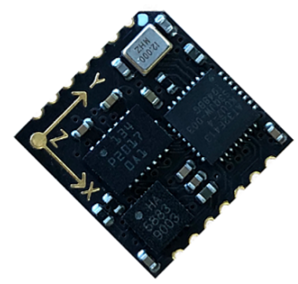
Hi229 User Manual

IMU/VRU/AHRS Attitude Measurement Module, Rev 1.0



1. Table of Contents

2. Introduction

Hi229 is a low-cost, high-performance, compact, low-latency Attitude and Heading Reference System (AHRS). The product integrates a 3-axis accelerometer, 3-axis gyroscope, 3-axis magnetic field sensor (magnetometer), and a microcontroller. It outputs 3D orientation data in the local geographic coordinate frame computed by the sensor-fusion algorithm, including heading (Yaw), pitch (Pitch), and roll (Roll). It can also output calibrated raw sensor data. The product provides a certain level of indoor magnetic-interference immunity and can still operate normally under geomagnetic disturbances of limited strength.

Typical applications:

- VR / Motion capture
- Attitude measurement / motion performance evaluation in high-dynamic environments
- UAV control

3. Key Features

3.1. On-board Sensors

- 3-axis gyroscope, full-scale range: ±2000°/s
- 3-axis accelerometer, full-scale range: ±8G
- 3-axis magnetic field sensor, full-scale range: 800mG (milli-Gauss)

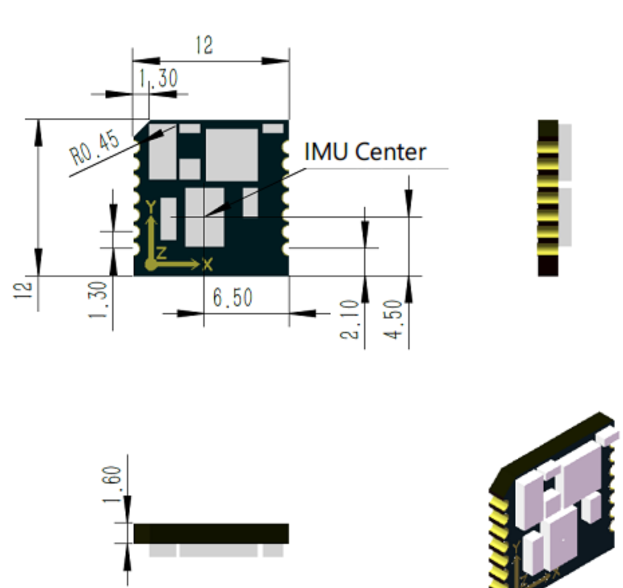
3.2. Communication Interface and Power

- Serial port (TTL compatible; can connect directly to 5V or 3.3V serial devices)
- Supply voltage: 3.3V (± 100mV)
- Peak current consumption: 32mA

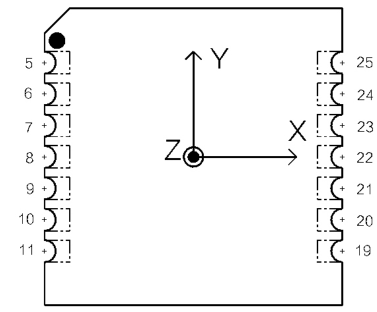
4. Hardware Specifications

Parameter	Description
Output interface	UART (TTL 1.8V - 5.0V)
Operating voltage	3.3V (± 100mV)
Power consumption	86mW @3.3V
Temperature range	-20℃ - 85 ℃
Max output rate	400Hz
Dimensions	12 x 12 x 2.6mm (W x L x H)
On-board sensors	3-axis accelerometer, 3-axis gyroscope, 3-axis magnetic field sensor

