# iNode Manufacturer Specific Data

# Uwagi

- bity w dokumencie liczone są od `1`, a nie od `0`.
- `rtto` to flaga wskazująca czy upłynęły 24h od ostatniego odczytu pamięci danych.
- pierwsze dwa bajty to wg specyfikacji *Company Identifier*. Pierwszy bajt zawiera różne flagi, a drugi to model urządzenia.
- *uint16le* to *unsigned 16-bit integer (little endian)* itp.
-

# iNode Beacon (0x80)

```
92 80 ??
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `80` | iNode Beacon |
| `??` | ? |

## iNode Energy Meter (0x82)

```
92 82 00 00 71 00 00 00 e8 03 00 00 00
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `82` | iNode Energy Meter |
| `00 00` | `rawAvg` (uint16le) |
| `71 00 00 00` | `rawSum` (uint32le) |
| `e8 03` | `options` (uint16le):<br>● bit 1-14: stała licznika<br>● bit 15-16: jednostka |
| `00` | `batteryAndLight` |
| `00 00` | `weekDayData` (uint32le) |

### Obliczanie średniej z poprzedniej minuty (`avg`) i sumy (`sum`):

```
unit = (options >> 14) & 3
constant = options & 0x3FFF
```

Jeżeli `unit == 0`, to:

```
    avgUnit = kWh
    sumUnit = kW
    constant = 100  (jeżeli constant == 0)
```

Jeżeli `unit == 1`, to:

```
    avgUnit = m^3
    sumUnit = m^3
    constant = 1000  (jeżeli constant == 0)
```

Jeżeli `unit != 0 && unit != 1`, to:

```
    avgUnit = cnt
    sumUnit = cnt
    constant = 1  (jeżeli constant == 0)
```

```
avg = 60 * rawAvg / constant
sum = rawSum / constant
```

## Obliczanie poziomu (`batteryLevel [%]`) i napięcia baterii (`batteryVoltage [V]`):

```
battery = (batteryAndLight >> 4) & 0x0F
if battery == 1
     batteryLevel = 100
else
     batteryLevel = 10 * (min(battery, 11) - 1)
batteryVoltage = (batteryLevel - 10) * 1.2 / 100 + 1.8
```

## Obliczanie poziomu światła (`lightLevel [%]`):

```
lightLevel = (batteryAndLight & 0x0F) * 100 / 15
```

## Obliczanie poprzedniego dnia tygodnia (`weekDay`) oraz sumy z poprzedniego dnia (`weekDayTotal`):

```
weekDay = weekDayData >> 13
weekDayTotal = weekDayData & 0x1FFF
```

# iNode Control ID (0x88)

```
92 88 ??
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `88` | iNode Control ID |
| `??` | ? |

# iNode Nav (0x89)

```
92 89 ??
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `89` | iNode Nav |
| `??` | ? |

# iNode Care Sensor #1 (0x91)

```
92 91 01 b0 00 00 17 00 a8 19 00 00 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `91` | iNode Care Sensor #1 |
| `01 b0` | `groupsAndBattery` (uint16le):<br>● bit 1-12 - każdy bit to oddzielna grupa (A-L).<br>● bit 13-16 - napięcie baterii |
| `00 00` | Alarmy (uint16le):<br>● bit 1: MOVE_ACCELEROMETER,<br>● bit 2: LEVEL_ACCELEROMETER,<br>● bit 3: LEVEL_TEMPERATURE,<br>● bit 4: LEVEL_HUMIDITY,<br>● bit 5: CONTACT_CHANGE,<br>● bit 6: MOVE_STOPPED,<br>● bit 7: MOVE_GTIMER,<br>● bit 8: LEVEL_ACCELEROMETER_CHANGE,<br>● bit 9: LEVEL_MAGNET_CHANGE,<br>● bit 10: LEVEL_MAGNET_TIMER |
| `17 00` | `rawPosition` (uint16le) |
| `a8 19` | `rawTemperature` (uint16le) |
| `00 00` | Nie używane |
| `04 00` | `rawTime1` (uint16le) |
| `f4 bb` | `rawTime2` (uint16le) |
| `ce 6e 77 a0`<br>`0b 97 d1 b5` | Cyfrowy podpis AES128 dla powyższych danych |

## Obliczanie grup (`groups`):

```
groups = groupsAndBattery & 0x0FFF
```

### Obliczanie poziomu (**batteryLevel [%]**) i napięcia baterii (**batteryVoltage [V]**):

