Welcome to our presentation!

3N2-Group 2

Do Xuan Vy

- 1801589

Tran Minh Hien

- 1601116

Nguyen Quoc Huy

- 1601121

THE IDEA

Preparing the tools

- Stm32l432kc
- Xbee
- Magnetfeld sensor

Stm32l432kc

- Why we choose Stm32l432kc ? :

- Cheaper than Raspberry.
- Also more featureful.
- Easy to add extra devices.

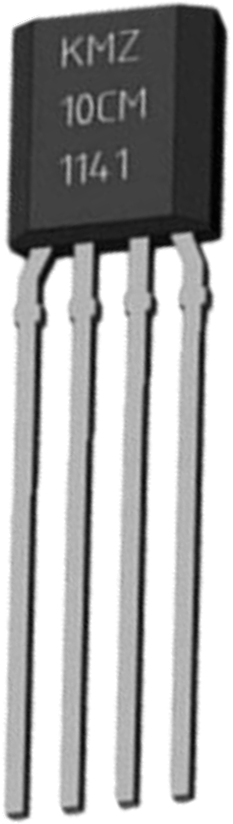
XBee

- Xbee has numerous advantages :
- Transmission frequently band 900Mhz and 2.4Ghz
- Indoor/Urban: up to 100’ (30 m)
- Outdoor line-of-sight: up to 300’ (90 m)
- Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Sensor
G-MRCO-028

- Basic information
- How it work?
- Pros and cons
- How we connect devices
- Principle of operation

Basic information

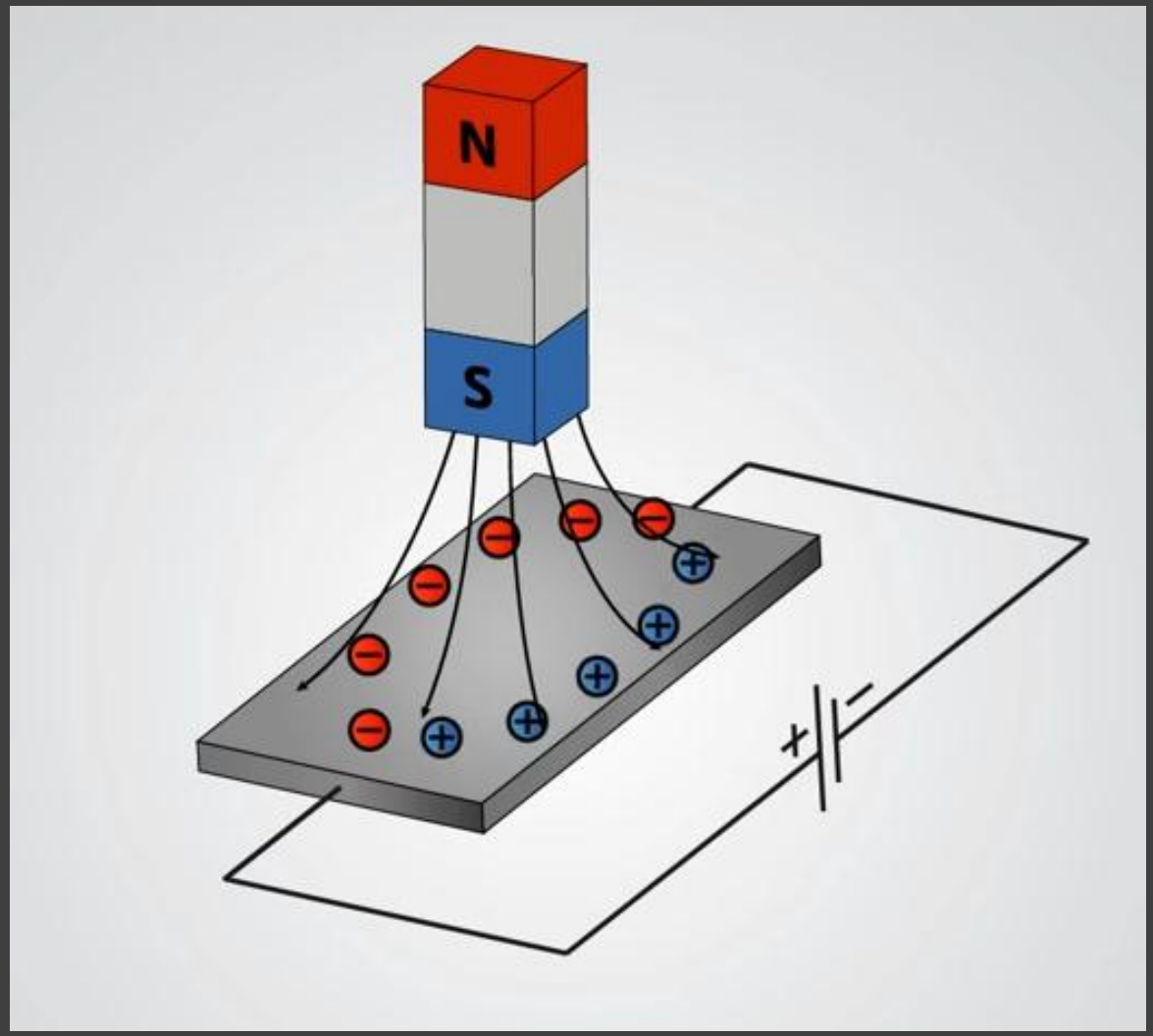


1 2 3 4

Pin	Symbol	Function
1	+Vo	positive output voltage
2	GND	negative supply voltage
3	-Vo	negative output voltage
4	+Vcc	positive supply voltage

