function Encode(fPort, obj) {

 var rst = [];

 if(fPort < 10 || fPort > 17){

 console.log("Unknown fPort,expect([10,17]),actual(" + fPort + ")");

 return rst;

 }

 if(null == obj){

 console.log("obj is null");

 return [];

 }

 if(obj.msgId <0 || obj.msgId > 255){

 return [];

 }

 var arrayIndex = 0;

 switch(fPort){

 //It's a parameters setting message.

 case 10:

 {

 var index;

 rst[arrayIndex++] = obj.msgId;

 var power = [0,1,2,3];

 index = power.indexOf(obj.txPower);

 if(-1 != index){

 rst[arrayIndex++] = 1;

 rst[arrayIndex++] = index;

 }

 var dr = [0,1,2];

 index = dr.indexOf(obj.dr);

 if(-1 != index){

 rst[arrayIndex++] = 2;

 rst[arrayIndex++] = index;

 }

 var auReport = ["disable","enable"];

 index = auReport.indexOf(obj.bleAuReport);

 if(-1 != index){

 rst[arrayIndex++] = 3;

 rst[arrayIndex++] = index;

 }

 var blePeriod = [0,5,10,20,30,60,120,300,600,900,1200,1800,3600,7200,21600,43200];

 index = blePeriod.indexOf(obj.blePeriod);

 if(-1 != index){

 rst[arrayIndex++] = 4;

 rst[arrayIndex++] = index;

 }

 var bleScan = [1,2,3,6,9,12,15,255];

 index = bleScan.indexOf(obj.bleScan);

 if(-1 != index){

 rst[arrayIndex++] = 5;

 rst[arrayIndex++] = index;

 }

 var scale = [0,1,2,3];

 index = scale.indexOf(obj.scale);

 if(-1 != index){

 rst[arrayIndex++] = 6;

 rst[arrayIndex++] = index;

 }

 var bleStepsOff = [0,1,2,3,4,5,6,7];

 index = bleStepsOff.indexOf(obj.bleStepsOff);

 if(-1 != index){

 rst[arrayIndex++] = 7;

 rst[arrayIndex++] = index;

 }

 var bleBleOff = [0,1,2,3,4,5,6,7];

 index = bleBleOff.indexOf(obj.bleBleOff);

 if(-1 != index){

 rst[arrayIndex++] = 8;

 rst[arrayIndex++] = index;

 }

 var warnBuzzer = ["disable","enable"];

 index = warnBuzzer.indexOf(obj.warnBuzzer);

 if(-1 != index){

 rst[arrayIndex++] = 9;

 rst[arrayIndex++] = index;

 }

 var warnVibrator = ["disable","enable"];

 index = warnVibrator.indexOf(obj.warnVibrator);

 if(-1 != index){

 rst[arrayIndex++] = 10;

 rst[arrayIndex++] = index;

 }

 var warnDistance = [2,4,6,8,10,15,255];

 index = warnDistance.indexOf(obj.warnDistance);

 if(-1 != index){

 rst[arrayIndex++] = 11;

 rst[arrayIndex++] = index;

 }

 var warnProximity = ["disable","enable"];

 index = warnProximity.indexOf(obj.warnProximity);

 if(-1 != index){

 rst[arrayIndex++] = 12;

 rst[arrayIndex++] = index;

 }

 var gnssPeriod = [0,10,20,30,60,120,300,600,1800,3600,7200,10800,21600,43200];

 index = gnssPeriod.indexOf(obj.gnssPeriod);

 if(-1 != index){

 rst[arrayIndex++] = 13;

 rst[arrayIndex++] = index;

 }

 var heartBeatPeriod = [60,300,600,1200,1800,3600,7200,21600,43200,86400];

 index = heartBeatPeriod.indexOf(obj.heartBeatPeriod);

 if(-1 != index){

 rst[arrayIndex++] = 14;

 rst[arrayIndex++] = index;

 }

 if(null != obj.time){

 if(obj.time.hour >= 0 && obj.time.hour <=23 &&

 obj.time.minute >= 0 && obj.time.minute <= 59 &&

 obj.time.second >=0 && obj.time.second <=59){

 rst[arrayIndex++] = 15;

 rst[arrayIndex++] = obj.time.hour;

 rst[arrayIndex++] = obj.time.minute;

 rst[arrayIndex++] = obj.time.second;

 }

 if(null != obj.time.date){

 rst[arrayIndex++] = 19;

 rst[arrayIndex++] = obj.time.date >> 8;

 rst[arrayIndex++] = obj.time.date & 0xff;

 }

 }

 else{

 if(null != obj.datetime){

 rst[arrayIndex++] = 15;

 rst[arrayIndex++] = (obj.datetime >> 24) & 0xff;

 rst[arrayIndex++] = (obj.datetime >> 16) & 0xff;

 rst[arrayIndex++] = (obj.datetime >> 8) & 0xff;

 rst[arrayIndex++] = obj.datetime & 0xff;

