



# **AN0517 - *swarm* bee UART communication**

1.0

NA-17-0356-0004

**Application Note**

AN0517 - swarm bee UART communication

Version: 1.0 Author: nanotron

**Document Information**

Document Title:	AN0517 - swarm bee UART communication
Document Version:	1.0
Current Date:	2017-03-20
Print Date:	2017-03-20
Document ID:	NA-17-0356-0004
Document Author:	nanotron

**Disclaimer**

Nanotron Technologies GmbH believes the information contained herein is correct and accurate at the time of release. Nanotron Technologies GmbH reserves the right to make changes without further notice to the product to improve reliability, function or design. Nanotron Technologies GmbH does not assume any liability or responsibility arising out of this product, as well as any application or circuits described herein, neither does it convey any license under its patent rights.

As far as possible, significant changes to product specifications and functionality will be provided in product specific Errata sheets, or in new versions of this document. Customers are encouraged to check the Nanotron website for the most recent updates on products.

**Trademarks**

All trademarks, registered trademarks, and product names are the sole property of their respective owners.

This document and the information contained herein is the subject of copyright and intellectual property rights under international convention. All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical or optical, in whole or in part, without the prior written permission of nanotron Technologies GmbH.

Copyright © 2017 nanotron Technologies GmbH.

## Contents

1. Introduction .....	4
2. What is the UART? .....	4
3. Setting the UART parameters .....	4
3.1. Changing the UART format .....	4
3.2. Changing the UART data rate .....	4
4. Effect of the UART parameters .....	5
4.1. Effect of the data format .....	5
4.2. Effect of the data rate .....	5
5. References.....	6
<b>Table 4-1: Transmission time for different message types .....</b>	<b>5</b>

## 1. Introduction

The *swarm* bee devices can communicate and interact via the AIR interface and also through their UART port. This application note addresses how to configure that port to adapt the communication through the UART to the application requirements.

The document covers *swarm* bee LE v1, *swarm* bee LE v2 and *swarm* bee ER.

To have more information about the format of sent/received messages over the UART, please check [1]

## 2. What is the UART?

The Universal Asynchronous Receiver/Transmitter (UART) is a hardware device for asynchronous serial communication in which the data format and transmission speeds are configurable. The UART settings are one start bit, 8 data bits, no parity bit and one stop bit (8, N, 1) and the idle, no data state is high-voltage, or powered. This means that each byte is sent as a logic low start bit, 8 data bits and 1 logic high stop bit. Since the start bit is logic low (0) and the stop bit is logic high (1) there are always at least two guaranteed signal changes between characters.

If the line is held in the logic low condition for longer than a character time, this is a break condition that can be detected by the UART.

For more information check [2] and [3].

## 3. Setting the UART parameters

The UART data rate and interface (ASCII or binary) can be configured by the user, but not saved; the API command SSET, used to save the settings in the EEPROM, has no effect on this parameter. This means that every time that the device is powered up, or reset, both parameters will be set to their default values: 115200 bps and ASCII interface.

### 3.1. Changing the UART format

On initialization default communication is done in ASCII format. To change from ASCII to binary format the following API command should be used:

**SBIN <void>**

This sets the communication format to binary. Once the device is using the binary format the only way to change back to ASCII is to reset the module. The SSET command has no effect on this setting; thus the device will automatically change the format to ASCII if a reset is performed. The reset can be done by pulling the /NRST pin low.

### 3.2. Changing the UART data rate

The UART data rate can only be changed over the UART itself; it is not possible to do this over the air. When a *swarm* device is powered up the UART data rate is set at default, 115200 bps. To change the speed the following API command is required:

**SUAS <Speed>**          where <Speed>    Range: 500 ... 2000000 bps  
    ASCII format: 7 byte (decimal)  
    Binary format: 4 byte (uint32\_t)

The set of possible data rates (speed) is a discrete number of values. If the data rate entered by the user is not supported, the closest possible value will be configured automatically. The error will be <1% of the desired value. For example an 115200 bps entered by the user will translate to 115108 bps configured in the device. This is a 0.08% of error.

Again, resetting the device will configure the default data rate back to 115200 bps, on the UART.

## 4. Effect of the UART parameters

### 4.1. Effect of the data format

All API commands consist of the instruction itself followed by the required parameters.

