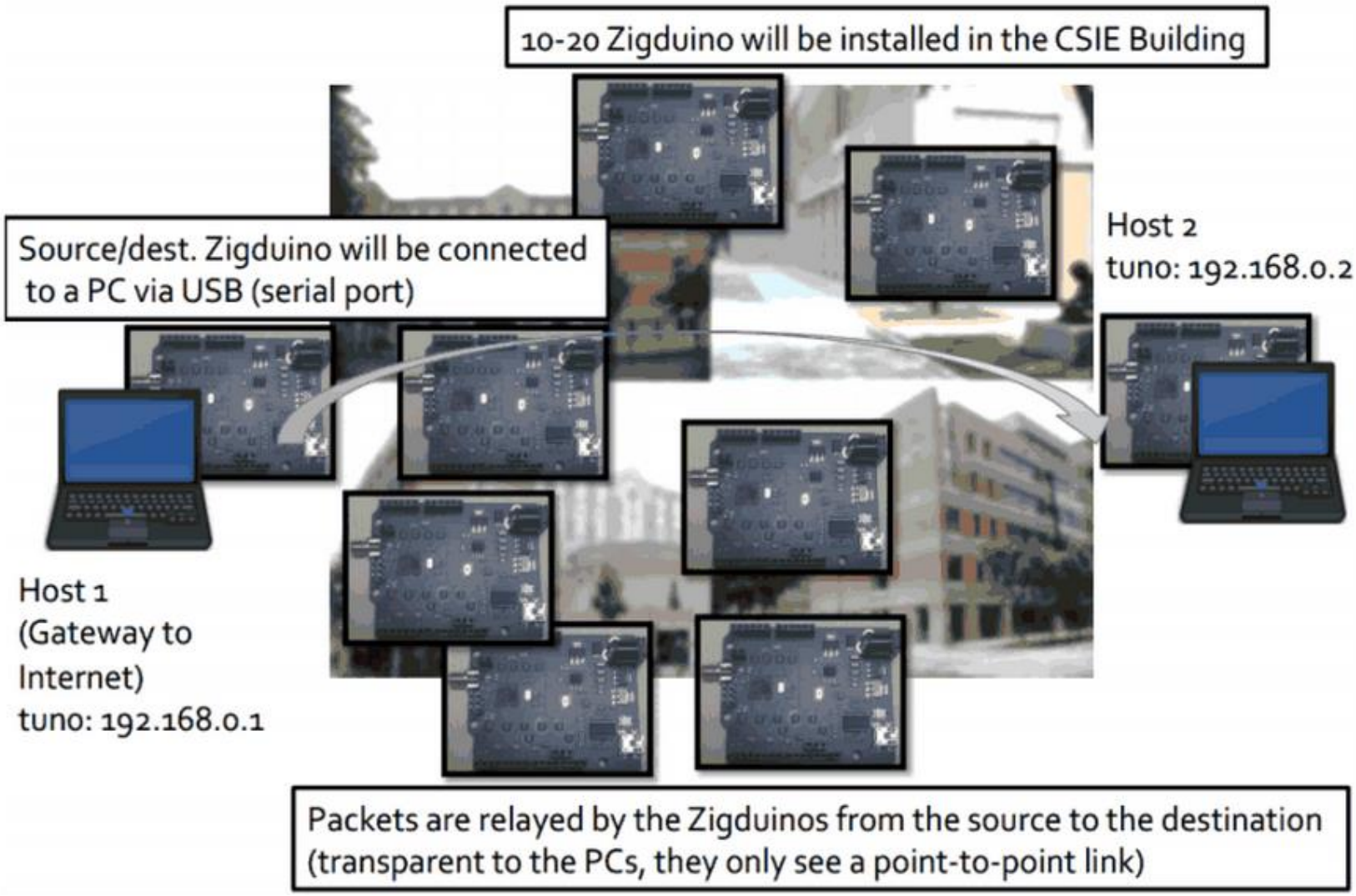
[2015-Fall] WNFA lab4 Wireless Sensor Network

