Aug 2, 2012: Draft
Sep 11, 2012: Release

Building Wireless Sensor Networks with XBee and Arduino

The University of Tokushima
Akinori TSUJI

Contact Information :
2-1, Minamijyosanjima-cho, Tokushima, 770-8506, Japan
TEL/FAX : +81-88-656-7485
E-mail: : a-tsuji@is.tokushima-u.ac.jp

Setup Environment

Day 1

2012/9/11(Tue) 10:00—12:00
Time Estimation: 2 hours

Agenda

1 Introduction

2 Wireless Sensor Networks

3 Building XBee Networks
- Setup software
- XBee modules
- Pair Network, 1 by 1 connection

1 Introduction

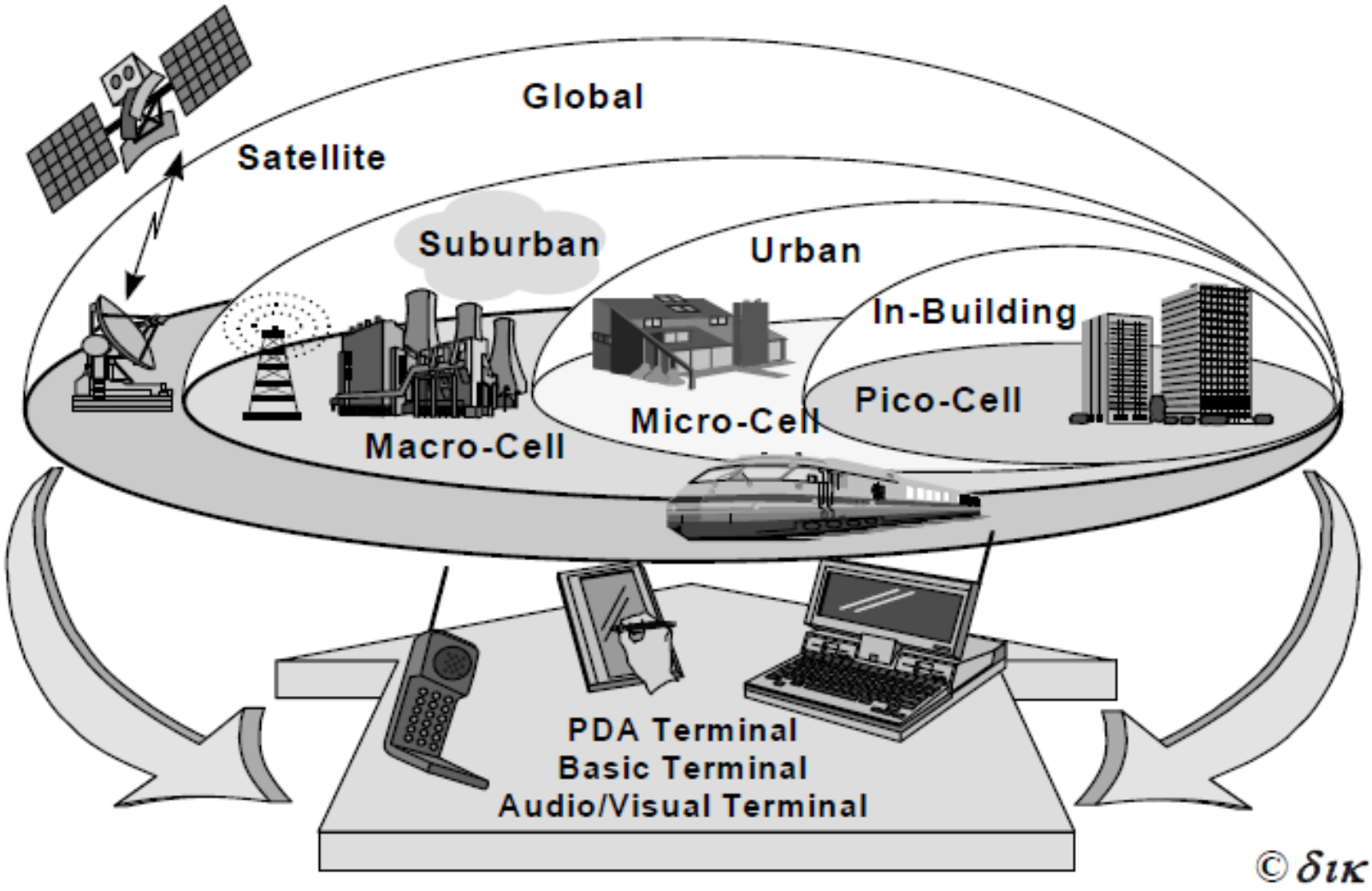
Wireless Sensor Networks

- Distributed sensor system
- Intelligent interactive devices
- Building robust network

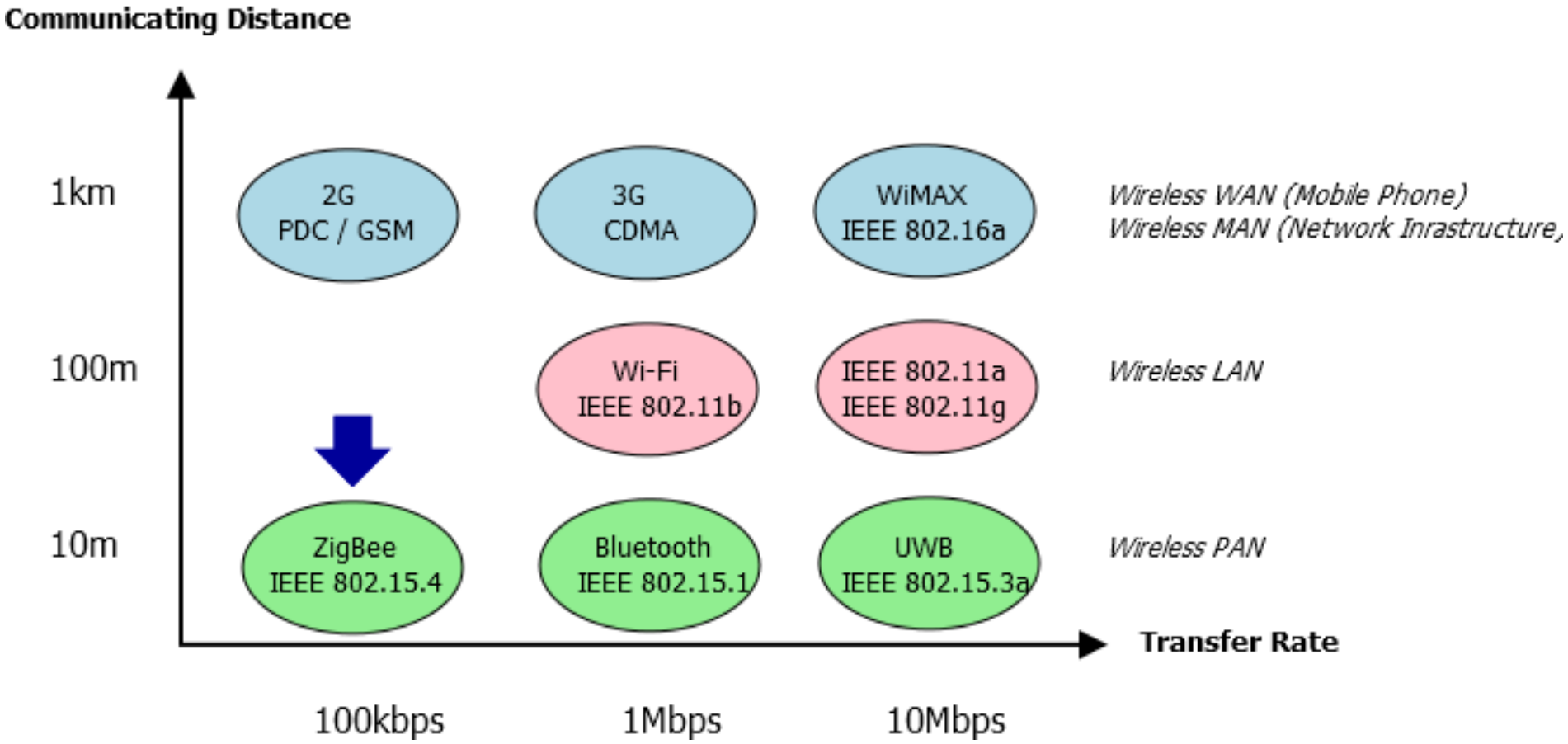
Applications

- Environmental monitoring (Earthquake, Tsunami, Weather, etc.)
- Clean energy (Smart meter, Energy management, etc)
- Consumer electronics (Medical device, Game, etc)
- Robots internal connection (Humanoid robot, Swarm Robot, Aerial Vehicle, etc)
- and more

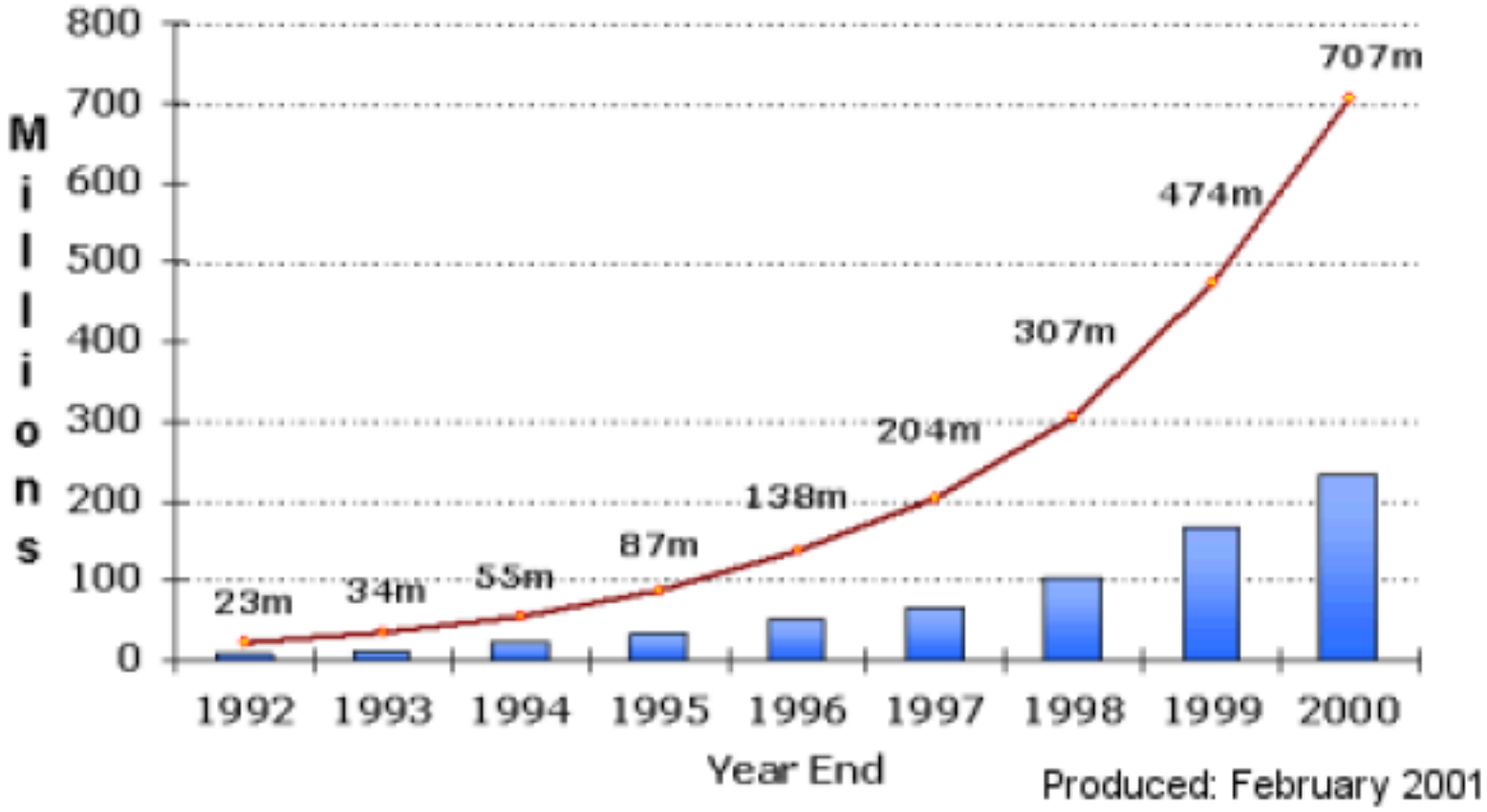
1 Introduction



1.1 Standard Wireless Networks



1.2 Subscribers of Mobile Phone



1.3 Specification of Wireless Networks

IEEE Standard	802.15.11b	802.15.1	802.15.4
Market	Wi-Fi	Bluetooth	ZigBee
Frequency	2.4 GHz	2.4 GHz	2.4 GHz
Modulation	CCK, PBCC	GFSK	O-QPSK
Comm. Dist.	100m	10m-30m	30m
Transfer Rate	11 Mbps	1 Mbps	240 kbps
Network Cap.	32 nodes	7 nodes	65,536 nodes
Battery Life	Several hours	Several days	Several years
Application	Wireless LAN	Wireless Audio	Measurement and Control

2 Wireless Sensor Networks

Wireless Sensor Networks (WSN)

- Wireless Sensor and Actuator Networks
- including sensors, actuators and wireless network module

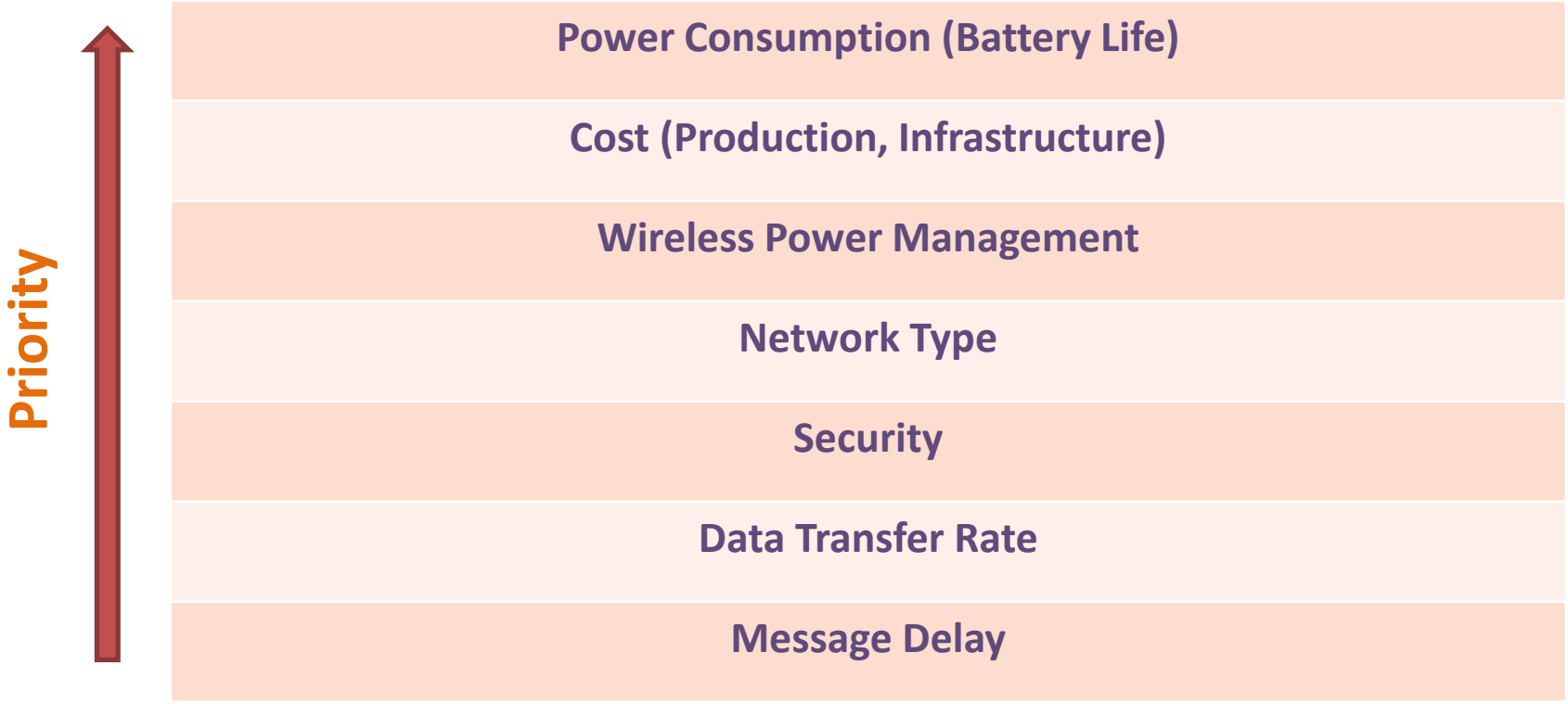
Advantage

- Low power consumption
- Low cost
- Near distance
- Easy to create a wireless network

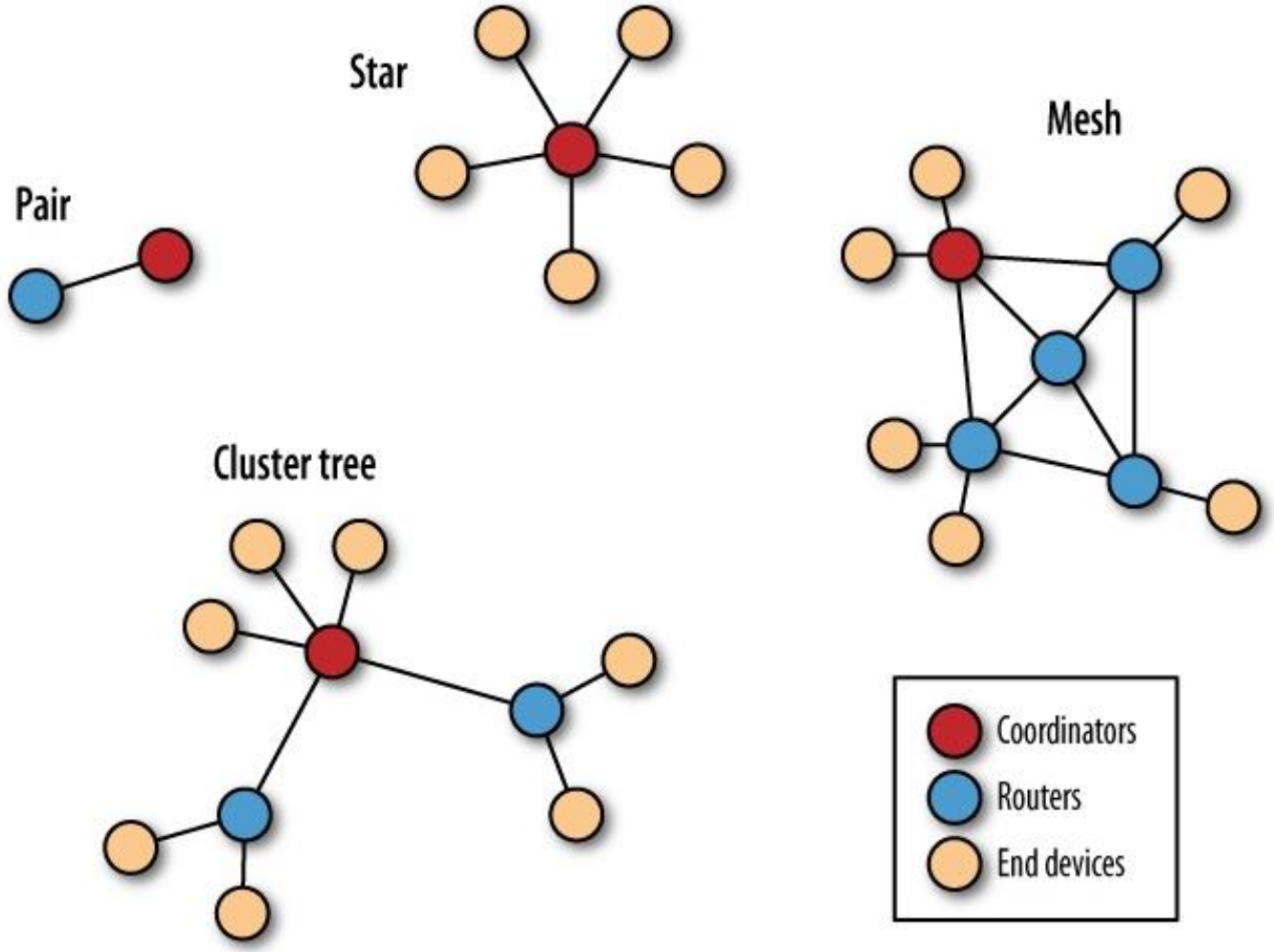
Disadvantage

- Implement protocol stack
- Cannot cover wide area
- Need many nodes

2.1 Requirements of Wireless Sensor Networks



2.2 Network Topology of ZigBee



2.2.1 ZigBee Wireless Networks

Pair

- Simplest network (two nodes)
- One node must be a coordinator
- The other can be configured as a router or an end device

Star

- Coordinator sets the center of the star topology
- Every message must pass through the coordinator
- The end device do not communicate each other directly

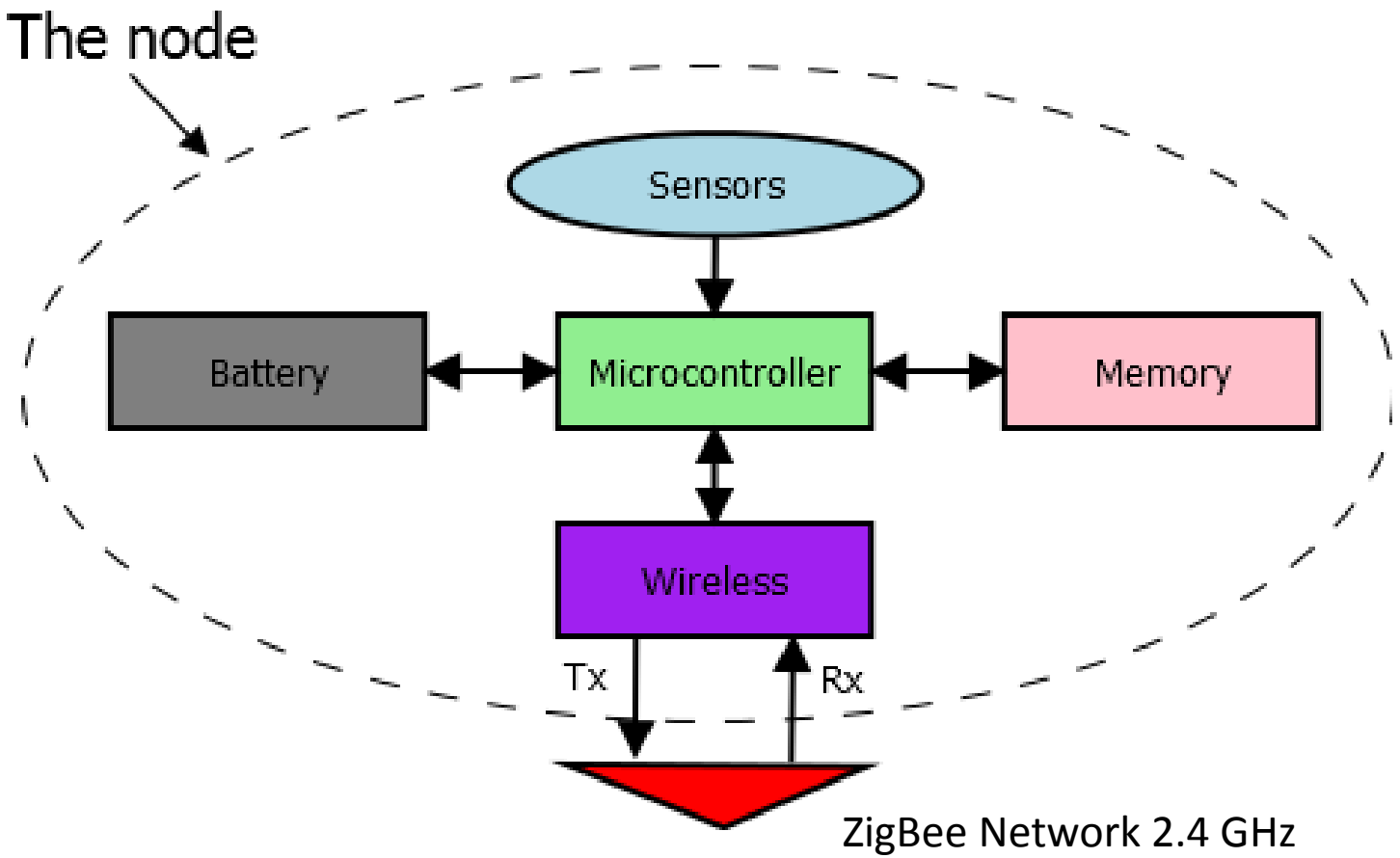
Mesh

- Router nodes in addition to the coordinator
- Coordinator acts to manage the network
- End devices are attached to any router or to the coordinator