4.1. Dimensions



4.2. Pin Assignment



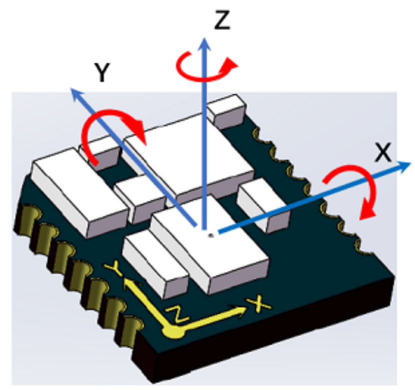
Pin No.	Name	Description
5	N/C	Reserved
6	VCC	Power supply 3.3V
7	SYNC_OUT	Data output sync, internal pull-up. High when there is no data output (idle). Goes low when a data frame starts transmitting and returns high (idle) after the frame completes. Leave floating if unused.
8	RXD	Module UART receive, RXD (connect to MCU TXD)
9	TXD	Module UART transmit, TXD (connect to MCU RXD)
10	SYNC_IN	Data input sync: internal pull-up. When the module detects a falling edge, it outputs one data frame. Leave floating if unused.
11	N/C	Reserved
19	GND	GND
20	RST	Reset, internal pull-up. Pull low for >10uS to reset the module. No external RC is required. Recommended to connect to an MCU GPIO for software reset.
21	N/C	Reserved
22	N/C	Reserved
23	N/C	Reserved
24	GND	GND
25	N/C	Reserved

5. Coordinate Frame Definition

The body frame uses a front-left-up (FLU) right-handed coordinate system, and the navigation/geographic frame uses a north-west-up (NWU) coordinate system. The Euler-angle rotation order is ZYX (rotate about Z first, then Y, and finally X). The definitions are:

- Rotation about Z axis: heading (also called yaw, or ψ (pronounced "psi")), range: -180° to 180°
- Rotation about Y axis: pitch (also called Pitch, or θ (pronounced "theta")), range: -90° to 90°
- Rotation about X axis: roll (also called Roll, or ϕ (pronounced "phi")), range: -180° to 180°

If you treat the module as an aircraft, the X axis should be considered the nose direction. When the sensor frame coincides with the inertial frame, the ideal Euler-angle output is: Pitch = 0°, Roll = 0°, Yaw = 0°.



6. Performance

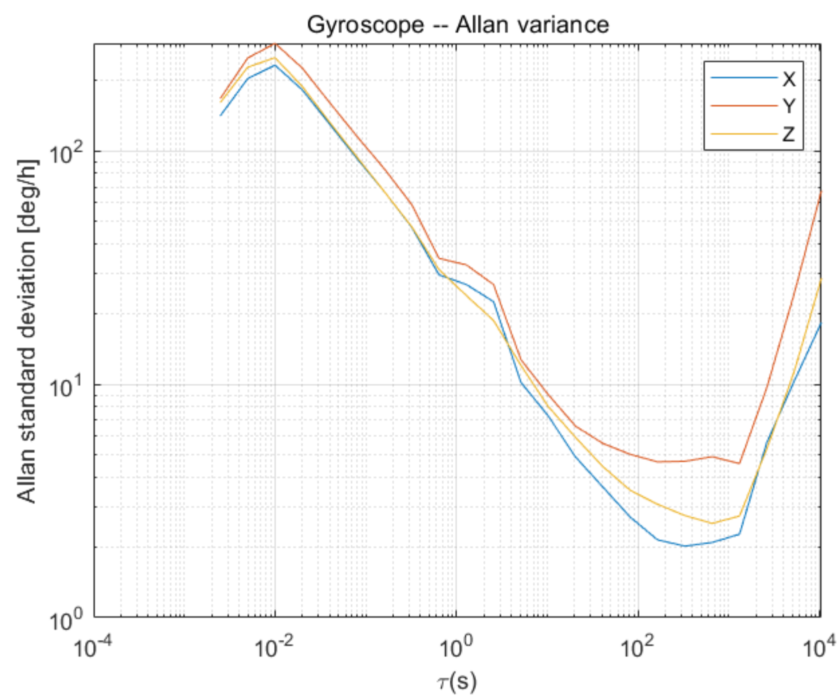
6.1. Attitude Angle Output Accuracy

Attitude angle	Typical
Roll/Pitch - static error	0.8°
Roll/Pitch - dynamic error	2.5°
Heading accuracy in motion (6-axis mode, 30min, smooth level motion, “robot vacuum” motion pattern)	<10°
Heading accuracy in motion (9-axis mode, no magnetic interference, after calibration)	3°