2015/5/7

Will Tseng

Project Scheme



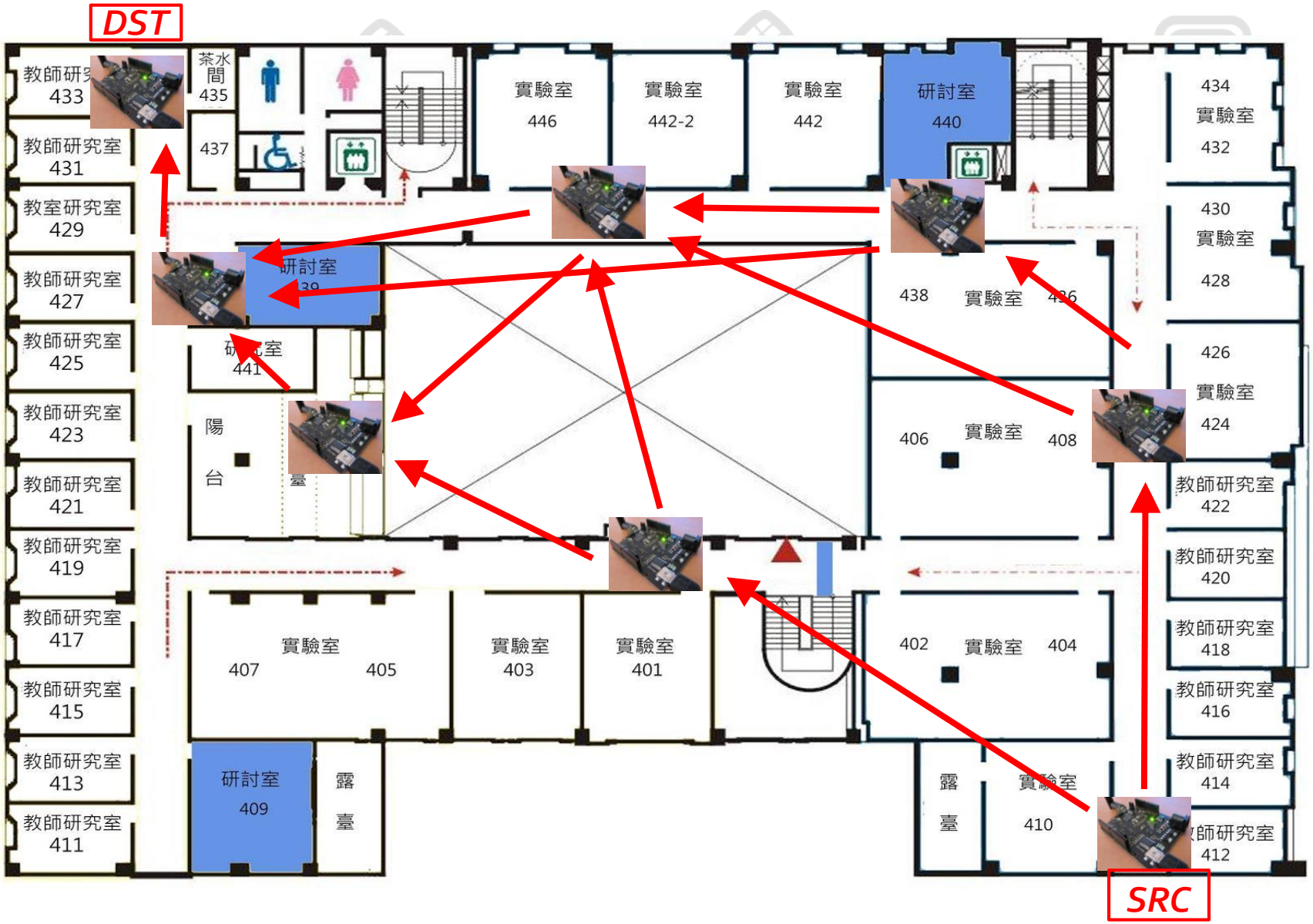
Device Map - CSIE 4F

Mobile and Vehicular Network Lab

Device Map - CSIE 4F

Mobile and Vehicular Network Lab

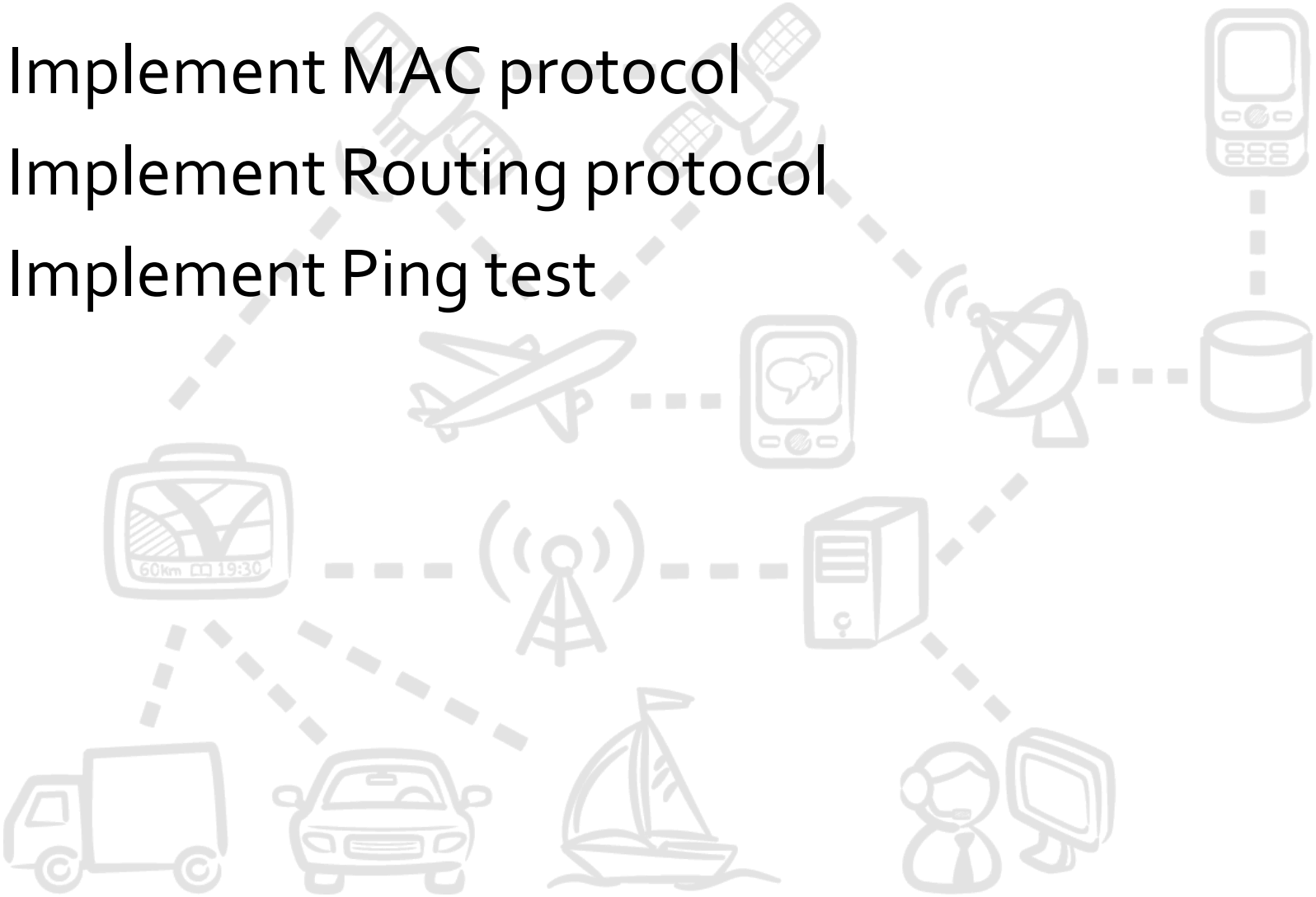
Mobile and Vehicular Network Lab



- Each group will get
  - 2 Zigduinos
  - 2 USB Line
  - 2 Antenna
- We have placed 2 sets at 4F for testing
  - lab4 routes reservation time slot
  - Use your own 2 Zigduino for src, dst.