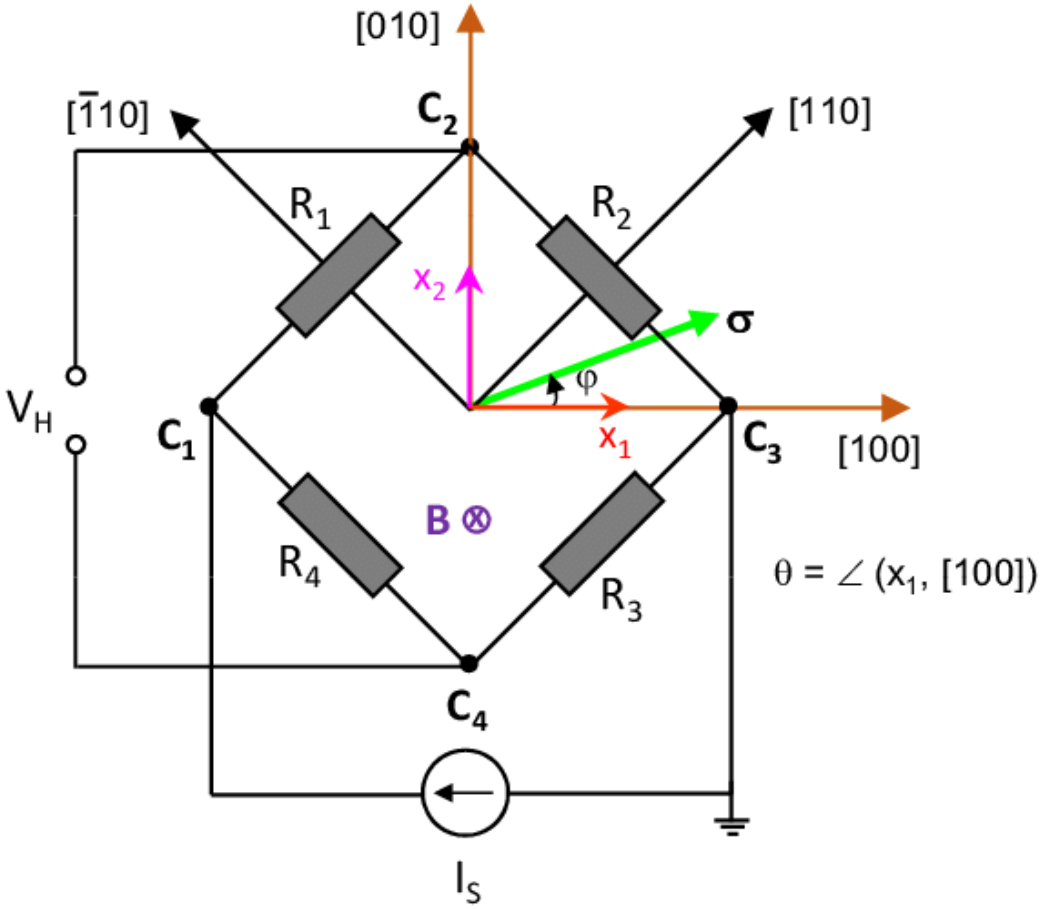
Hall Effect

- When a conductor with an running current is placed into a magnetic field, a voltage difference appears perpendicular to the current



Hall Effect Sensor

- Based on this effect, we can use a Wheatstone bridge to detect the magnetic field in the enviroment
- The strength of the magnetic field is determined by measuring the voltage differential between the two bias
- In this project, the sensor is connected to the differential ADC on the STM32



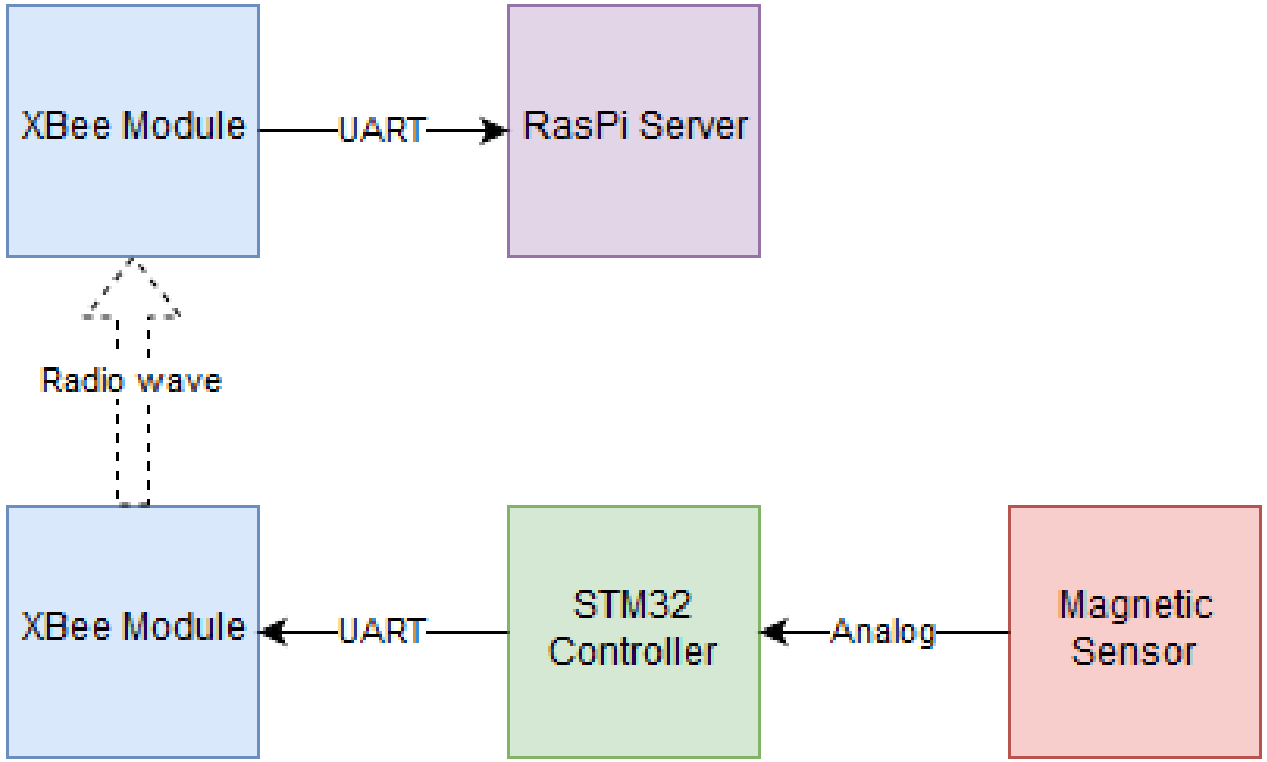
- Linear signal output
- Over increased field range
- Very low hysteresis
- High sensitivity

- Passive component

(Requite a individual circuit to cut the power of the sensor)

- Require amplifiers to increase the power of signal.