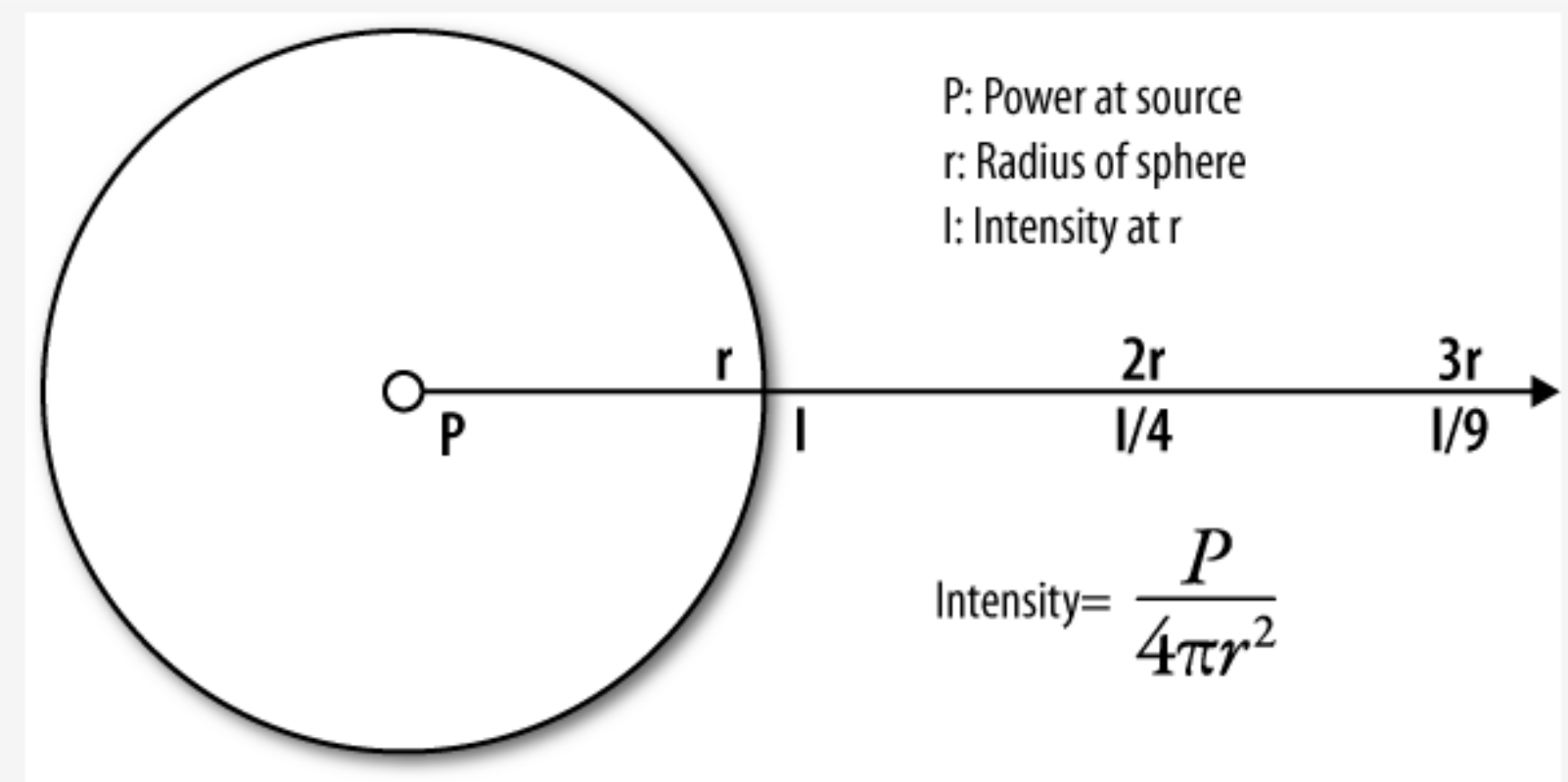
Cluster tree

- Routers from a backbone of sorts with end devices clustered around each router

2.3 Typical Sensor Nodes



2.3.1 Inverse Square Law for Wireless Communication



Parts

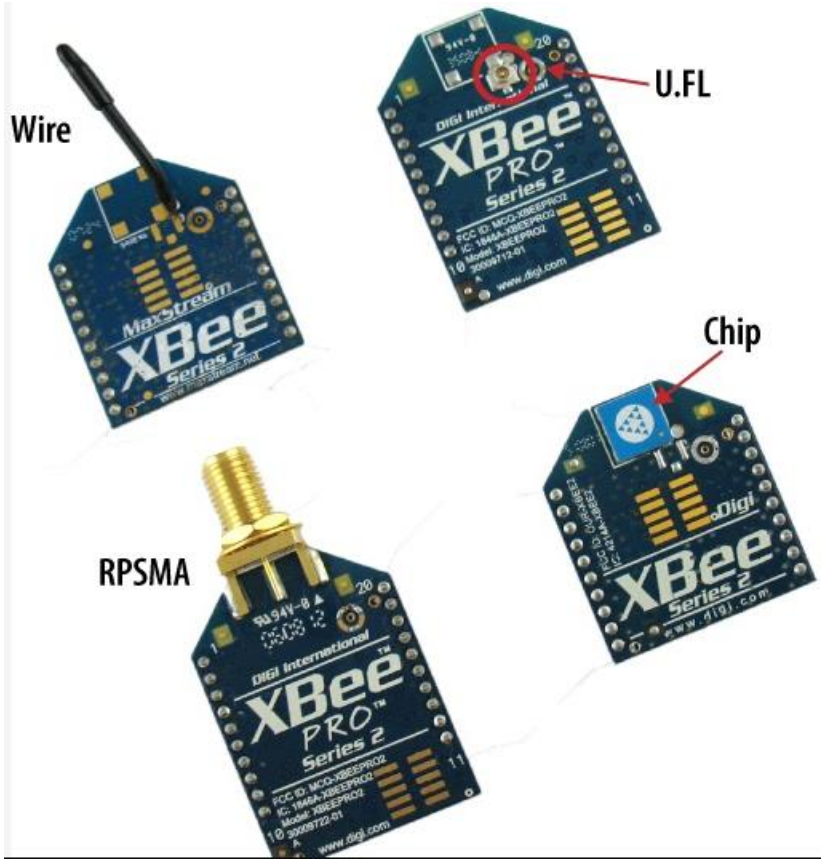
Arduino board compatible with Duemilanove ATmega168/328 x1
XBeeZB Module, Chip antenna x2
XBee 2.54mm conveter x2
Breadboard EIC-801 x1
USB Serial converter cable (3.3V) x1
Battery 006P 6F22 9V x1
Jumper wires x1
LCD Display Module 16x2, no backlight x1
Resistors 10kΩ x4, 4.7kΩ x2, 1kΩ x1
Push switch x1
Piezo Buzzer PKM13EPYH4000-A0 x1
Temperature sensor LM60BIZ (TO-92) -25--+125 x1

3 Building XBee Networks

Specification of XBee module, S2

Typical range (indoor): 40 meters
Best range: 120 meters
Transmit/Receive current: 40/40 mA
Network: ZB ZigBee mesh
Digital In/Out: 11
Analog Inputs: 4
Features: Low power, Low bandwidth, Low cost, Small,
Interoperable mesh routing, Ad-hoc network creation
Point-point, Star network, Mesh network, Cluster network
Easy to update Firmware

3.1 Series of XBee Module



3.2 Pin Assignment of XBee Module

Do NOT supply +5V to the XBee module

1 Vcc (+3.3V)
2 DOUT (TXD)
3 DIN (RXD)
4 DIO12
5 RESET
6 PWM0/RSSI/DIO10
7 DIO11
8 Reserved
9 DTR/SLEEP_RQ/DIO8
10 GND

11 DIO4
12 CTS/DIO7
13 ON/SLEEP
14 VREF
15 ASSOC/DIO5
16 RTS/DIO6
17 AD3/DIO3
18 AD2/DIO2
19 AD1/DIO1
20 AD0/DIO0/COMMIS

3.3 Development Tools for WSNs

Development Tools
- X-CTU for updating the firmware of XBee modules
- TeraTerm for monitoring and logging communication

XBee Boards (X-Cite XBee 24)
- more than 2 set

Development Board
- Arduino with ATMega328P (AVR Corporation)

Flash Writer
- AVR ISP-MKII or USBasp

  Writing Arduino firmware to ATMega328P

3.4 Setup Software

Download and install software

X-CTU (ver. 5.2.7.5)
http://ftp1.digi.com/support/utilities/40003002_B.exe

FTDI Driver (Window 2012-4-26/ x86(32-Bit), x64(64-Bit)
http://www.ftdichip.com/Drivers/VCP.htm

TeraTerm (ver. 4.75)
http://sourceforge.jp/projects/ttssh2/releases/

Arduino (ver. 1.0.1)
http://arduino.cc/hu/Main/Software

3.4 Setup Software

X-CTU

1. Download 40003002_B.exe (X-CTU program) from above direct link.
2. Run the setup program by default setting.

* After the installation, run the X-CTU program and apply the latest firmware as the following steps.

1. Run the X-CTU program
2. Click on the "Modem Configuration" tab
3. Click on the "Download and new versions" button
4. Click on the "Web" button

... wait for several minutes to complete the updates

FTDI Driver (2.08.24 Window 2012-4-26/ x86(32-Bit), x64(64-Bit)

1. Download the driver specified for your Windows OS 32-bit or 64-bit.
2. Extract the archive
3. The driver will be manually installed when you insert the USB-Serial converter cable at first time.

3.4 Setup Software

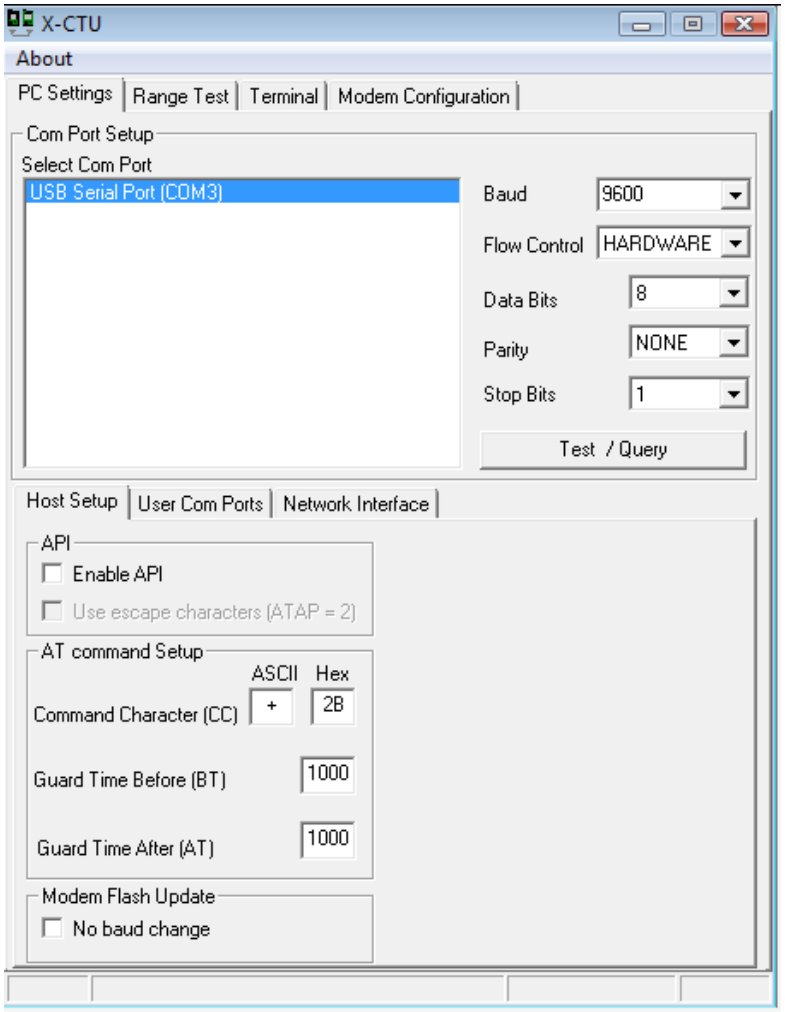
TeraTerm (ver. 4.74)

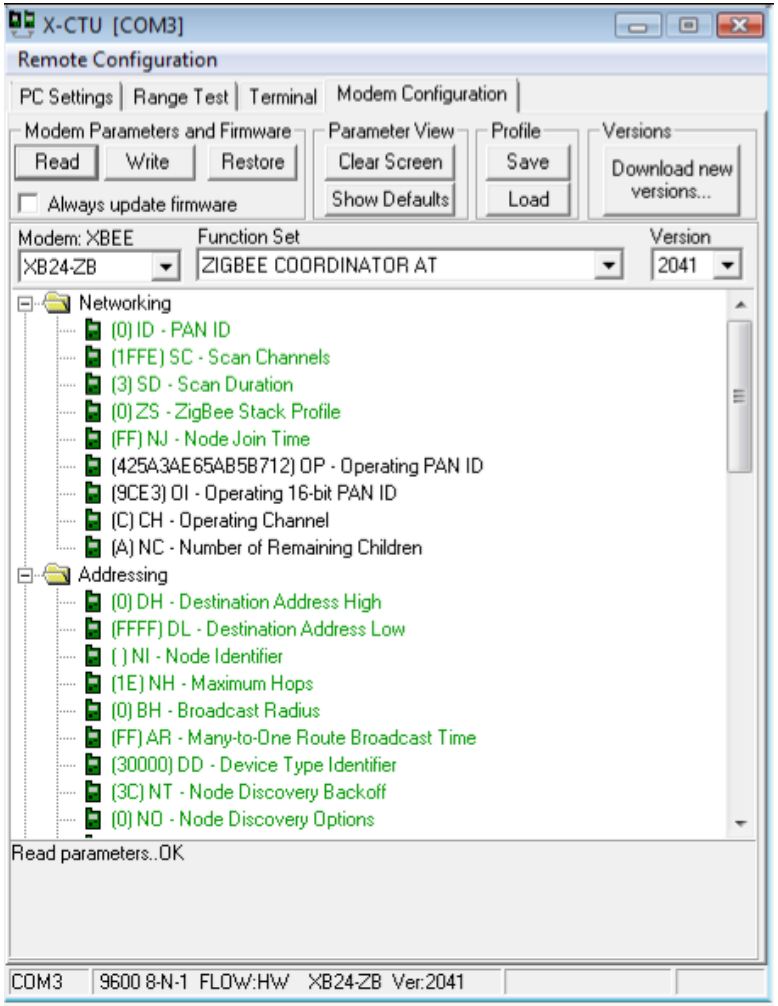
1. Download TeraTerm-4.74.exe or the latest version.
2. Run the setup program by default setting.

Arduino (ver. 1.0.1)

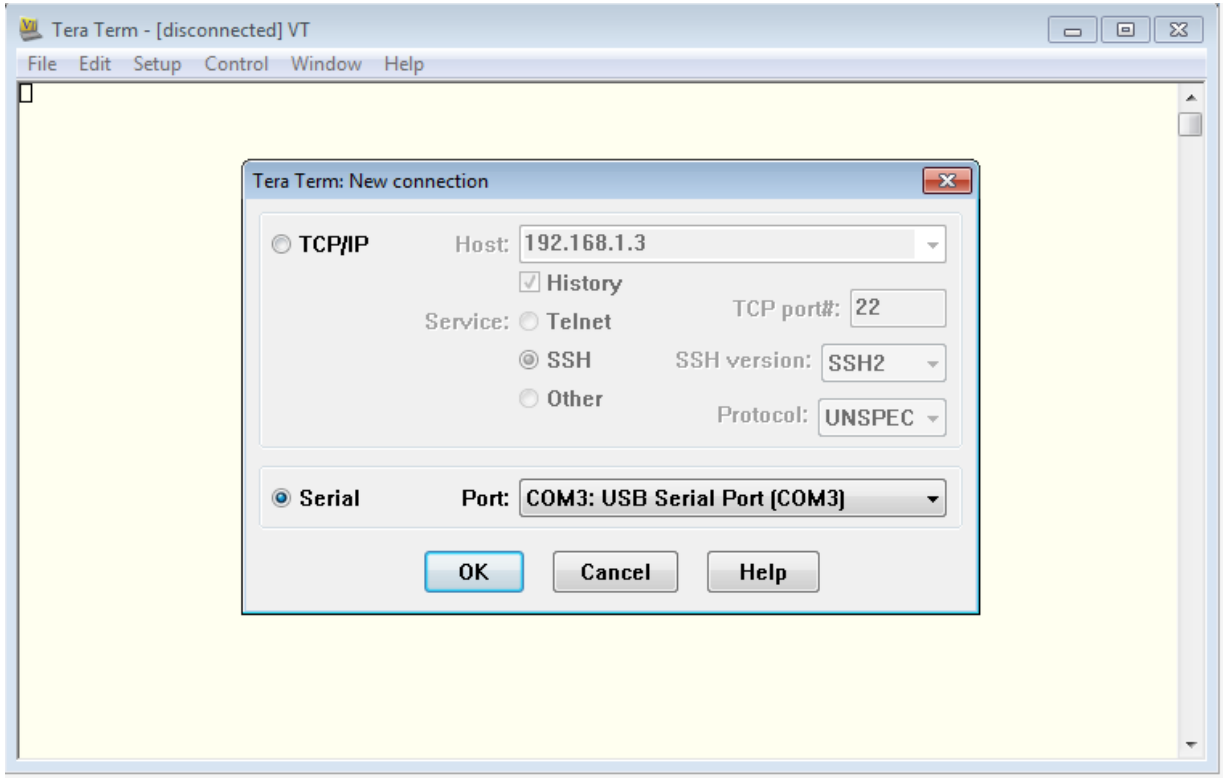
1. Download the Arduino-1.0.1-windows.zip
2. Extract the archive
3. Move the arduino-1.0.1 folder to under the C:¥ drive, then the arduino program is placed on C:¥arduino-1.0.1
4. Make a short cut of C:¥arduino-1.0.1¥arduino.exe on the desktop or your preferred place, for easy to launch the program.

3.4.1 X-CTU


X-CTU
About
PC Settings
Range Test
Terminal
Modem Configuration
Com Port Setup
Select Com Port
USB Serial Port (COM3)
Baud
9600
Flow Control
HARDWARE
Data Bits
8
Parity
NONE
Stop Bits
1
Test / Query
Host Setup
User Com Ports
Network Interface
API
Enable API
Use escape characters (ATAP = 2)
AT command Setup
ASCII
Hex
Command Character (CC)
+
2B
Guard Time Before (BT)
1000
Guard Time After (AT)
1000
Modem Flash Update
No baud change


X-CTU [COM3]
Remote Configuration
PC Settings
Range Test
Terminal
Modem Configuration
Modem Parameters and Firmware
Read
Write
Restore
Always update firmware
Parameter View
Clear Screen
Show Defaults
Profile
Save
Load
Versions
Download new versions...
Modem: XBEE
XB24-ZB
Function Set
ZIGBEE COORDINATOR AT
Version
2041
Networking
(0) ID - PAN ID
(1FFE) SC - Scan Channels
(3) SD - Scan Duration
(0) ZS - ZigBee Stack Profile
(FF) NJ - Node Join Time
(425A3AE65AB5B712) OP - Operating PAN ID
(9CE3) OI - Operating 16-bit PAN ID
(C) CH - Operating Channel
(A) NC - Number of Remaining Children
Addressing
(0) DH - Destination Address High
(FFFF) DL - Destination Address Low
( ) NI - Node Identifier
(1E) NH - Maximum Hops
(0) BH - Broadcast Radius
(FF) AR - Many-to-One Route Broadcast Time
(30000) DD - Device Type Identifier
(3C) NT - Node Discovery Backoff
(0) NO - Node Discovery Options
Read parameters..OK
COM3
9600 8-N-1 FLOW:HW XB24-ZB Ver:2041

3.4.2 Tera Term



Default Setting

Baud	9600
Data	8 bit
Parity	None
Stop bits	1
Flow control	None
Line feed	CR+LF or Auto Line Feed
Local echo	On

3.4.3 Arduino

Arduino development environment

- has started the development in Italy
- easy to use for beginners, no need software or electronics experience
- C / C++ language
- IDE (Integrated Development Environment)
- APIs

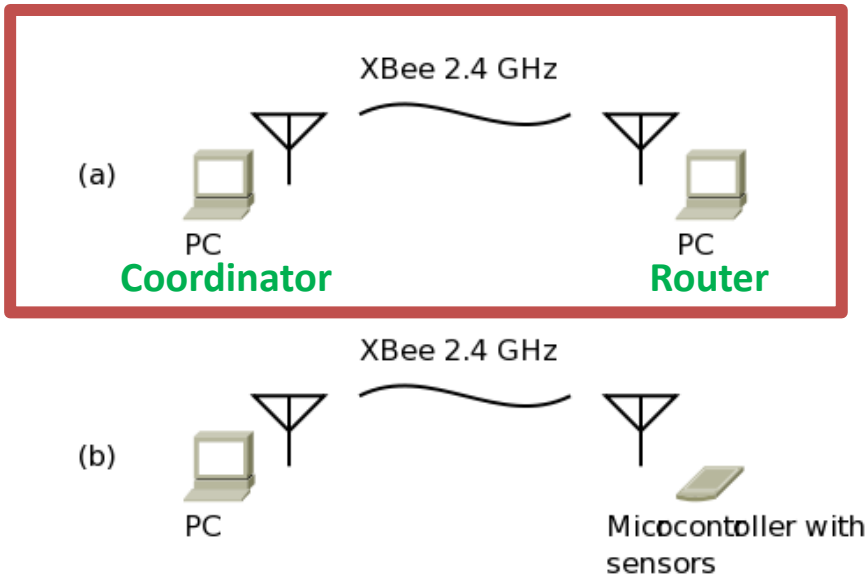
Default Settings:

- **Tools -> Board -> Arduino Duemilianove w/Atmega328**
- **Tools -> Serial Port -> (USB Serial Converter)**
- **Tools -> USBasp**

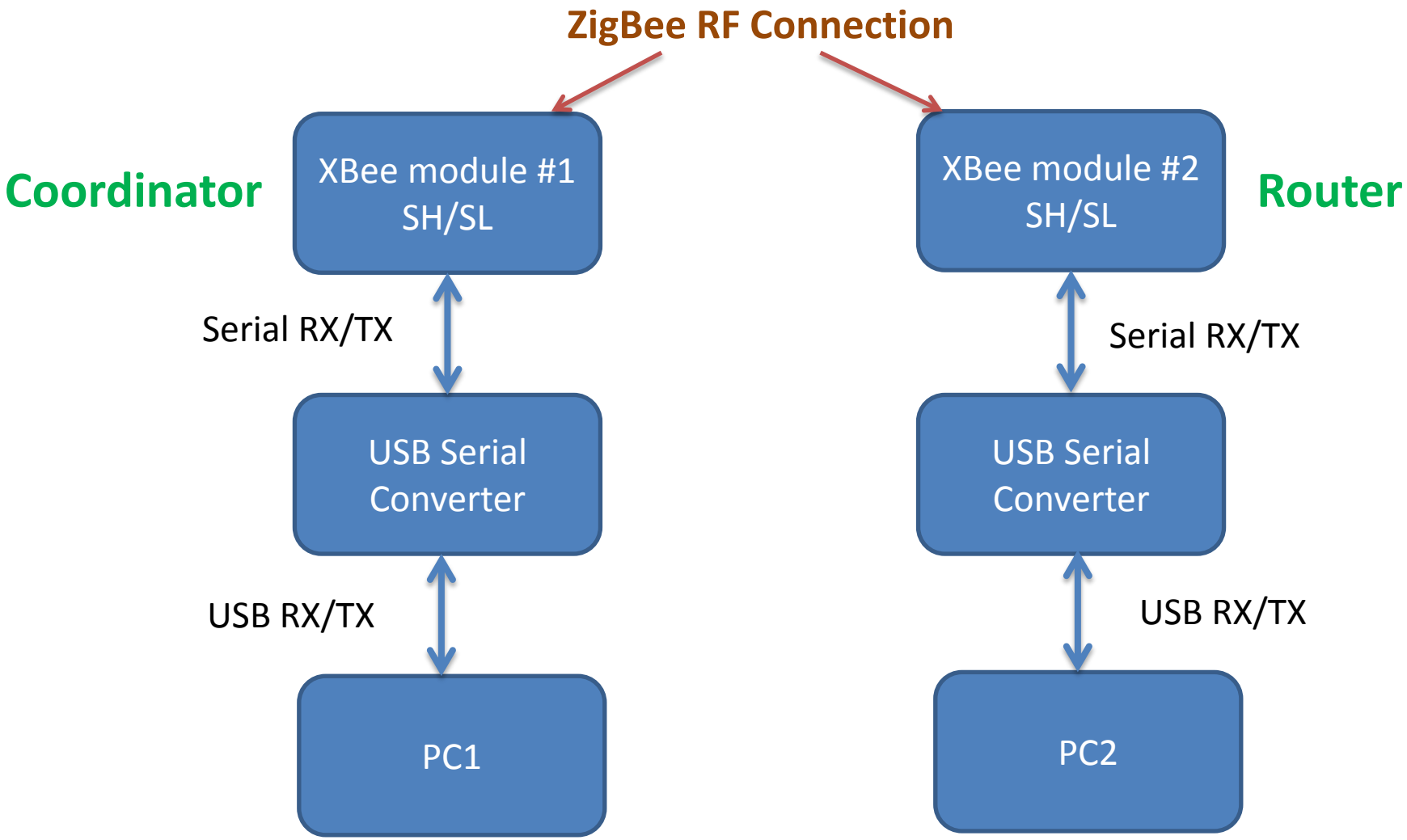
3.5 Test XBee Module

Four Steps to start XBee

1. Check Serial Number of modules
2. Write Firmware for Coordinator or Router/Enddevice
3. Set up the PAN ID and the Destination address
4. Reconnect the USB cable for restart the Xbee
5. Build a Pair Network (1 by 1)



3.5 Test XBee Module



3.5.1 Check Serial Number

Set PAN ID
_____ (64 digits)

Check Serial Number of Modules
Coordinator
SH 0013A200
SL _____

Router or Enddevice
SH 0013A200
SL _____

＊ Mark Coordinator and Router on each module

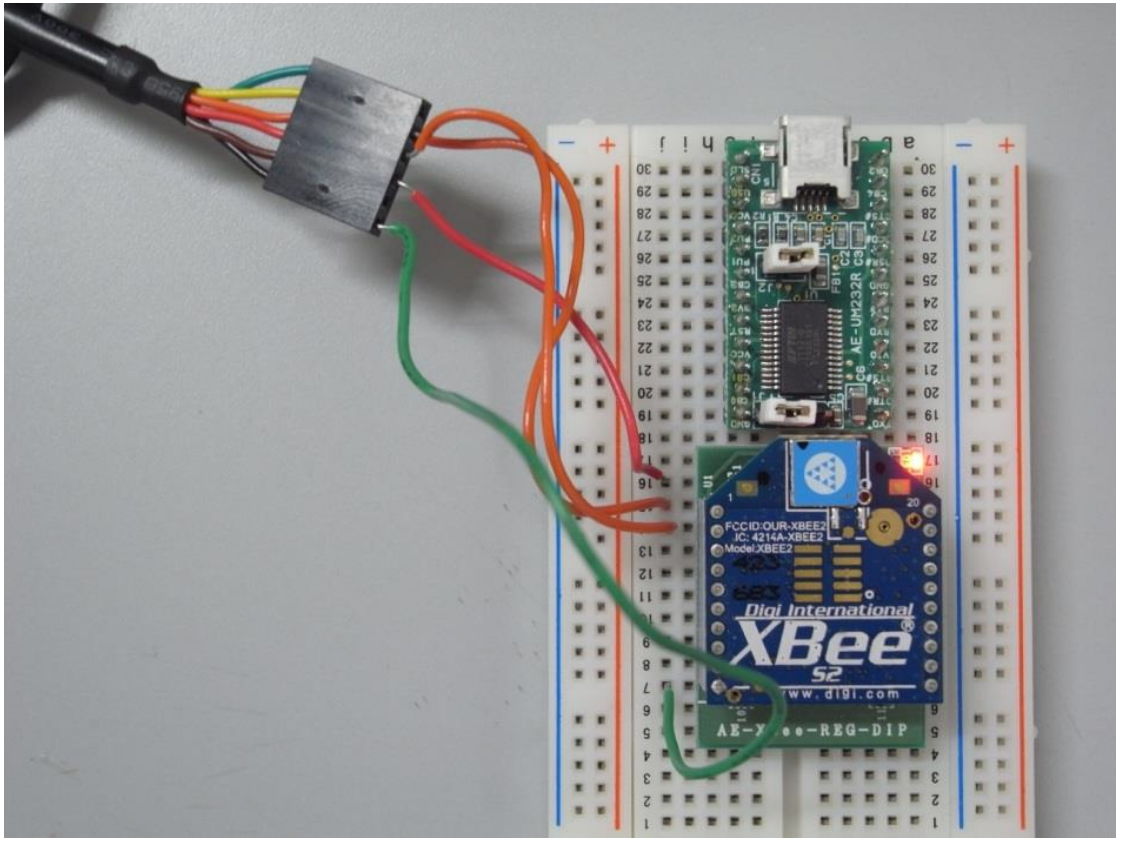
Bottom of Module

3.5.2 Connect USB Cable and Module

Connect the USB Serial cable and XBee module

USB-FTDI	XBee Module
5V (Red)	----- 1 VCC
RXD (Yellow)	<---- 2 DOUT (TX)
TXD (Orange)	----> 3 DIN (RX)
GND (Black)	------10 GND

