

```

// Decode decodes an array of bytes into an object.
// - fPort contains the LoRaWAN fPort number
// - bytes is an array of bytes, e.g. [225, 230, 255, 0]
// The function must return an object, e.g. {"temperature": 22.5}
function Decode(fPort, bytes) {
    var decoded = {};
    decoded.cmd = bytes[0];
    if (decoded.cmd === 16 || decoded.cmd === 32 || decoded.cmd === 48)
    {
        decoded.hours = bytes[1];
        decoded.minutes = bytes[2];

        var index = 4;
        var pair = true;
        var header = bytes[3];

        if (header & 0x80)
        {
            decoded.pio0Event = bytes[index];
            index += 1;
        }
        if (header & 0x40)
        {
            decoded.pio1Event = bytes[index];
            index += 1;
        }
        if (header & 0x20)
        {
            if (pair )
            {
                decoded.pio2Event = (bytes[index]*16 + (bytes[index+1] &
0xF0)/16);
                index += 1;
                pair = false;
            } else {
                decoded.pio2Event = (bytes[index+1] + (bytes[index] &
0x0F)*256);
                index += 2;
                pair = true;
            }
        }
        if (header & 0x10)
        {

```

```

        if (pair )
        {
            decoded.pio3Event = (bytes[index]*16 + (bytes[index+1] &
0xF0)/16);
            index += 1;
            pair = false;
        } else {
            decoded.pio3Event = (bytes[index+1] + (bytes[index] &
0x0F)*256);
            index += 2;
            pair = true;
        }
    }
    if (header & 0x08)
    {
        if (pair )
        {
            decoded.intTemp = ((bytes[index]*16 + (bytes[index+1] &
0xF0)/16)/16);
            index += 1;
            pair = false;
        } else {
            decoded.intTemp = ((bytes[index+1] + (bytes[index] &
0x0F)*256)/16);
            index += 2;
            pair = true;
        }
    }
    if (header & 0x04)
    {
        if (pair )
        {
            decoded.extTemp = (bytes[index]*16 + (bytes[index+1] &
0xF0)/16);
            index += 1;
            pair = false;
        } else {
            decoded.extTemp = (bytes[index+1] + (bytes[index] & 0x0F)*256);
            index += 2;
            pair = true;
        }
    }
    if (header & 0x02)

```

```

    {
        if (pair )
        {
            decoded.Vbat = (bytes[index]*16 + (bytes[index+1] & 0xF0)/16);
            index += 1;
            pair = false;
        } else {
            decoded.Vbat = (bytes[index+1] + (bytes[index] & 0x0F)*256);
            index += 2;
            pair = true;
        }
    }
    if (header & 0x01)
    {
        if (pair )
        {
            decoded.Vext = (bytes[index]*16 + (bytes[index+1] & 0xF0)/16);
            index += 1;
            pair = false;
        } else {
            decoded.Vext = (bytes[index+1] + (bytes[index] & 0x0F)*256);
            index += 2;
            pair = true;
        }
    }
}
if (decoded.cmd === 00)
{
    decoded.config_exist = bytes[1];
    decoded.hours = bytes[2];
    decoded.minutes = bytes[3];
    decoded.guardTimeDay = bytes[4];
    decoded.guardTimeNight = bytes[5];
    decoded.KeepAliveValue = bytes[6];
    decoded.pio0EventThresholdDay = bytes[7];
    decoded.pio0EventThresholdNight = bytes[8];
    decoded.pio1EventThresholdDay = bytes[9];
    decoded.pio1EventThresholdNight = bytes[10];
    decoded.pio2EventThresholdDay = (bytes[11]*16 + (bytes[12] & 0xF0)/16);
    decoded.pio2EventThresholdNight = (bytes[13] + (bytes[12] & 0x0F)*256);
    decoded.pio3EventThresholdDay = (bytes[14]*16 + (bytes[15] & 0xF0)/16);
    decoded.pio3EventThresholdNight = (bytes[16] + (bytes[15] & 0x0F)*256);
}

```

```
        decoded.nightStartHours = bytes[17];
        decoded.nightStartMinutes = bytes[18];
        decoded.dayStartHours = bytes[19];
        decoded.dayStartMinutes = bytes[20];
        decoded.en_Register = bytes[21];
    }
    return decoded;
}
```