HOW WE CONNECT!



Principle of operation

Record the change of magnectic field

STM process the signal

Xbee send it to the another Xbee which connected to server

Server analysit data and display result on screen

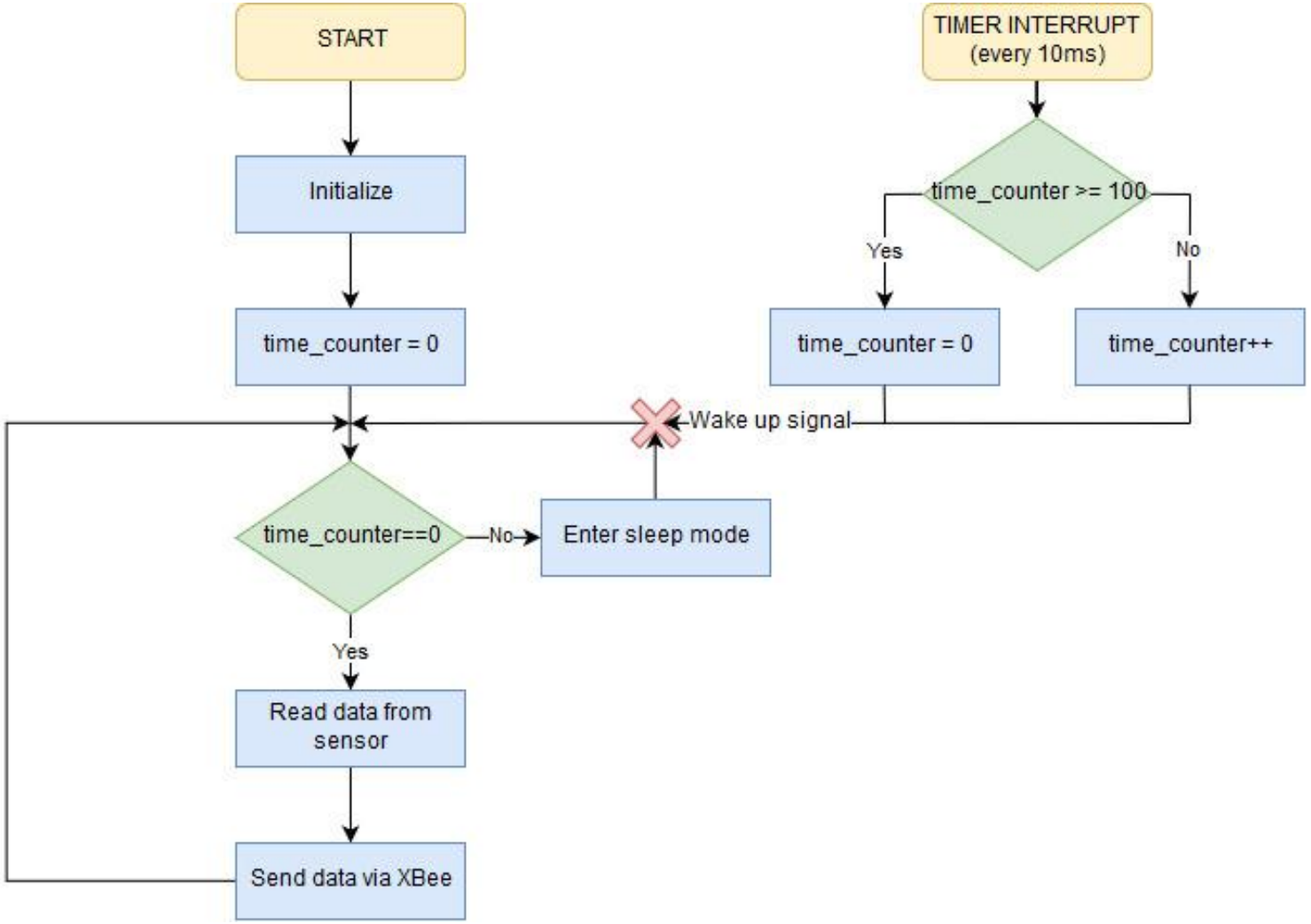
Step 1

Step 2

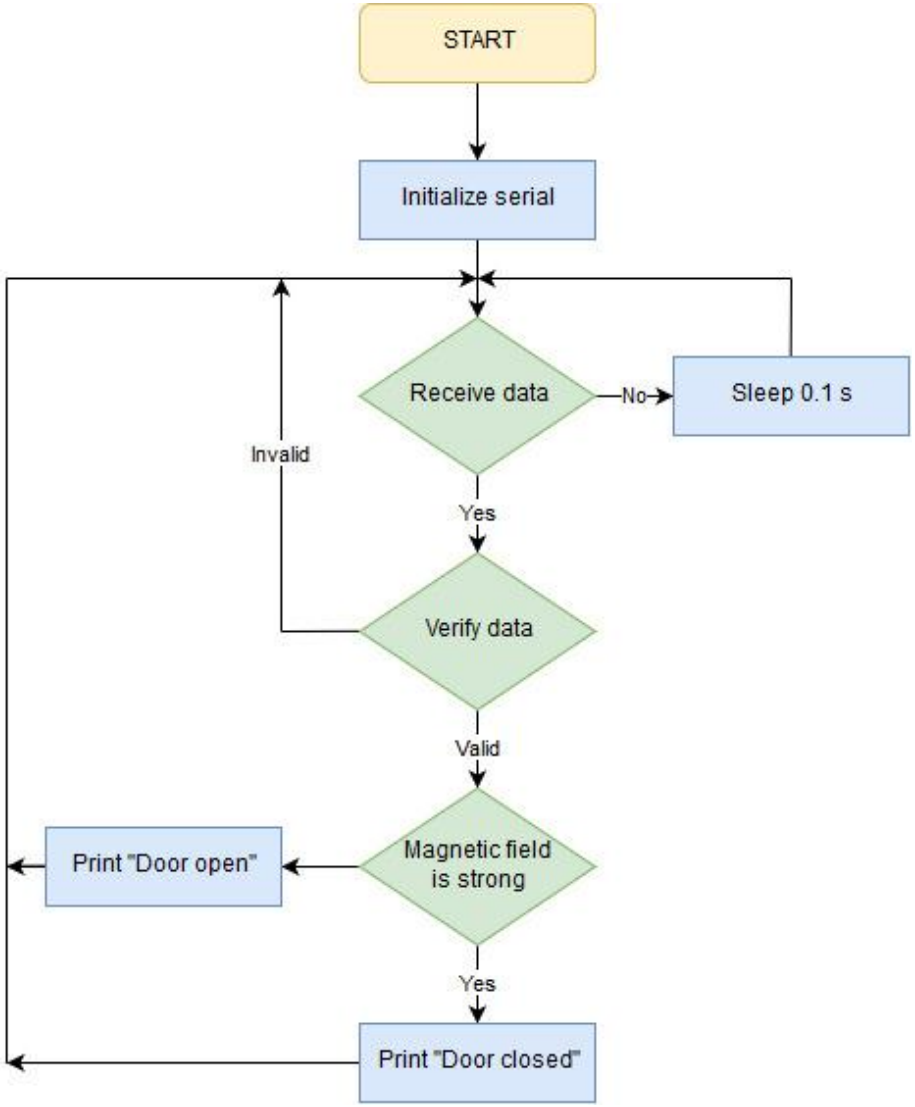
Step 3

Step 4

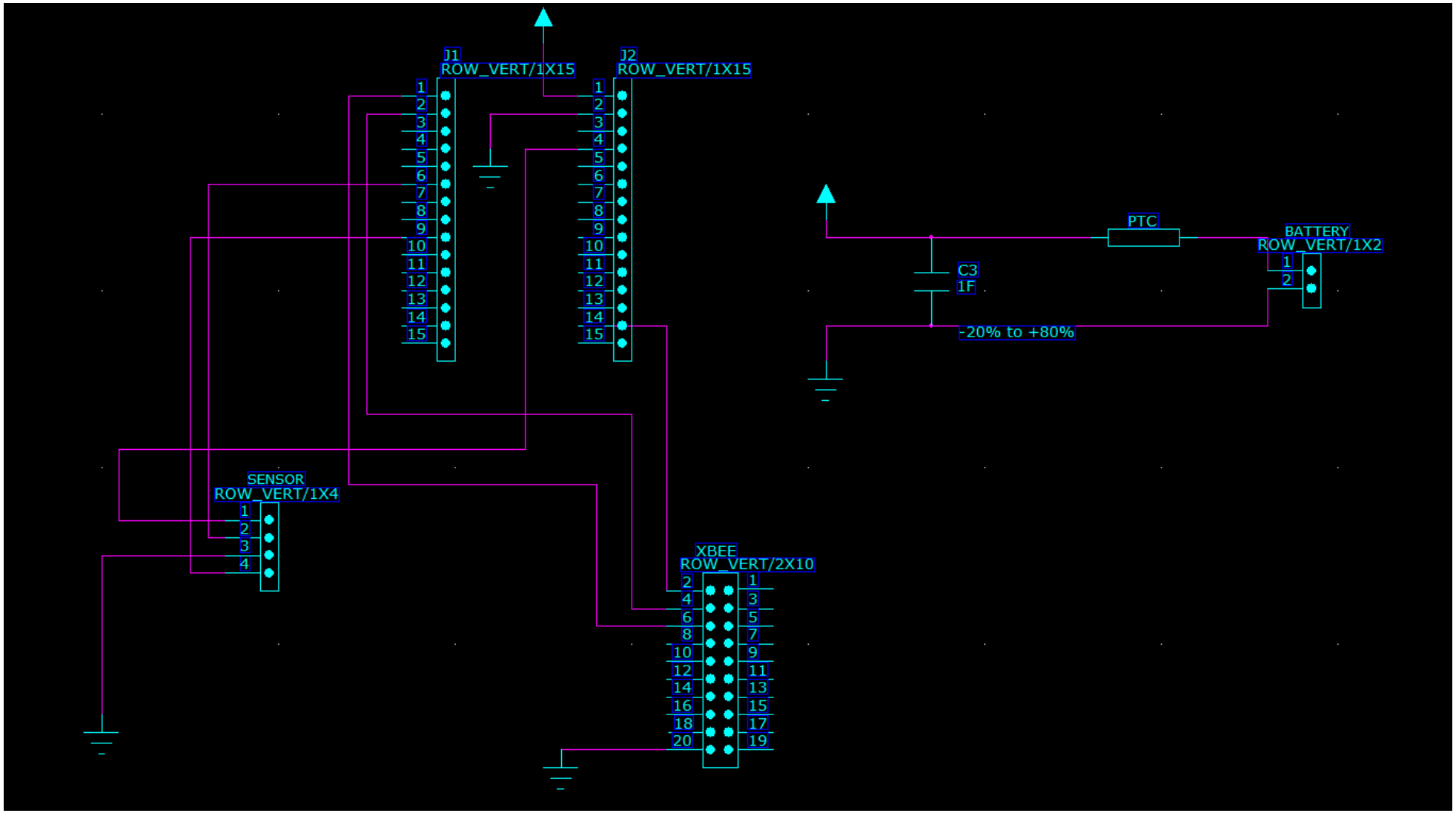
</Here is our Block Diagram>
(Client side)



