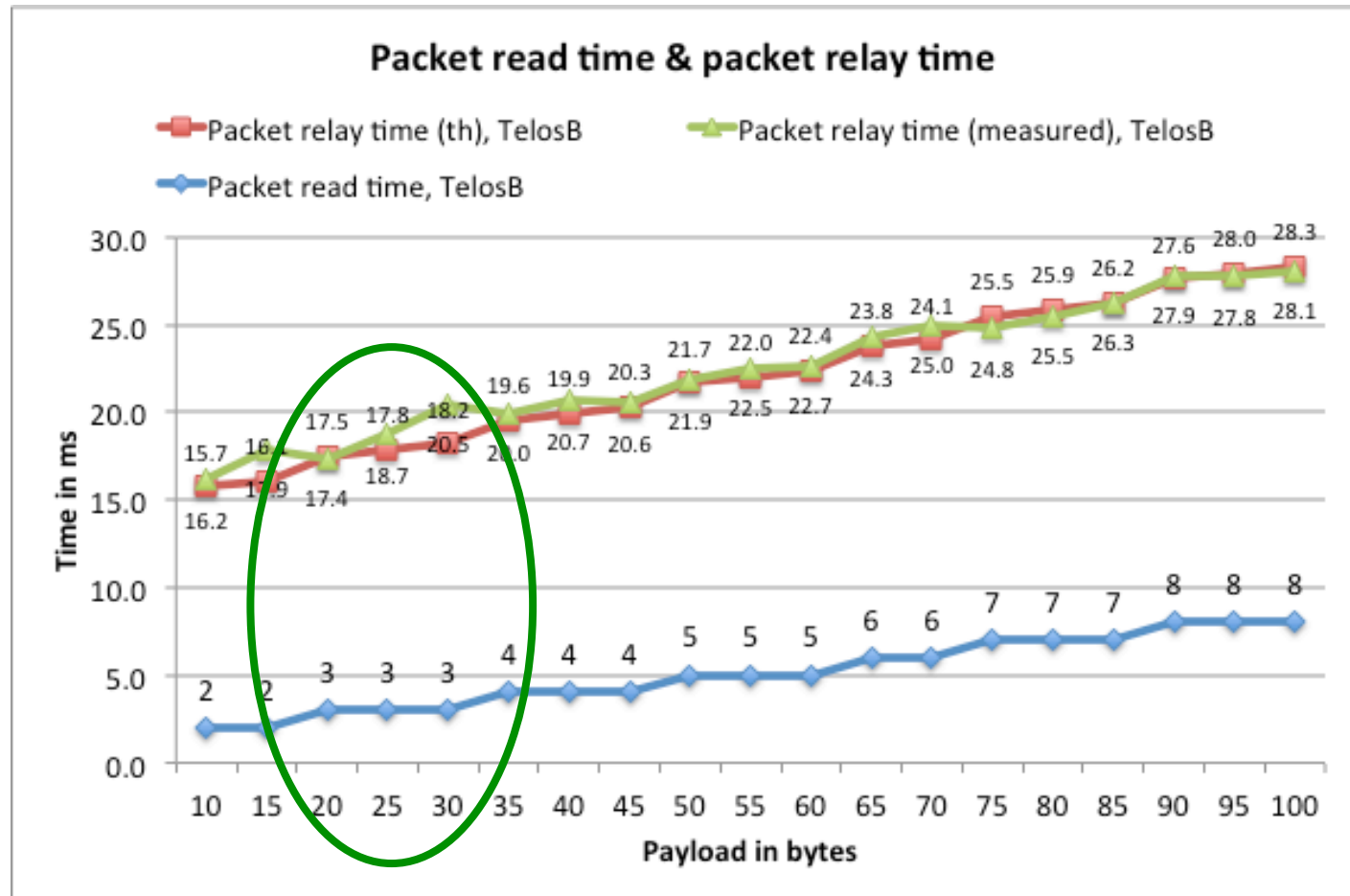
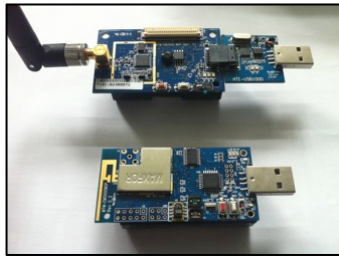


```
python 115200SerialToStdout.py | speex_sampledec_wframing | play --buffer 100 -t raw -r 8000 -s -2 -
```