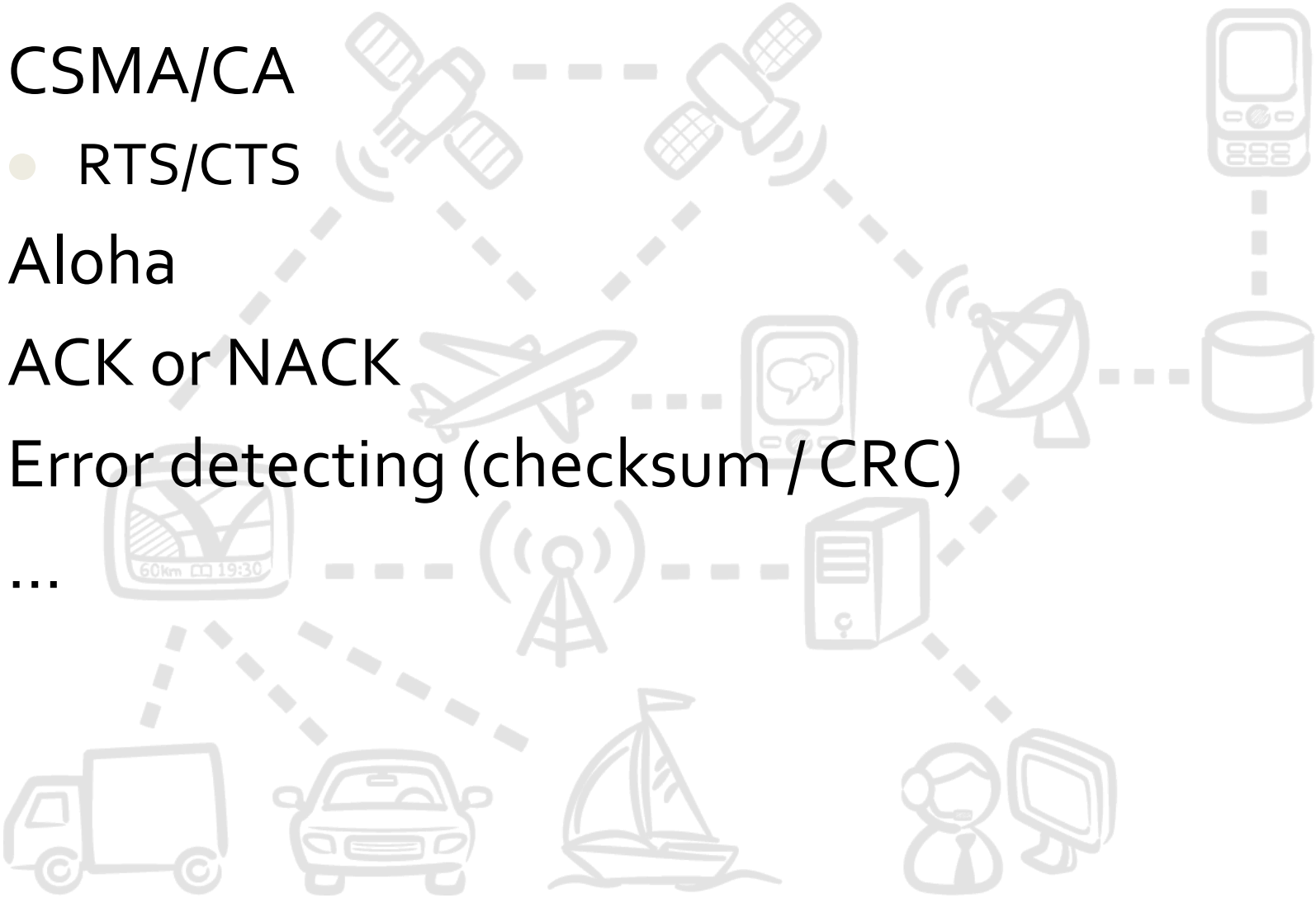
What You Need to Do...

- Implement MAC protocol
- Implement Routing protocol
- Implement Ping test



MAC Protocol

- CSMA/CA
  - RTS/CTS
- Aloha
- ACK or NACK
- Error detecting (checksum / CRC)
- …



Routing Protocol

- AODV
- DSR
- DSDV
- …
- Ref : Routing 2014.pdf

Ping Tests


```
huiyu@MBA:~> [18:10] ping -c 4 140.112.91.208
PING 140.112.91.208 (140.112.91.208): 56 data bytes
64 bytes from 140.112.91.208: icmp_seq=0 ttl=62 time=10.124 ms
64 bytes from 140.112.91.208: icmp_seq=1 ttl=62 time=15.155 ms
64 bytes from 140.112.91.208: icmp_seq=2 ttl=62 time=13.931 ms
64 bytes from 140.112.91.208: icmp_seq=3 ttl=62 time=1.963 ms

--- 140.112.91.208 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.963/10.293/15.155/5.155 ms
```