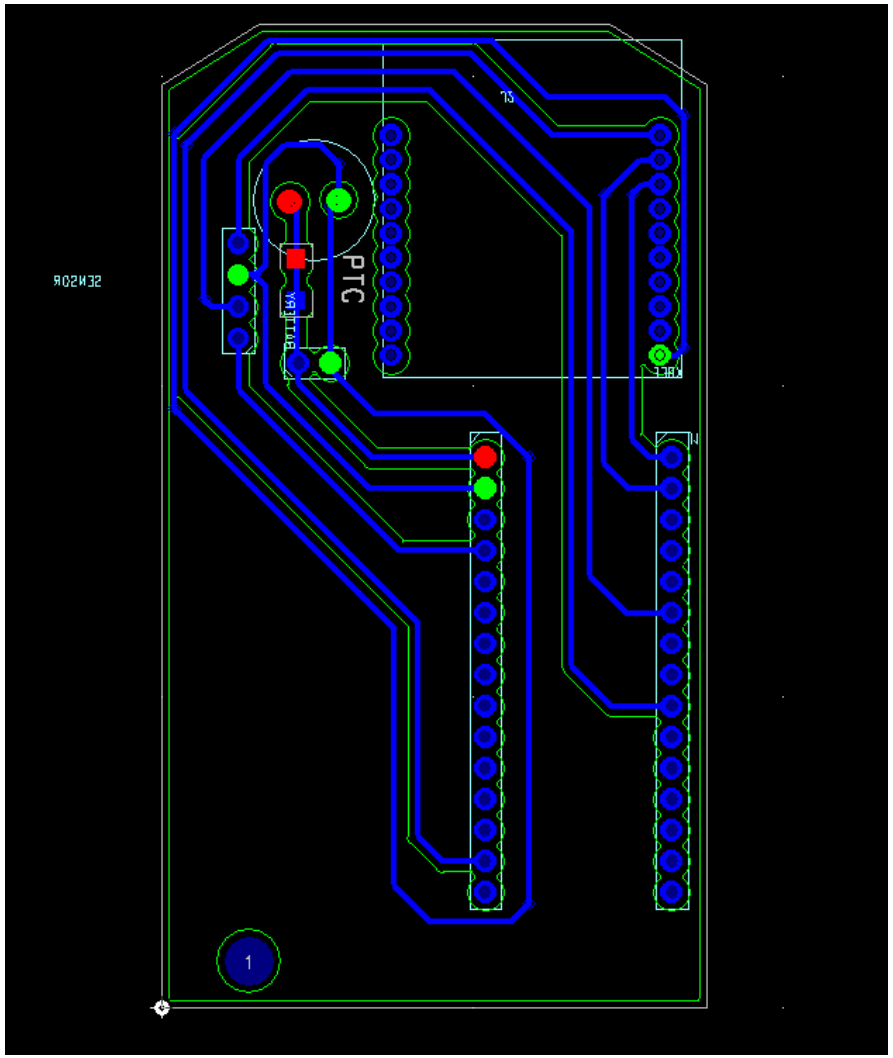
</Here is our Block Diagram>
(Server side)

