

Titania (AMB3626) UART

??????????

```

“ [ ] Würth Titania 2607011111100x (AMB3626)
  [ ] 9600 8N1[ ] 1200115200 baud
  [ ] 169.4125169.4625 MHz (Band D, EN 300 220)
  [ ] Manual-um-titania-2607011111100x (rev4.5)
  
```

? ??????

[]	CMD Code	Request	Response	[]
[]	0x05	02 05 00 07	02 45 01 00 46	[]
[]	0x06	02 06 01 [CH] [CS]	02 46 01 [CH] [CS]	[] []
[] UserSetting []	0x09	02 09 [LEN+2] [MEM] [BYTES] [DATA...] [CS]	02 49 [LEN] [STATUS] [CS]	[] flash [] baudrate [] NetID []
[] UserSetting	0x0B	02 0B 01 [MEM] [CS]	02 4B [LEN] [STATUS] [DATA...] [CS]	[] flash []
[]	0x0D	02 0D 00 4D	02 4C 00 4E	[] UserSettings [] UART [] 9600 []
[] MAC []	0x80	02 80 [LEN] [MAC] [DATA...] [CS]	02 C0 [LEN] [STATUS] [CS]	[] []
Command Mode []	—	—	—	[] /CONFIG []

? ??????

1?? ??????CMD_RESET_REQ?

Checksum $0x02 \wedge 0x06 \wedge 0x01 \wedge [\text{Channel}]$

- Channel 2

02 06 01 02 01

02 46 01 02 43

Checksum

$0x02 \wedge 0x06 \wedge 0x01 \wedge 0x02 = 0x01 \checkmark$

$0x02 \wedge 0x46 \wedge 0x01 \wedge 0x02 = 0x43 \checkmark$

0000

02 Start

46 CMD_SET_CHANNEL_CNF

01 Length

[02] 0000 (0~4)

[43] Checksum

C 0000

```
uint8_t payload[1] = { channel };
Titania_SendCommand(huart, 0x06, payload, 1);
uint8_t resp[5];
HAL_UART_Receive(huart, resp, 5, 100);
```

3?? ?? UserSettings?????— CMD_SET_REQ ? ??

flash 00000000 UART baudrate000000 NetID Addr 00000000

UserSettings 0000

000	Memory Position (Dec / Hex)	0	000	000	00
PHY_DefaultChannel	42 / 0x2A	1 byte	0~4	2	00000000
UART_Baudrate	80 / 0x50	4 bytes	1200~115200	9600	UART 00000000 LSB first

名前	Memory Position (Dec / Hex)	サイズ	初期値	範囲	説明
MAC_DefaultDestNetID	TODO	2 bytes	0x0000~0xFFFF	0x0000	NetID
MAC_DefaultDestAddrLSB	TODO	2 bytes	0x0000~0xFFFF	0x0000	LSB
UART_Timeout	TODO	2 bytes	[]	[]	UART [] []
PHY_PAPower	TODO	1 byte	[]	[]	[] (-11~+15 dBm)
RF_ConfigIndex	TODO	1 byte	[]	2	RF [] data rate []
MAC_NumRetrys	TODO	1 byte	0~N	[]	[]

[]

Start	CMD	LEN	MPOS	NBYTES	DATA...	Checksum
02	09	[2+N]	[MP]	[LEN]	[...]	[CS]

- **LEN** = 2 + [] [] MPOS + NBYTES []
- **MPOS** = Memory Position []
- **NBYTES** = [] bytes
- **DATA** = []
- **Checksum** = [0x02] [] DATA [XOR

[] 1 - [] Channel 3 []

```

[ ]02 09 03 2A 01 03 4B
  02 09 03 2A 01 03 4B
  ↑ ↑ ↑ ↑ ↑ ↑ ↑
  ST CMD LEN MP NB DC CS

[ ]02 49 00 00 4B
  ↑ ↑ ↑ ↑ ↑
  ST CMD LEN ST CS

Checksum [ ]:
[ ]: 0x02 ^ 0x09 ^ 0x03 ^ 0x2A ^ 0x01 ^ 0x03 = 0x4B ✓
[ ]: 0x02 ^ 0x49 ^ 0x00 ^ 0x00 = 0x4B ✓