Ping Tests


```
huiyu@MBA:~> [18:10] ping -c 4 140.112.91.208
PING 140.112.91.208 (140.112.91.208): 56 data bytes
64 bytes from 140.112.91.208: icmp_seq=0 ttl=62 time=10.124 ms
64 bytes from 140.112.91.208: icmp_seq=1 ttl=62 time=15.155 ms
64 bytes from 140.112.91.208: icmp_seq=2 ttl=62 time=13.931 ms
64 bytes from 140.112.91.208: icmp_seq=3 ttl=62 time=1.963 ms

--- 140.112.91.208 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.963/10.293/15.155/5.155 ms
```

Environment Installation

- Platform
  - Zigduino hardware + Arduino software
- Steps :
  1. Follow lab1 Zigduino+Arduino installation guide
  2. Download ZigduionoRadio library on course website
  3. Move ZigduionoRadio directory to
     PATHWHEREARDUINOINSTALLED/Arduino/libraries (for Window)
     ~/Document/Arduino/libraries (for OS X)
  4. **Restart** Arduino Software, then you can see 2 examples through "File -> Examples -> ZigduinoRadio"

ZigduinoRadio

- Arduino library
  - Allow users using Zigduino's built-in 2.4 GHz radio transceiver easily , simple API for radio propogation
- Examples
  - ZigduinoRadioExample
  - SpectrumAnalyzer

HelloWorld

- Compile and upload ZigduinoRadioExample for both Zigduino board
- Modify NODE_ID(line10)and receiver address (line60) pkt_Tx(addr,teststr)
  - EX
    - Board 1 : NODE_ID 0x0001, pkt_Tx(0x0002, teststr)
    - Board 2:NODE_ID 0x0002, pkt_Tx(0x0001, teststr)
- Then you will see the transmitting and receiving messages between them
  - Take Board 1'Serial Terminal as an example

```
-------------------------------------------------
|TxDone: 0 (OK) |
| |
| Rx: .....TT 2.... |
| LQI: 255, RSSI: -37 dBm, ED: -90 dBm |
| |
| ca fail with rssi = -87 |
| |
| TxDone: 1 NO ACK |
| |
| TxDone: 0 (OK) |
| |
| Rx: .....TT 2.... |
| LQI: 255, RSSI: -37 dBm, ED: -90 dBm |
| |
-------------------------------------------------
```

Functions You Should Know

- Application layer examples :
  - examples/ZigduinoRadioExample/ZigduinoRadioExample.pde
  - examples/SpetrumAnalyzer/SpectrumAnalyzer.pde
- For detail :
  - ZigduinoRadio.cpp
  - ZigduinoRadio.h
  - radio_rfa.c
  - radio.h

API - Serial Port Functions

- **Use Tools -> Serial Monitor**
  - Zigduino reset itself when press the "reset" button.
- **Serial I/O functions**
  - Serial.begin(int baudrate) // default 9600
  - Serial.available()
  - Serial.read()
  - Serial.print()
  - Serial.println(uin8_t inchar)
- Zigduino has small buffer - crash if too fast I/O
  - Solution : **higher baudrate** or **call delay()**

API - About Packet TX

- uint8_t pkt_Tx(uint16_t addr, uint8_t* msg, uint8_t pkt_len)
  - In ZigduinoRadioExample.pde
  - A simple packet transmit function.
  - Feel free to modify it.
  - Set addr oxffff for broadcast.
  - Return 1 if get HW ACK. (broadcast will get 0)

API - About Packet TX

- ZigduinoRadio.attachTxDone(void(*funct)(radio_tx_done_t))
- onXmitDone(radio_tx_done_t x)
  - Register a handler
  - Called after a packet delivery.
  - You should check tx status in that handler.
  - (note: The TX_CCA_FAILED isn't implemented)

API - About Packet RX

- ZigduinoRadio.attachReceiveFrame(uint8_t* (*funct)(uint8_t, uint8_t*, uint8_t, uint8_t))
- uint8_t* pkt_Rx(uint8_t len, uint8_t* frm, uint8_t lqi, uint8_t crc_fail)
  - pkt_Rx() is called automatically when Zigduino got a packet. No calling by yourself.
  - pkt_Rx() will put packet in RxBuffer , then set RX_available.
  - Use has_RX() to check if got packet
  - uint8_t fcs_failed : a variable indicates incorrect packet

API - About Packet TX/RX

- The whole process :
  - TX :
    - set TX_available=1 → need_TX() → pkt_Tx() → Fill payload → Carrier Sense → txFrame() → onXmitDone() → Finished when receiving HW ACK ,or retransmitting until > retry times)
  - RX :
    - (Hardware and pkt_Rx() will automatically run and send HW ACK) →Use has_RX() to Check RX_available==1 or not →Data is in RxBuffer

API - About Protocol, Settings (p.1)

- ZigduinoRadio.begin(channel_t chan, uint8_t* frameHeader)
  - Initilizer, 11 <= channel <= 26.
  - You can see correspond frequency list in SpectrumAnalyzer.pde (the other example).
  - Header must follow the IEEE802.15.4 standard.