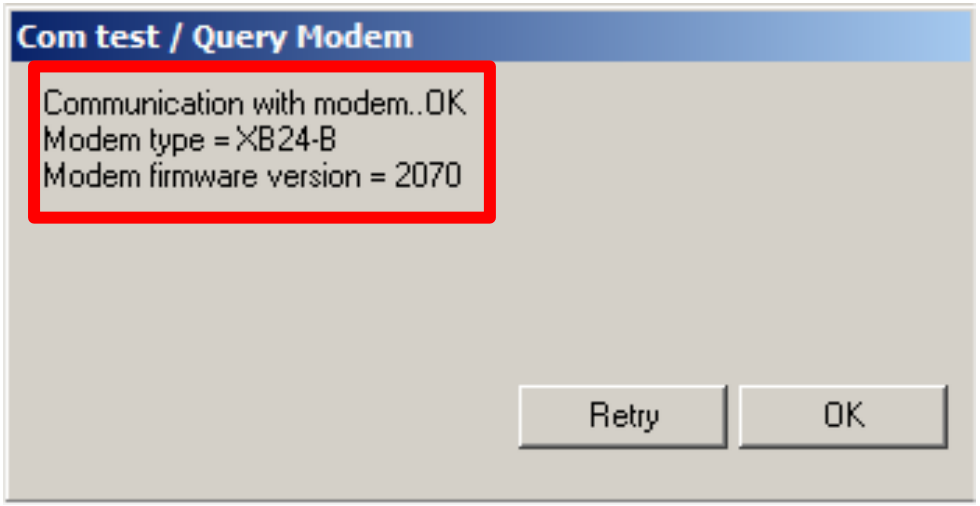
USB-FTDI Cable

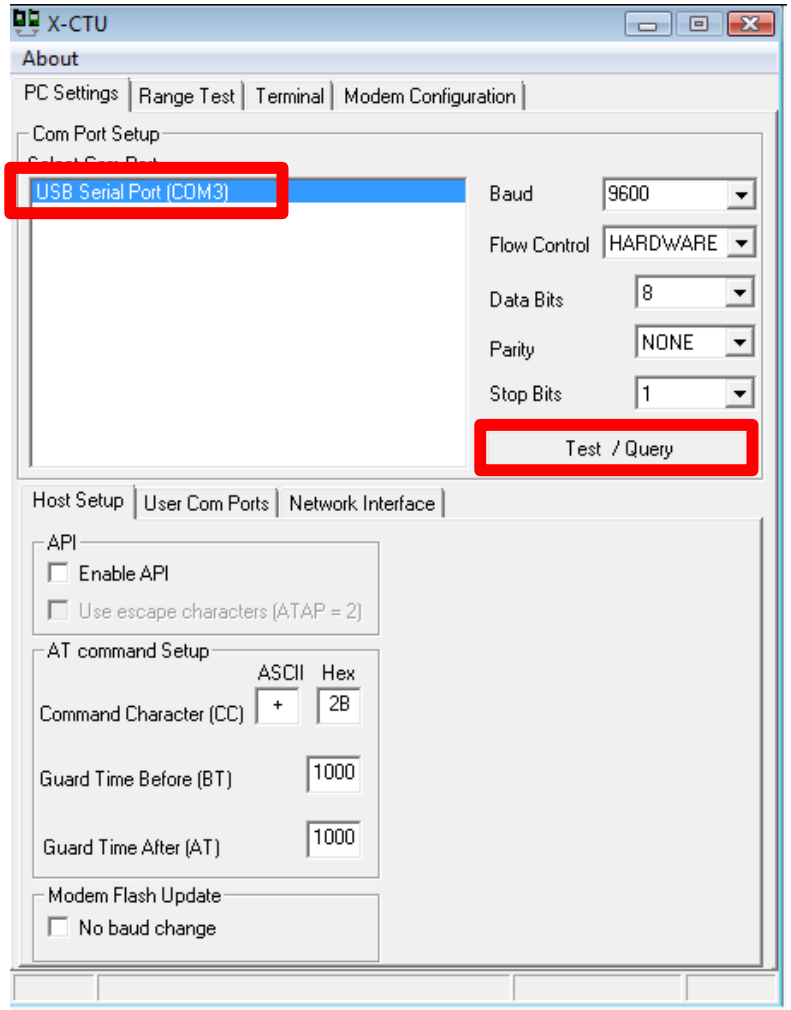


3.5.3 Confirm Connection

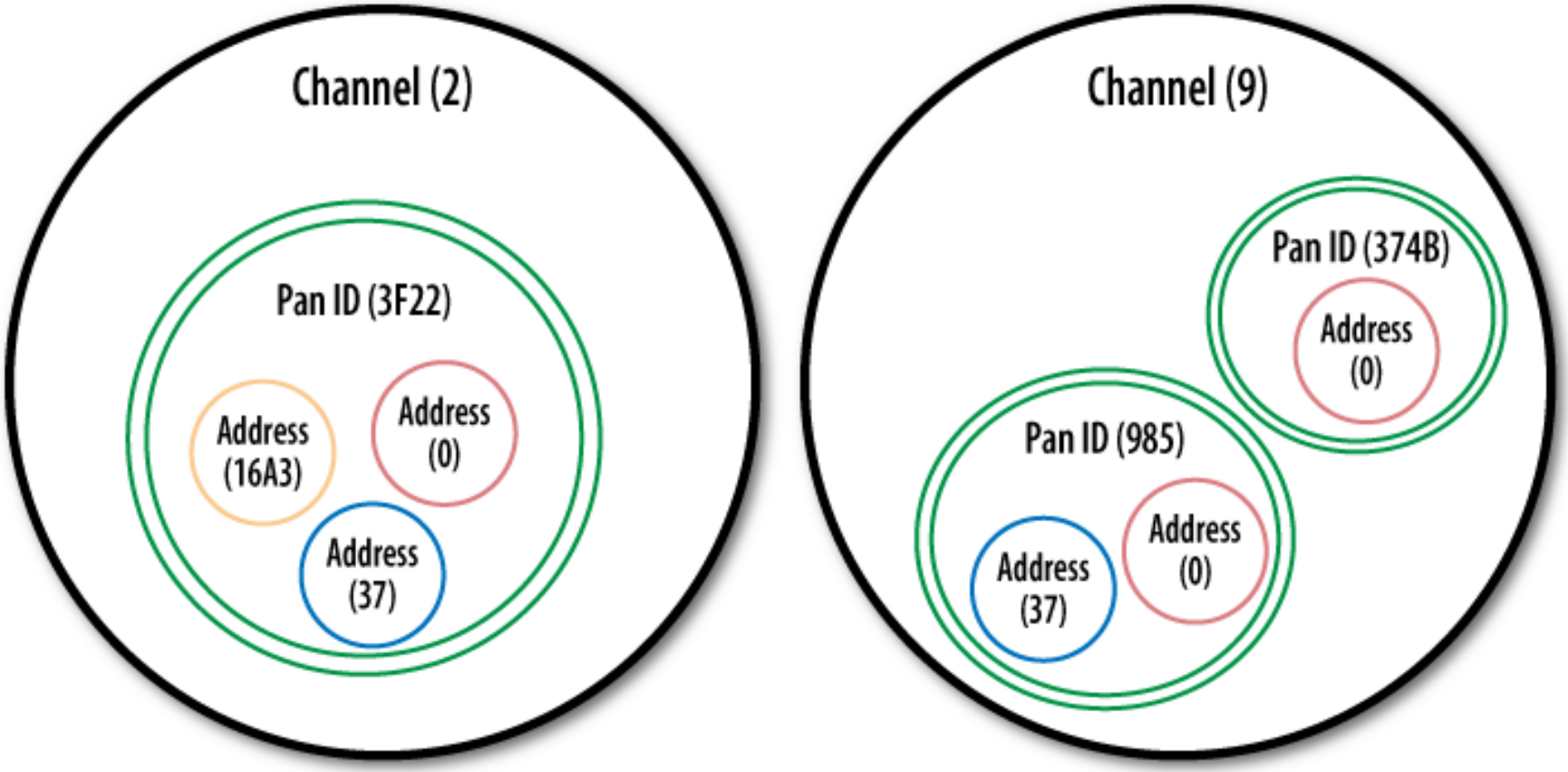
Test and Query of Module

1. Connect XBee Module to PC
   - Wait for installing the driver at first time connection
2. Run X-CTU
3. Select USB Serial Port
4. Click on [Test / Query]
5. Confirm Modem type and Firmware Version





3.5.4 Private Area Network (PAN)

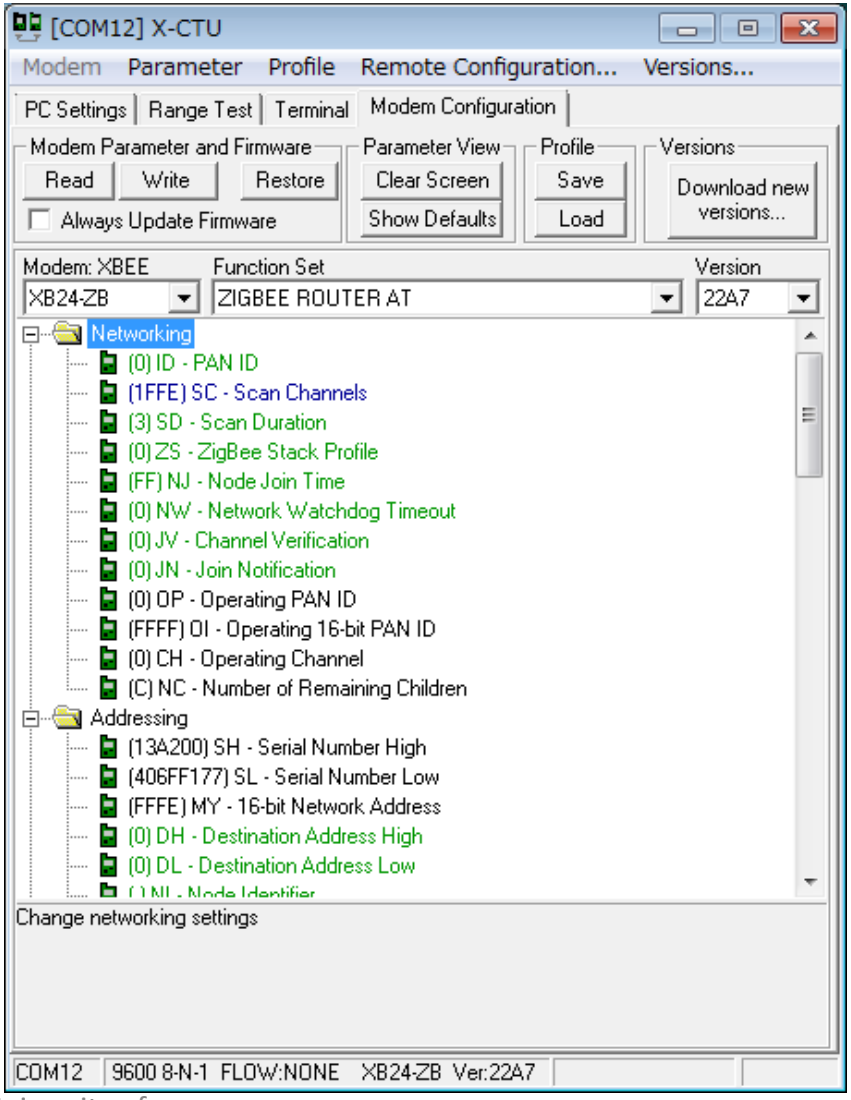


3.5.5 Writing Firmware

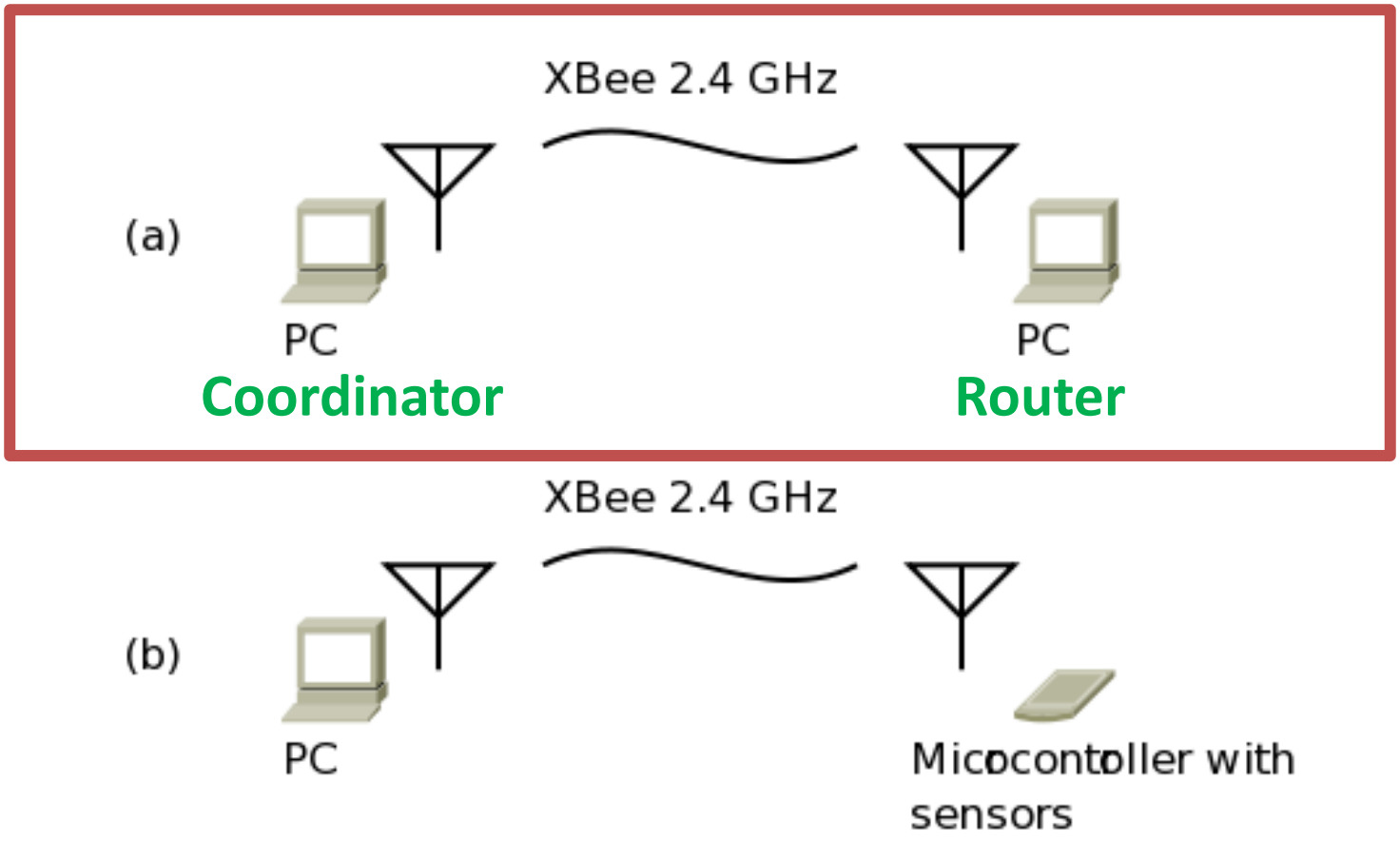
1 Run X-CTU
2. Click on Modem Configuration Tab
3. Select Modem XBEE
 - XB24-ZB ・・・ ZigBee protocol
4. Function Set
 - ZIGBEE COORDINATOR AT
 - ZIGBEE ROUTER AT
5. Set PAN ID
6. Set Destination Address
 - DH, DL (Coordinator and Router)
7. Click on [Write]
 ・・・ Wait for around 40 seconds
8. Click on [Read]
 - Confirm the parameters

※ DO NOT remove the USB cable while writing the firmware

※ REMOVE the USB cable when you change the modules



3.5.6 Building Pair Network (1 by 1)



3.5.7 Send Characters

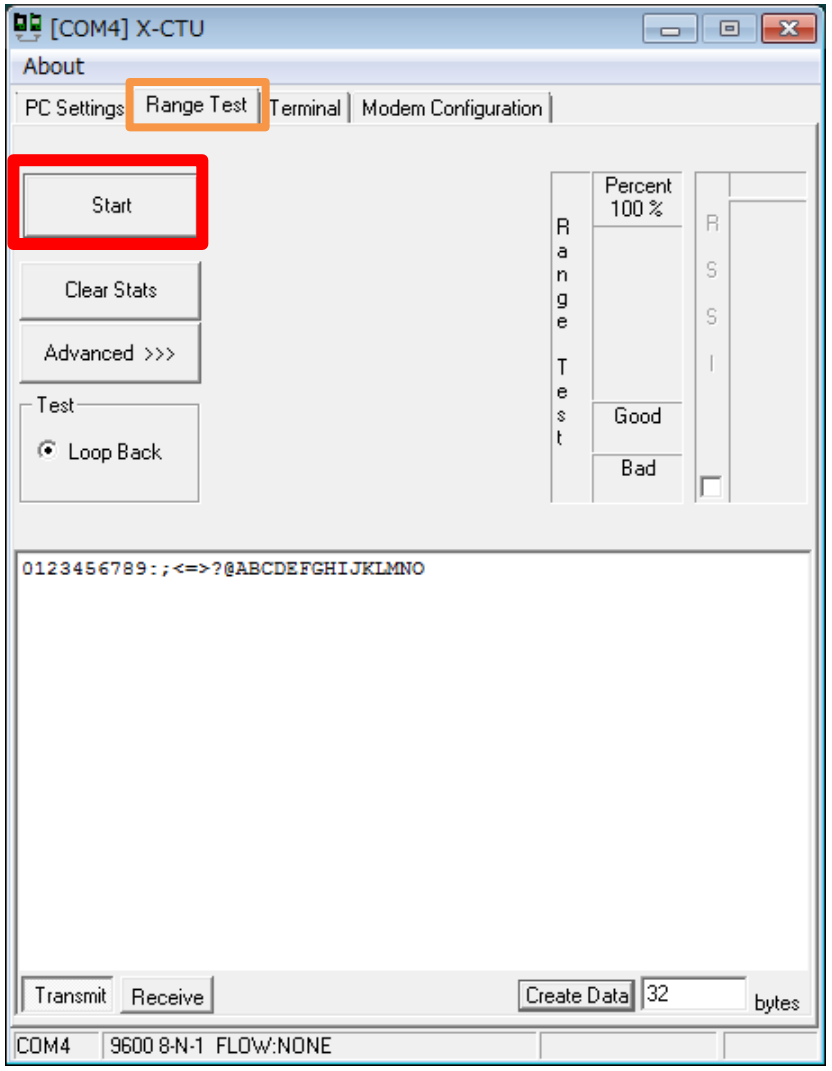
1. Click on Terminal Tab
2. Put the characters in the terminal
3. Confirm the received characters on Coordinator and Routers

3.6 Loop back Test

ZigBee module: Router

Connect TX and RX Pin on XBee module

USB-FTDI	XBee Module
5V (Red)	------ 1 VCC
RXD (Yellow)	<---- 2 DOUT (TX)
TXD (Orange)	----> 3 DIN (RX)
GND (Black)	------10 GND



PC: Coordinator

1. Run X-CTU
2. Click on [Range Test] tab
3. Click on Start button
4. Check Error

3.7 Building Star Network (1 by multi)

Coordinator -> Broadcast to Routers

PANID 123

DH 0

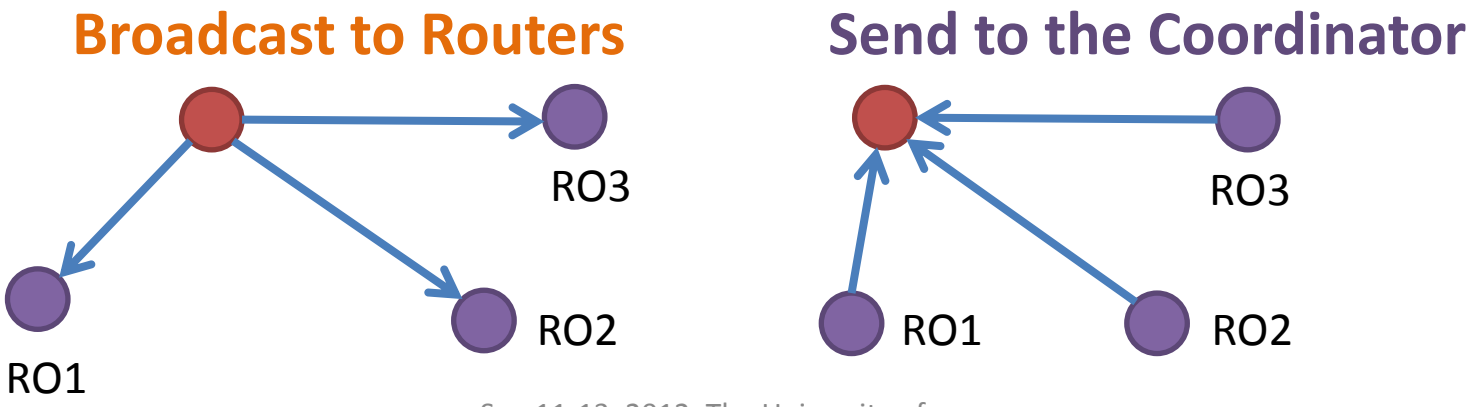
DL FFFF

Router -> Send to the Coordinator

PANID 123

DH 0

DL 0



Appendix: AT Commands

Configure XBee module by the AT commands

+++ Entering command mode Do not need to enter after +++

ATID PAN (Personal Area Network) ID 0x0-0xFFFF

ATSH/ATSL 64-Bit serial number as permanent address

ATDH/ATDL 64-Bit serial number assigned the destination address

ATCN Quit the command mode immediately

ATWR Complete current configuration

ATMY Shows 16-Bit address, Coordinator assigns this address dynamically

Aug 2, 2012: Draft

Building Wireless Sensor Networks with XBee and Arduino

The University of Tokushima
Akinori TSUJI

Contact Information :
2-1, Minamijyosanjima-cho, Tokushima, 770-8506, Japan
TEL/FAX : +81-88-656-7485
E-mail: : a-tsuji@is.tokushima-u.ac.jp

Building Pair Networks

Day 2

2012/9/12(Wed) 10:00—12:00, 13:30—16:00
Time Estimation: 4.5 hours

Agenda

1 Basics of Arduino
- Blink the on board LED

2 Building Pair Network (PC by MCU)
- Pair Network Test

3. Basics of Sensors and Actuators
- Temperature Monitor

4. Building Pair Network (MCU by MCU)
- Switch and Buzzer

1 Basics of Arduino

Arduino development environment

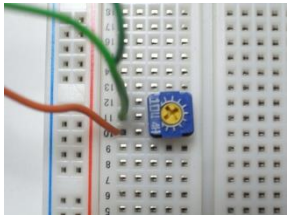
- has started the development in Italy
- easy to use for beginners, no need software or electronics experience
- C / C++ language
- IDE (Integrated Development Environment)
- APIs

Default Setting:

- Tools -> Board -> Arduino Duemilianove w/Atmega328
- Tools -> Serial Port -> (USB Serial Converter)
- Tools -> USBasp

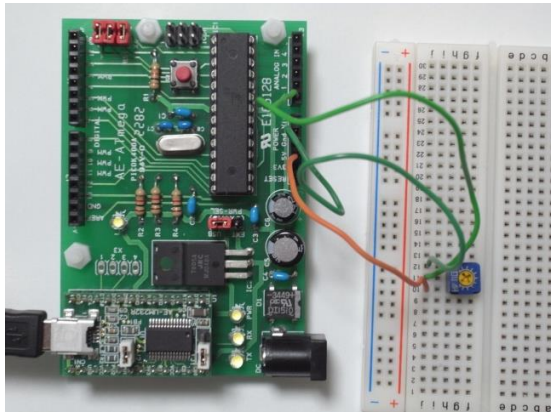
1.1 How it works

1. Make a circuit on the bread board



Note: DO NOT SHORT +5V and GND

2. Connect the circuit to the Arduino development board by wire

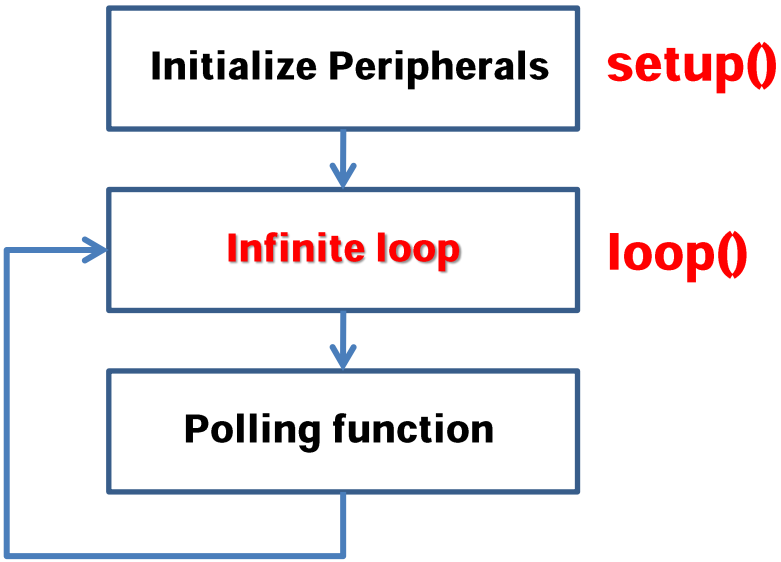


Note: DO NOT SUPPLY POWER

3. Programming on the PC
4. Connect a USB cable to the PC (Power Supply)
5 Upload a program to the microcontroller

1.2 Programming Scheme

Basic structure



Polling

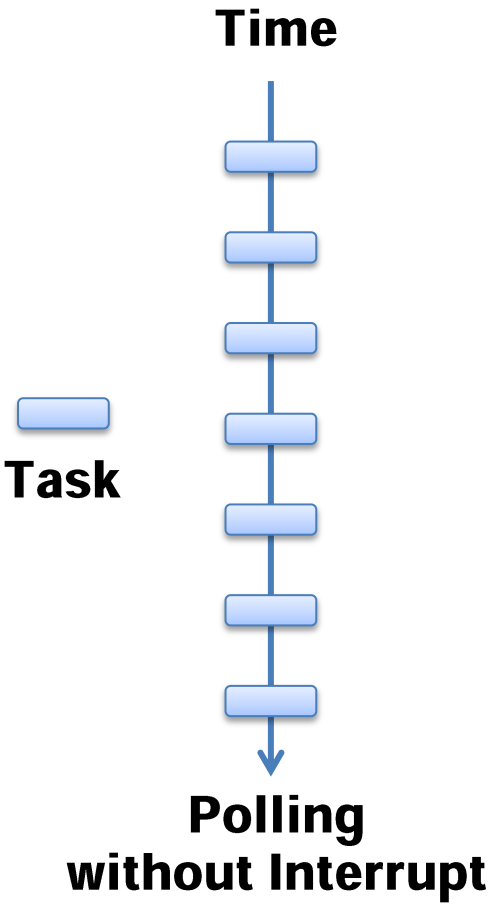
Interrupt function

Interrupt requested from internal or external hardware

MCU has to do an interrupt function Immediately even if a polling function Is not finished.

