

HackMD ??????

- [03_03](#)
- [\[REDACTED\]](#)
- [\[REDACTED\]](#)
- [\[REDACTED\] Wurth](#)
- [\[REDACTED\]](#)
- [\[REDACTED\] \(20250731\)](#)
- [Hackmd\[REDACTED\]](#)
- [Tip](#)

03_03

03/04

image

image

IG

OR ()

????????

????????

??

Step 1

??

?????

□□□ : □□□ /□□□ /5~9□

? ?????

□□ **STM32 MCU** □□□□□□□□

- **MPXV7002** □□□□ □□□ ADC □□□□□□□□□□
- □□ UART □□□□□□□ Python GUI □□□□

? ?????

□□□ /□□	□□	□□
MPXV7002	□□□	image
stm32	□□	image
GUI □□	□□□	USARTx_TX (UART □□□□)

? ???? ???? ?

□□□□□ 2025-05-21 164842

?????

MPXV7002 ↓ ADC ↓ □□ /□□□□ ↓ STM32 □□□□ ↓ UART □□□ GUI

? ???? ???? ?

???? ???? ?

```
pressure_kPa = (voltage - 2.5) / 1.0; // ±2kPa □□ 0.5V~4.5V
```

????????????????

```
airspeed_mps = sqrt(2 * abs(pressure_kPa) * 1000 / 1.225);  
airspeed_kmh = airspeed_mps * 3.6;
```

? STM32 ??????

1 ADC [] [] [] [] [] 12-bit [] [] 112~114 []

```
HAL_ADC_Start(&hadc1);  
HAL_ADC_PollForConversion(&hadc1,100);  
if(HAL_IS_BIT_SET(HAL_ADC_GetState(&hadc1),HAL_ADC_STATE_REG_EOC))  
ADC_Value = HAL_ADC_GetValue(&hadc1);
```

2 [] [] / [] [] [] [] 118~121 []

```
ADC_Volt = (float)ADC_Value/4096*3.3;  
float pressure = (ADC_Volt - V_OFFSET) / SENSITIVITY; //kPa  
float airspeed = sqrt(2 * fabs(pressure) * 1000.0f / AIR_DENSITY);  
float airspeed_kmh = airspeed * 3.6f;
```

3 [] [] [] [] [] [] [] [] UART [] [] 123~126 []

```
if(HAL_GetTick()%100 == 0){ //USRT Send  
    char buf[100];  
    sprintf(buf, "Volt:%.2lf V \t P:%.2lf kPa \t S:%.2lf Kmh\n",  
ADC_Volt,pressure,airspeed_kmh);  
    HAL_UART_Transmit(&huart1, (uint8_t*)buf, strlen(buf), HAL_MAX_DELAY);  
}
```

? STM32 ????

image


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

□□□□□

Sensor	□□□□	□□□□
WSEN-HIDS	HIDS.temperature_c, HIDS.humidity_rh	□□ , □□
WSEN-PADS	PADS.pressure_kpa, PADS.temperature_c	□□ , □□
WSEN-PDUS	PDUS.pressure_kpa, PDUS.temperature_c	□□ , □□

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

1. WSEN?HIDS (I²C 0x44) ?????

□□ 0xFD □□□□□□□□□□ 8.4 ms□ □□□□ 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) ??????

0x28-0x2A (24 bit)
0x2B-0x2C (16 bit)
P(Pa) = raw / 40960.0f
T(°C) = raw * 0.01f

```

#define PADS_ADDR (0x5C << 1)

void PADS_Init(I2C_HandleTypeDef *hi2c)
{
    uint8_t cfg[2] = {0x10, 0b00011000}; // ODR=25Hz,
    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) ??????

2.2 ms
4 byte
P + T (15 bit)
P(kPa) = (rawP - OUTPMIN) × SENP + PMIN
T(°C) = (rawT - 8192) × 4.272e-3

2513130810301 (0-100 kPa @ 5 V)

```

#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();
```



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

text

????

- [] Titania [] 9600 baud []
 - [] MCU [] UART []
 - Titania [] (Command Mode) []
 - []
-

????

- Würth Elektronik WSEN-PADS / WSEN-HIDS / WSEN-PDUS []
 - Titania []
-

??????(20250731)

??????????

1. NAS
2. MOUSER
3.
4.

? ??????????????????

image image image

? ??????? MOUSER ???????????

MOUSER MOUSER

[MOUSERE](#)

image

? ??????????????????

image

? ??????? NAS ?????

NAS

MOUSER

[NAS](#) image

? ??????

Tip