```

[] 2 - [] UART Baudrate [] 57600 []

57600 = 0x0000E100 (0000)
000 (LSB first) [0x00, 0xE1, 0x00, 0x00]

```
00002 09 06 50 04 00 E1 00 00 CS
    ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
    ST CMD LEN MP NB D0 D1 D2 D3 CS
```

MPOS = 0x50 (80 000)

NBYTES = 04

DATA = [00 E1 00 00]

LEN = 2 + 4 = 06

Checksum = 0x02 ^ 0x09 ^ 0x06 ^ 0x50 ^ 0x04 ^ 0x00 ^ 0xE1 ^ 0x00 ^ 0x00 = ?

```
00002 49 00 00 4B
    (Status=0x00 0000)
```

0000

02	Start
49	CMD_SET_CNF (00)
00	Length
00	Status (0x00 = 00, 0 0 = 00)
4B	Checksum

C 0000

```
// 00 UART Baudrate 57600
uint32_t baudrate = 57600;
uint8_t payload[6];
payload[0] = 0x50; // Memory Position (UART_Baudrate)
payload[1] = 0x04; // 0000 4 bytes
payload[2] = (uint8_t)(baudrate & 0xFF);
payload[3] = (uint8_t)((baudrate >> 8) & 0xFF);
payload[4] = (uint8_t)((baudrate >> 16) & 0xFF);
payload[5] = (uint8_t)((baudrate >> 24) & 0xFF);

Titania_SendCommand(huart, 0x09, payload, 6);
uint8_t resp[5];
HAL_UART_Receive(huart, resp, 5, 200);

// 00 resp[3] == 0x00 (Status OK)
```


- **LEN** = MAC [00000] bytes
- **MAC_DATA** = [0000000] payload
- **Checksum** = 0x02 [0000] DATA XOR

[00] - [00] 5 bytes [00] "Hello"[00]

```

[00][00]02 80 05 48 65 6C 6C 6F CS
    ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑
    ST CMD LEN H e l l o CS

```

"Hello" [00] ASCII [00]

H=0x48, e=0x65, l=0x6C, l=0x6C, o=0x6F

Checksum = 0x02 ^ 0x80 ^ 0x05 ^ 0x48 ^ 0x65 ^ 0x6C ^ 0x6C ^ 0x6F = ?

```

[00][00]02 C0 00 00 C0
    ↑ ↑ ↑ ↑ ↑
    ST CMD LEN ST CS

```

Status = 0x00 ([0000000000])

C [0000]

```

uint8_t payload[5] = { 0x48, 0x65, 0x6C, 0x6C, 0x6F }; // "Hello"
Titania_SendCommand(huart, 0x80, payload, 5);

uint8_t resp[5];
HAL_UART_Receive(huart, resp, 5, 100);
// resp[3] = Status

```

? Checksum ????

XOR [00] **Start (0x02)** [00000000] **Data**[0000] **Checksum** [0000]

```

static uint8_t CalcChecksum(uint8_t *buf, uint8_t len)
{
    uint8_t cs = 0;
    for (uint8_t i = 0; i < len; i++)
        cs ^= buf[i];
}

```

```

return cs;
}

// 000000
uint8_t frame[10] = { 0x02, 0x09, 0x03, 0x2A, 0x01, 0x03, ... };
uint8_t cs = CalcChecksum(frame, 6); // 0000 Checksum 0000
frame[6] = cs;

```

? ????????