Interrupt

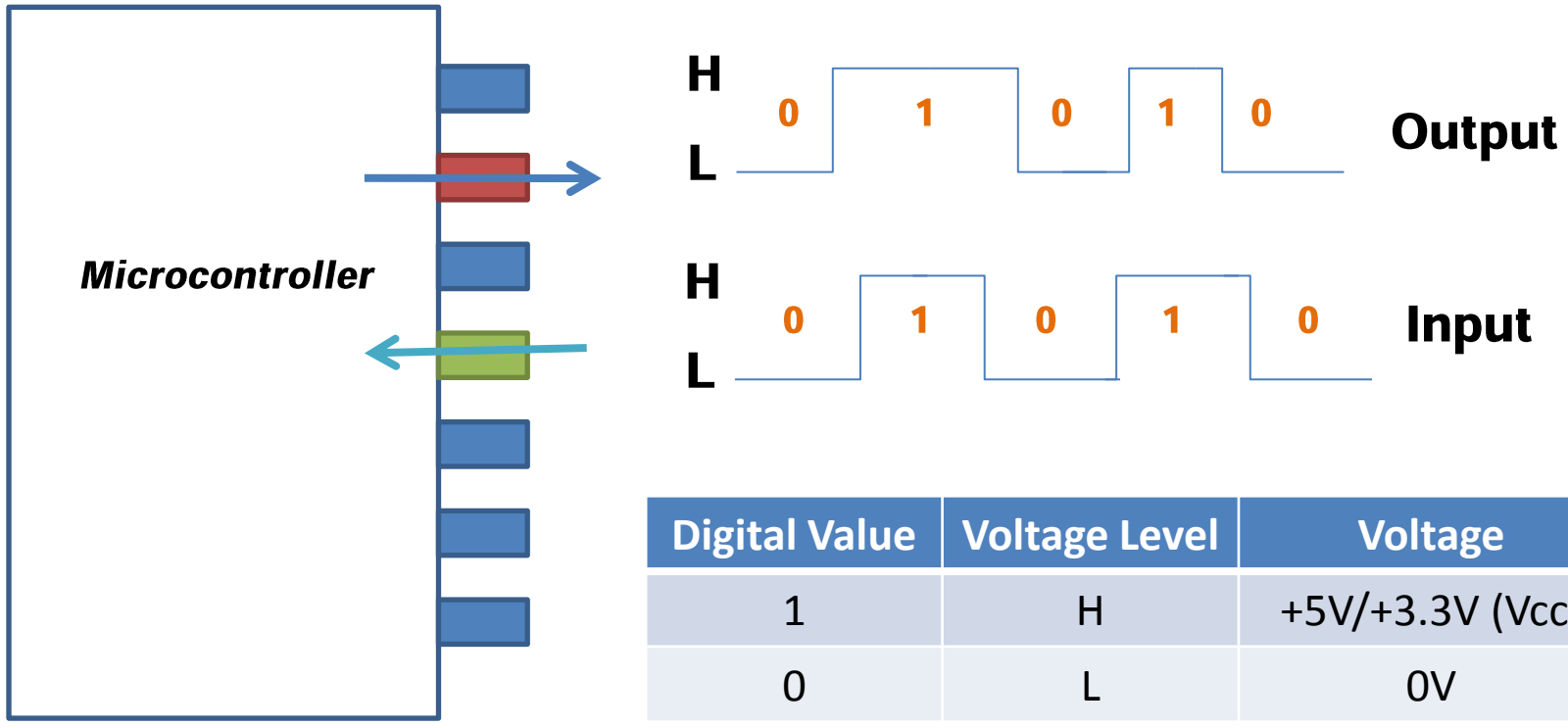
1.3 Polling




```
int main()
{
    init();         //  initializes a hardware

    setup();        //  setup your sketch's function

    while (1) {  //  do the task forever
      loop();    //  Task, polling function
    }
     // no terminate, no return
}
```

1.4 I/O Port

I/O Port = Digital Input / Output port
has Direction, Input or Output. handles 0 or 1, Voltage level: High or Low level

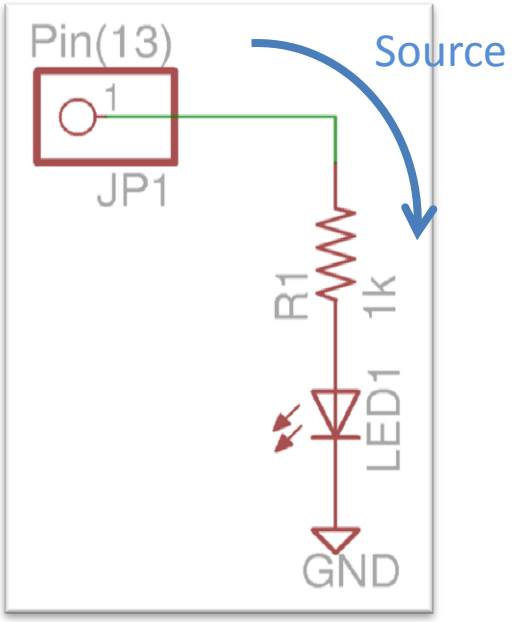


Digital Value	Voltage Level	Voltage
1	H	+5V/+3.3V (Vcc)
0	L	0V

Exercise 1: Blink the on board LED

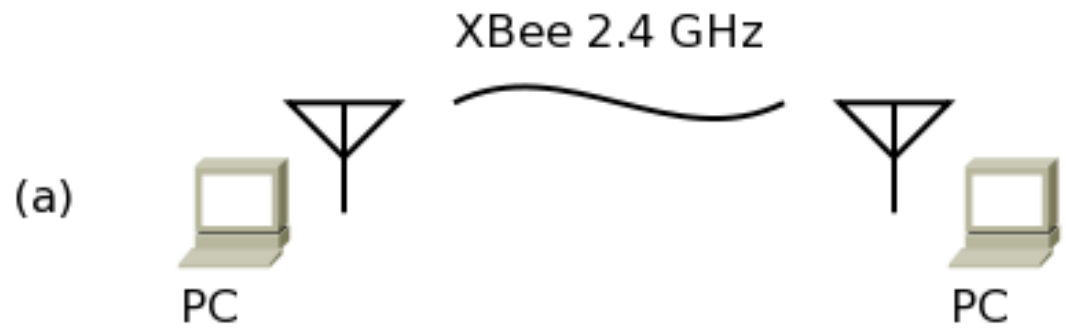
BlinkPin13: Blink the on board LED (connected to Pin 13)

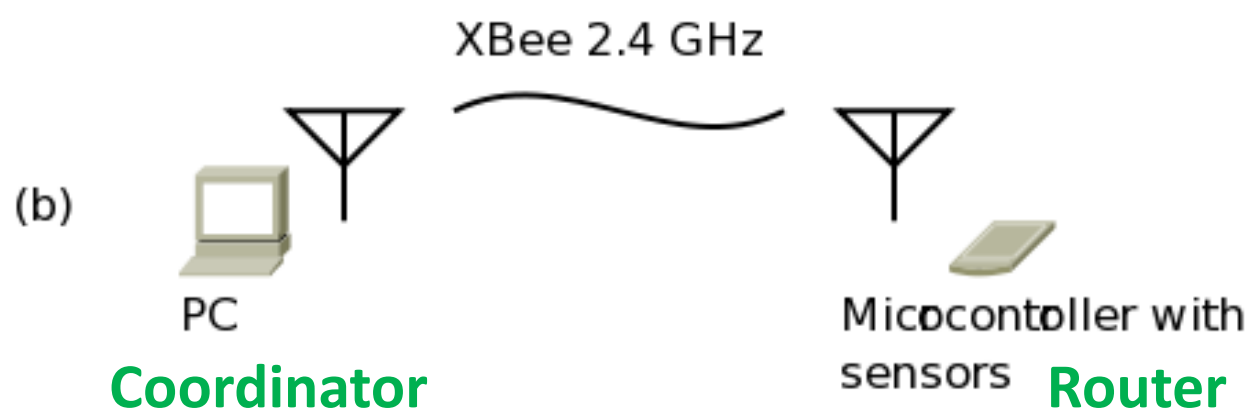
```
const int ledPin = 13;

void setup() {    // Initialize peripherals
  pinMode(ledPin, OUTPUT);
}

void loop() {  // Infinite loop
  digitalWrite(ledPin, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(ledPin, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```



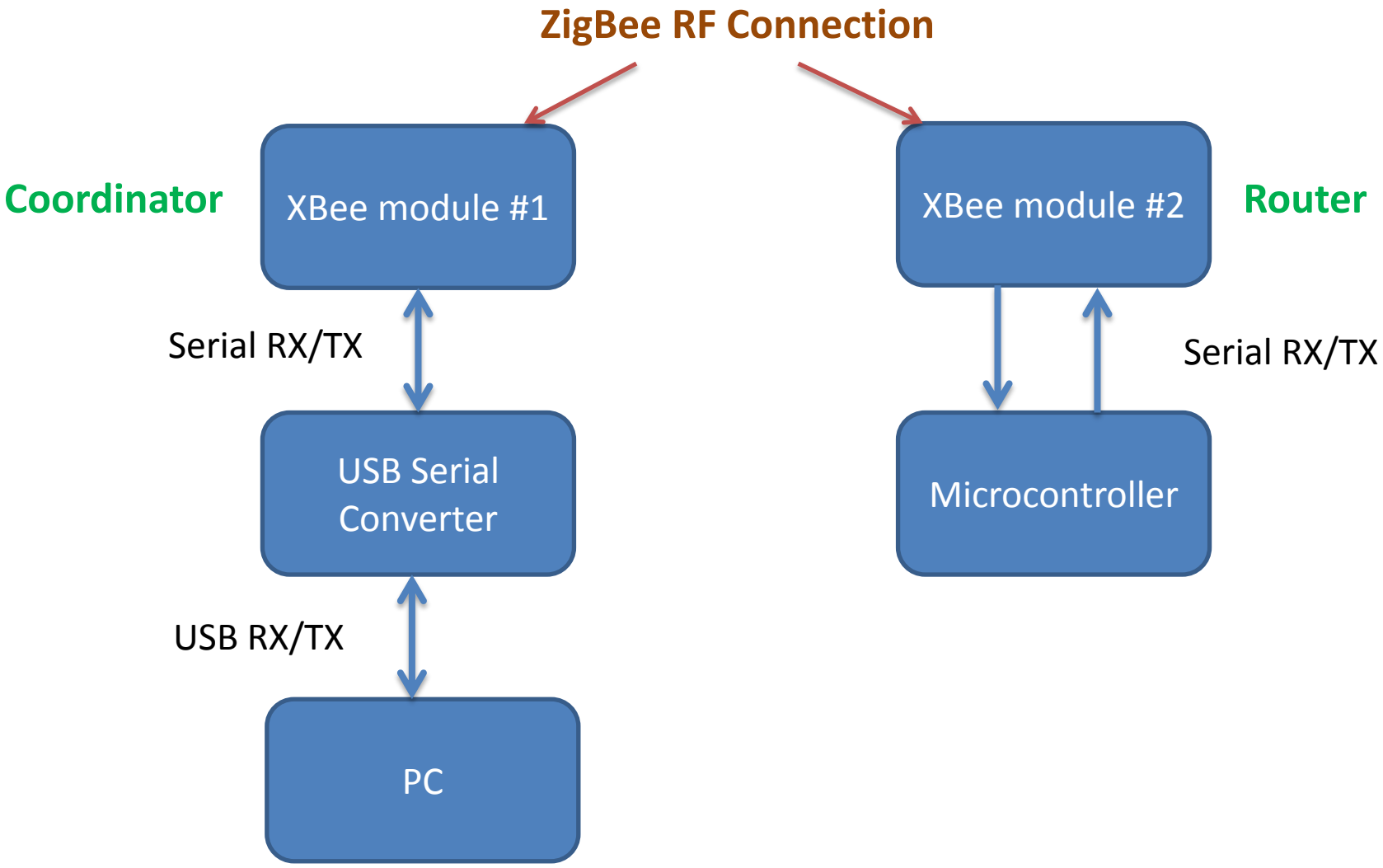
On board LED

2 Building Pair Network (PC by MCU)





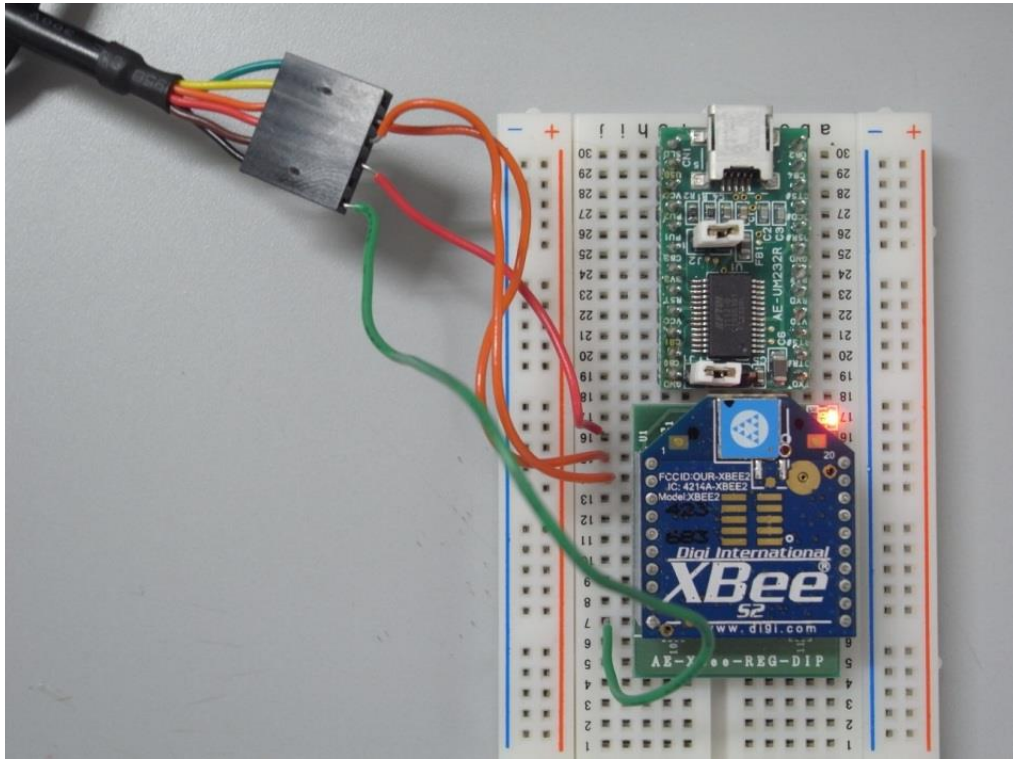
2.1 Pair Network of XBee Module



2.1.1 Pair Network Test of XBee Module

Connect the XBee Module to PC (Coordinator)

USB-FTDI	XBee Module
5V (Red)	------ 1 VCC
RXD (Yellow)	<---- 2 DOUT (TX)
TXD (Orange)	----> 3 DIN (RX)
GND (Black)	------10 GND

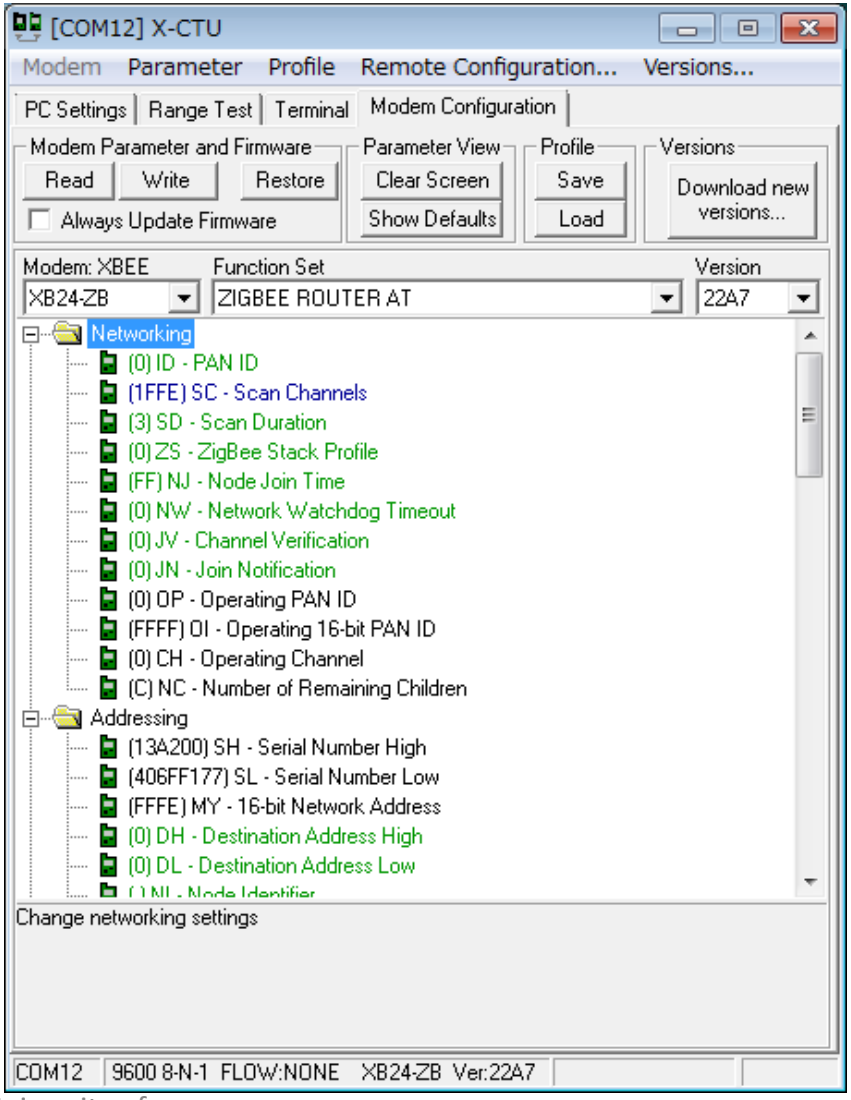


2.1.2 Writing Firmware

1 Run X-CTU
2. Click on Modem Configuration Tab
3. Select Modem XBEE
 - XB24-ZB
5. Set PAN ID
6. Function Set
 - ZIGBEE COORDINATOR AT
 - ZIGBEE ROUTER AT
7. Set Destination Address
 - DH, DL (Coordinator and Router)
8. Click on [Write]
 … Wait for around 40 seconds
9. Click on [Read]
 - Confirm the parameters

※ DO NOT remove the USB cable while writing the firmware

※ REMOVE the USB cable when you change the XBee modules

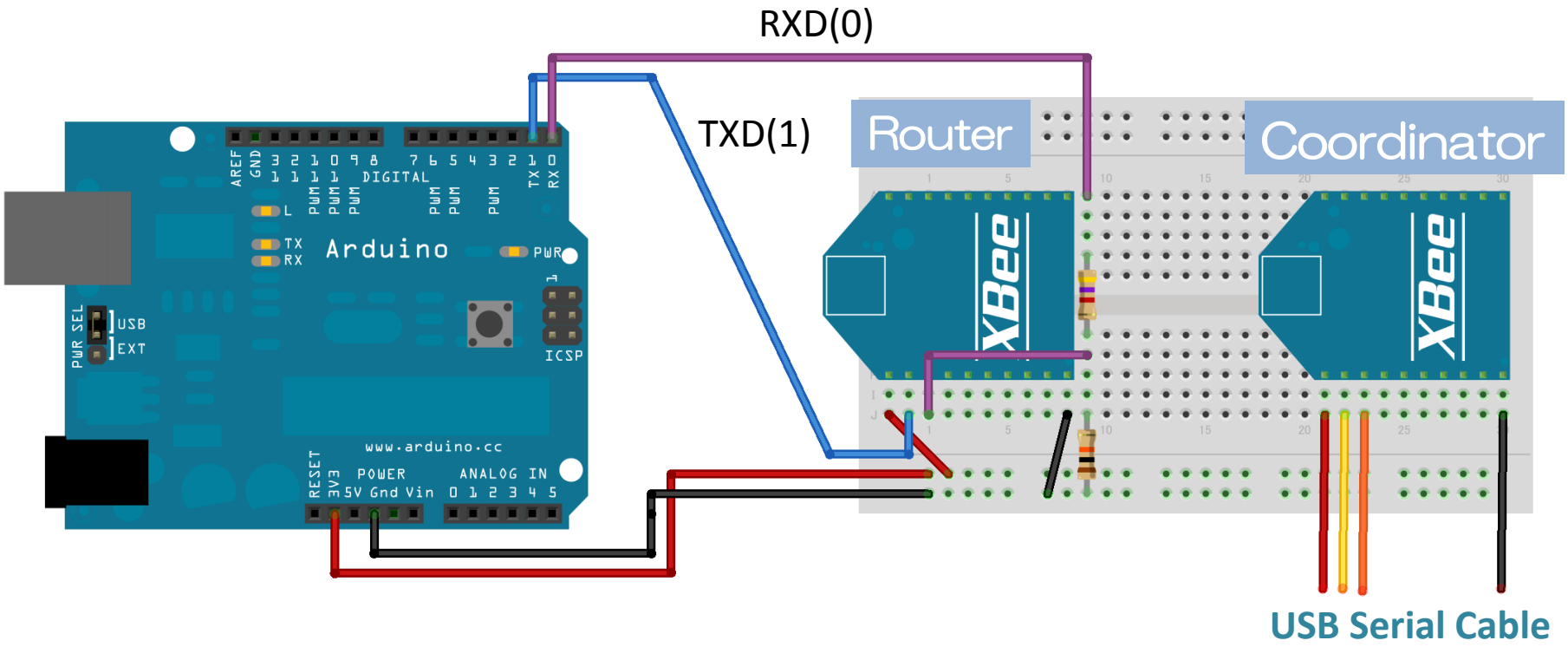


2.2 Arduino with XBee (Pair Network Test)

XBee		Arduino
1 VCC	-----	3.3V
2 DOUT (TX)	----	TXD (1)
3 DIN (RX)	----	RXD (0)
10 GND	-----	GND

※ This connection (RX-RX, TX-TX) is special case for loop back test

※ insert resistors RX-4.7k-RXD-10k-GND



Exercise 2: Pair Network Test

PairNetwork: Nothing todo in the setup() and loop() function

This code means that the serial port TXD and RXD pins are floating connection