 }

 }

 if(null != obj.sleepy){

 if(obj.sleepy.start >= 0 && obj.sleepy.start <=23 &&

 obj.sleepy.end >= 0 && obj.sleepy.end <= 23 &&

 obj.sleepy.degree >=0 && obj.sleepy.degree <=7){

 rst[arrayIndex++] = 16;

 rst[arrayIndex++] = obj.sleepy.degree;

 rst[arrayIndex++] = obj.sleepy.start;

 rst[arrayIndex++] = obj.sleepy.end;

 }

 }

 var bleThres= [0,1,2,3,4,5,6,7];

 index = bleThres.indexOf(obj.thres);

 if(-1 != index){

 rst[arrayIndex++] = 17;

 rst[arrayIndex++] = index;

 }

 var bleAck = [0,1];

 index = bleAck.indexOf(obj.bleAck);

 if(-1 == index){

 var bleAckStr= ["disable","enable"];

 index = bleAckStr.indexOf(obj.bleAck);

 if(-1 != index){

 rst[arrayIndex++] = 18;

 rst[arrayIndex++] = index;

 }

 }

 else{

 rst[arrayIndex++] = 18;

 rst[arrayIndex++] = index;

 }

 return rst;

 }

 break;

 case 11:

 {

 rst[arrayIndex++] = obj.msgId;

 rst[arrayIndex++] = obj.time >> 8;

 rst[arrayIndex++] = obj.time & 0xFF;

 return rst;

 }

 break;

 //command

 case 12:

 {

 rst[arrayIndex++] = obj.msgId;

 if(obj.cmd >= 0 && obj.cmd <= 9){

 rst[arrayIndex] = obj.cmd;

 return rst;

 }

 }

 break;

 //ack

 case 13:

 {

 rst[arrayIndex] = obj.msgId;

 return rst;

 }

 break;

 //Positioning beacon UUID setting

 case 14:

 //Asset beacon UUID setting

 case 15:

 {

 var uuidNum = obj.uuidList.length;

 rst[arrayIndex++] = obj.msgId;

 rst[arrayIndex++] = uuidNum;

 if(uuidNum >= 1 && uuidNum <=5){

 for(var i=0; i<uuidNum; i++){

 var uuidIndex = obj.uuidList[i].index;

 if(uuidIndex >= 0 && uuidIndex <= 4){

 var uuid = obj.uuidList[i].uuid;

 if(uuid.length == 32){

 rst[arrayIndex++] = uuidIndex;

 var pos=0;

 for(var j=0; j<16; j++)

 {

 var s = uuid.substr(pos, 2);

 var v = parseInt(s, 16);

 rst[arrayIndex++] = v;

 pos += 2;

 }

 }

 else{

 return [];

 }

 }

 }

 return rst;

 }

 }

 break;

 case 16:

 {

 rst[arrayIndex++] = obj.msgId;

 var filterPort = obj.port;

 if(filterPort >= 21 && filterPort <= 25){

 var filterLen = obj.filterLen;

 var filter = obj.filter;

 if(filter.length == (filterLen<<1)){

 rst[arrayIndex++] = filterPort;

 rst[arrayIndex++] = obj.start;

 rst[arrayIndex++] = obj.end;

 rst[arrayIndex++] = obj.filterStart;

 rst[arrayIndex++] = obj.filterLen;

 var filterPos=0;

 for(var j1=0; j1<filterLen ; j1++)

 {

 var s1 = filter.substr(filterPos, 2);

 var v1 = parseInt(s1, 16);

 rst[arrayIndex++] = v1;

 filterPos += 2;

 }

 }

 return rst;

 }

 }

 break;

 case 17:

 {

 var beaconNum = obj.number;

 rst[arrayIndex++] = obj.msgId;

 rst[arrayIndex++] = beaconNum;

 if(beaconNum >= 1 && beaconNum <=20){

 for(var beaconI=0; beaconI<beaconNum; beaconI++){

 var beaconIndex = obj.beaconList[beaconI].index;

 if(beaconIndex >= 0 && beaconIndex <= 19){

 var beaconMajor= obj.beaconList[beaconI].major;

 var beaconMinor= obj.beaconList[beaconI].minor;

 rst[arrayIndex++] = beaconIndex;

 rst[arrayIndex++] = beaconMajor >> 8;

 rst[arrayIndex++] = beaconMajor & 0xff;

 rst[arrayIndex++] = beaconMinor >> 8;

 rst[arrayIndex++] = beaconMinor & 0xff;

 }

 }

 return rst;

 }

 }

 break;

 default:

 }

 return [];

}