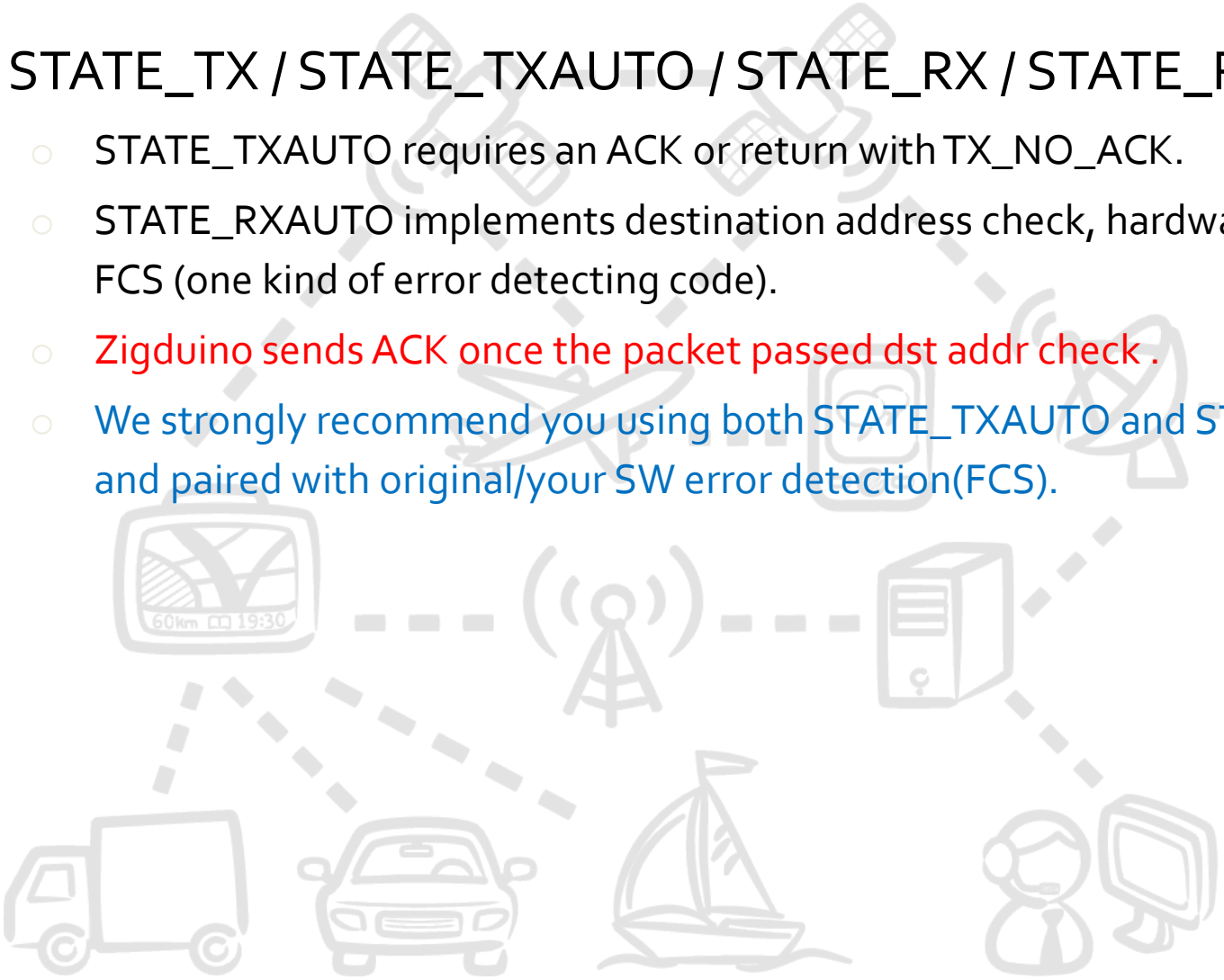
IEEE802.15.4 Header

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source PAN Identifier	Source Address	Auxiliary Security Header	Frame Payload	FCS
		Addressing fields						
MHR							MAC Payload	MFR

Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame Type	Security Enabled	Frame Pending	Ack. Request	PAN ID Compression	Reserved	Dest. Addressing Mode	Frame Version	Source Addressing Mode

API - About Protocol, Settings (p.2)

- STATE_TX / STATE_TXAUTO / STATE_RX / STATE_RXAUTO.
  - STATE_TXAUTO requires an ACK or return with TX_NO_ACK.
  - STATE_RXAUTO implements destination address check, hardware ACK, but no FCS (one kind of error detecting code).
  - Zigduino sends ACK once the packet passed dst addr check .
  - We strongly recommend you using both STATE_TXAUTO and STATE_RXAUTO, and paired with original/your SW error detection(FCS).



API - About Protocol, Settings (p.3)

- Error detection schemes
  - checksum, CRC, FCS, ……
  - Hardware FCS is broken, please ignore that
  - uint16_t cal_fcs(uint8_t* frm, uint8_t len)
    - in ZigduinoRadioExample.pde
    - kind of checksum, very simple
    - not good in some situations

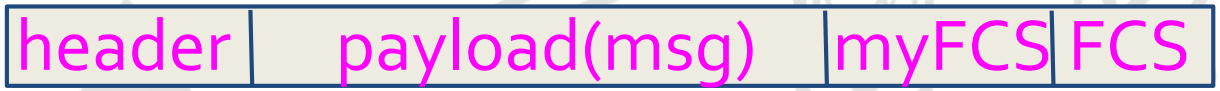
API - About Protocol, Settings (p.4)

- MAC layer design already in ZigduinoRadioExample.pde
  - *#define NODE_ID 0x0001 // node id of this node. change it with different boards*
  - *#define CHANNEL 26 // check correspond frequency in SpectrumAnalyzer*
  - *#define TX_DO_CARRIER_SENSE 1*
  - *#define TX_SOFT_FCS 1*
    - They are defined in ZigduinoRadioExample.pde, feel free to modify it
    - Set 0 as unused, 1 as used
    - Change NODE_ID with different boards (used as node id)
- What you can implement
  - Retransmission
  - Error indicating packet - RX got bad packet, then tell TX retransmitting latest packet.
  - .....

