

APOLLON Q QUICK START GUIDE EN

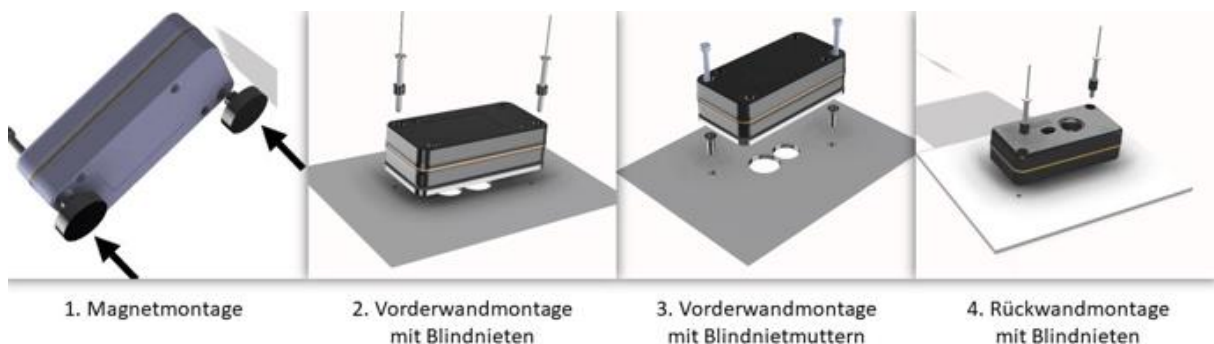


The Apollon-Q series comprises wireless IoT level sensors with precise measurement results thanks to a unique measuring principle combining optical measurement and radar. The sensors detect fill levels up to 2.50 meters - suitable for piece, liquid or bulk goods in containers, shafts, ducts or in the smart waste sector. They provide reliable data even in small containers such as waste paper baskets. They offer a long range, short measuring and transmission intervals and extended functions. Communication standards such as MIOTY®, NB-IoT, LoRaWAN® and LTE-CAT-M1 are supported.

Please observe the warnings and instructions in the operating instructions to ensure the safety of you, your surroundings and the sensor!

MOUNTING AND INSTALLATION

Install the sensor on an interior wall at a height of between 1.50m and 1.80m. Make sure that the LEDs are on the bottom right-hand side. Ensure that the sensor is at least 20 cm away from people and is located in an area below 2000 m above sea level. Use either magnets, rivets or suitable screws for mounting. The detailed steps as well as drilling templates can be found in the operating instructions.



COMISSIONING

There are two Hall sensors (magnetic field switches) on the sensor. The following diagram shows the position of the Hall sensors and the recommended placement of the magnets.



The upper, central magnetic field switch (1) can be operated in three different modes:

- container is closed when the magnet is applied
- the container is open when the magnet is applied
- the sensor counts an opening when the magnet passes through twice

For newer sensors, the opening detection function is taken over by the accelerometer.

COMMISSIONING THE SENSOR VIA BLE

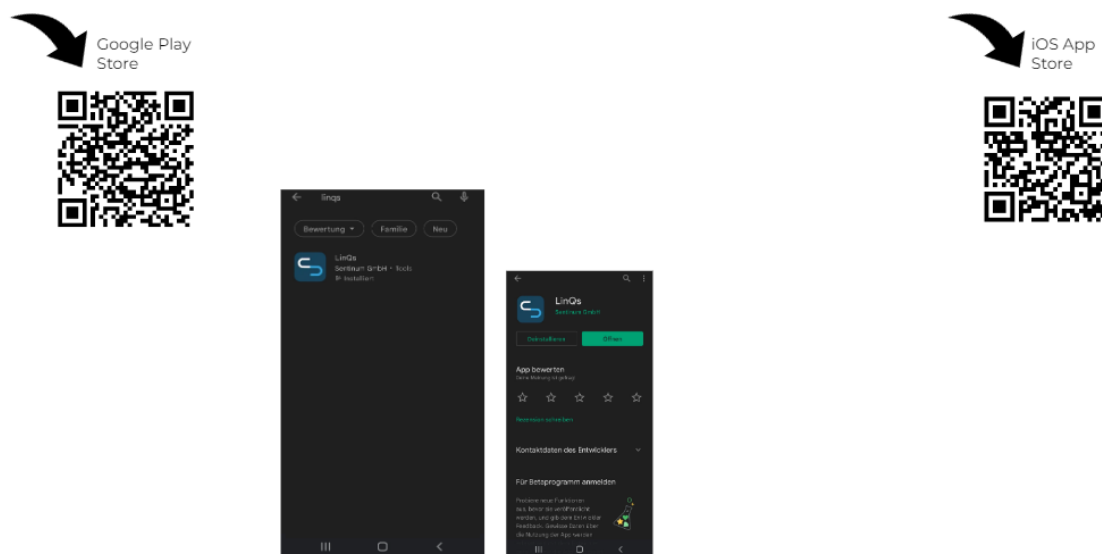
BLE is not available for sensors with ToF and without radar. For these versions, please follow the steps in section via NFC to commission and configure them.