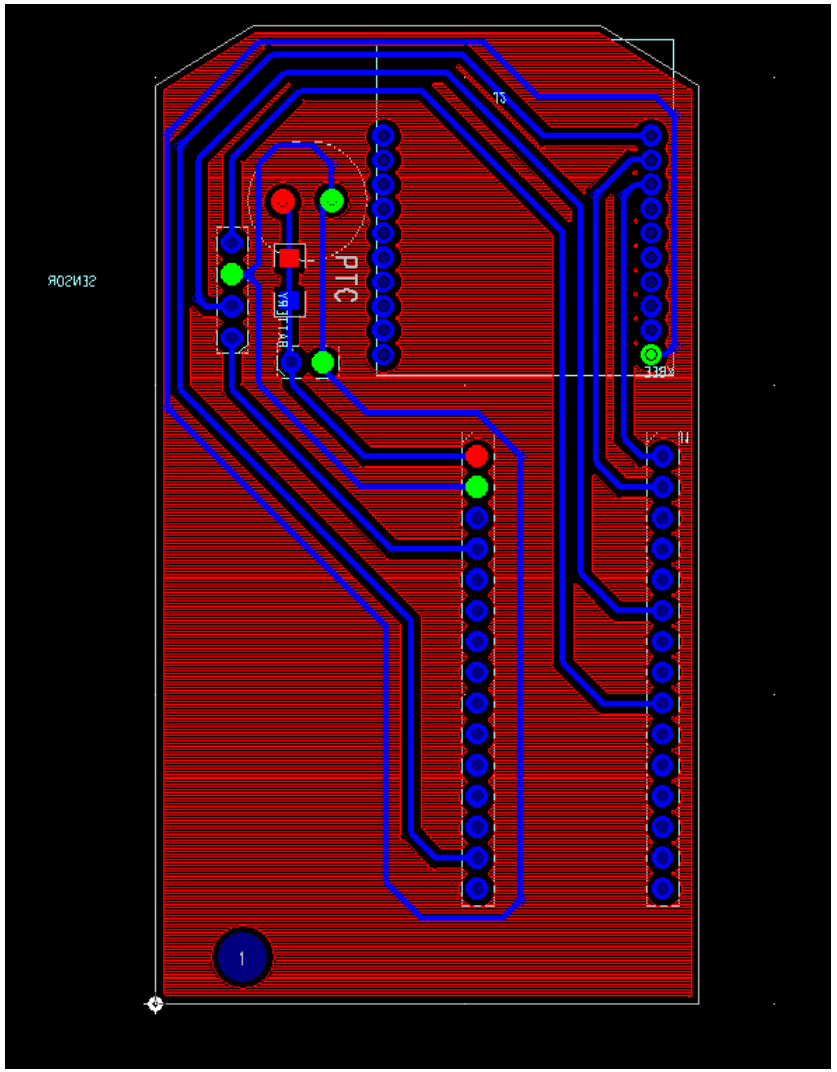


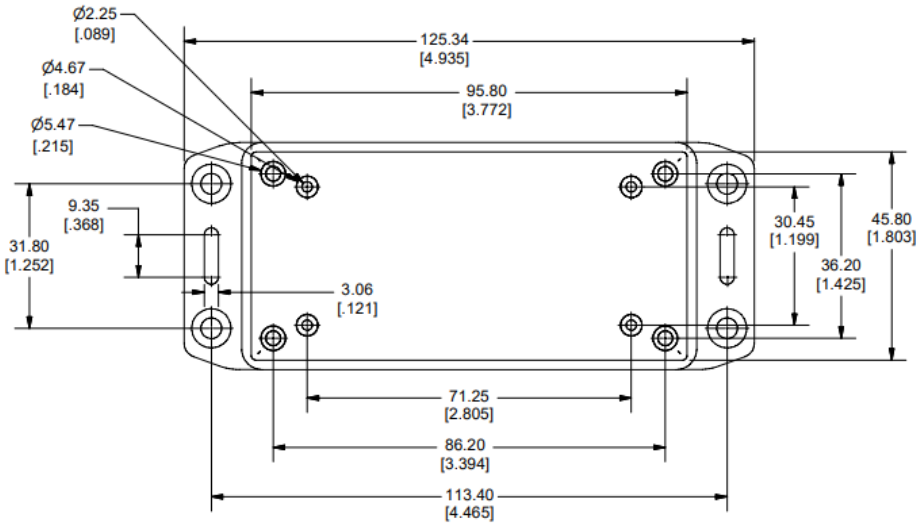
Layout schematic

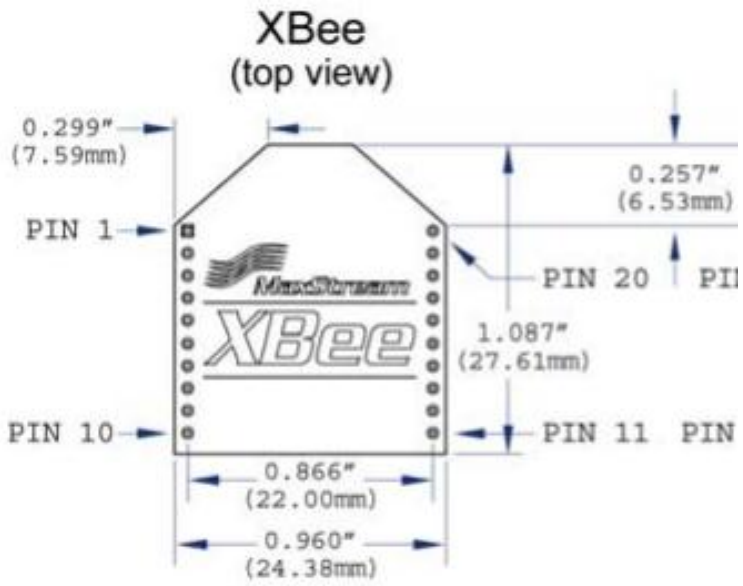


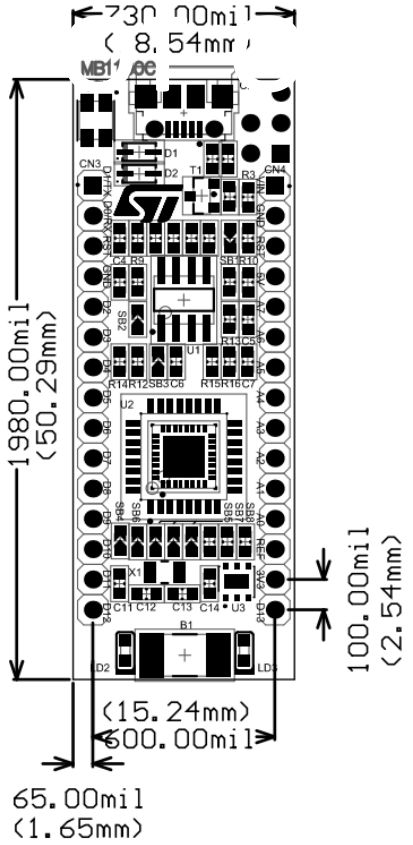
PCB Design

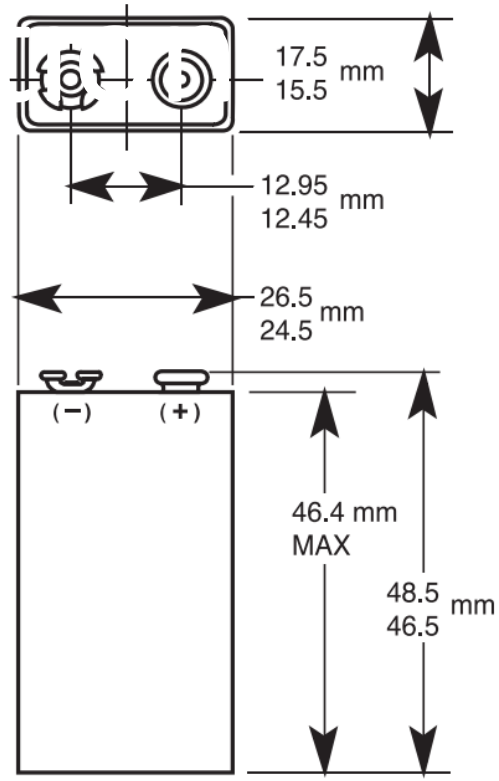




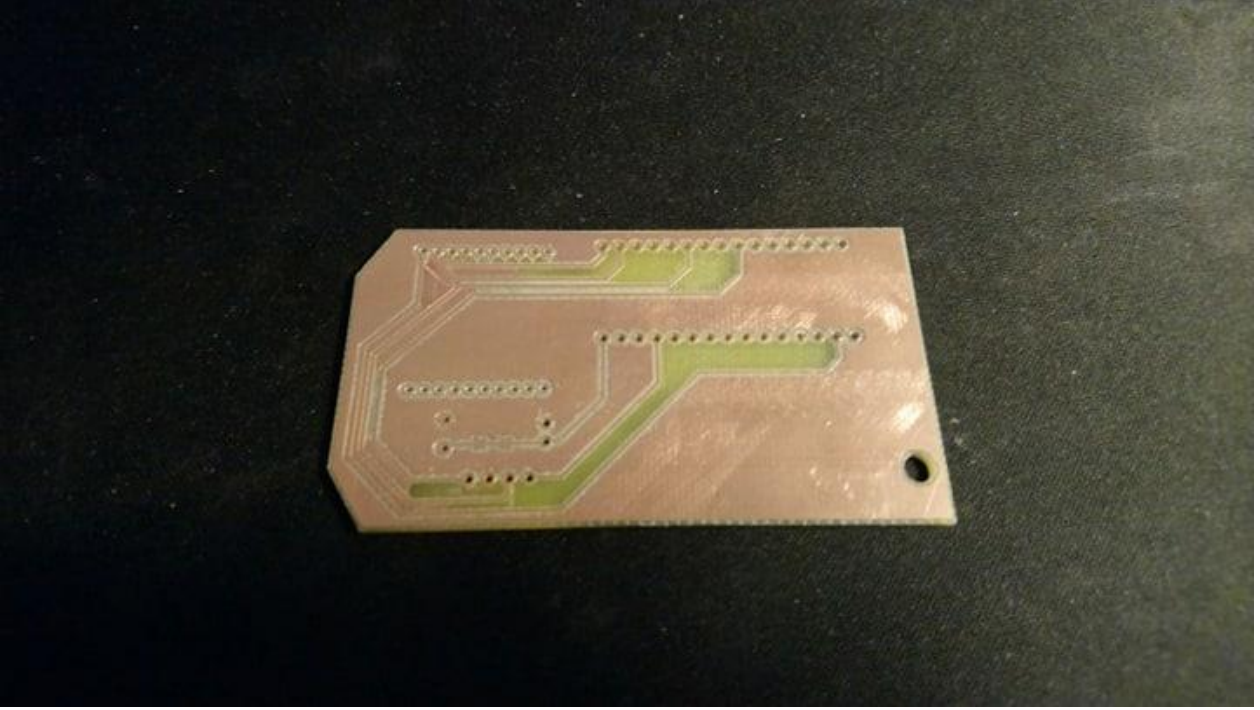


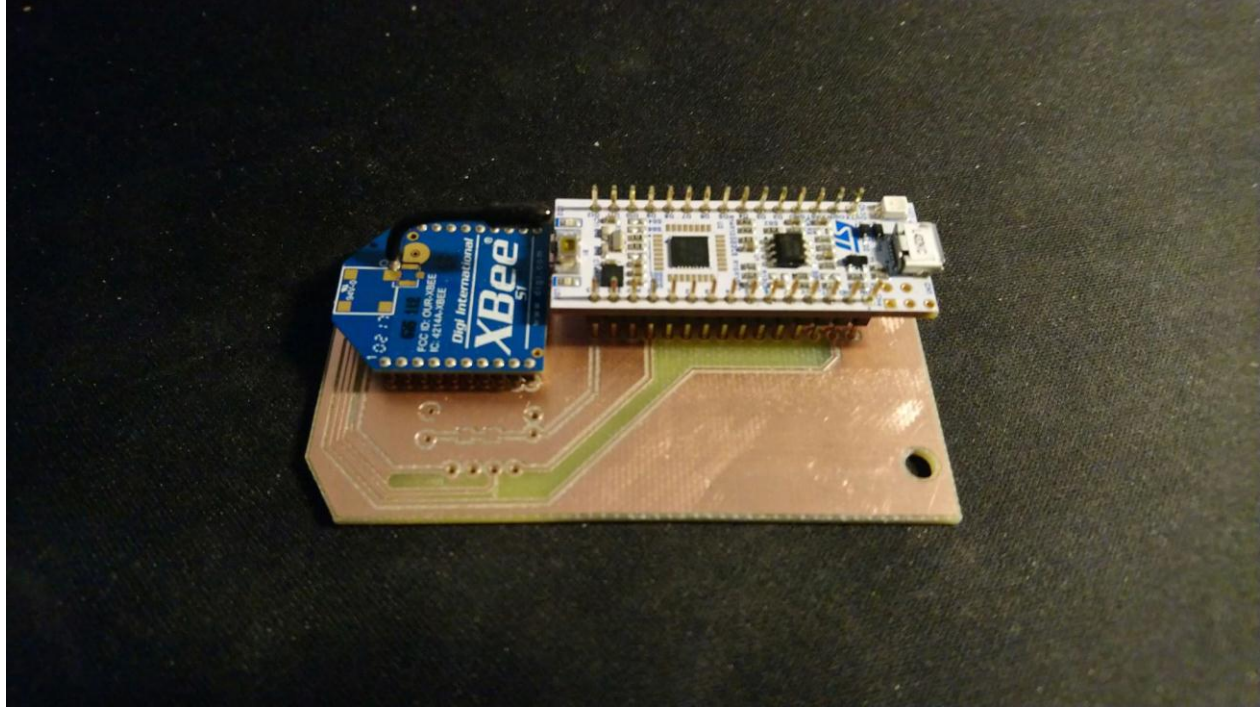


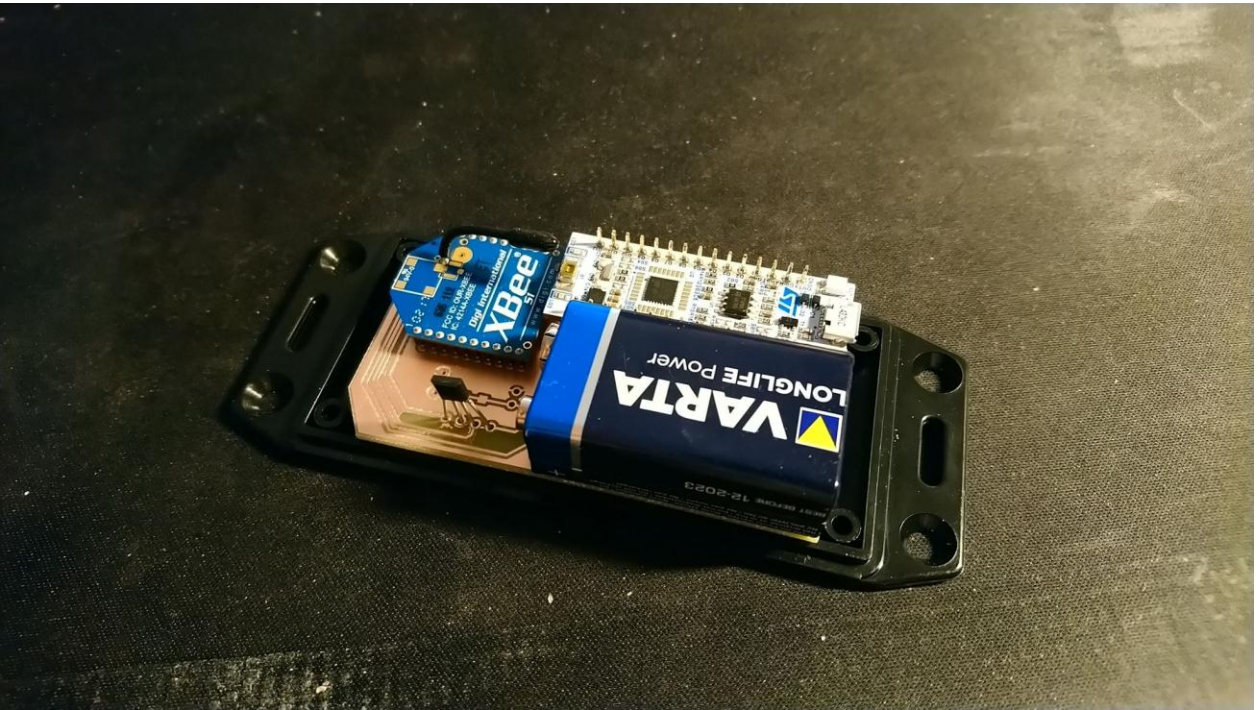




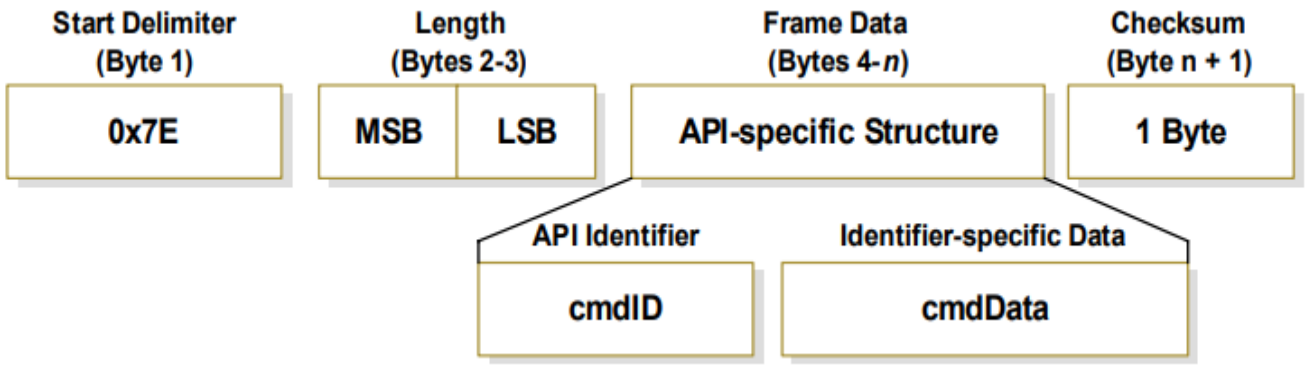
Design PCB to fit the box




XBee
Digi International


XBee
VARTA
LONGLIFE Power

UART frame and checksum programming




```c
void put_int16(uint8_t* pos, uint16_t num) {
    pos[0] = (num & 0xFF00) >> 8;
    pos[1] = num & 0xFF;
}


void init_XBee(UART_HandleTypeDef* huart) {
    huart_g = huart;
}

void send_XBee(uint8_t *data, size_t len) {
    uint8_t buf[XBEE_BUFFER_LENGTH];
    uint16_t sum = 0;

    buf[0] = 0x7E;
    put_int16(buf + 1, len);