```
void setup() {    // Initialize peripherals
  // nothing to do
}
void loop() {  // Infinite loop
 // nothing to do
}
```

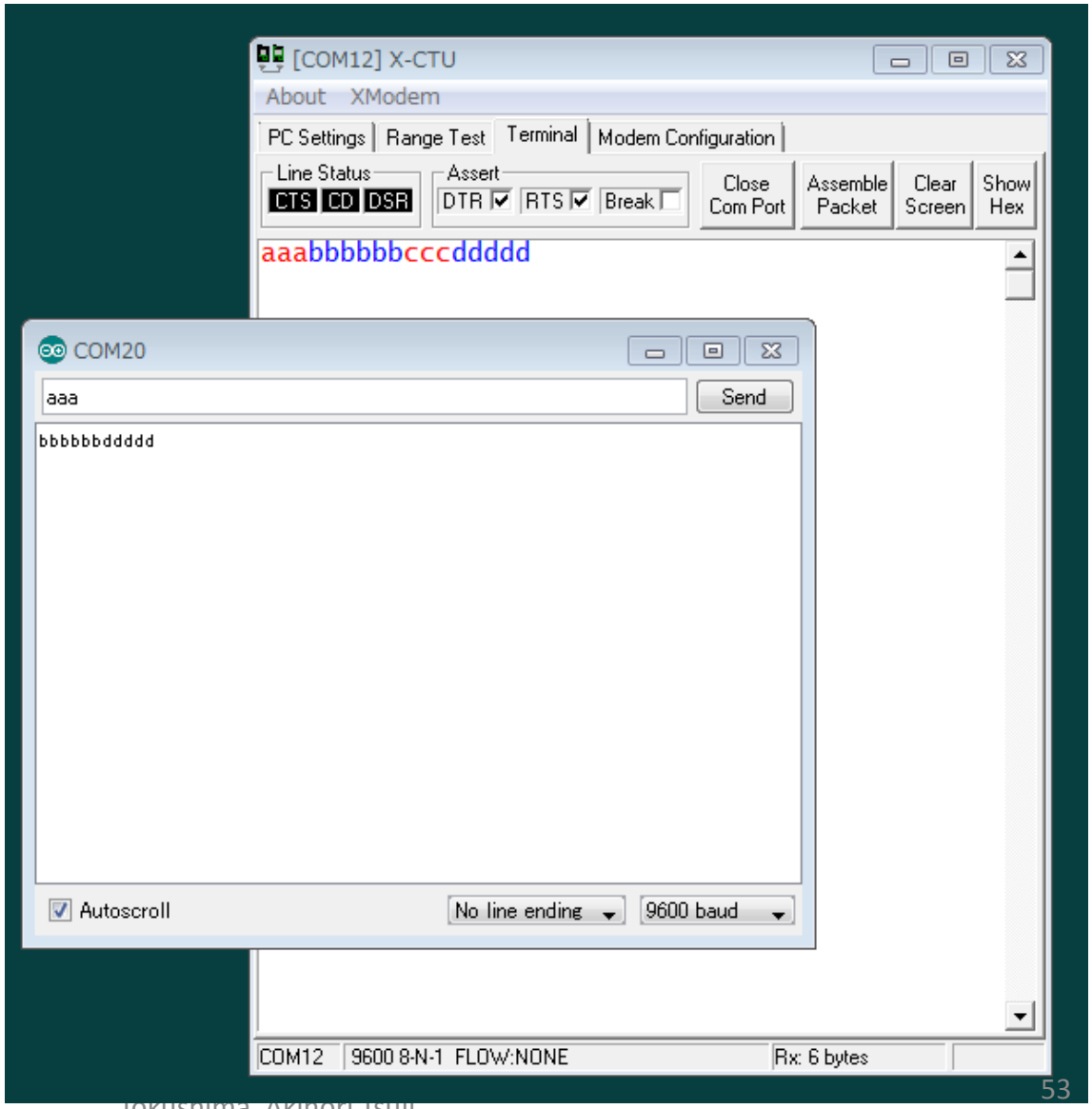
NOTE: Before writing the Arduino program, remove the XBee pin (RXD) to prevent collisions of serial communication

2.2.1 Arduino Seiral Mon and X-CTU Terminal

Coordinator:
1 Run X-CTU
2. Open Terminal

Router:
3 Run Arduino
4. Open Serial Monitor
Tools -> Serial Monitor
5. Put characters and
Click on [Send]
-> Show characters on
the X-CTU Terminal

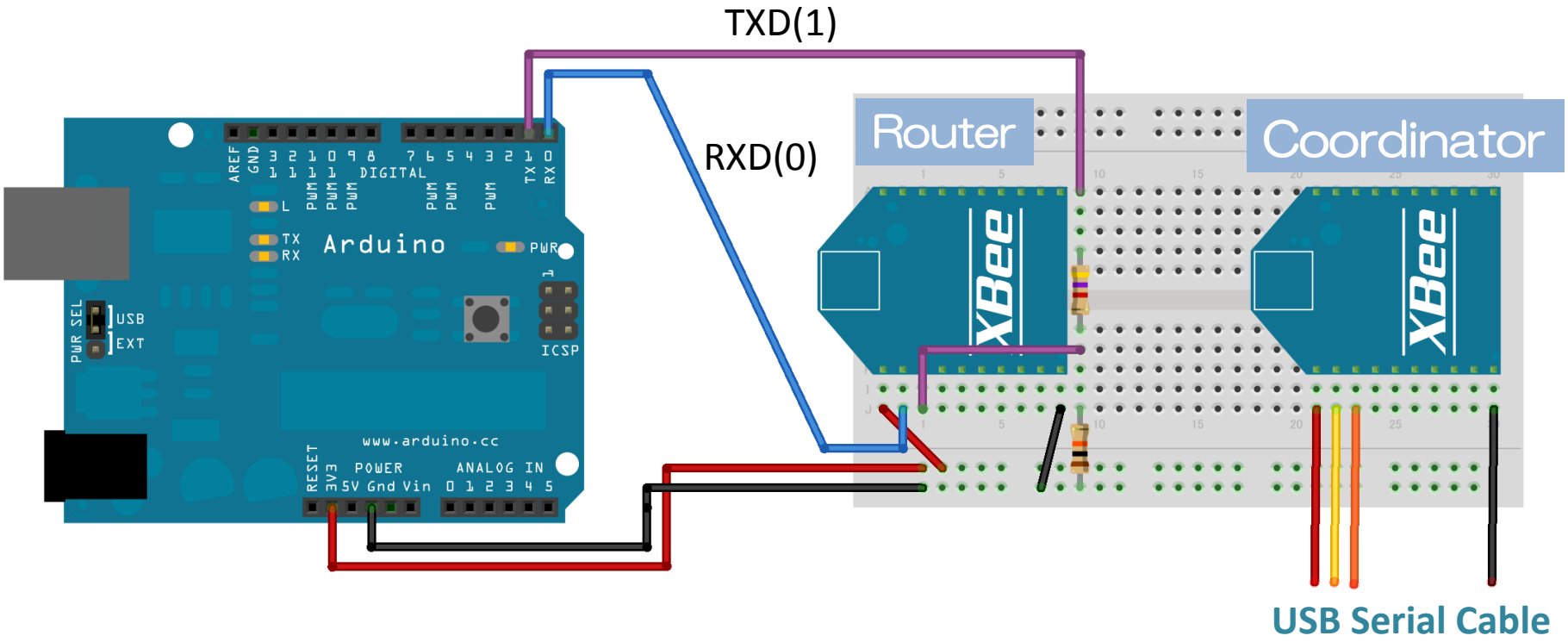
Coordinator:
6. Put characters in the
Terminal
-> Show characters on
Serial Monitor



2.3 Arduino with XBee (Send Characters)

XBee		Arduino
1 VCC	-----	3.3V
2 DOUT (TX)	---->	RXD (0)
3 DIN (RX)	<----	TXD (1)
10 GND	-----	GND

※ Insert resistors RX-4.7k-TXD-10k-GND



Exercise 3: Pair Network Send Characters

SendChars: send characters from router to coordinator

```
void setup() {    // Initialize peripherals
   Serial.begin(9600);
}
void loop() {  // Infinite loop
   int  n = 10;
   Serial.print(n); // number
   Serial.println("Hello World");  // characters
   delay(1000); // every 1 second
}
```

NOTE: Before writing the Arduino program, remove the XBee pin (RX) to prevent collisions of serial communication

2.3.1 X-CTU Terminal (Coordinator)

Router:

1 Power Supply

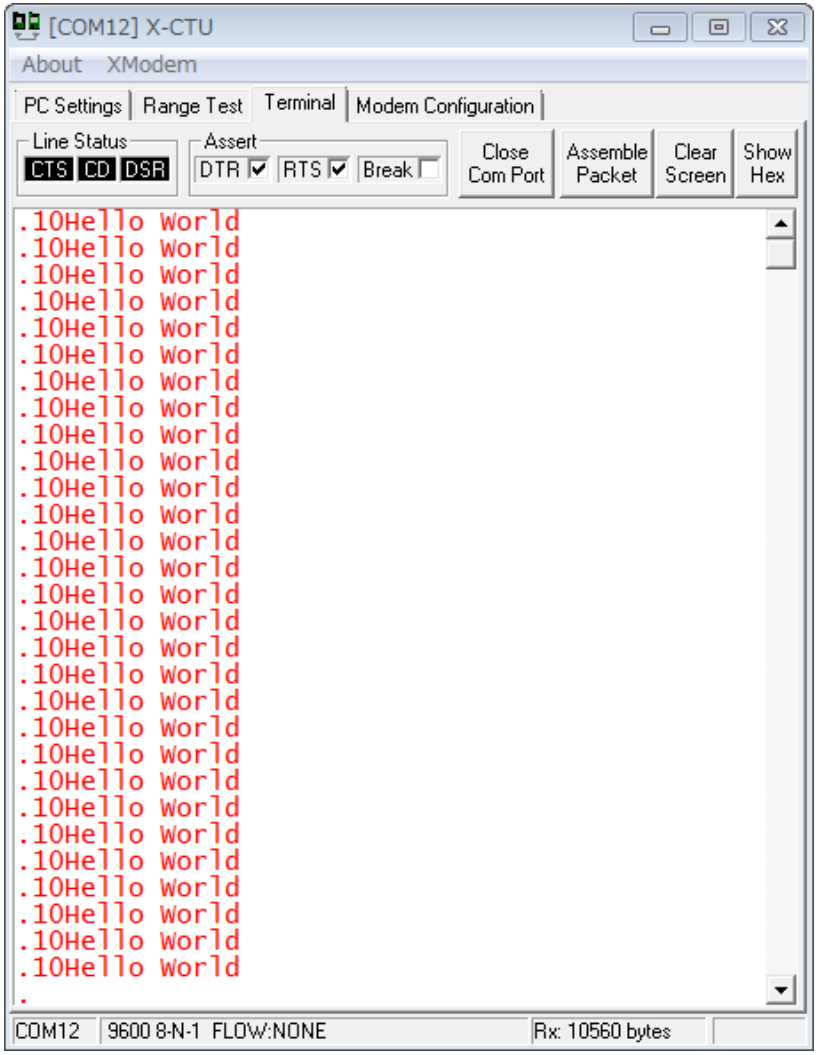
-> Send characters to Coordinator

Coordinator:

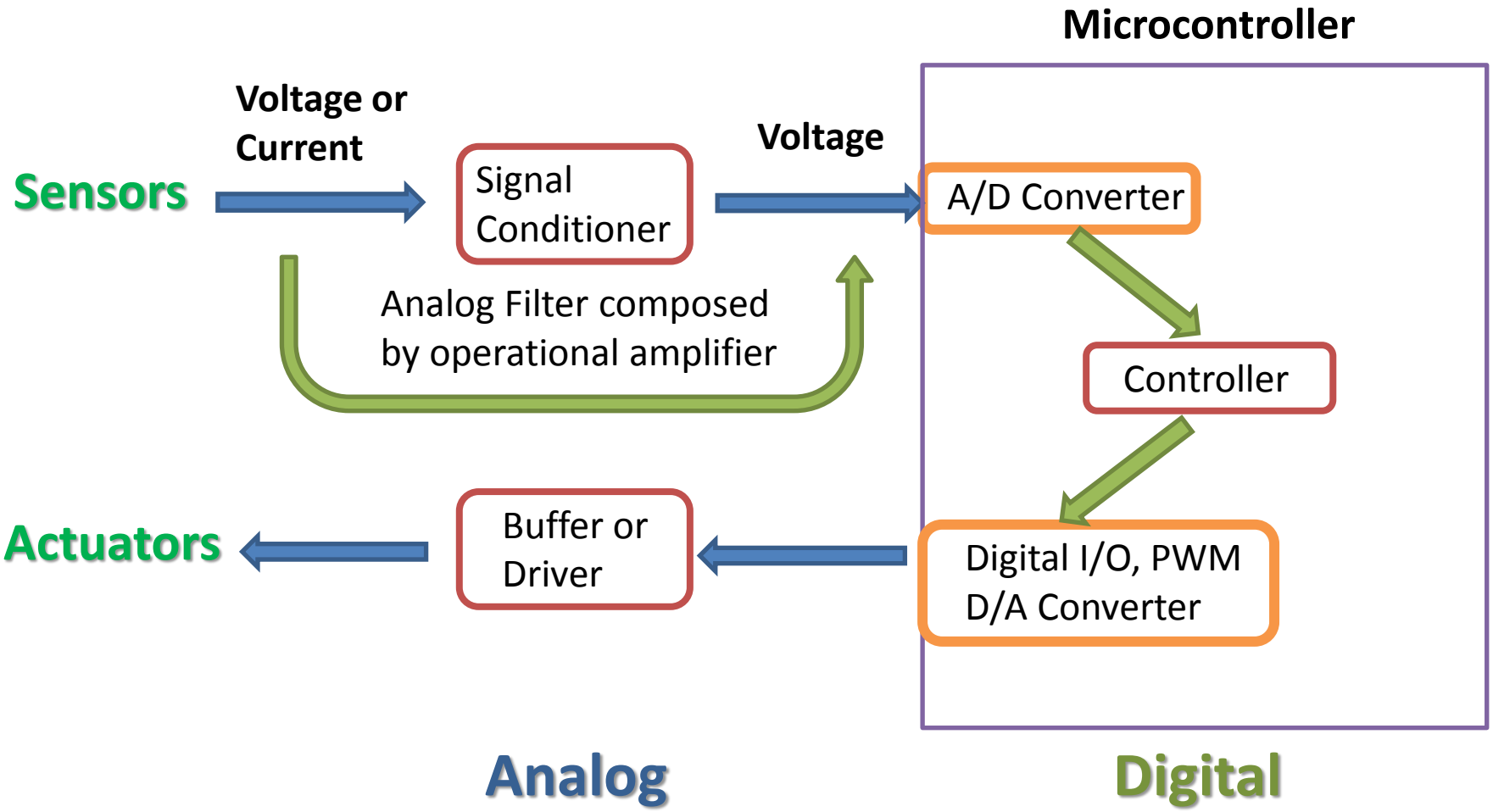
2. Run X-CTU

3. Open Terminal

-> Show characters received from Router on Serial Monitor



3 Basics of Sensors and Actuators



3.1 Temperature Sensor

National Semiconductor LM60BIZ

Output: Analog

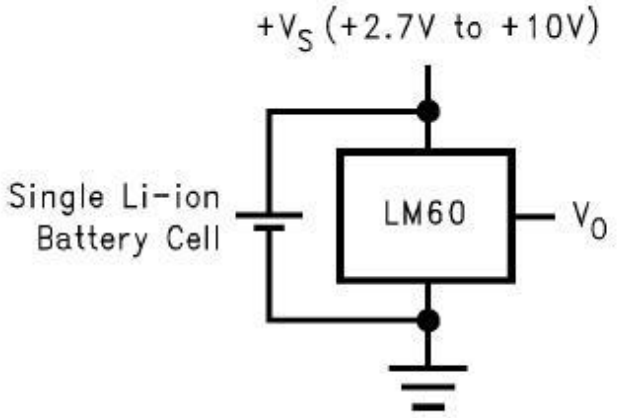
Operating Voltage: DC 2.7V – 10V

Measurement Temperature: -25 deg. -- +125 deg.

6.25 mV / deg.

Torrelance: ±2 deg. (@ 25 deg.)

Temperature (T)	Typical V_O
+125˚C	+1205 mV
+100˚C	+1049 mV
+25˚C	+580 mV
0˚C	+424 mV
–25˚C	+268 mV
–40˚C	+174 mV



$$V_O = (+6.25\ \text{mV/℃} \times T\ ℃) + 424\ \text{mV}$$

3.1.1 Measurement of Temperature

Temprature to Voltage conversion

Voltage Output (Vo):

Vo = +6.25mV x T + 424 mV

6.25 mV x T = Vo – 424 mV

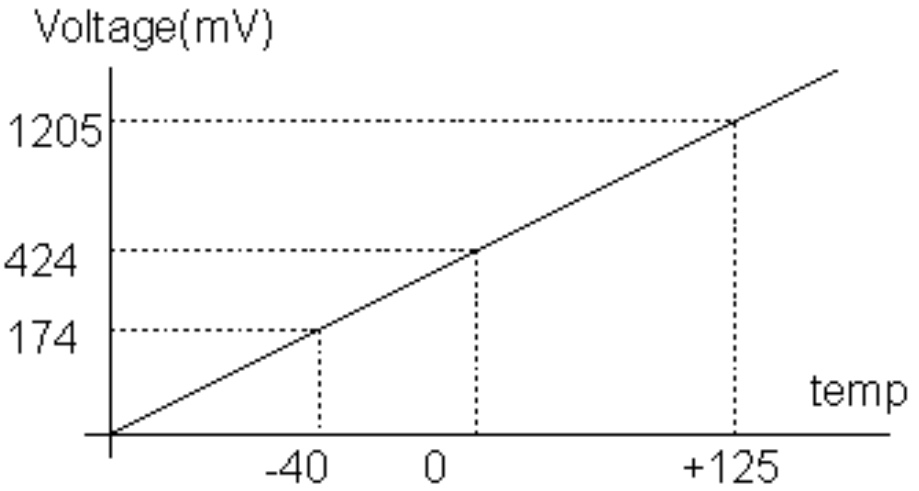
Temperature:

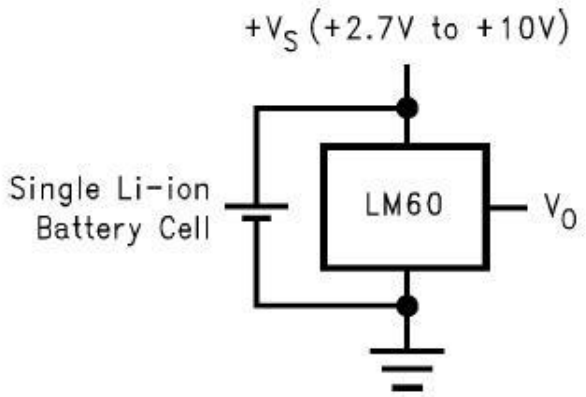
T = (Vo – 424 mV) / 6.25 mV

Analog Input:

Vo = AIN0 x 5.0 / 1024 (V)

= AIN0 x 5000 / 1024 (mV)



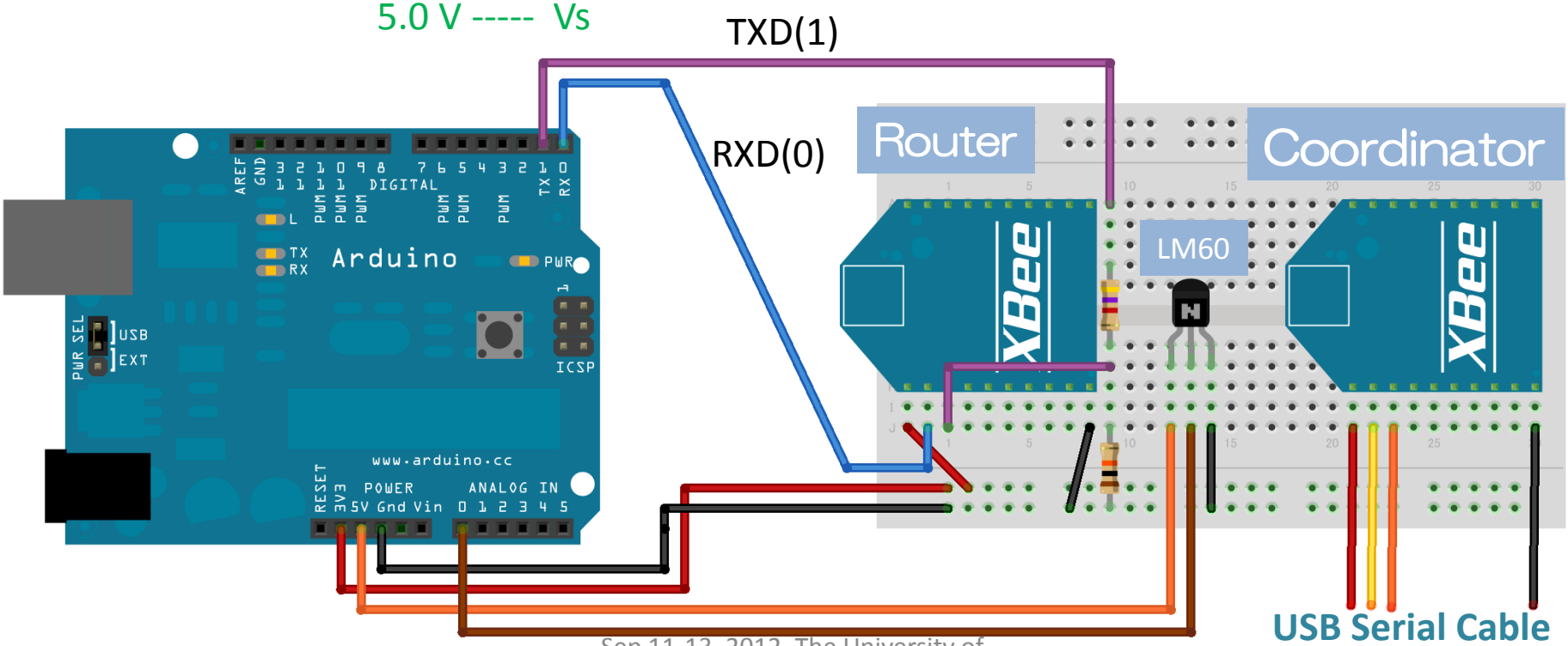


$$V_O = (+6.25\ mV/℃ \times T℃) + 424\ mV$$

3.2 Arduino with XBee (Temperature)

XBee		Arduino	LM60
1 VCC	-----	3.3V	
2 DOUT (TX)	---->	RXD (0)	
3 DIN (RX)	<----	TXD (1)	
10 GND	-----	GND	---- GND
		A0	----- Vout
		5.0 V	----- Vs

※ Insert resistors RX-4.7k-RXD-10k-GND



Exercise 4: Pair Network Send Temperature

TempMeas: Read the output voltage of the temperature sensor and send to the Coordinator

```
const int analogInPin = A0;  // Analog input pin that the temperature sensor is connected
unsigned int sensorValue = 0;     // value read from the temperature sensor