API - About Protocol, Settings (p.5)

- ZigduinoRadio.getRssiNow()
- cZigduinoRadio::doCca()
  - for RSSI and carrier sense
  - Should add *"ZigduinoRadio.setParam(phyCCAMode, (uint8_t)3);"* in setup().
  - You can implement CSMA using these function

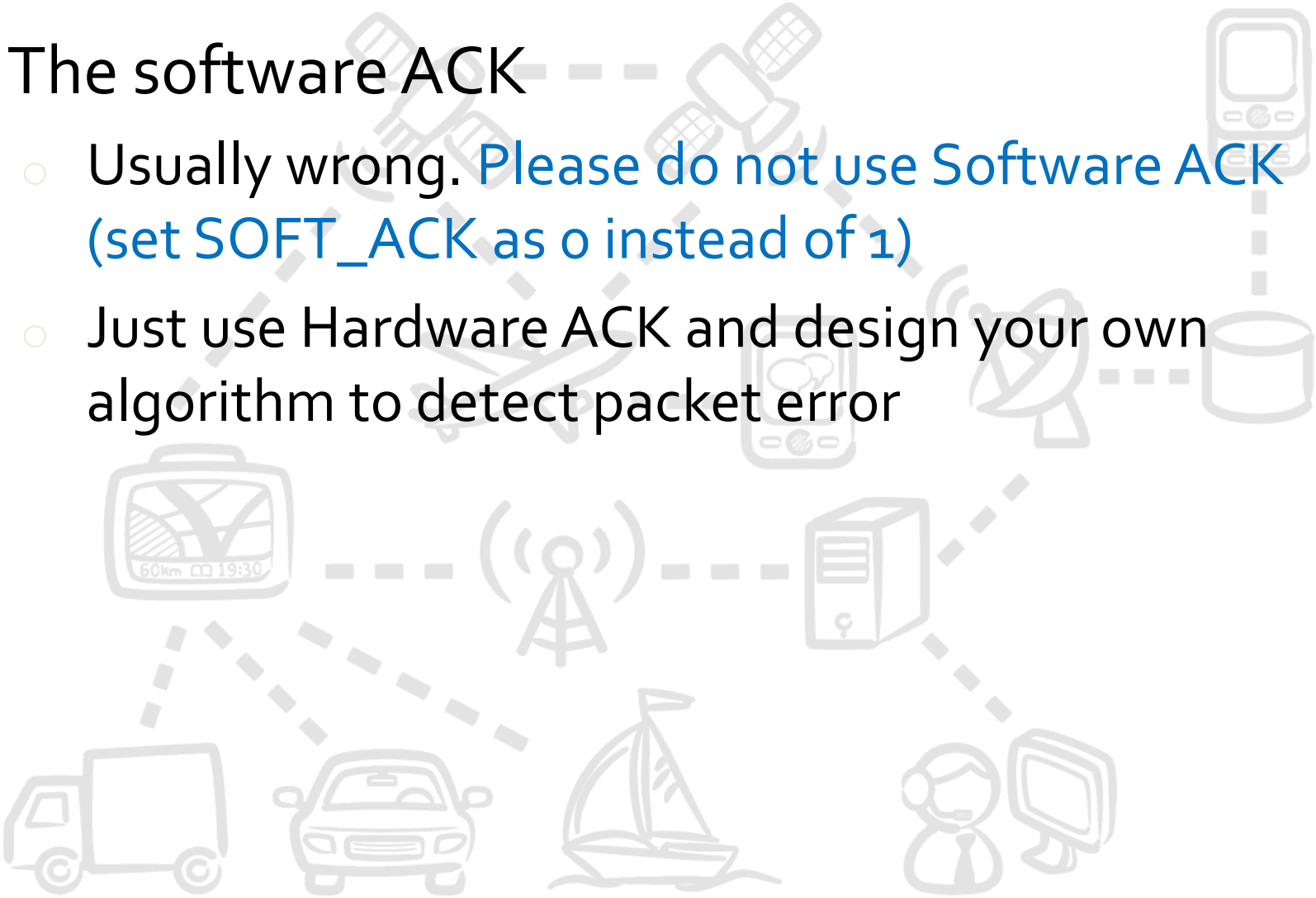
API - Problems

- The hardware FCS
  - Usually wrong. Suggest you do it by yourself.
  - EX. Pad 2 bytes after data, and ignore the last 2 bytes in pkt (the original FCS part in the protocol)
  - 

- The hardware ACK
  - The HW ACK have no sender information / FCS.
  - RX still send ACK even if packet content error (FCS failed.)

API - Problems

- The software ACK
  - Usually wrong. Please do not use Software ACK (set SOFT_ACK as 0 instead of 1)
  - Just use Hardware ACK and design your own algorithm to detect packet error



Ping Tests

- Write your own ping()
- What we focus on :
  - 1. Average RTT (round-trip time)
  - 2. Ping success rate (we'll ping 100 times)
  - 3. the route this pkt has passed through
- Calculate and print them in your code
  - show above information during pinging

Calculate Consumed Time

Use millis() to get the system time

Review - all you have to do (1)

- MAC protocol
  - change CHANNEL
  - design carrier sense and retry machenism
    - modify pkt_Tx() function
    - random backoff time
  - STATE_TX(AUTO), STATE_RX(AUTO)
    - use hardware ACK
    - modify time period waiting ACK
      - add delay in ISR(TRX24_TX_END_vect) in radio_rfa.c
  - design error detecting code (FCS / checksum / …)
  - ……

Review - all you have to do (2)

- Routing table - the algorithm
  - "reset" packet (clear table quickly, save time :P)
  - Route rebuild
  - Traffic estimate: packet send rate / packet loss rate
  - route optimize : ETX (hop count vs. success rate)
  - .....