```
battery = (groupsAndBattery >> 12) & 0x0F
if battery == 1
     batteryLevel = 100
else
     batteryLevel = 10 * (min(battery, 11) - 1)
batteryVoltage = (batteryLevel - 10) * 1.2 / 100 + 1.8
```

### Obliczanie pozycji:

```
inMotion = rawPosition & 0x8000
x = (rawPosition >> 10) & 0x1F
y = (rawPosition >> 5) & 0x1F
z = rawPosition & 0x1F
x = x - (x & 0x10 ? 0x1F : 0)
y = y - (y & 0x10 ? 0x1F : 0)
z = z - (z & 0x10 ? 0x1F : 0)
```

### Obliczanie temperatury (**T [°C]**):

```
T = rawTemperature
if T > 127
     T = T - 8192
if T < -30
     T = -30
if T > 70
     T = 70
```

### Obliczanie czasu (**t [s]**):

```
t1 = rawTime1 << 16 // 04 00 00 00
t2 = rawTime2       // bb f4
t = t1 | t2         // 04 00 00 00 bb f4
```

## iNode Care Sensor #2 (0x92)

```
92 92 01 b0 00 00 17 00 a8 19 00 00 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| 92 | bit 2: rtto<br>bit 3: lowBattery |
| 92 | iNode Care Sensor #2 |
| 01 b0 | groupsAndBattery (uint16le); jak w iNode Care Sensor #1 |
| 00 00 | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| 17 00 | rawPosition (uint16le); jak w iNode Care Sensor #1 |
| a8 19 | rawTemperature (uint16le) |
| 00 00 | Nie używane |
| 04 00 | rawTime1 (uint16le); jak w iNode Care Sensor #1 |
| f4 bb | rawTime2 (uint16le) |
| ce 6e 77 a0<br>0b 97 d1 b5 | Cyfrowy podpis AES128 dla powyższych danych |

### Obliczanie temperatury (T [°C]):

```
MSB = rawTemperature[0] // a8
LSB = rawTemperature[1] // 19
T = MSB * 0.0625 + 16 * (LSB & 0x0F)
if LSB & 0x10
     T = T - 256
if T < -30
     T = -30
if T > 70
     T = 70
```

# iNode Care Sensor #3 (0x93)

```
92 93 01 b0 00 00 17 00 a8 19 e8 18 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| 92 | bit 2: `rtto`<br>bit 3: `lowBattery` |
| 93 | iNode Care Sensor #3 |
| 01 b0 | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| 00 00 | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| 17 00 | `rawPosition` (uint16le); jak w iNode Care Sensor #1 |
| a8 19 | `rawTemperature` (uint16le) |
| e8 18 | `rawHumidity` (uint16le) |
| 04 00 | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| f4 bb | `rawTime2` (uint16le) |
| ce 6e 77 a0<br>0b 97 d1 b5 | Cyfrowy podpis AES128 dla powyższych danych |

## Obliczanie temperatury (`T [°C]`):

```
T = (175.72 * rawTemperature * 4 / 65536) - 46.85
if T < -30
     T = -30
if T > 70
     T = 70
```

## Obliczanie wilgotności (`H [%]`):

```
H = (125 * rawHumidity * 4 / 65536) - 6
if H < 1
     H = 1
if H > 100
     H = 100
```

# iNode Care Sensor #4 (0x94)

```
92 94 01 b0 00 00 17 00 a8 19 00 00 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery`<br>bit 4: `input` |
| `94` | iNode Care Sensor #4 |
| `01 b0` | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| `17 00` | `rawPosition` (uint16le); jak w iNode Care Sensor #1 |
| `a8 19` | `rawTemperature` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Nie używane |
| `04 00` | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| `f4 bb` | `rawTime2` (uint16le) |
| `ce 6e 77 a0 0b 97 d1 b5` | Cyfrowy podpis AES128 dla powyższych danych |

## iNode Care Sensor #5 (0x95)

```
92 95 01 b0 00 00 17 00 a8 19 00 00 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery`<br>bit 4: `magneticFieldDirection` |
| `95` | iNode Care Sensor #5 |
| `01 b0` | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| `17 00` | `rawPosition` (uint16le); jak w iNode Care Sensor #1 |
| `a8 19` | `rawTemperature` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | `magneticField` (uint16le) |
| `04 00` | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| `f4 bb` | `rawTime2` (uint16le) |
| `ce 6e 77 a0 0b 97 d1 b5` | Cyfrowy podpis AES128 dla powyższych danych |

# iNode Care Sensor #6 (0x96)