When working in ASCII format that instruction is always 4 characters followed by a space (5 characters). The parameters are decimal or hexadecimal numbers separated by spaces. Each digit, decimal or hexadecimal, is an ASCII character. Each character of the command will be translated into 8 bits (1 byte).

When the binary format is used, each instruction will consist of only 2 bytes, each of them expressed in hexadecimal. The first byte indicates the command type and the second one the instruction code itself. The parameters will always be represented as hexadecimal numbers and no white spaces will be required in between them. However, a 2-byte synchronization code and a 2-byte CRC are required in each message, at the beginning and at the end respectively.

The following example shows the difference in message size when sending an API command according to the ASCII or binary interface, for instance the command to range with other devices: immediate RATO (RATO with option 0). For all commands in the API, the *swarm bee* will confirm its correct reception with a response. In this case the response will also include the estimated distance between the devices. When working in ASCII format the response will start by a '='.

Command: RATO <Option=0> <ID>

Response: <Option=0>: =<ErrorCode>,<RangingResult>,<RSSI>

#### ASCII

Command: RATO\_0\_0000656EB681            19 bytes  
Response: =0, 001843,-56                14 bytes

#### Binary:

Command: 7F0955120000000000135C8FB        13 bytes  
Response: 7F0857120000000045CB5007        12 bytes

Sending the same command (RATO) will require a 19-byte message when using ASCII format and a 13-byte message with binary format.

The process is similar when receiving notifications. In binary format a notification consist of the 5 characters indicating the notification followed by the parameters separated by, in this case, commas. All this is represented with ASCII characters. In binary format only hexadecimal codes are used, which reduces the length of the message.

The next example shows this difference in the case of a typical Range Result Notification (\*RRN) in which no sensor data passed, only the RSSI value:

ASCII: \*RRN:1F3123123133,1F3CFF322133,0,001843,04,-56  
Each ASCII character is 1 byte → 46 bytes

Binary: 7F1361621F31231231331F3CFF3221330000073004C8D870  
Each hexadecimal character is 4bits → 24 bytes

Again using the binary format significantly reduces the message size.

### 4.2. Effect of the data rate

The UART data rate is configurable and how it is set will determine the transmission time required for the exchange of messages. It can be set from 500 bps up to 2 Mbps. The table below shows the maximum, default and minimum transmission time for sending and receiving an API command or notification. The same command and notification as in previous sections have been used. As it was shown that the message format has an impact on its size, this factor has also been included in the table.

**Table 4-1:** Transmission time for different message types

Message type	Format	Length (bytes)	Transmission time (ms)		
			@ 500bps	@ 115200bps	@ 2Mbps
RATO	ASCII	19	304	1.32	0.08
	Binary	13	208	0.90	0.05
Response to RATO	ASCII	13	208	0.90	0.05
	Binary	12	192	0.83	0.05
*RRN	ASCII	46	736	3.20	0.18
	Binary	24	384	1.67	0.10

## Application Note

AN0517 - swarm bee UART communication

Version: 1.0 Author: nanotron

---

## 5. References

- [1] *swarm* API – User Guide V3.0, nanotron Technologies 2016
- [2] *swarm* bee LE - Data Sheet V1.6, nanotron Technologies 2016
- [3] *swarm* bee ER - Data Sheet V0.8, nanotron Technologies 2016

**Document History**

Date	Author	Version	Description
3/1/2017	MLA	1.0	Initial version

### **Life Support Policy**

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nanotron Technologies GmbH customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify nanotron Technologies GmbH for any damages resulting from such improper use or sale.

### **About Nanotron Technologies GmbH**

Today nanotron's *embedded location platform* delivers location-awareness for safety and productivity solutions across industrial and consumer markets. The platform consists of chips, modules and software that enable precise real-time positioning and concurrent wireless communication. The ubiquitous proliferation of interoperable location platforms is creating the location-aware Internet of Things.

### **Further Information**

For more information about products from nanotron Technologies GmbH, contact a sales representative at the following address:

nanotron Technologies GmbH  
Alt-Moabit 60  
10555 Berlin, Germany  
Phone: +49 30 399 954 – 0  
Fax: +49 30 399 954 – 188  
Email: [sales@nanotron.com](mailto:sales@nanotron.com)  
Internet: [www.nanotron.com](http://www.nanotron.com)