6.2. Gyroscope

Parameter	Value	Notes
Measurement range	±2000°/s	
Resolution	0.01°/s	
Internal sampling rate	1KHz	
Bias stability	8°/h	@25°C, 1σ
Bias repeatability	0.12°/s	@25°C, 1σ
Non-orthogonality error	±0.1%	@25°C, 1σ
Angle random walk	$0.6^\circ/\sqrt{hr}$	@25°C, 1σ
Scale-factor nonlinearity	±0.1%	(max) at full scale
Scale-factor error	±0.4%	After factory calibration
Acceleration sensitivity	0.1°/s/g	

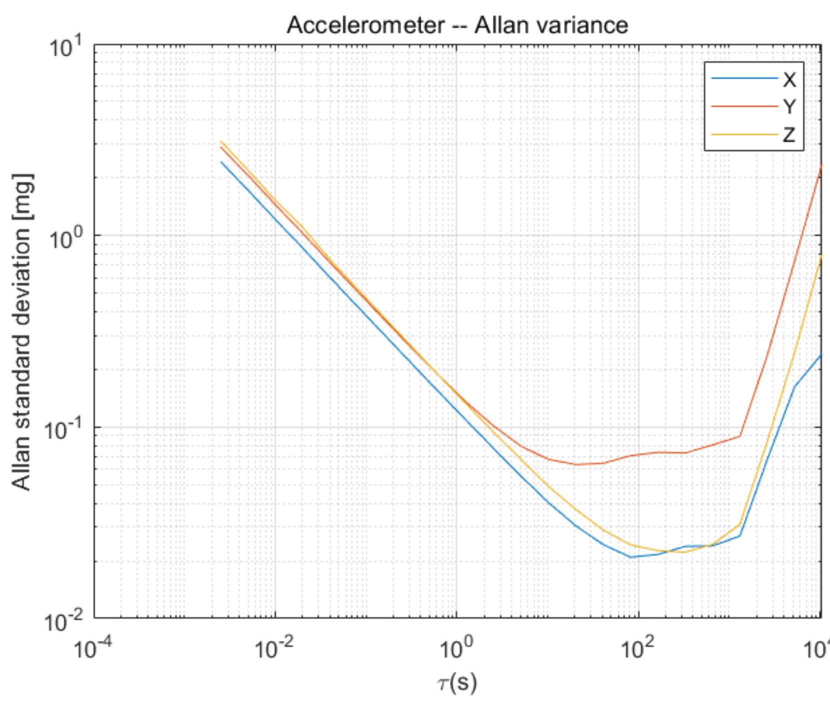
Gyro Allan variance curve



6.3. Accelerometer

Parameter	Value	Notes
Measurement range	±8G (1G = 1x gravitational acceleration)	
Resolution	1uG	
Internal sampling rate	1KHz	
Bias stability	60uG	@25°C, 1σ
Bias repeatability	4.8mG	@25°C, 1σ
Non-orthogonality error	±0.1%	
Velocity random walk	0.08$m/s\sqrt{h}$	@25°C, 1σ
Scale-factor error	±0.3% (at full scale)	After factory calibration
Bias variation over full temperature range	2mg	-20 - 85°

Accelerometer Allan variance curve



6.4. Magnetometer Parameters

Parameter	Value
Measurement range	±8G (Gauss)
Nonlinearity	±0.1%
Resolution	0.25mG

6.5. Module Data Interface Parameters

Parameter	Value
UART baud rate	9600/115200/460800/921600 (selectable)
Frame output rate	1/25/50/100/200/400Hz (selectable)

7. Sensor Calibration

7.1. Accelerometer and Gyroscope

The accelerometer and gyroscope are factory calibrated for scale-factor error, non-orthogonality error, and bias error, and the calibration parameters are stored inside the module. In addition, the gyroscope is temperature-compensated at the factory, and the compensation parameters are stored inside the module.