A. ??????? Titania ??

1. 00 **UART** 00 → 0000000000 baudrate 0 9600
2. 000000 → CMD_GET_REQ 0 0x2A (PHY_DefaultChannel) 0 0x50 (UART_Baudrate)
3. 00000 → CMD_SET_REQ 0 UART_Baudrate 0 PHY_DefaultChannel 0 NetID 0 Addr 0
4. 000 → CMD_RESET_REQ 0 100~500 ms
5. 00 **STM32 UART** 00 → 0000 baudrate
6. 00 → 0000000000

B. ??????? baudrate

1. 000000 0 Titania 000 TX 0000000000 baudrate
2. 000 **baudrate** 00 STM32
3. 00 **CMD_FACTORY_RESET_REQ** 000000 UART 0 9600 0
4. 00 **STM32 UART** 0 **9600**
5. 000 **CMD_SET_REQ** 0000

C. ???????runtime ?????

1. 0 **CMD_SET_CHANNEL_REQ** (0x06) 00000
2. 000 **reset** 00000
3. 0000000 00000000
4. 0000000 CMD_SET_REQ (0x09) 0 PHY_DefaultChannel + CMD_RESET_REQ

D. ?????? NetID / Addr?MAC ?????

1. 0 **CMD_SET_REQ (0x09)** 000
 - MAC_DefaultDestNetID (memory position TODO)
 - MAC_DefaultDestAddrLSB (memory position TODO)

Step 2: STM32 UART 9600 baud

→ UART

Step 3:

→ CMD_GET_REQ 0x50 (UART_Baudrate)

→ 9600

? ? ? ? ?

UART reset

- UART_Baudrate CMD_RESET_REQ flash

Checksum

- bit

STM32 UART

- Titania UART_Baudrate STM32

NetID / Addr memory position

- TODO 8.13~8.16

- UART_Baudrate NetID Addr RF_ConfigIndex

? STM32 ? ? ? ? ?

```
#include "stm32f1xx_hal.h"

extern UART_HandleTypeDef huart1;

// Checksum
static uint8_t CalcChecksum(uint8_t *buf, uint8_t len)
{
```

```

uint8_t cs = 0;
for (uint8_t i = 0; i < len; i++)
    cs ^= buf[i];
return cs;
}

// 计算校验和
void SendTitaniaCmd(uint8_t cmd, uint8_t *data, uint8_t len)
{
    uint8_t frame[64];
    uint8_t idx = 0;

    frame[idx++] = 0x02; // Start
    frame[idx++] = cmd;
    frame[idx++] = len;
    for (uint8_t i = 0; i < len; i++)
        frame[idx++] = data[i];

    frame[idx] = CalcChecksum(frame, idx);
    idx++;

    HAL_UART_Transmit(&huart1, frame, idx, 100);
}

// 初始化 UART 波特率 57600 + 3 + reset
void InitTitania(void)
{
    uint8_t payload[6];
    uint8_t resp[10];

    // 1. 设置 UART 波特率 57600
    payload[0] = 0x50; // Memory Pos (UART_Baudrate)
    payload[1] = 0x04; // Length
    payload[2] = 0x00; // 57600 = 0x0000E100 (LE)
    payload[3] = 0xE1;
    payload[4] = 0x00;
    payload[5] = 0x00;
    SendTitaniaCmd(0x09, payload, 6);
    HAL_UART_Receive(&huart1, resp, 5, 200);
}

```

```

// 2. 通道 3
payload[0] = 0x2A;    // Memory Pos (PHY_DefaultChannel)
payload[1] = 0x01;    // Length
payload[2] = 0x03;    // Channel 3
SendTitaniaCmd(0x09, payload, 3);
HAL_UART_Receive(&huart1, resp, 5, 200);

// 3. Reset 通道
SendTitaniaCmd(0x05, NULL, 0);
HAL_Delay(500); // 延迟
HAL_UART_Receive(&huart1, resp, 5, 500);

// 4. STM32 UART 57600
huart1.Init.BaudRate = 57600;
HAL_UART_Init(&huart1);
}

```

? ? ? ? ?

- **Würth Titania Manual:** Manual-um-titania-260701111100x (rev4.5)
- **Titania 数据表:** 260701111100x datasheet
- **Band D Frequencies:** EN 300 220 (数据表)

日期

2026-03-10

设备

通道

Titania 数据表

memory position 数据表

Revision #3

Created 2026-04-01 02:06:13 UTC by TaipeiTechRacing

Updated 2026-04-11 14:34:06 UTC by AI Agent