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
void loop() {
  sensorValue = analogRead(analogInPin);
  float Vo = sensorValue * 5000.0 / 1024;  // converts to Voltage (mV)
  float T = (Vo - 424) / 6.25; // converts the temperature (Centigrade Degree)
  Serial.println(T); // Send the measurement result
  delay(1000);  // wait 1 second for the AD converter to settle after the last reading
}
```

3.2.1 X-CTU Terminal (Coordinator)

Router:

1 Power Supply

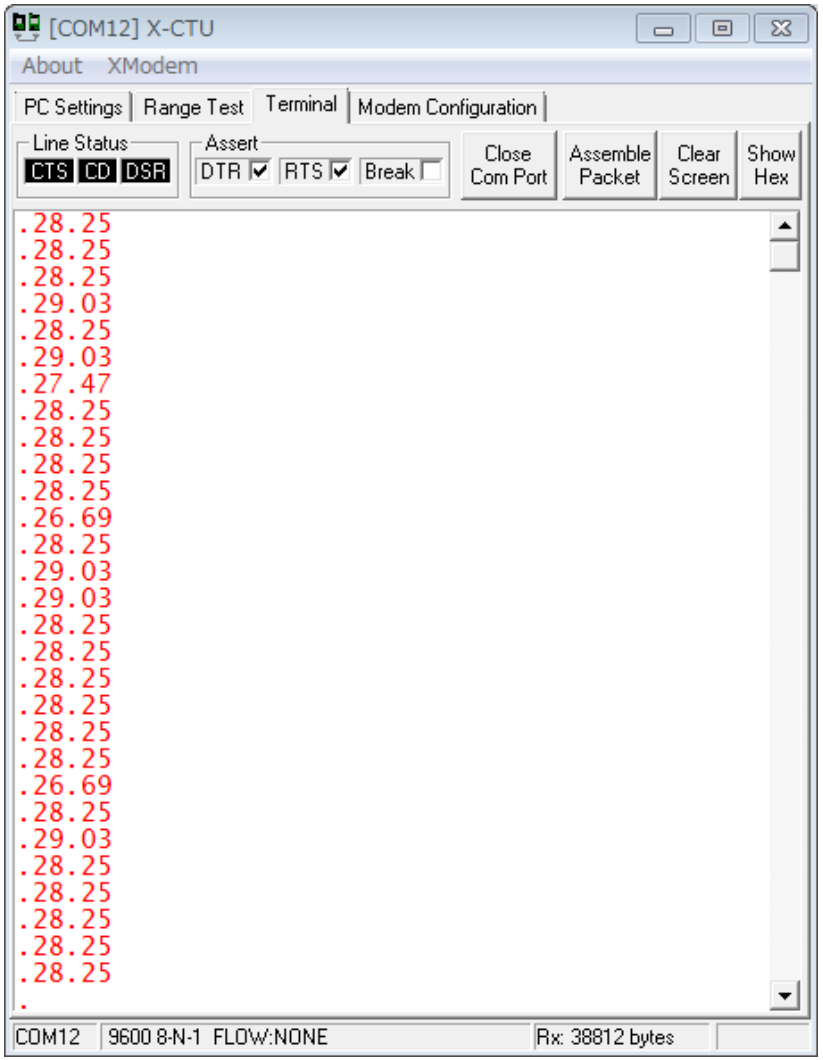
-> Send characters to Coordinator

Coordinator:

2. Run X-CTU

3. Open Terminal

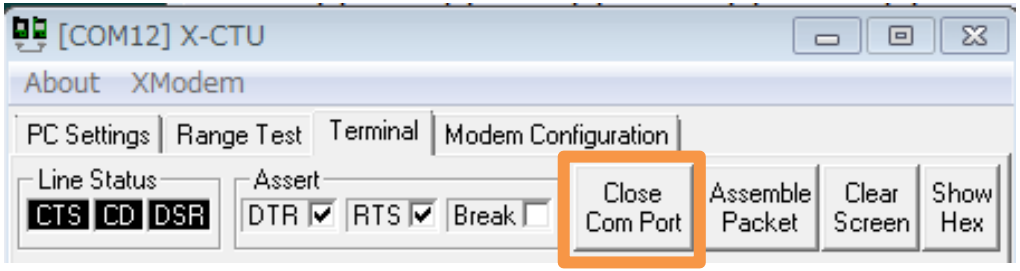
-> Show characters received from Router on Serial Monitor



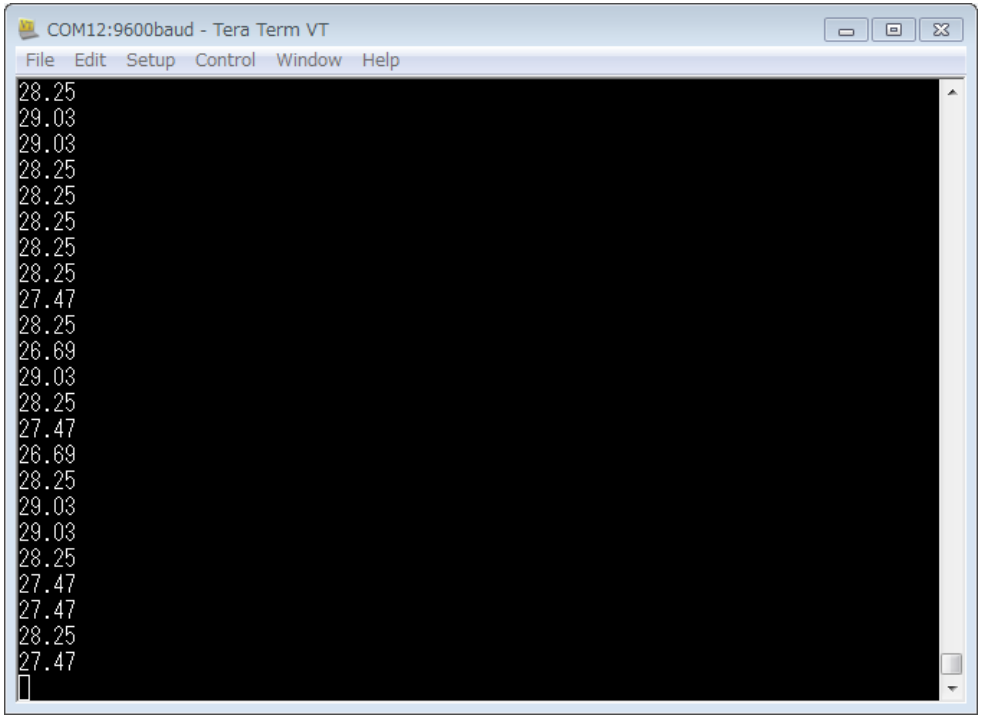
3.2.2 Data Acquisition using TeraTerm

Coordinator:

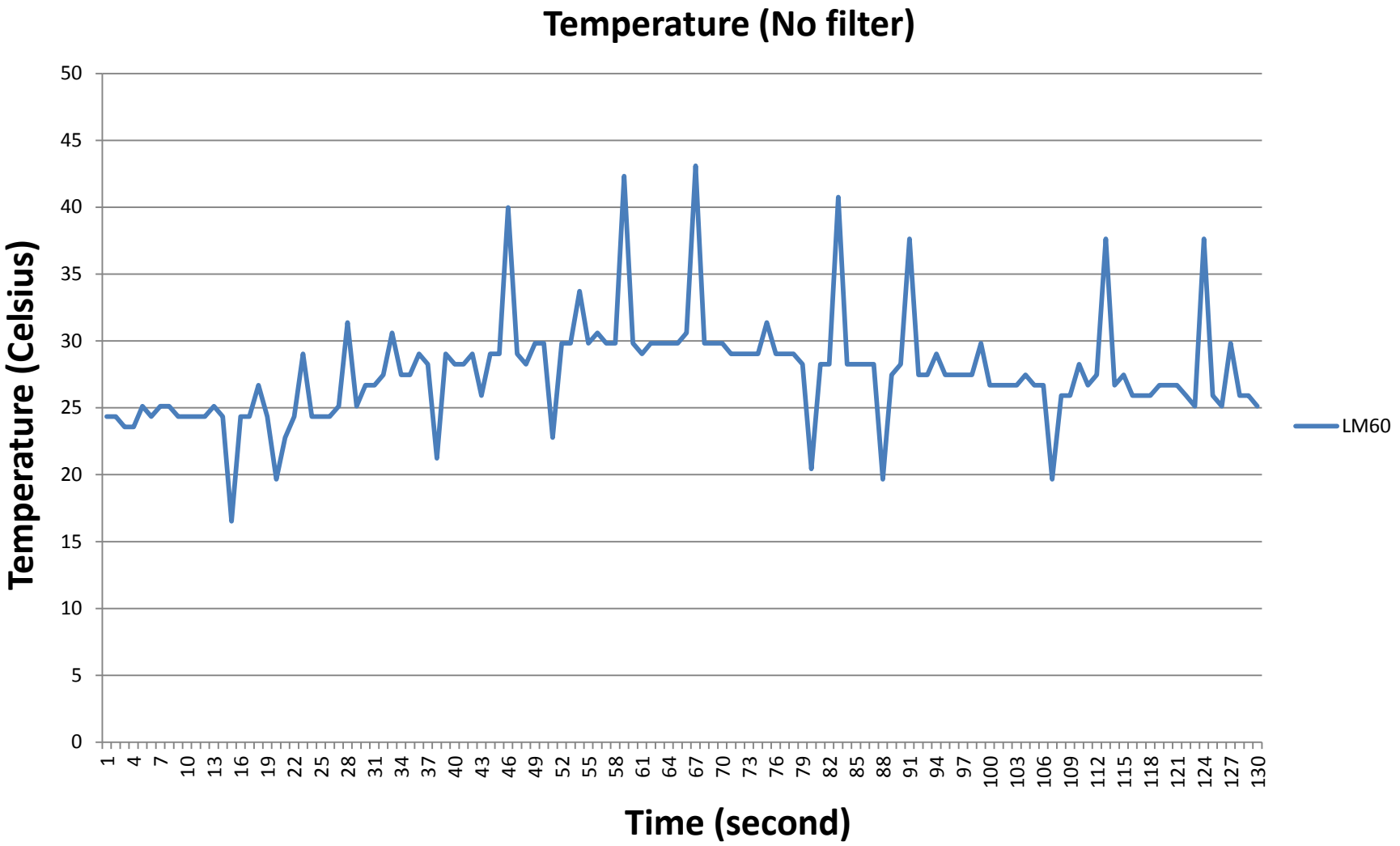
1. Run TeraTerm
2. Open USB Serial Port
3. Setup Serial Port
   9600 bps, 8bit, Parity None,
   Stop 1 bit, Flot Cntrl None
   -> Show characters received
   from Router on Serial Monitor
4. Logging the data
   File -> Log
   Ex) Save as filename: log.csv
5. Stop logging
   click on Close button on the
   log window
6. Open log file by Excel
7. Make a graph



Before run Teraterm, close the Comport on X-CTU



Measurement Result



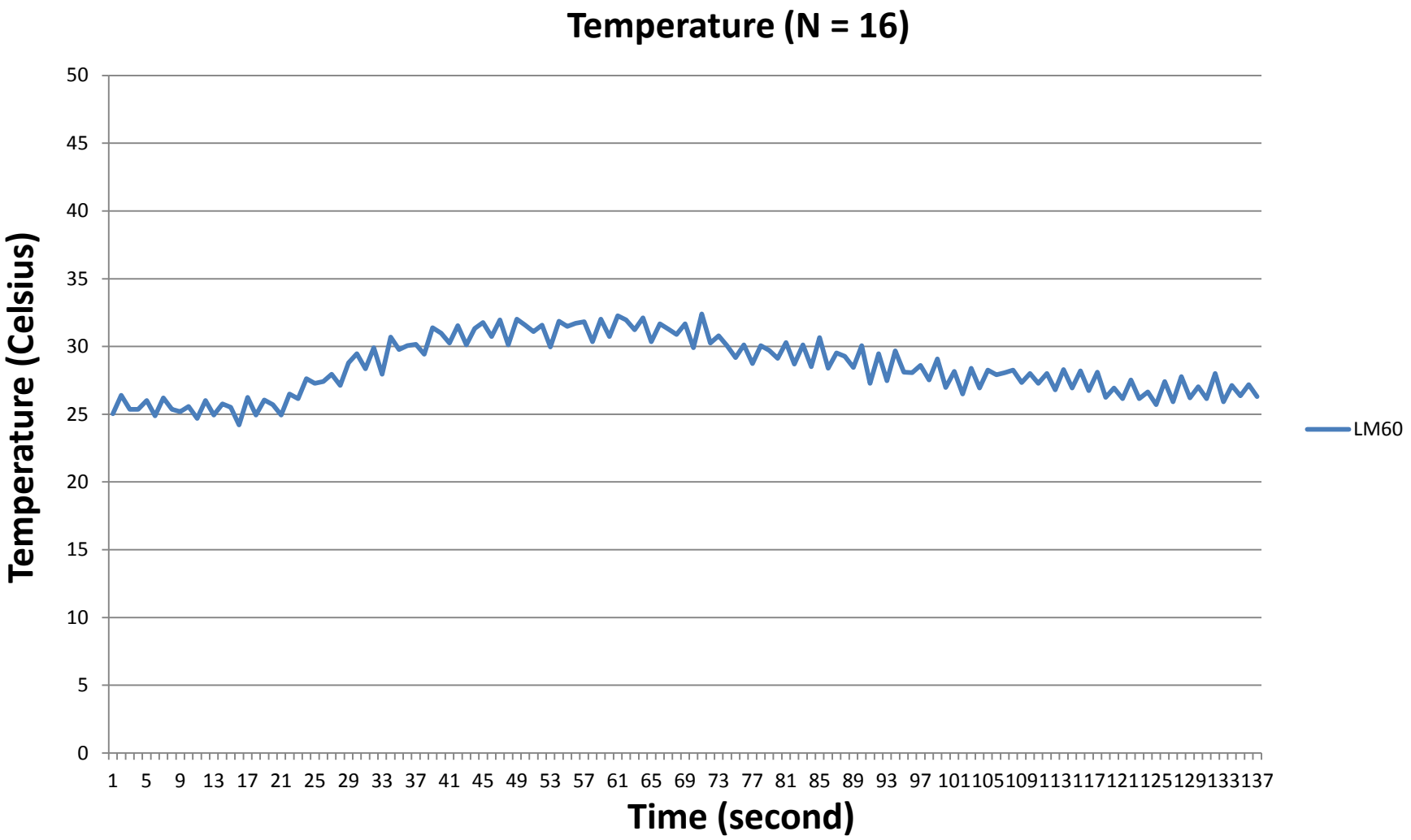
Exercise 5: Pair Network Send Temperature

TempMeasWithFilter: Read the output voltage of the temperature sensor

```
const int analogInPin = A0;  // Analog input pin that the temperature sensor is connected
const int N = 16;  // Sample number for average filter
int sensorValue = 0;        // value read from the temperature
void setup() {
  Serial.begin(9600);
}
void loop() {
  float T;
  T = 0.0;
  for (int i=0; i<N; i++) {
    sensorValue = analogRead(analogInPin);
    float Vo = sensorValue * 5000.0 / 1024;
    Vo = (Vo - 424) / 6.25;
    T += Vo;
    delay(10);
  }
  Serial.println(T / N);
  delay(840);
}
```

Average filter

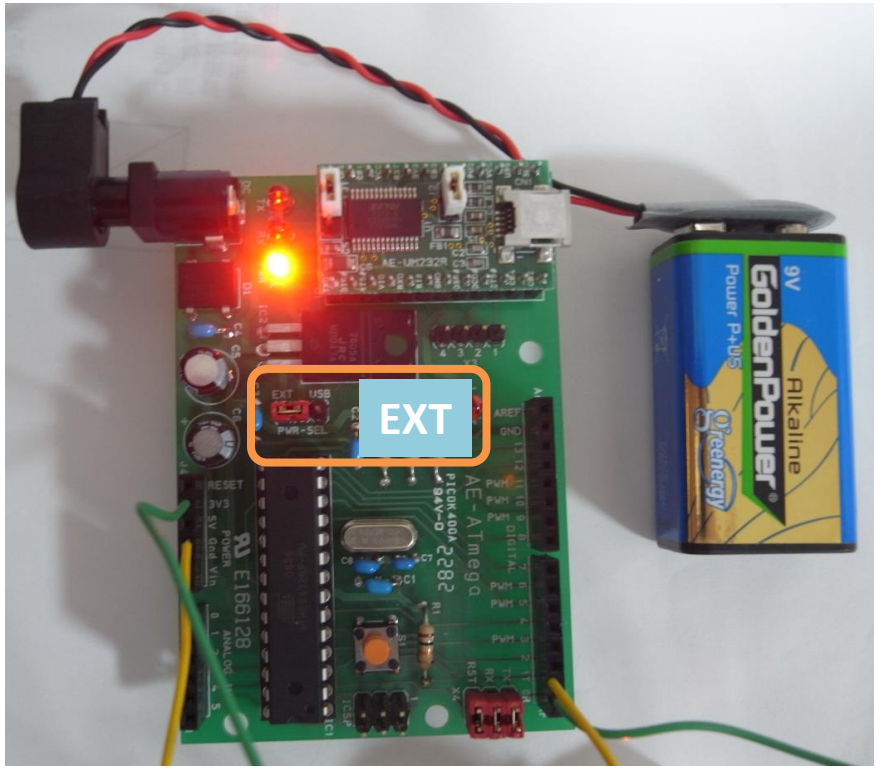
Measurement Result (With Filter)



Power Supply (USB or Battery) for Arduino

Supply from the USB port

Supply from the battery 006p (9V)



Must change Short Pin place
USB or EXT

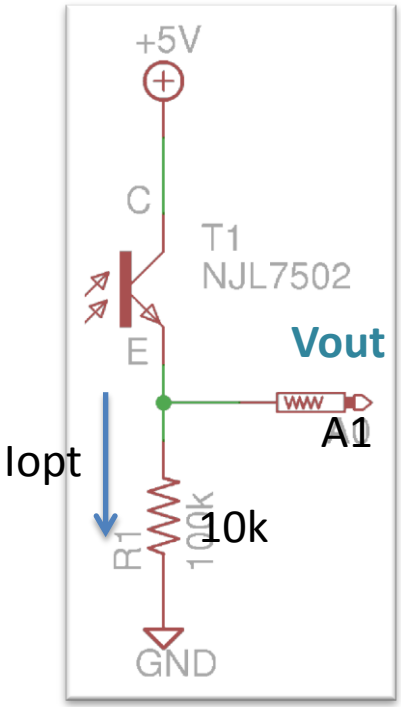
3.3 Light Sensor

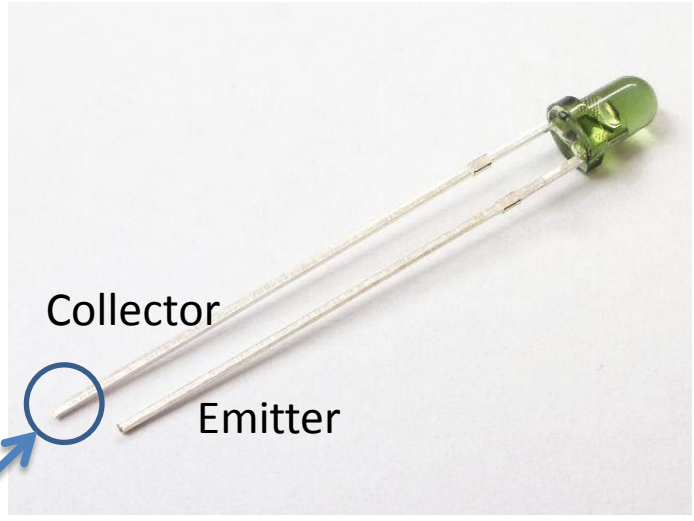
Photo Transistor

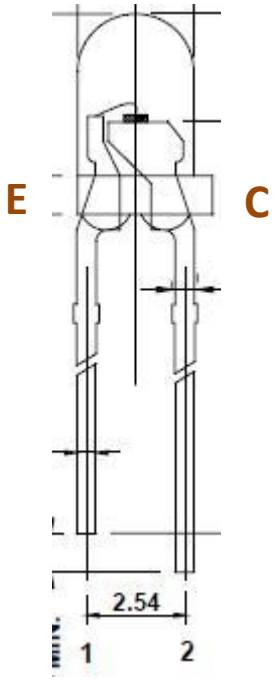
JRC NJL7502L

Peak Sensitivity 560 nm

Optical Current 33 uA





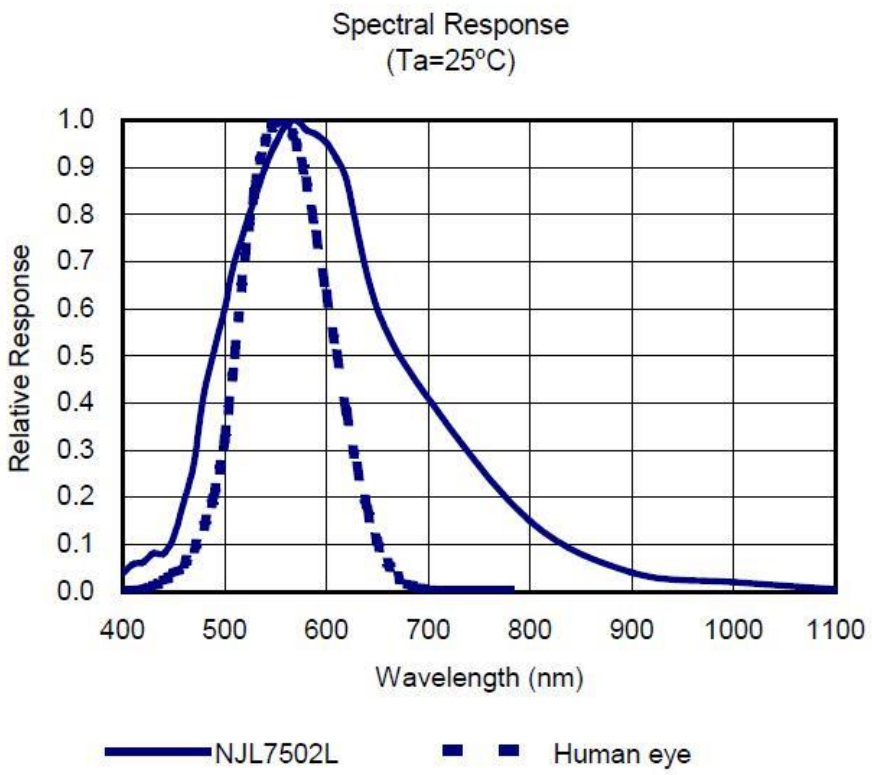


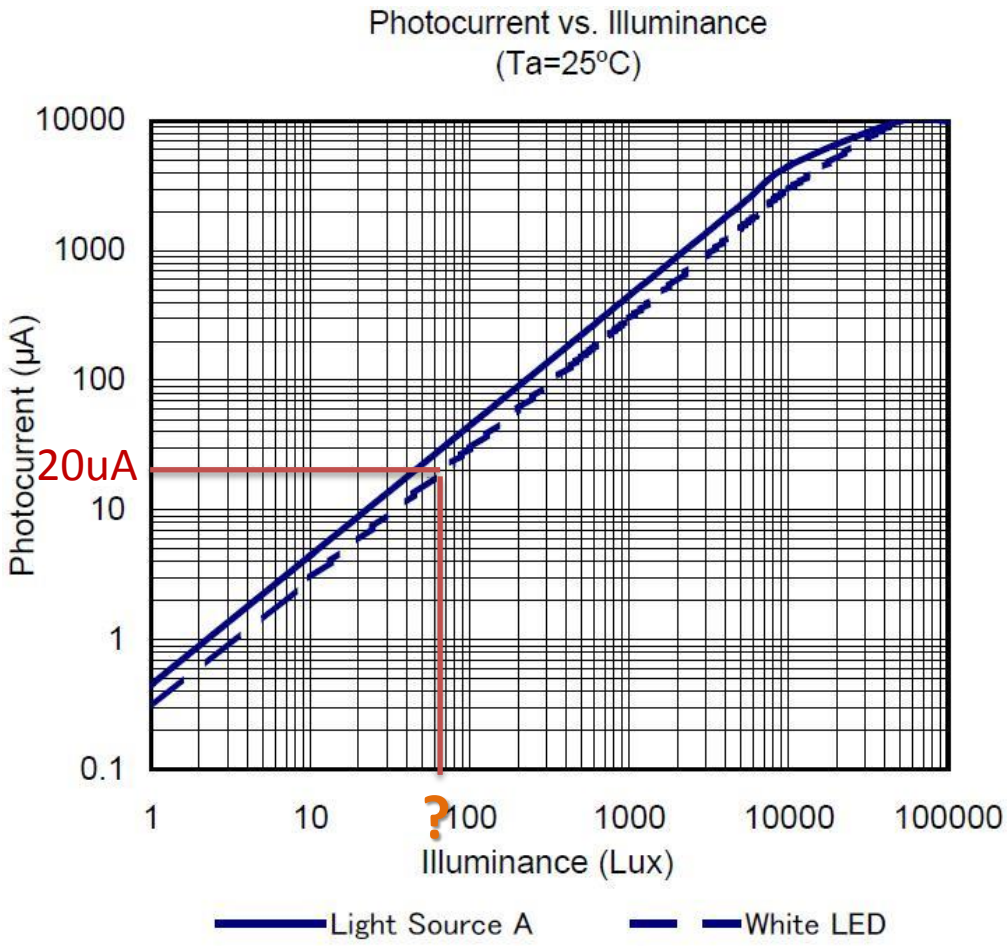
Package

Vout = 10k * Iopt

Iopt = Vout * 100 [μA]

3.3.1 Illuminance





3.3.2 Measurement of Light

Illuminance and photo-current

Lux = 2.22 x I_{opt}

＝2.22 x Vout / R

＝2.22 x AIN1 x (Vref / 1024) / R

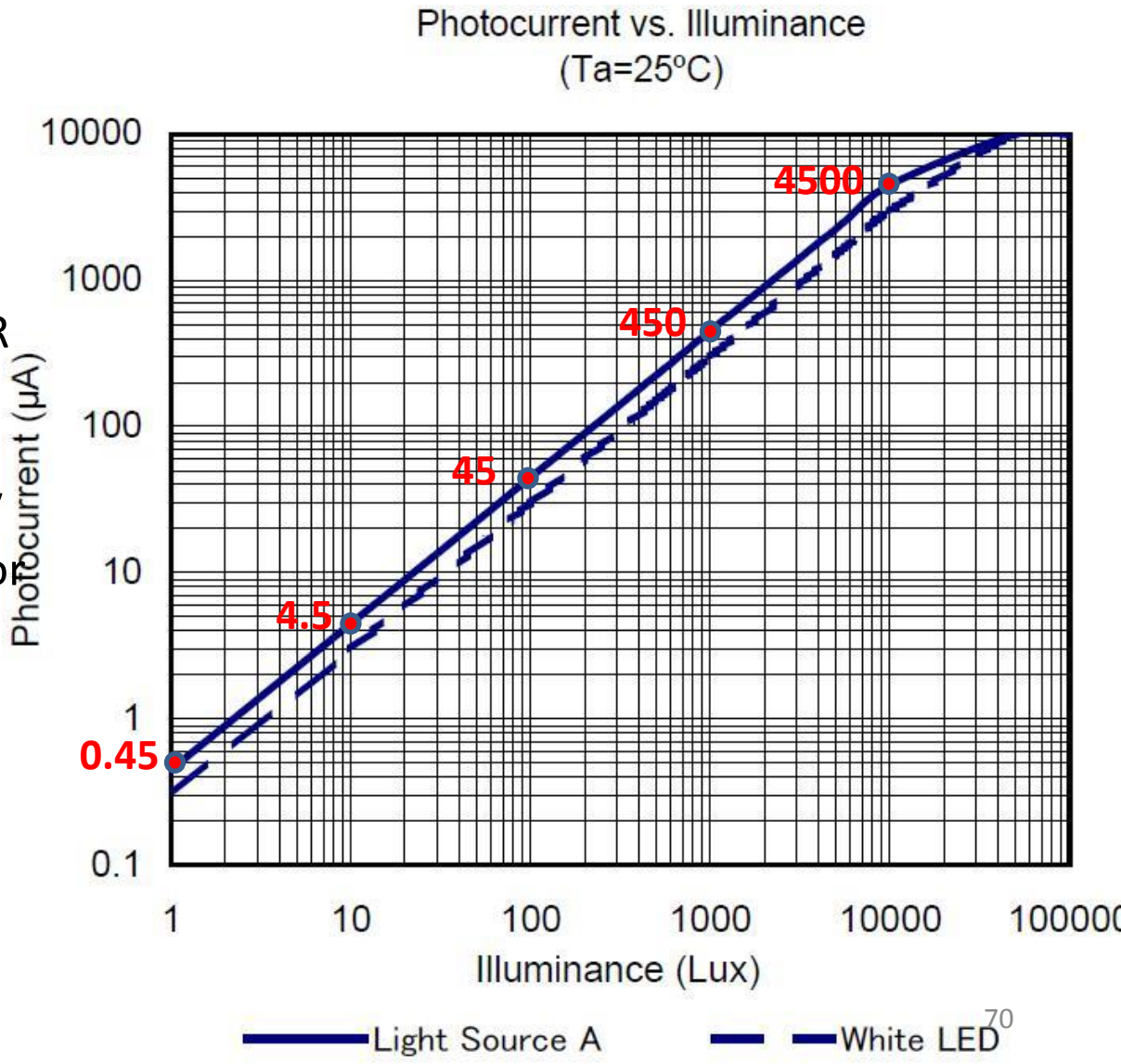
Vref : Reference Voltage

5V / 3.3V or 5000mV /3300 mV

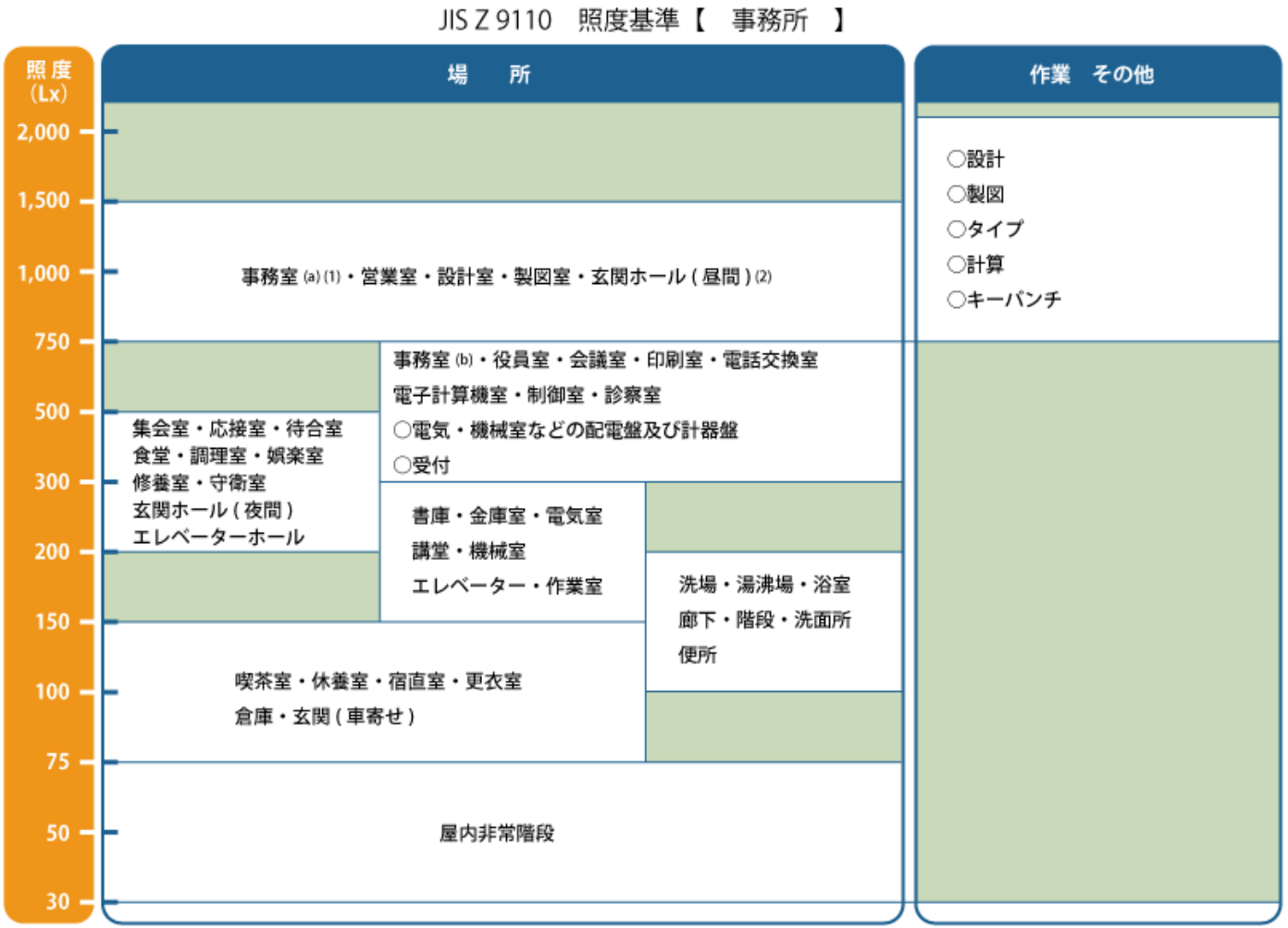
R : Resistor with Photo Transistor

10kΩ

$$\begin{cases} I_{opt} = Vout / R \\ Vout = AIN1 \times (Vref / 1024) \end{cases}$$



3.3.3 Typical Illuminance (From JIS)



注 (1) 事務室は細かい視作業を伴う場合及び昼光りの影響により窓外が明るく、室内が暗く感ずる場合は、(a) を選ぶことが望ましい。
(2) 玄関ホールでは、昼間の屋外自然光による数万 lx の照明に目が順応していると、ホール内部が暗く見えるので、照度を高くすることが望ましい。なお、玄関ホール (夜間) と (昼間) は段階点滅で調節してもよい。

Exercise 6: Pair Network Send illuminance

PhotoTrans1: Read the output voltage of the light sensor and convert to lux