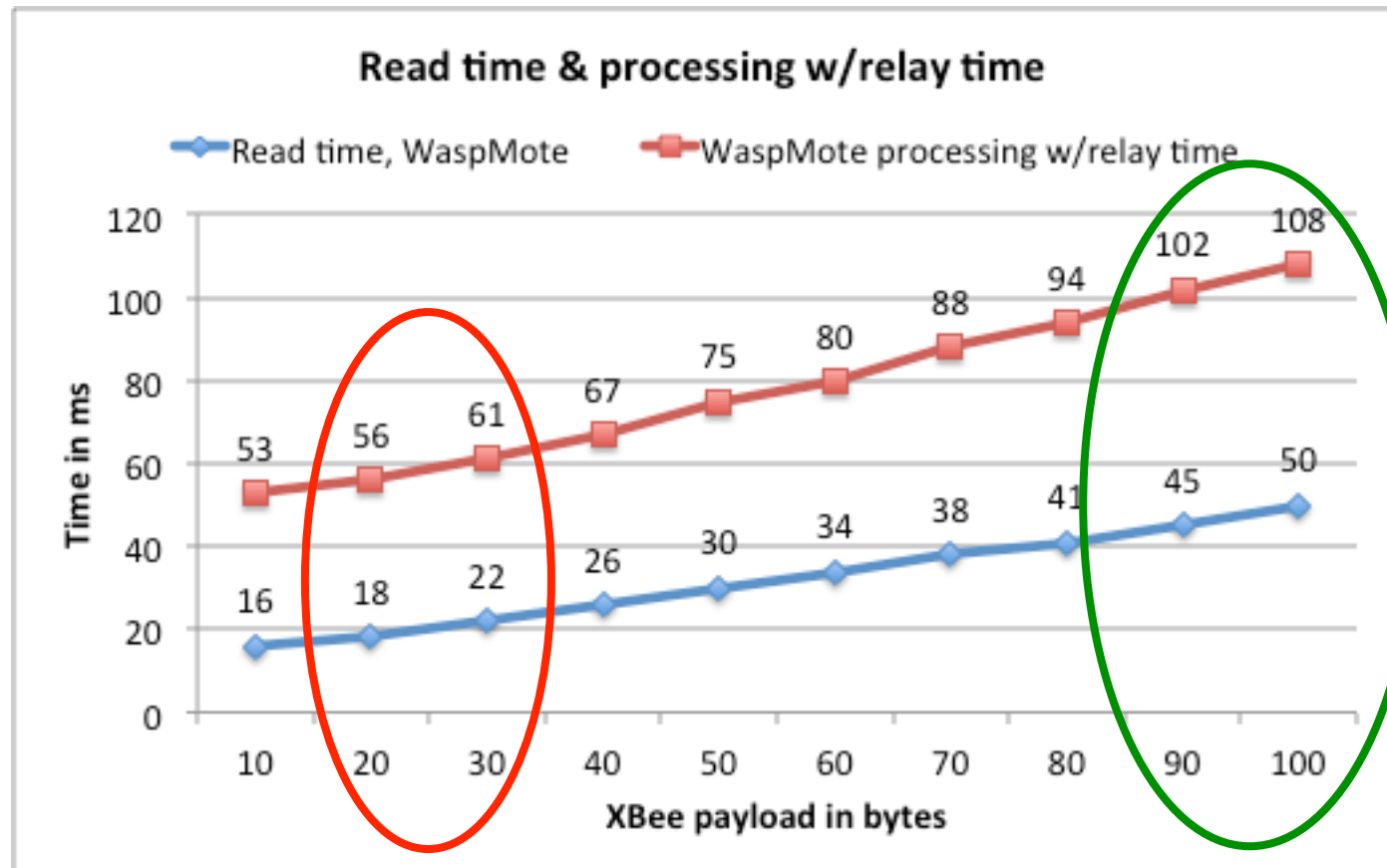
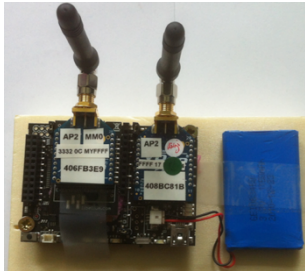
speex at 8kbps