Review - all you have to do (3)

- The ping test
  - packet design (sequence number, get route, ping period)
  - timing to rebuild path ?
  - calculate time
  - whole I/O design
    - reset routing table
    - start ping
    - print time+route
- Tips
  - Use LED to debug !!!
  - Note the "reset" button on Zigduino board

GRADE

Grade

- Deadline:5/28(Thur.) 23:59 **email to wn@csie.ntu.edu.tw**
  - Email subject:[WN]lab4_teamXX
  - [WN]lab4_teamXX.zip
    - Source code and Modified Library Files (ex ZigduinoRadio.cpp)
    - Report(pdf)
- **Demo-5/28(Thur.)**
  - Please register the demo slot. And come to CSIE R424 at that time. http://ppt.cc/mA4Zs

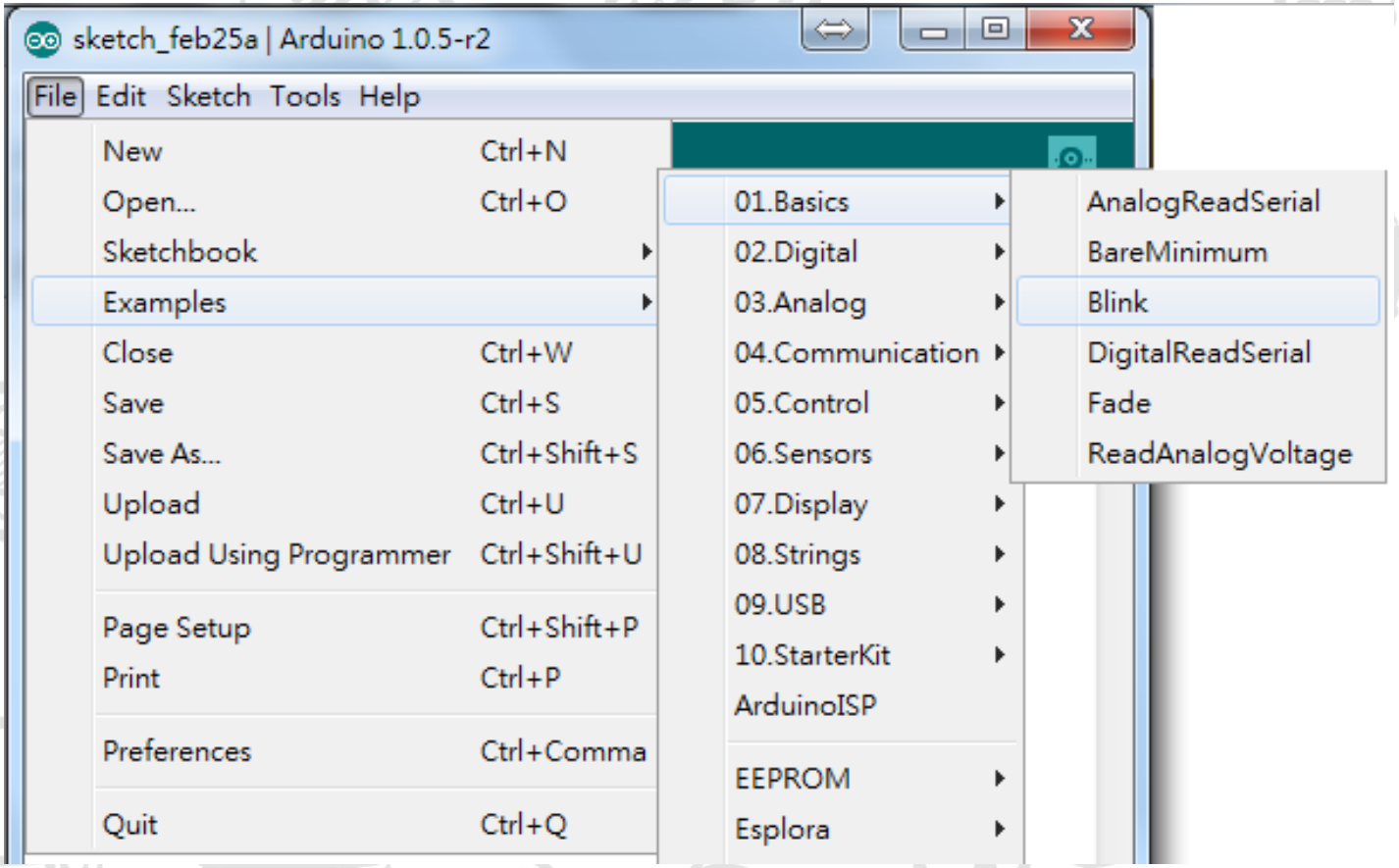
Grade

- Demo - the ping test (70%)
  - 40% : base line, a success ping
  - 10% : average RTT, get by ranking
  - 10% : success rate (100 pings, 0.2% for each sec)
  - 10% : path rebuild after removing 1 node
- Report (20%)
  - Describe what you have done, why choosing this way, and problems encountered
- Peer Review (10%)
- Bonus (5%)
  - any other things you have done

