???Wurth

????

本文檔示範如何使用 STM32 HAL 同時讀取三顆 Würth WSEN 環境感測器，並透過 Titania 無線模組將感測器資料以 UART 傳輸發送。

支援感測器:

- WSEN-PADS: 環境壓力與溫度
- WSEN-HIDS: 濕度與環境溫度
- WSEN-PDUS: 差壓與溫度

Titania 無線模組預設 UART 波特率 9600，支持命令模式下切換波特率至 115200 提升通信速率。

????

I²C ????

Sensor	**7-bit I²C 地址**	備註
WSEN-HIDS	0x44	濕度 + 溫度
WSEN-PADS	0x5C / 0x5D	氣壓 + 溫度 (SAO 腳選)
WSEN-PDUS	0x78	差壓 + 溫度

??????

1. ???????

I²C讀取感測器數據

```
HIDS_Read(&hi2c1, &HIDS);
PADS_Read(&hi2c1, &PADS);
PDUS_Read(&hi2c1, &PDUS);
```

讀取後的數據

Sensor	變數名稱	資料內容
WSEN-HIDS	`HIDS.temperature_c`, `HIDS.humidity_rh`	濕度 ，溫度
WSEN-PADS	`PADS.pressure_kpa`, `PADS.temperature_c`	氣壓 ，溫度
WSEN-PDUS	`PDUS.pressure_kpa`, `PDUS.temperature_c`	差壓 ，溫度

2. ??????????

1. WSEN?HIDS (I²C 0x44) ????

命令 0xFD 為高精度測量, 轉換時間約 8.4 ms。接著讀取 6 bytes: T + CRC + RH + CRC。

```
#define HIDS_ADDR (0x44 << 1)
HAL_StatusTypeDef HIDS_Read(I2C_HandleTypeDef *hi2c, SensorData_t *data)
{
    uint8_t cmd = 0xFD;
    uint8_t rx[6];
    HAL_I2C_Master_Transmit(hi2c, HIDS_ADDR, &cmd, 1, HAL_MAX_DELAY);
    HAL_Delay(9); // wait conversion
    HAL_I2C_Master_Receive(hi2c, HIDS_ADDR, rx, 6, HAL_MAX_DELAY);

    uint16_t Traw = (rx[0] << 8) | rx[1];
    uint16_t RHraw = (rx[3] << 8) | rx[4];

    data->temperature_c = -45.0f + 175.0f * (float)Traw / 65535.0f;
    data->humidity_rh = 100.0f * (float)RHraw / 65535.0f;
    return HAL_OK;
}
```

2. WSEN?PADS (I²C 0x5C) ?????

壓力暫存器 0x28–0x2A (24 bit), 溫度 0x2B–0x2C (16 bit)。壓力轉換式: P(Pa) = raw / 40960.0f , 溫度: T(°C) = raw * 0.01f。

```
#define PADS_ADDR (0x5C << 1)

void PADS_Init(I2C_HandleTypeDef *hi2c)
{
    uint8_t cfg[2] = {0x10, 0b00011000}; // ODR=25Hz, 連續模式
    HAL_I2C_Master_Transmit(hi2c, PADS_ADDR, cfg, 2, 100);
}

HAL_StatusTypeDef PADS_Read(I2C_HandleTypeDef *hi2c, SensorData_t *data)
{
    uint8_t reg = 0x28;
    uint8_t rx[5];
    HAL_I2C_Master_Transmit(hi2c, PADS_ADDR, &reg, 1, 100);
    HAL_I2C_Master_Receive(hi2c, PADS_ADDR, rx, 5, 100);

    int32_t Praw = (int32_t)((rx[2] << 16) | (rx[1] << 8) | rx[0]);
    int16_t Traw = (int16_t)((rx[4] << 8) | rx[3]);
    data->pressure_kpa = Praw / 4096.0f / 10.0f; // convert Pa → kPa
    data->temperature_c = Traw * 0.01f;
    return HAL_OK;
}
```

3. WSEN?PDUS (I²C 0x78) ?????

每 2.2 ms 更新, 4 byte: P + T (各 15 bit)。氣壓轉換式: `P(kPa) = (rawP - OUTPMIN) × SENP + PMIN` 溫度: `T(°C) = (rawT - 8192) × 4.272e-3`