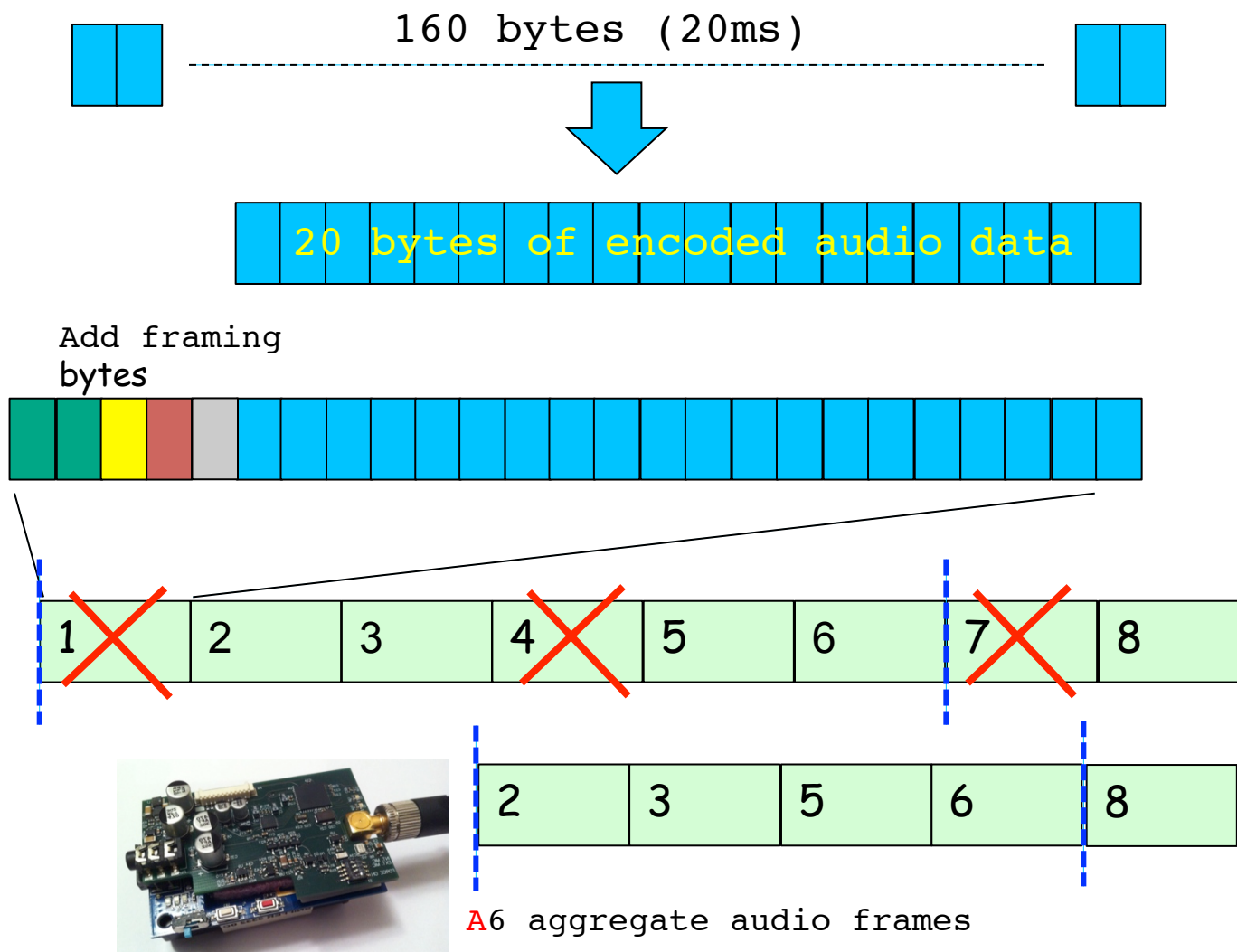
Relay node performances



Relay node performances



speex at 8kbps on slow relay nodes



Capture 6 audio frames (120ms) but only send 4

Need to be able to relay 100-byte pkt every 120ms