7.2. Magnetometer Calibration

The magnetometer (supported on some variants) is factory ellipsoid-calibrated. However, the magnetometer is easily affected by external magnetic disturbances, so customers generally need to recalibrate it:

The module includes an active, automatic magnetometer calibration system. The user does not need to send any commands. The system automatically collects magnetic-field data over a period of time in the background, analyzes and compares the data, removes outliers, and attempts magnetometer calibration once enough valid data has been collected. Therefore, when using 9-axis mode, **no user intervention is required to complete magnetometer calibration**. The module still provides an interface for checking the current calibration status. Automatic calibration requires sufficient attitude maneuvers (changes in module orientation) sustained for a period of time, so the internal calibration system can collect magnetic-field information under different orientations and complete calibration. **Magnetometer calibration cannot be performed while the module is stationary.**

When using the module for the first time and you need 9-axis mode, perform the following calibration procedure:

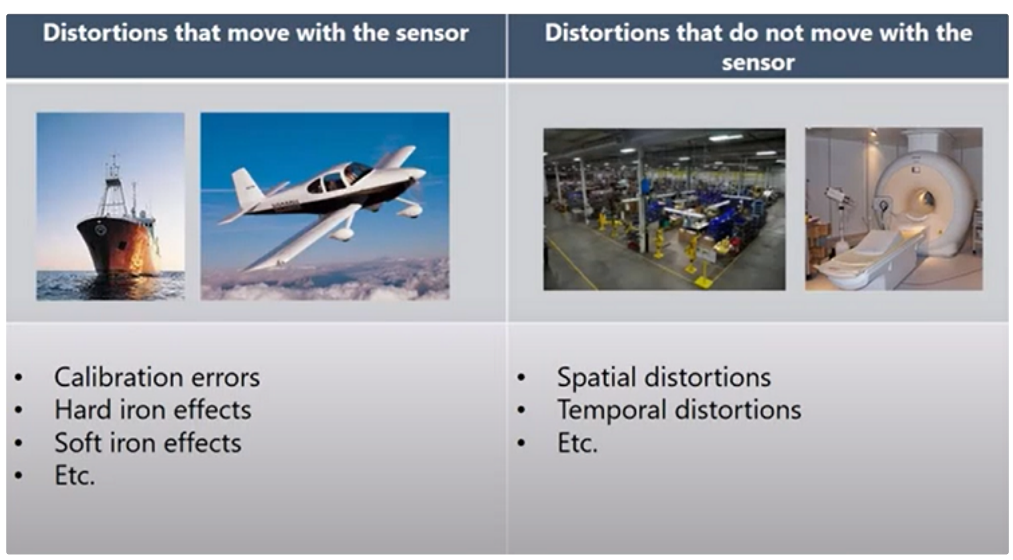
1. Check whether the surroundings are magnetically clean. Indoors, near lab benches, and near large iron/steel frame structures are common interference areas. It is recommended to take the module outdoors to an open area. If that is not possible, keep the module away from (>0.5m) lab benches/computers and other objects that may create magnetic interference.
2. In as small an area as possible (do not change position, only rotate), slowly rotate the module so it experiences as many orientations as possible (rotate at least 360° about each axis; about 1 minute total). In most cases this completes calibration. If calibration still cannot succeed, it indicates strong magnetic interference in the environment.
3. You can check whether calibration succeeded via AT command: send `AT+INFO=HSI`. The module will print the current status of the magnetometer calibration system:

Only the `fiterr` item matters: a value below 0.03 indicates the calibration result is already good enough. If `fiterr` remains >0.1, it indicates strong magnetic interference and you should recalibrate to obtain better results. The fit residual will slowly increase over time.