```
92 96 01 b0 00 00 17 00 a8 19 00 00 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| `92` | bit 1: `output`<br>bit 2: `rtto`<br>bit 3: `lowBattery`<br>bit 4: `input` |
| `96` | iNode Care Sensor #6 |
| `01 b0` | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| `17 00` | `rawPosition` (uint16le); jak w iNode Care Sensor #1 |
| `a8 19` | `rawTemperature` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Nie używane |
| `04 00` | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| `f4 bb` | `rawTime2` (uint16le) |
| `ce 6e 77 a0 0b 97 d1 b5` | Cyfrowy podpis AES128 dla powyższych danych |

## iNode Care Sensor T (0x9A)

```
92 9A 01 b0 00 00 00 00 a8 19 00 00 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `9A` | iNode Care Sensor T |
| `01 b0` | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Nie używane |
| `a8 19` | `rawTemperature` (uint16le); jak w iNode Care Sensor #2 |
| `00 00` | Nie używane |
| `04 00` | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| `f4 bb` | `rawTime2` (uint16le) |
| `ce 6e 77 a0 0b 97 d1 b5` | Cyfrowy podpis AES128 dla powyższych danych |

# iNode Care Sensor HT (0x9B)

```
92 9B 01 b0 00 00 00 00 a8 19 00 00 04 00 f4 bb ce 6e 77 a0 0b 97
d1 b5
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `9B` | iNode Care Sensor HT |
| `01 b0` | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| `00 00` | Nie używane |
| `a8 19` | `rawTemperature` (uint16le); jak w iNode Care Sensor #3 |
| `00 00` | `rawHumidity` (uint16le); jak w iNode Care Sensor #3 |
| `04 00` | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| `f4 bb` | `rawTime2` (uint16le) |
| `ce 6e 77 a0 0b 97 d1 b5` | Cyfrowy podpis AES128 dla powyższych danych |

## iNode Care Sensor PT (0x9C)

```
12 9C 00 B0 00 00 4E 3E 40 D8 00 00 00 00 AC 53 54 65 11 26 68 75
16 3A
```

| | |
|---|---|
| 12 | bit 2: `rtto`<br>bit 3: `lowBattery` |
| 9C | iNode Care Sensor PT |
| 00 b0 | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| 00 00 | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| 4e 3e | `rawPressure` (uint16le) |
| 40 d8 | `rawTemperature` (int16le) |
| 00 00 | - |
| 00 00 | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| ac 53 | `rawTime2` (uint16le) |
| 54 65 11 26<br>68 75 16 3a | Cyfrowy podpis AES128 dla powyższych danych |

### Obliczanie ciśnienia (`P [mbar]`):

```
P = rawPressure / 16
```

### Obliczanie temperatury (`T [°C]`):

```
T = 42.5 + rawTemperature / 480
```

# iNode Care Sensor PHT (0x9D)

```
12 9D 01 C0 00 00 4F 3E 3F 19 95 12 03 00 3C C0 91 99 BB A2 CC 23
AC 82
```

| | |
|---|---|
| 92 | bit 2: `rtto`<br>bit 3: `lowBattery` |
| 9D | iNode Care Sensor PHT |
| 01 c0 | `groupsAndBattery` (uint16le); jak w iNode Care Sensor #1 |
| 00 00 | Alarmy (uint16le); jak w iNode Care Sensor #1 |
| 4f 3e | `rawPressure` (uint16le); jak w iNode Care Sensor PT |
| 3f 19 | `rawTemperature` (uint16le); jak w iNode Care Sensor #3 |
| 95 12 | `rawHumidity` (uint16le); jak w iNode Care Sensor #3 |
| 03 00 | `rawTime1` (uint16le); jak w iNode Care Sensor #1 |
| 3c c0 | `rawTime2` (uint16le) |
| 91 99 bb a2 cc 23 ac 82 | Cyfrowy podpis AES128 dla powyższych danych |

# iNode Control Point (0xB2)

```
92 B2 ??
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `B2` | iNode Control Point |
| `??` | ? |

# iNode Control Relay (0xB3)

```
92 B3 ??
```

| | |
|---|---|
| `92` | bit 1: `output`<br>bit 2: `rtto`<br>bit 3: `lowBattery` |
| `B3` | iNode Care Relay |
| `??` | ? |

# iNode Transceiver UART (0xB5)

```
92 B2 ??
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `B5` | iNode Transceiver UART |
| `??` | ? |

# iNode Transceiver USB (0xB6)

```
92 B6 ??
```

| | |
|---|---|
| `92` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `B6` | iNode Transceiver USB |
| `??` | ? |

# iNode GSM (0xB7)

```
10 B7
```

| | |
|---|---|
| `10` | bit 2: `rtto`<br>bit 3: `lowBattery` |
| `B7` | iNode GSM |