For all other sensors, you can use the convenient variant via BLE as follows:

1. all other variants can be activated via BLE with your cell phone. Simply use our LinQs APP, which you can find in the [APP Store](#) and [Google Play Store](#).
2. activate the sensor by briefly triggering the lower left Hall sensor (2) with a magnet for at least 2 seconds. The sensor should now emit an acoustic tone sequence. Advertising mode is then started automatically.
3. you can now conveniently control the sensor via BLE from your end device. Make sure that it is switched on on your cell phone and that you are within range of the sensor so that you can reach and control it.

NFC COMMISSIONING, PARAMETERIZATION AND LOCATION OF THE NFC TAG

Activation can be carried out via an NFC app. A smartphone is required for this. The app can be downloaded from the [APP Store](#) and [Google Play Store](#). Simply search for "Sentinum LinQs" and download the LinQs app.



First locate the tag on the sensor and then the reader on your end device. You will find the location of the NFC tag at the position of the orange arrow.



Open the app and activate the sensor. To start the sensor in the basic settings, click on the "Activate sensor" button in the app's start menu. Now place your device on the NFC marker of the sensor.



When the sensor is activated, "Sensor updated!" is displayed. You can then continue with the activation of the other sensors.

PAYLOAD DECODER APOLLON Q SERIE

```
// Up to Date Apollon decoder
function Decoder(bytes, port) {

    // Conversion of signed integers
    function uncomplement(val, bitwidth) {
        var isnegative = val & (1 << (bitwidth - 1));
        var boundary   = (1 << bitwidth);
        var minval      = -boundary;
        var mask        = boundary - 1;
        return isnegative ? minval + (val & mask) : val;
    }

    var decoded = {};

    if (port === 1) {

        // Attributes
        decoded.base_id          = bytes[0] >> 4;
        decoded.major_version    = bytes[0] & 0x0F;
        decoded.minor_version     = bytes[1] >> 4;
        decoded.product_version  = bytes[1] & 0x0F;

        // Telemetry
        decoded.up_cnt            = bytes[2];
        decoded.battery_voltage   = ((bytes[3] << 8) | bytes[4]) /
1000.0;
        decoded.internal_temperature = bytes[5] - 128;
        decoded.alarm              = bytes[6] ? "ALARM" : "NO ALARM";
        decoded.master_value       = ((bytes[7] << 8) | bytes[8]); // in
mm

        // Start of version specific fields
        var byte_cnt = 9;

        // ToF onboard
        if (bytes[1] & 0x01) {
            decoded.tof_status    = bytes[byte_cnt++];
            decoded.tof_distance  = ((bytes[byte_cnt++] << 8) |
bytes[byte_cnt++]); // in mm
            decoded.tof_index     = bytes[byte_cnt++];
        }

        // Radar onboard
        if (bytes[1] & 0x02) {
            decoded.radar_status    = bytes[byte_cnt++];
            decoded.radar_no_peaks   = bytes[byte_cnt++];
            decoded.radar_distance_1 = ((bytes[byte_cnt++] << 8) |
bytes[byte_cnt++]);
            decoded.radar_amplitude_1 = uncomplement((bytes[byte_cnt++] <<
8) | bytes[byte_cnt++], 16);
            decoded.radar_distance_2 = ((bytes[byte_cnt++] << 8) |
bytes[byte_cnt++]);
            decoded.radar_amplitude_2 = uncomplement((bytes[byte_cnt++] <<
8) | bytes[byte_cnt++], 16);
        }
    }
}
```

```
        decoded.radar_distance_3 = ((bytes[byte_cnt++] << 8) |
bytes[byte_cnt++]);
        decoded.radar_amplitude_3 = uncomplement((bytes[byte_cnt++] <<
8) | bytes[byte_cnt++], 16);
    }

    // ACC onboard
    if (bytes[1] & 0x04) {
        decoded.acc_status      = bytes[byte_cnt++] ? "Fehler" : "OK";
        decoded.acc_orientation = bytes[byte_cnt++];
        decoded.acc_open        = bytes[byte_cnt++];
        decoded.acc_open_cnt     = ((bytes[byte_cnt++] << 8) |
bytes[byte_cnt++]);
        decoded.acc_impact       = bytes[byte_cnt++] ? "Vandalismus" :
"OK";
    }

    // Hall onboard
    if (bytes[1] & 0x08) {
        decoded.hall_open        = bytes[byte_cnt++];
        decoded.hall_open_cnt    = ((bytes[byte_cnt++] << 8) |
bytes[byte_cnt++]);
    }
}
return decoded;
}
```