    for (size_t i = 0; i < len; i++) {
        buf[i + 3] = data[i];
        sum += data[i];
        sum &= 0xFF;
    }
    buf[len + 3] = 0xFF - sum;
    HAL_UART_Transmit(huart_g, buf, len + 4, 100);
}

void concat_XBee(uint8_t *buf, char* data, size_t offset) {
    for (size_t i = 0; data[i]; i++) {
        buf[i + offset] = data[i];
    }
}

void tx_req_XBee(uint8_t id, uint16_t addr, uint8_t opt, char* data) {
    uint8_t buf[XBEE_BUFFER_LENGTH];

    buf[0] = 0x01;
    buf[1] = id;
    put_int16(buf + 2, addr);
    buf[4] = opt;

    concat_XBee(buf, data, 5);
    send_XBee(buf, strlen(data) + 5);
}
```

Sleep mode programming

Sever programming

File Edit Search Options Help

```
#!/usr/bin/python3

import serial
from time import sleep

# Configuration
zero = 1988
range = 10

xbee = serial.Serial ("/dev/ttyS0", 9600)

while True:
        while xbee.inWaiting() < 1:
                sleep(0.01)
        c = xbee.read()
        if c == b"\x7e":
                buf = xbee.read(2)
                len = buf[0]*256 + buf[1]
                apid = int.from_bytes(xbee.read(1), "big")
                if apid == 0x81:
                        addr = int.from_bytes(xbee.read(2), "big")
                        str = int.from_bytes(xbee.read(1), "big")
                        opt = int.from_bytes(xbee.read(1), "big")
                        data = xbee.read(len-5).decode("UTF-8").rstrip()
                        if addr == 0xAB02 and data[:2] == "2,":
                                value = abs(int(data[2:])-zero)
                                if value >= range:
                                        print("Door closed")
                                else:
                                        print("Door open")
```

Testing

thanks for listening!