```
const unsigned int sensorPin = A1;    // select the input pin for the potentiometer

void setup() {
  Serial.begin(9600);
}

void loop() {
  unsigned int sensorValue ;  // variable to store the value coming from the sensor
  sensorValue = analogRead(sensorPin);    // read the value from the sensor:
  float vout = sensorValue * 5000.0 / 1024;  // converts to the voltage
  float lux = vout * 222.0 / 1000;
  Serial.println(lux);
  delay(100);
}
```

Example) measure illuminance in the room
Voltage output is Vout =1.0 [V], then photo current is Iopt=100[uA]. From the photo current & illuminance equ., get the 222 [lux].

Exercise 7: Pair Network Send illuminance

PhotoTrans2: if illuminance smaller than 100 [lux] then turn off the LED

```
const unsigned int sensorPin = A1;    // select the input pin for the light sensor
const unsigned int ledPin = 13; // pin 13, on board LED

void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  unsigned int sensorValue ;  // variable to store the value coming from the sensor
  sensorValue = analogRead(sensorPin);    // read the value from the sensor:
  float vout = sensorValue * 5000.0 / 1024;  // converts to the voltage
  float lux = vout * 222.0 / 1000;
  if (lux < 100) { digitalWrite(ledPin, LOW);   // set the LED off
  } else { digitalWrite(ledPin, HIGH); }   // set the LED on
  Serial.println(lux);
  delay(250);
}
```

Aug 2, 2012: Draft

Building Wireless Sensor Networks with XBee and Arduino

The University of Tokushima
Akinori TSUJI

Contact Information :
2-1, Minamijyosanjima-cho, Tokushima, 770-8506, Japan
TEL/FAX : +81-88-656-7485
E-mail: : a-tsuji@is.tokushima-u.ac.jp

Building Wireless Sensor Networks

Day 3

2012/9/13(Thu) 10:00—12:00
Time Estimation: 2 hours

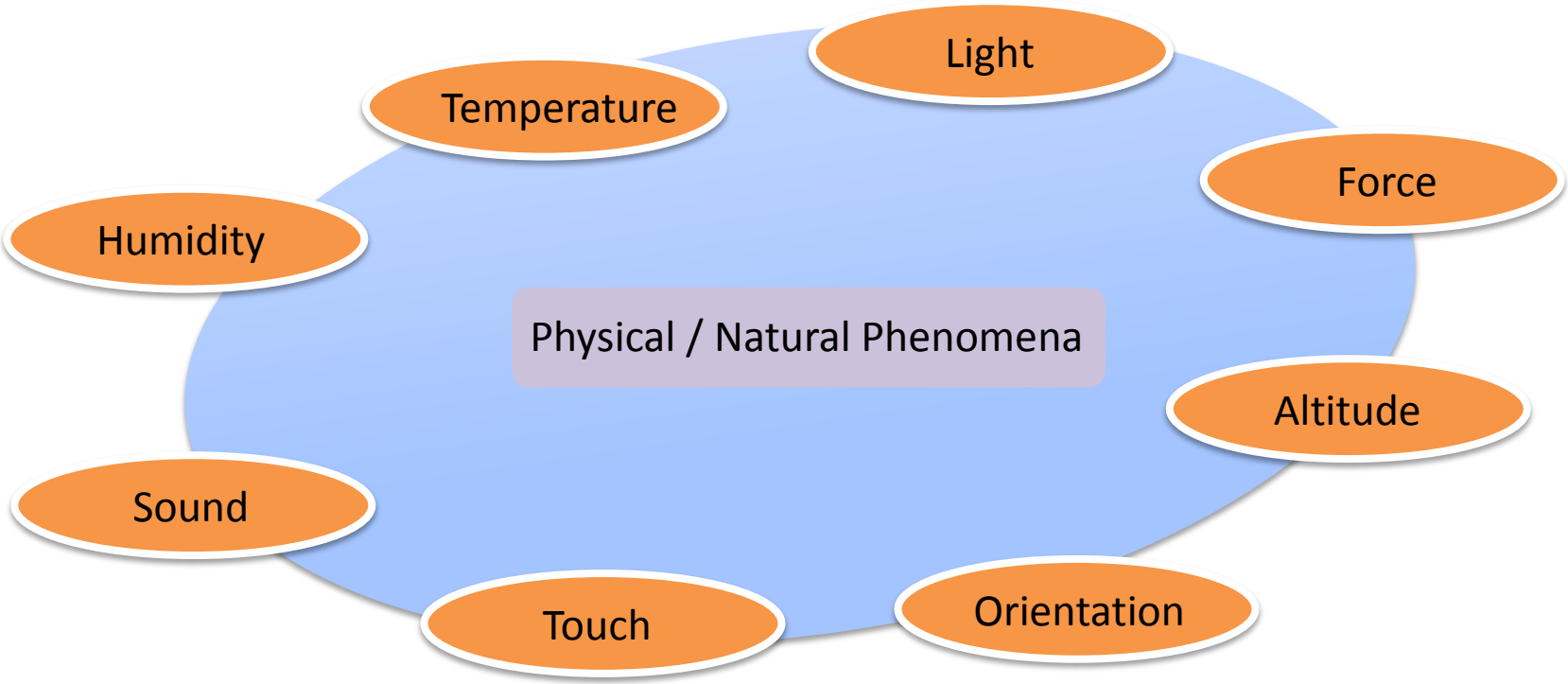
Agenda

1 Sensors and Actuators

2 Building Pair Network (MCU by MCU)
- Switch
- Buzzer
- LCD

3 Building Wireless Sensor Network
- Data Acquisition
- Monitoring
- Controll

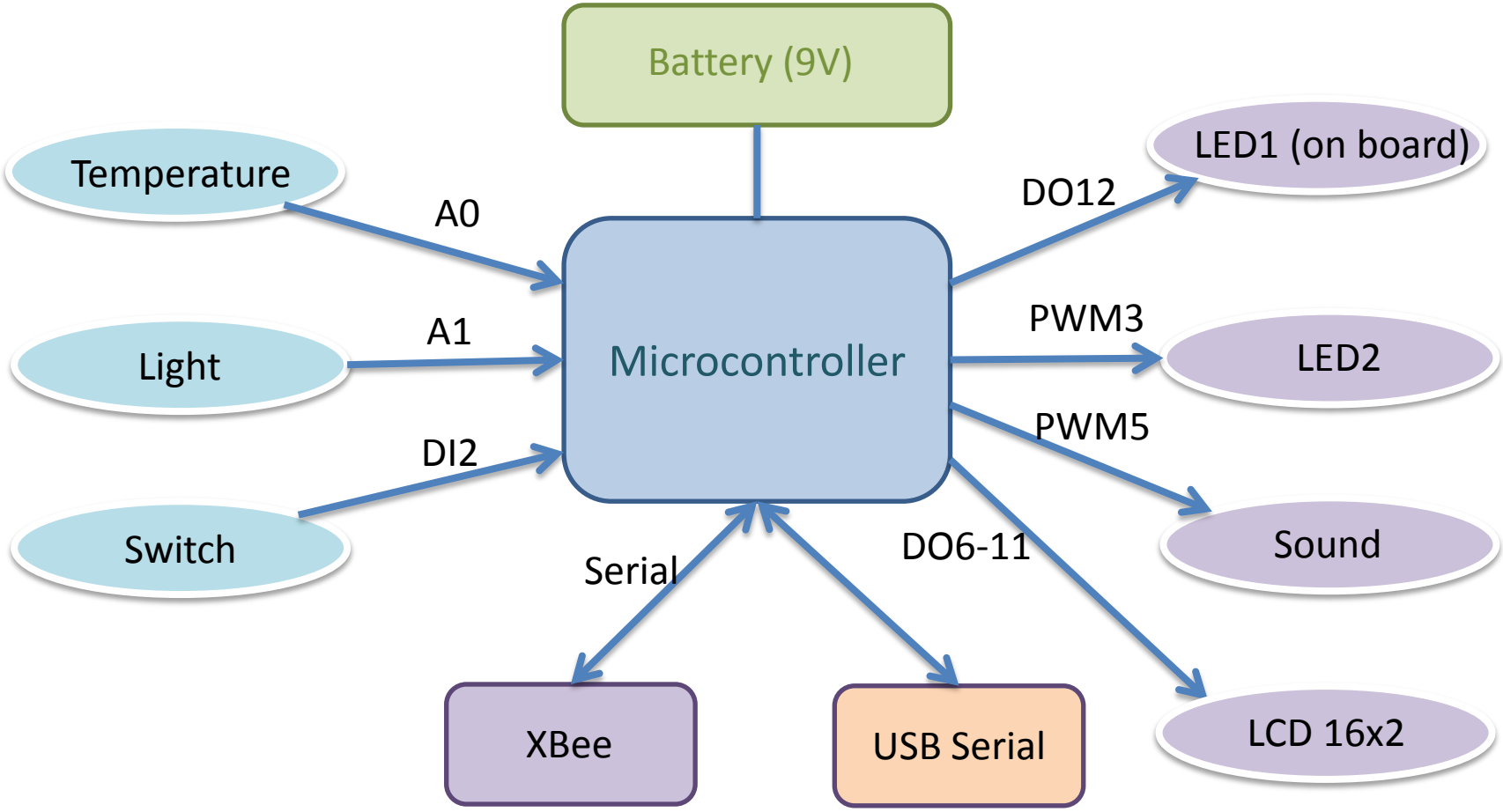
1 Sensors and Actuators



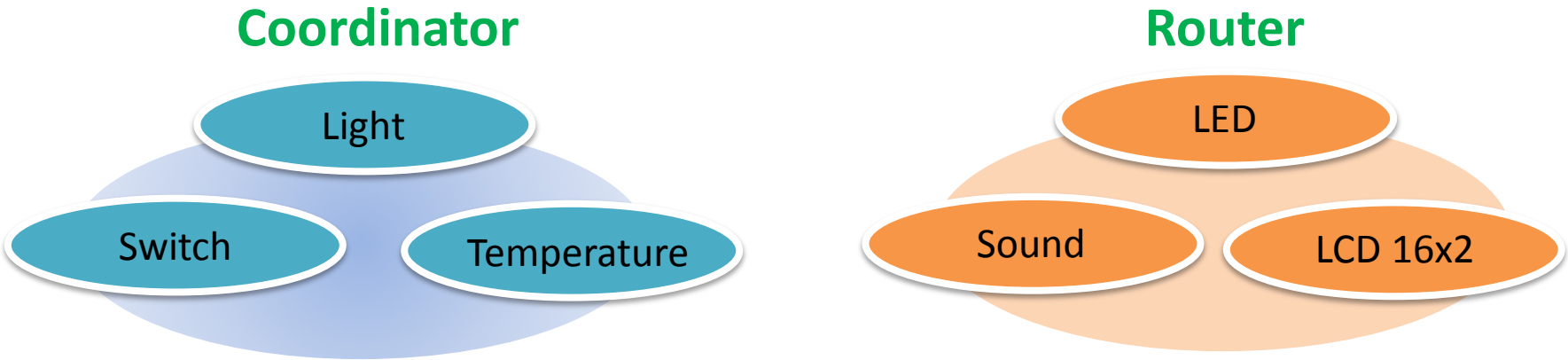
1.1 Sensors

Accelerometer	Accelerations	**Potentiometer**	Rotation on an analog scale
Capacitance	Electrical properties	**Pulse**	Heartbeat rate
Color	Wavelengths of light	**Range finder**	Distance
Pressure	Physical pressure	**Rotary encoder**	Rotation on a digital scale
Gas	Alcohol, Methane, Co2, CO	**Smoke**	Airborne particles
GSR	Galvanic skin	**Stretch**	Physical deformation
Gyroscope	Rotation	**Thermistor**	Temperature
Hall	Magnetic field	**Tilt**	Angular attitude
Microphone	Acoustic sound	**Temperature**	Temperature
Motion	Changes in relative distance		
Phototransistor	Light		

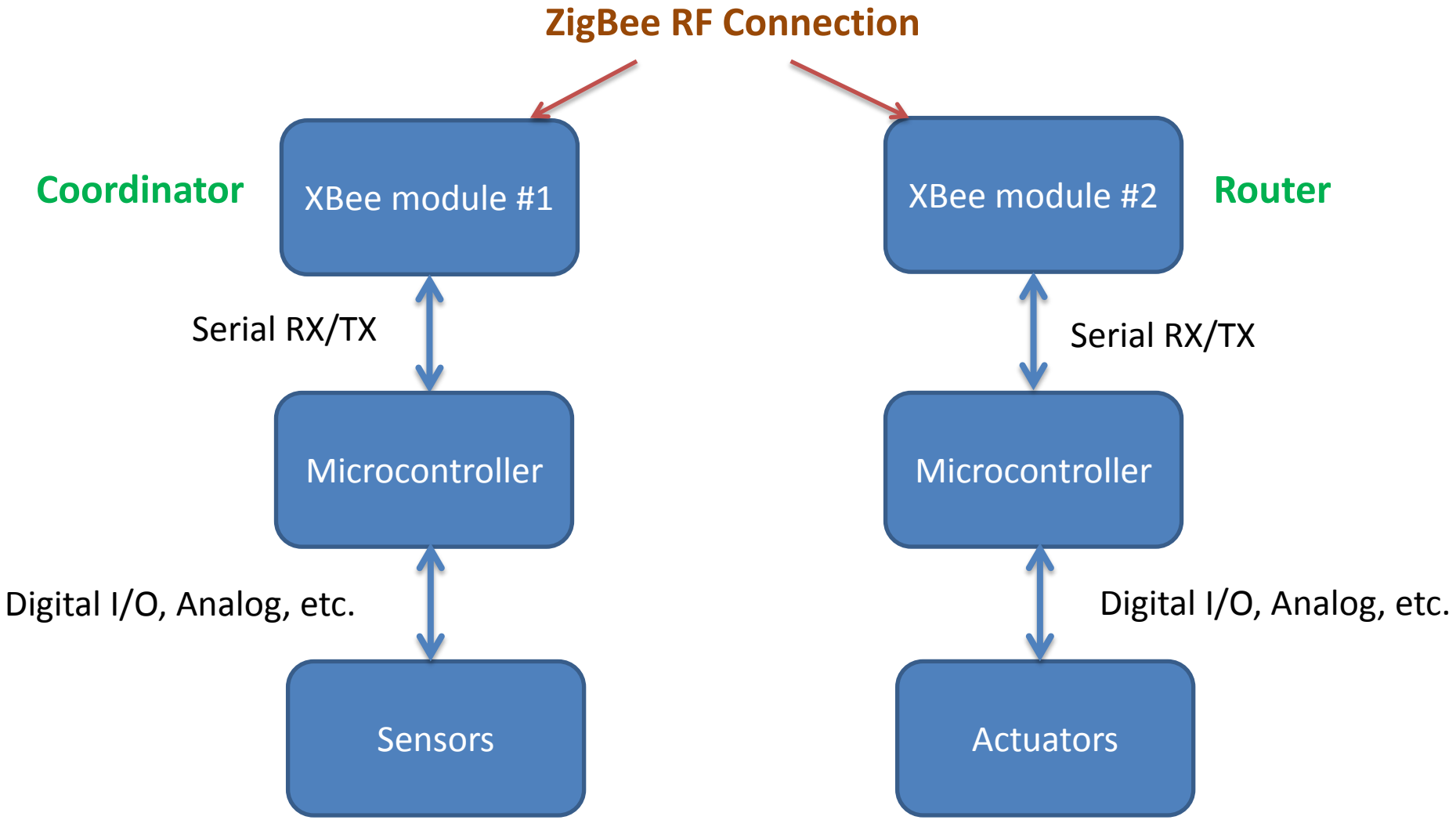
1.2 Wireless Sensor Node



2 Building Pair Network (MCU by MCU)



2.1 Signal Flow of Pair Network (MCU by MCU)

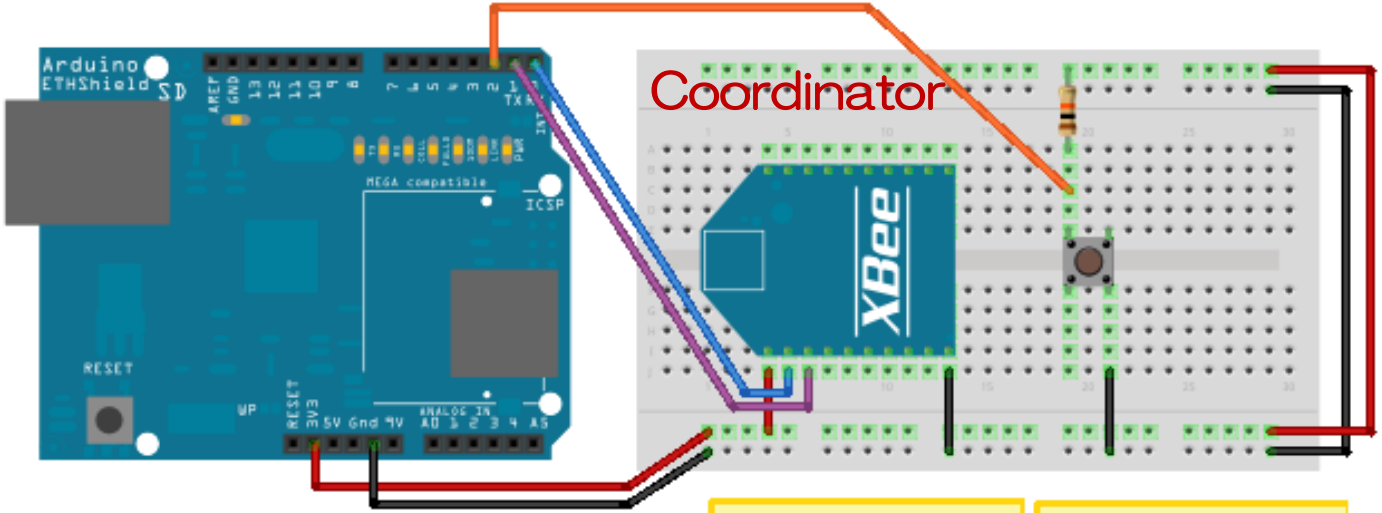


2.2 Prepare XBee Firmware (Coordinator and Router)

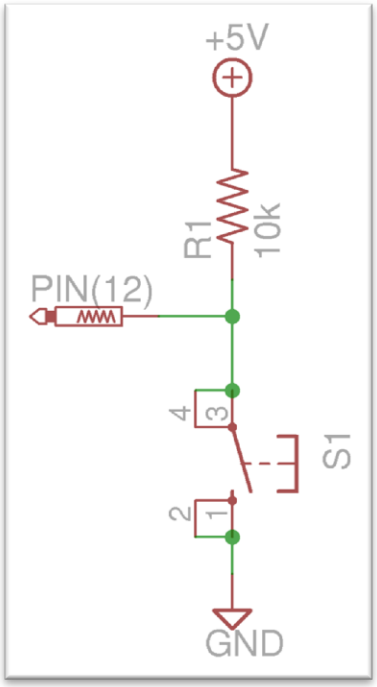
1. Check Serial Number(SH,SL) of each module
2. Write Firmware, Coordinator AT or Router/Enddevice AT
3. Set up same PAN ID for each module
4. Set Destination address (DH, DL)
5. Reconnect the USB cable for restarting the XBee module
6. Communication test on the Terminal tab on X-CTU

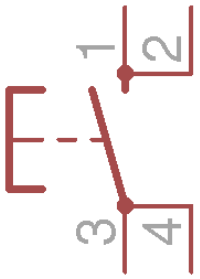
2.3 XBee and Push Switch (Coordinator)

XBee		Arduino	Switch
1 VCC	-----	3.3V	
2 DOUT (TX)	----->	RXD(0)	
3 DIN (RX)	<--- -	TXD(1)	※ insert resistors RX-4.7k-TXD-10k-GND
10 GND	-----	GND	----------Switch
		Digital In (2)	----- Switch----------|
		3.3V	----------Resistor 10k ---|

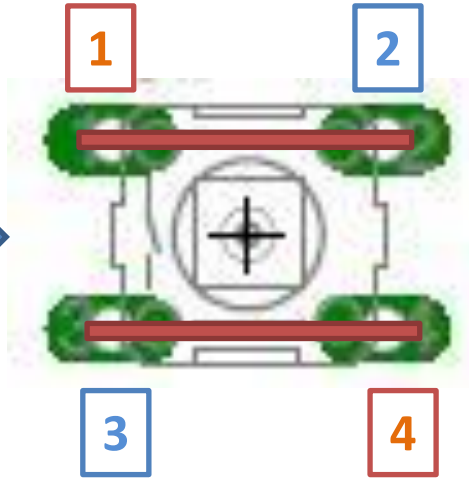


Appendix: Push Switch





Schematic



Parts

Use Diagonal Pins

Switch state	Voltage Level	Digital Value
ON	LOW	0
OFF	HIGH	1

Exercise 1: XBee and Push Switch (Coordinator)

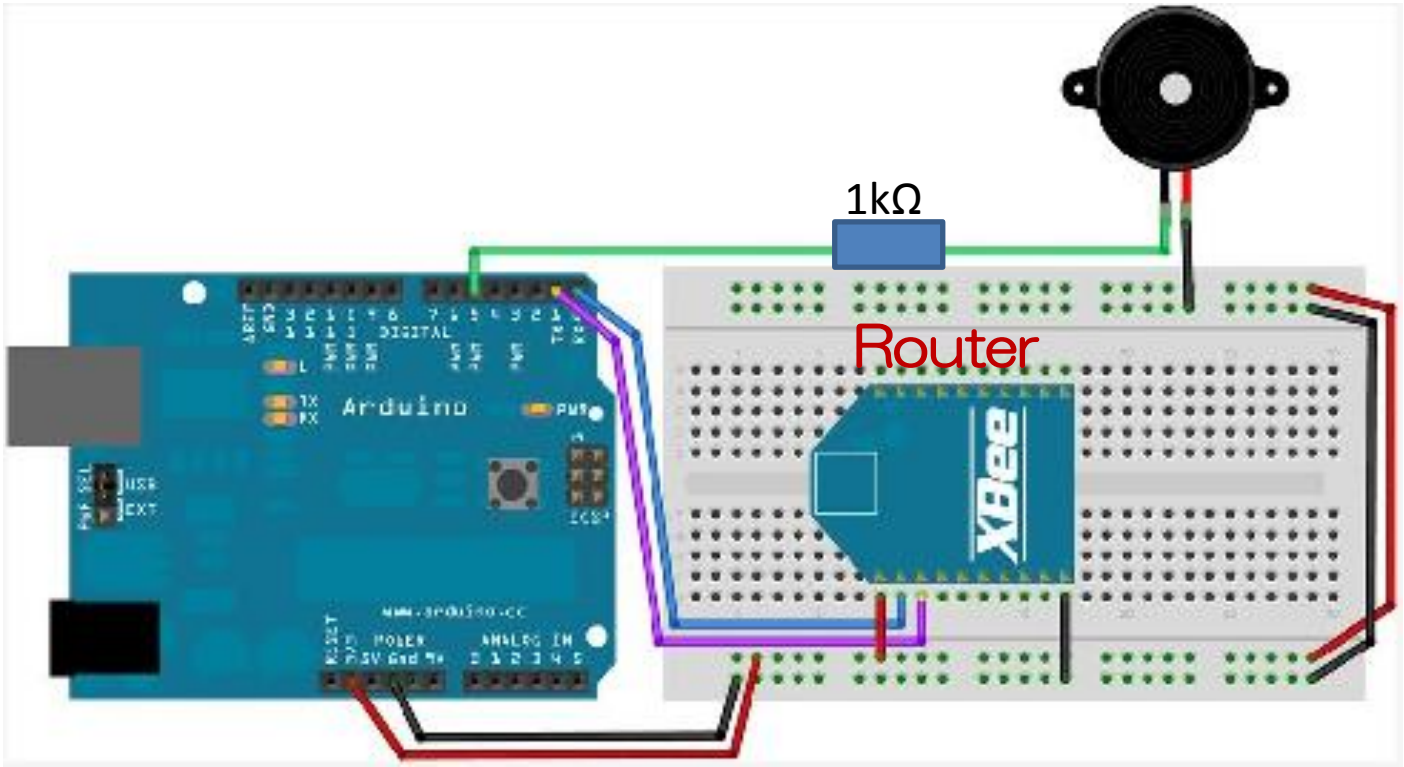
CoordinatorSwitch: send the character 'D' to the router when user push the switch