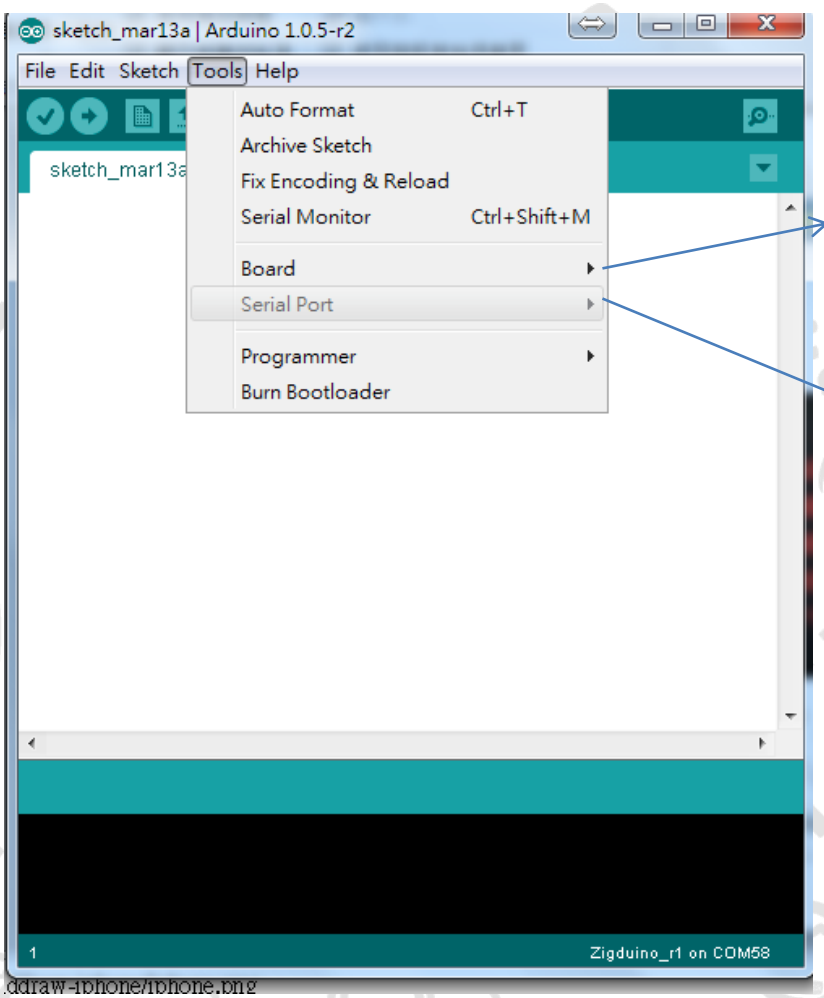
- Contact to TAs :
  - facebook https://www.facebook.com/groups/wn15spring/
  - Email : wn@csie.ntu.edu.tw
  - Office hour : Thur. 16:00~17:00@ CSIE R424

APPENDIX

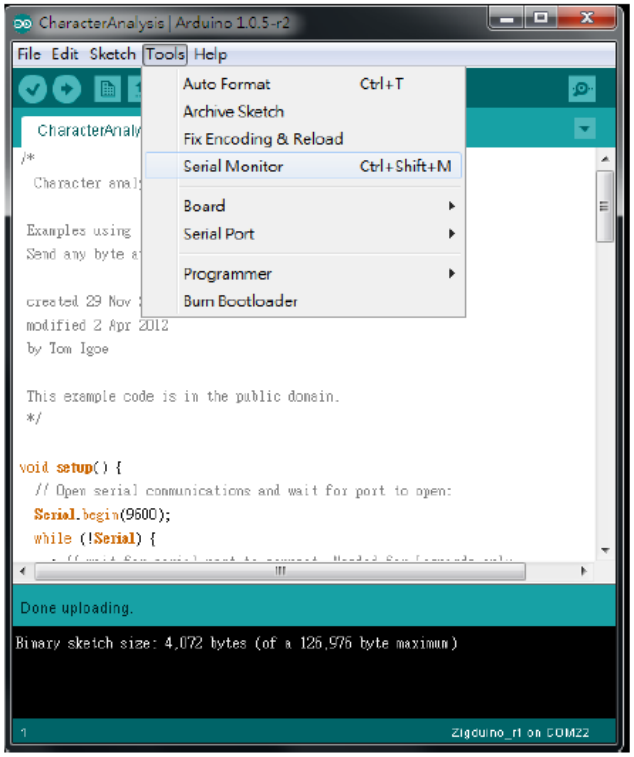
- First program – Blink.ino

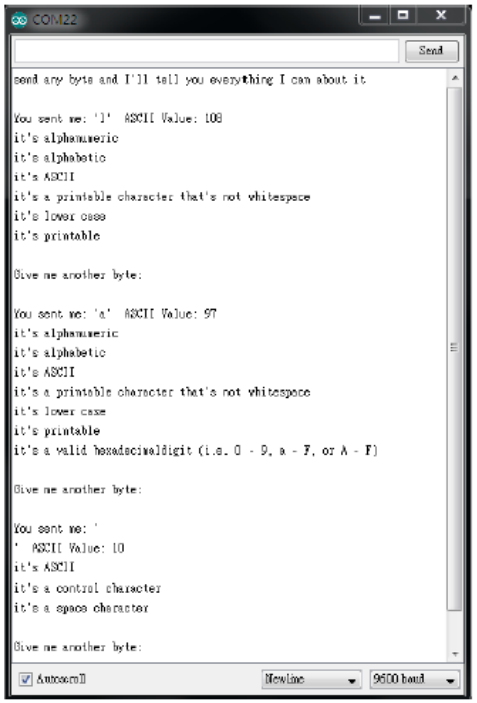


Zigduino


sketch_mar13a | Arduino 1.0.5-r2
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
sketch_mar13a
1
Zigduino_r1 on COM58
Zigduino
COMx

- serial port monitor – text I/O
  - see Examples 08. Strings
  - Use the serial monitor to communicate





Reference

- Zigduino-radio
  - https://code.google.com/p/zigduino-radio
  - http://www.daxia.com/bibis/upload/ZigDuino%B5%C4Arduino%D1%A7%CF%B0%B1%CA%BC%C7.640.pdf
- Error detecting code
  - FCS
  - http://cs.nju.edu.cn/yangxc/dcc_teach/fcs-calc.pdf
  - CRC
  - checksum
  - http://en.wikipedia.org/wiki/Checksum

Q&A