以下以型號 2513130810301 (0-100 kPa @ 5 V) 為例:

```
#define PDUS_ADDR (0x78 << 1)
#define OUTPMIN  3277
#define SENP     3.815e-3
#define PMIN     0.0

HAL_StatusTypeDef PDUS_Read(I2C_HandleTypeDef *hi2c, SensorData_t *data)
{
```

```
    uint8_t rx[4];
    HAL_I2C_Master_Receive(hi2c, PDUS_ADDR, rx, 4, 100);

    uint16_t Praw = ((rx[0] << 8) | rx[1]) & 0x7FFF;
    uint16_t Traw = ((rx[2] << 8) | rx[3]) & 0x7FFF;

    data->pressure_kpa = (Praw - OUTPMIN) * SENP + PMIN;
    data->temperature_c = (Traw - 8192) * 4.272e-3f;
    return HAL_OK;
}
```

2. Titania UART ???????

```
HAL_StatusTypeDef Titania_SetBaudRate115200(UART_HandleTypeDef *huart)
{
uint8_t cmd[] = {
0x02, // Start
0x09, // CMDSETREQ
0x06, // Length
0x50, // Memory address UARTBaudrate
0x04, // Data length 4 bytes
0x00, 0xC2, 0x01, 0x00, // 115200 LSB first
0x00 // Checksum placeholder
};
uint8_t cs = 0;
for(int i=0; i<sizeof(cmd)-1; i++) cs ^= cmd[i];
cmd[sizeof(cmd)-1] = cs;


return HAL_UART_Transmit(huart, cmd, sizeof(cmd), HAL_MAX_DELAY);
}
```

text

3. ??????

int main(void) { HAL_Init(); SystemClock_Config(); MX_GPIO_Init(); MX_I2C1_Init();

text // □□ UART □□□ 9600bps MX_USART1_UART_Init_9600();

SensorData_t HIDS, PADS, PDUS;

// 初始化 Titania (如果需要) // ...

// 傳送改波特率指令 if(Titania_SetBaudRate115200(&huart1) == HAL_OK) { HAL_Delay(100); // 等待模組切換

```
// MCU UART 也切換 115200
MX_USART1_UART_Init_115200();
```

}

while(1) { HIDS_Read(&hi2c1, &HIDS); PADS_Read(&hi2c1, &PADS); PDUS_Read(&hi2c1, &PDUS);

```
// 整合資料格式
char payload;
int len = snprintf(payload, sizeof(payload),
    "HIDS: %.2fC %.1f%%, PADS: %.2fkPa %.2fC, PDUS: %.2fkPa %.2fC",
    HIDS.temperature_c, HIDS.humidity_rh,
    PADS.pressure_kpa, PADS.temperature_c,
    PDUS.pressure_kpa, PDUS.temperature_c);

// 傳送資料到 Titania
HAL_UART_Transmit(&huart1, (uint8_t*)payload, len, HAL_MAX_DELAY);

HAL_Delay(500);
```

} }

text

4. UART ????? (??)

void MX_USART1_UART_Init_9600(void) { huart1.Instance = USART1; huart1.Init.BaudRate = 9600; huart1.Init.WordLength = UART_WORDLENGTH_8B; huart1.Init.StopBits = UART_STOPBITS_1; huart1.Init.Parity = UART_PARITY_NONE; huart1.Init.Mode = UART_MODE_TX_RX; huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE; huart1.Init.OverSampling = UART_OVERSAMPLING_16; HAL_UART_Init(&huart1); }

void MX_USART1_UART_Init_115200(void) { huart1.Instance = USART1; huart1.Init.BaudRate = 115200; huart1.Init.WordLength = UART_WORDLENGTH_8B; huart1.Init.StopBits = UART_STOPBITS_1; huart1.Init.Parity = UART_PARITY_NONE; huart1.Init.Mode = UART_MODE_TX_RX; huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE; huart1.Init.OverSampling

```
= UART_OVERSAMPLING_16; HAL_UART_Init(&huart1); }
```

text

????

- 請確保 Titania 模組初始是 9600 baud, 否則指令無法正常送達。
- 波特率切換後 MCU 需立即同步調整 UART 設定。
- Titania 透過命令模式 (Command Mode) 接收波特率修改命令。
- 讀取感測器與無線發送任務可根據實際需求加中斷或任務分配。

????

- Würth Elektronik WSEN-PADS / WSEN-HIDS / WSEN-PDUS 官方手冊
- Titania 無線模組使用手冊（附件）

Revision #4
Created 2026-04-01 02:06:20 UTC by TaipeiTechRacing
Updated 2026-04-11 14:41:04 UTC by AI Agent