```
const unsigned int swPin = 2;

void setup() {    // Initialize peripherals
  pinMode(swPin, INPUT);
  Serial.begin(9600);
}

void loop() {  // Infinite loop
  if (digitalRead(swPin) == LOW) { // if switch is pressed
     Serial.print('D');  // send character 'D' to the router
     delay(200); // more than 200ms wait to prevent chattering
  }
}
```

NOTE: Before writing the Arduino program, remove the XBee pins (RX) to prevent collisions of serial communication

2.4 XBee and Buzzer (Router)

XBee		Arduino	
1 VCC	-----	3.3V	
2 DOUT (TX)	-----	RXD(0)	
3 DIN (RX)	-----	TXD(1)	* Insert Register RX-4.7k-TXD-10k-GND
10 GND	-----	GND	---------- Buzzer
		Digital Out (5)	----1kΩ---- Buzzer



Exercise 2: XBee and Buzzer (Router)

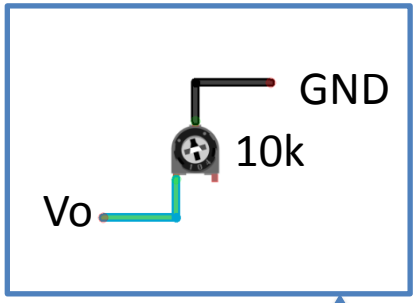
RouterSpeaker: Buzzer turns on when the router received the character 'D'

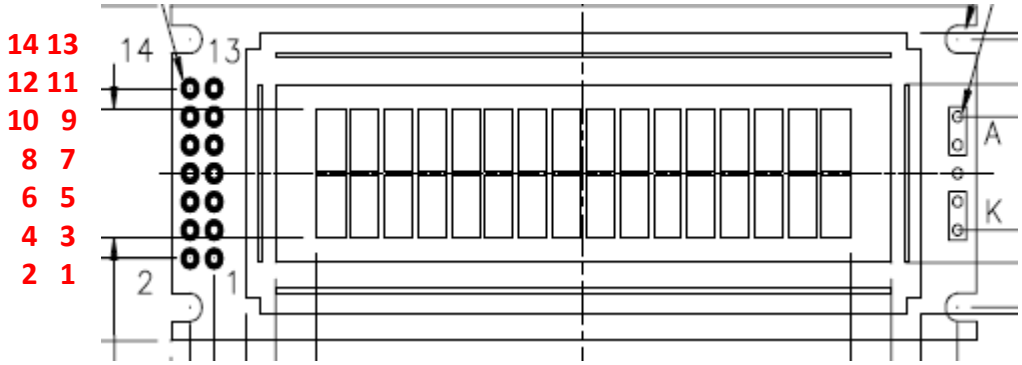
```
const unsigned int speakerPin = 5;  // digital pin 5, PWM port

void setup() {    // Initialize peripherals
  pinMode(speakerPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {  // Infinite loop
  if (Serial.available() > 0) {
    if (Serial.read() == 'D') { // receive character 'D' to the router
      analogWrite(speakerPin, 127); // buzzer on, PWM sets duty cycle 0—255, 490Hz
      delay(100);                   // duty cycle 50% (127)
      analogWrite(speakerPin, 0);  // buzzer off, PWM sets duty cycle 0—255, , 490Hz
    }                               // duty cycle 0% (0)
  }
}
```

2.5 LCD Character Display

LCD SC1602BS-B(-XA-GB-K)

Number of Characters: 16 Characters x 2 lines
Drive Method: 1/5 bias, 1 / 16 duty
Operating Voltage: 4.5V – 5.5V
Controller: HD47780 Compatible





LCD Pin	Arduino Pin
1 Vdd	+5V
2 Vss	GND
3 Vo (Contrast Adjustment)	
4 RS	D11
5 R/W	GND
6 E	D10
11 DB4	D9
12 DB5	D8
13 DB6	D7
14 DB7	D6

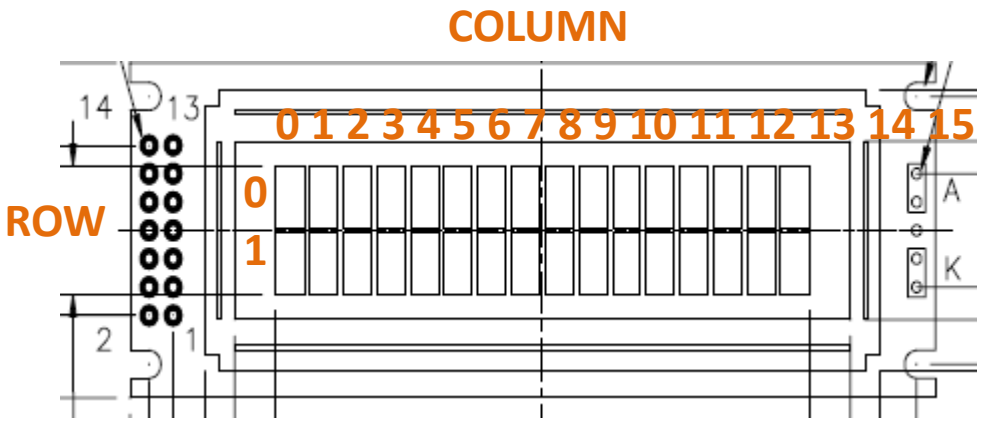
Arduino Function:
LiquidCrystal(rs, enable, d4, d5, d6, d7)
LiquidCrystal(11, 10, 9, 8, 7, 6)

Exercise 3: LCD test1 (1/2)

LCDtest1: Initialize and clear the LCD, display test to show characters and numbers

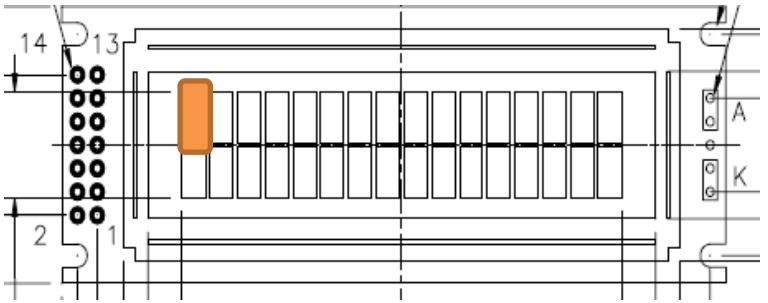
```
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(11, 10, 9, 8, 7, 6); // RS, E, D4, D5, D6, D7

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  lcd.clear();  // clear the display
}
```

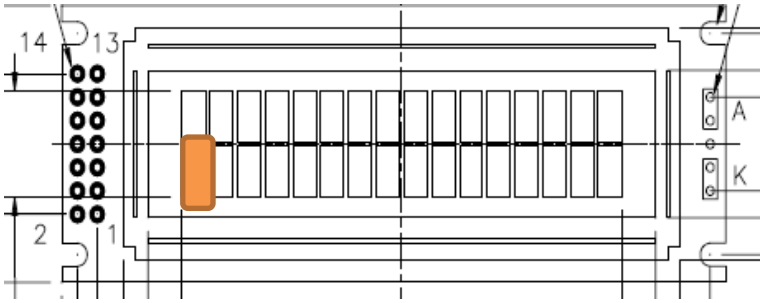
<continue next page>



Exercise 3: LCD test1 (2/2)

```
void loop() {
  static int n = 0;
  float f = 1.2;

 // set the cursor to column 0, line 0
  lcd.setCursor(0, 0);
  lcd.print(“Hello, world!");
  lcd.print(f);

  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  lcd.print(n++);
  delay(200);
}
```

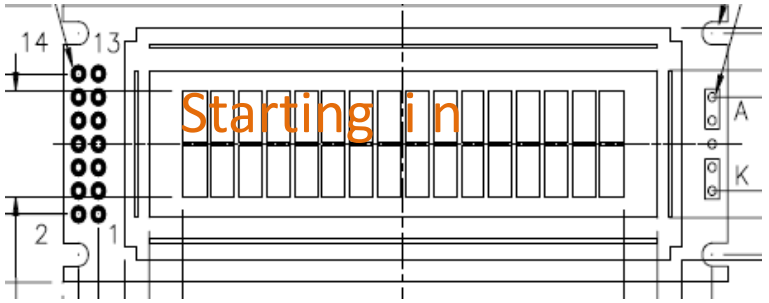


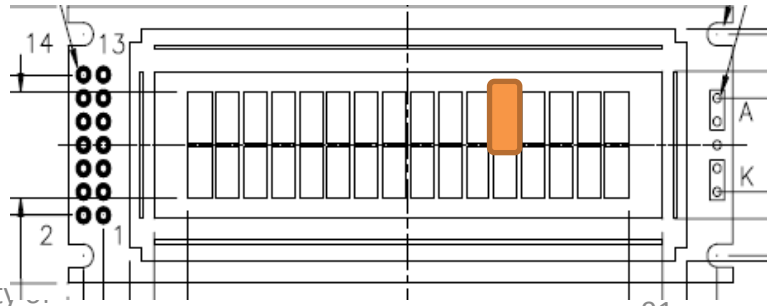


Exercise 4: LCD test (1/2)

LCDtest2: Display shows counting down number and shows three data in one line

```
#include <LiquidCrystal.h>
const int numRows = 2; // lines
const int numCols = 16;
int count;
LiquidCrystal lcd(11, 10, 9, 8, 7, 6); // RS, E, D4, D5, D6, D7

void setup() {
 lcd.begin(16, 2);
 lcd.print("Starting in "); // this string is 12 characters
  for(int i=9; i > 0; i--) // count down from 9
  {
    // the top line is row 0
    lcd.setCursor(12,0); // move the cursor to the end of the string
    lcd.print(i);
    delay(1000);
  }
}
```

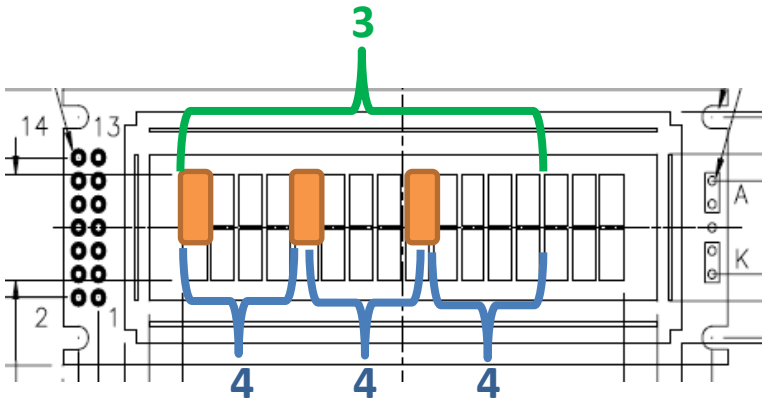




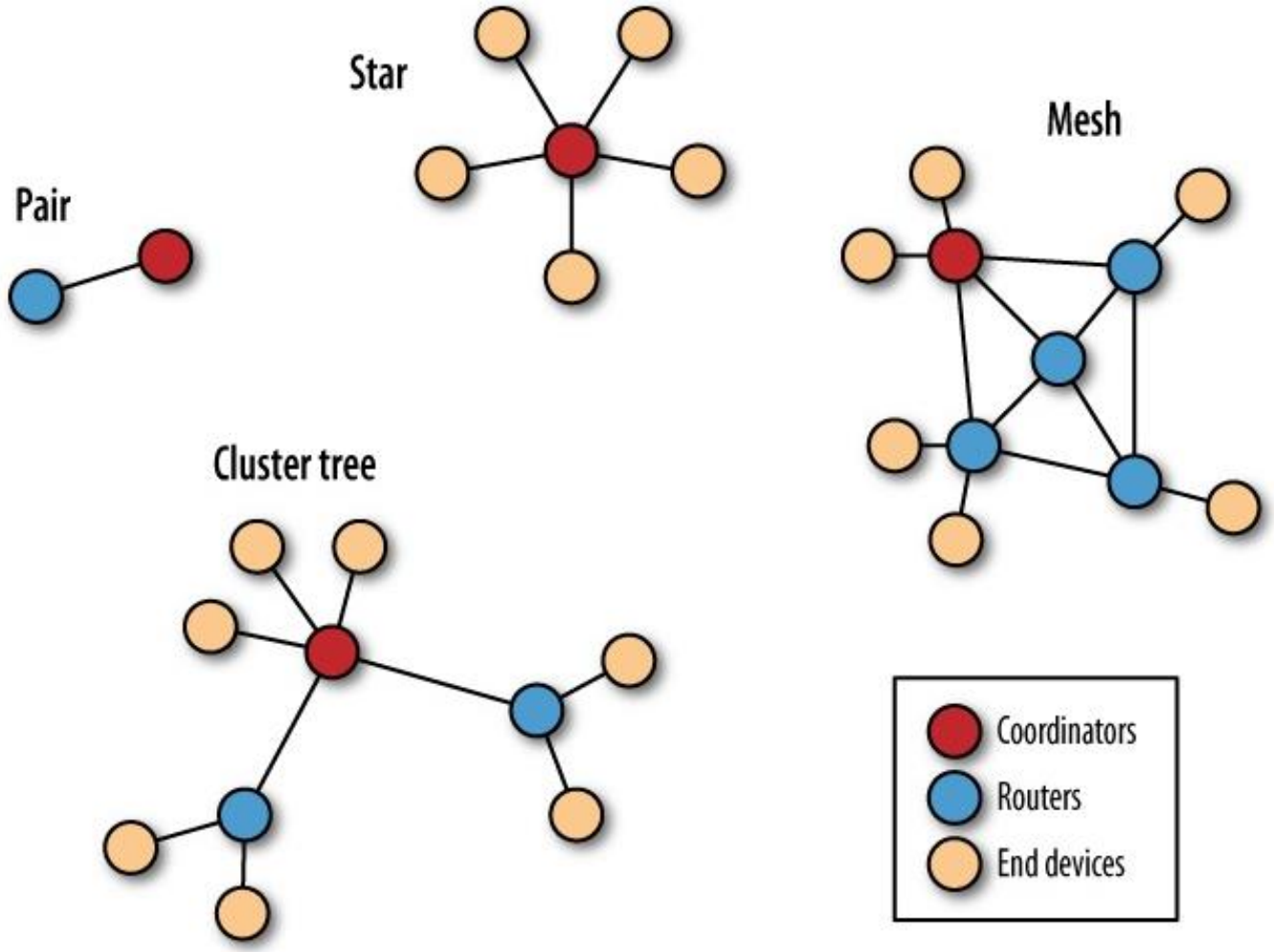
<continue next page>

Exercise 4: LCD test (2/2)

```
void loop()
{
  int columnWidth = 4; //spacing for the columns
  int displayColumns = 3; //how many columns of numbers
  lcd.clear();  // clear the display
  for( int col=0; col < displayColumns; col++)
  {
    lcd.setCursor(col * columnWidth, 0);
    count = count+ 1;
    lcd.print(count);
  }
  delay(1000);
}
```

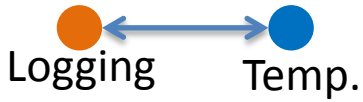


3 Building Your Wireless Sensor Networks

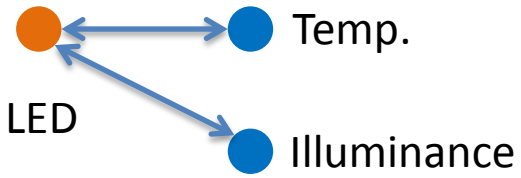


3.1 Examples of Wireless Sensor Networks

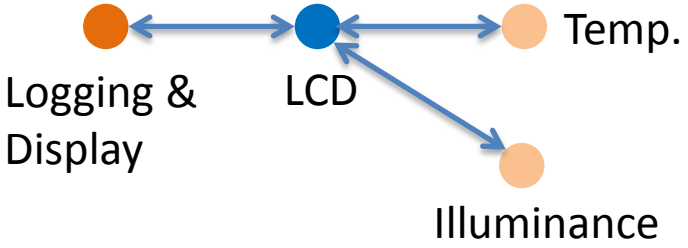
Data Acquisition

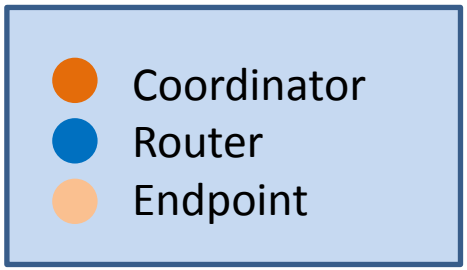


Controller



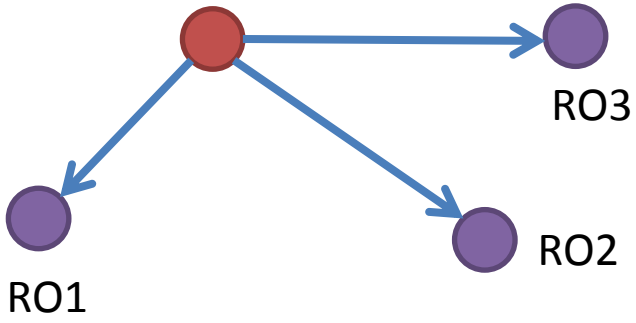
Monitoring





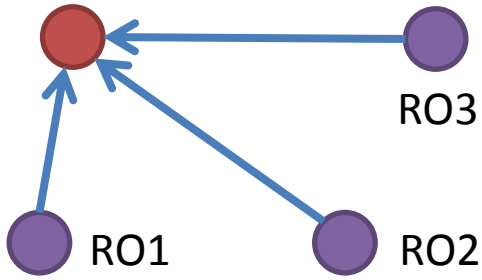
3.2 Broadcast

Broadcast to all nodes



Coordinator
DH 0
DL FFFF

Send to the Coordinator



Router
DH 0
DL 0

3.3 Communication Protocol

Data Format (Coordinator):

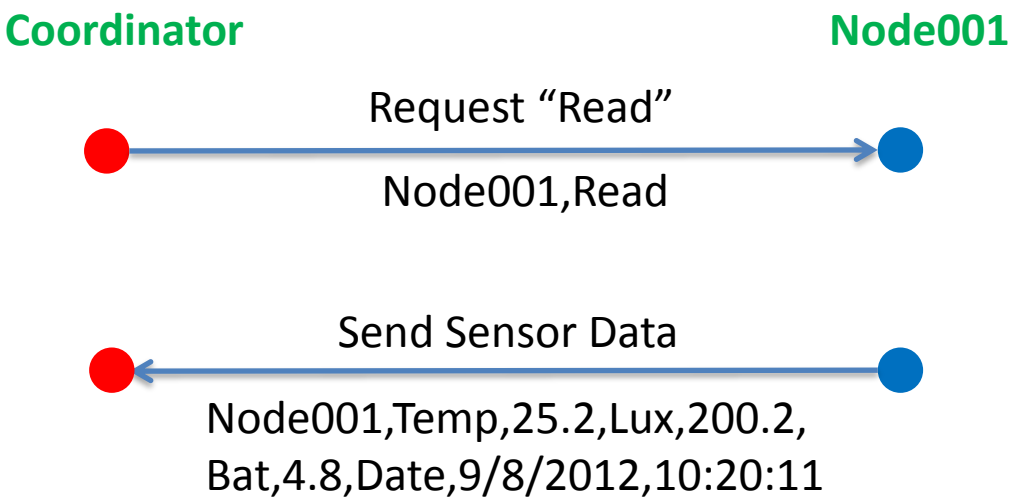
Node Number,CMD

Example) Node001,Read

Data Format (Router):

Node Number,<sensor 1>, <sensor 2>, …, <sensor N>,<battery power>, <date>, <time>

Example) Node001,Temp,25.2,Lux,200.2,Bat,4.8,Date,9/8/2012,10:20:11



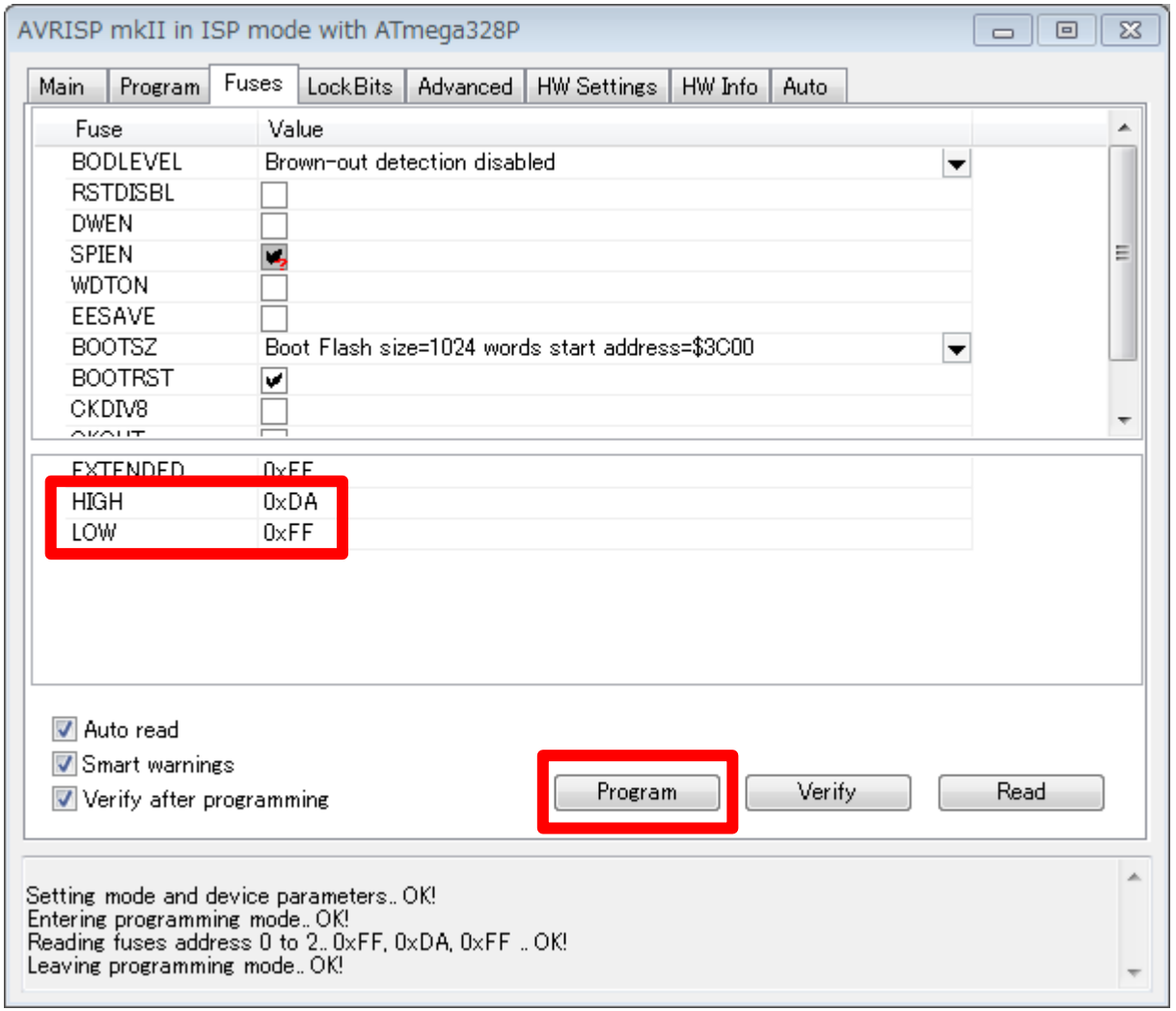
Appendix How to write Arduino firmware

For Arduino Duemilianove w/Atmega328

1. **Connect AVR-ISPMKII to the ICSP pin header**
2. **Power supply from a USB connector**
3. **Run the AVR Studio**
4. **Click on the AVR Button**

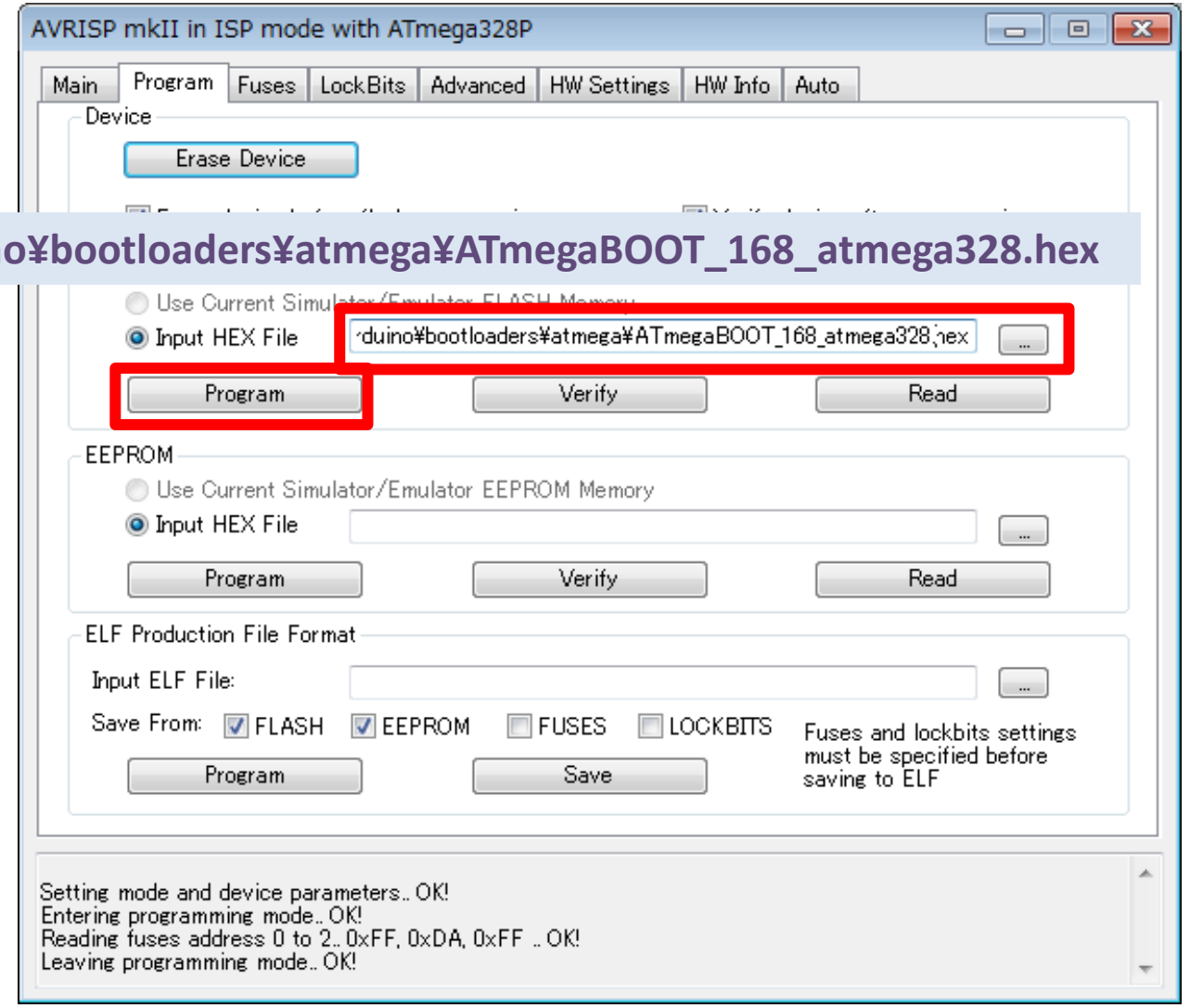
Appendix How to write Arduino firmware

5. Program Fuses Bits
HIGH: 0xDA
LOW: 0xFF



Appendix How to write Arduino firmware

6. Program Arduino Firmware



Revised History

2012/8/2(Sat) Created by Akinori Tsuji
2012/8/11(Sat) Day1, Day2, and Day3 write briefly
2012/8/12(Sun) Day1 almost complete
2012/8/15(Tue) Day2, Day3 write in detail
2012/9/10(Mon) Day1 & Day2 write
2012/9/11(Tue) Released 1st edition
2012/9/12(Wed) Day1, Day 2, and Day3 modified
2012/9/13(Thu) Day3 modified
2012/9/14(Fri) Day1, Day2, and Day3 made some correction

A part of this workshop is supported by the KAKEN SUISIN budget titled in "Building Battery-less Sensing System based on the Energy Harvesting Technology" in Tokushima University