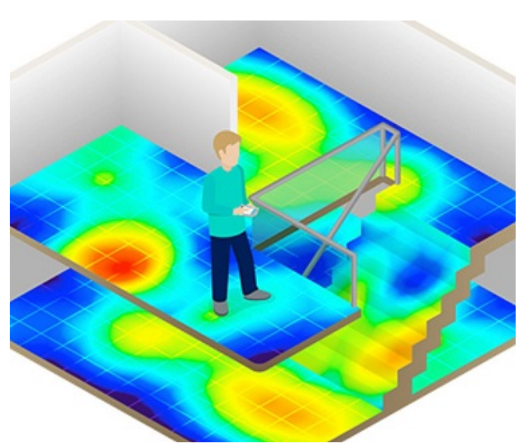
4. Although magnetometer parameter estimation can automatically collect data online and dynamically fit calibration parameters, if the surrounding magnetic environment changes (e.g., moving to a different room, switching between indoor and outdoor, or the module is installed/soldered into a new environment), you should repeat steps 1–3.
5. Although magnetometer calibration does not require manually starting/stopping the system (it runs automatically in the background), the user can still manually enable/disable the magnetometer calibration system. Use `AT+MCALCTL=0` to disable and `AT+MCALCTL=1` to enable. The command takes effect immediately and is saved across power cycles. After enabling magnetometer calibration, the user can use `AT+INFO=HSI` to judge calibration quality. Once calibration completes successfully, use `AT+MCALCTL=0` to disable the calibration system and lock the calibration values. In general, once you have calibrated successfully in a magnetically clean area, you do not need to recalibrate again later.
6. If the customer's installation changes (for example, the previous calibration was done while holding the module alone, but during actual use it is installed on the target device), then recalibration must be performed with the target device. See the mobile robot case study.

7.2.1. More About Magnetic Interference

Category	Definition	Typical sources	Impact	Mitigation
Spatial magnetic distortions (Distortions that do not move with sensor)	The distortion does not move with the sensor; it is fixed in the world frame	Fixed sources such as furniture, household appliances, cables, steel reinforcement in buildings, etc. Anything that does not move with the magnetometer	Regardless of how well the magnetometer is calibrated, these spatial distortions (i.e., non-uniform environmental field) will distort the local geomagnetic field. Magnetic compensation becomes wrong and a correct heading cannot be obtained. This is the main reason indoor magnetometer fusion is difficult to use. This distortion cannot be calibrated out and will severely impact magnetometer performance. It is especially severe indoors.	Avoid these sources as much as possible
Sensor-frame distortions (Distortions that move with sensor)	The distortion source moves with the sensor	The module PCB, boards fixed to the module, instruments, products, etc. They can be regarded as a rigid body together with the magnetometer and move with it	Causes hard-iron and soft-iron distortions to the sensor. These distortions can be effectively removed by the magnetometer calibration algorithm.	Module automatic magnetometer calibration



The figure below shows a typical indoor magnetic-field distribution. You can see that spatial magnetic-field distortion is generally severe indoors (spatial geomagnetic distortion that cannot be compensated by calibration).



Note

In indoor environments, spatial magnetic distortion is especially severe, and it cannot be eliminated by calibration. Although the module has built-in homogeneous-field detection and shielding mechanisms, the heading accuracy in 9-axis mode depends heavily on how distorted the indoor magnetic environment is. If the indoor magnetic environment is poor (e.g., near computer rooms, electromagnetic labs, workshops, underground garages, etc.), even after calibration, the heading accuracy in 9-axis mode may be worse than in 6-axis mode, and large heading errors may occur.

7.2.2. Differences Between 6-axis and 9-axis Modes

Because the geomagnetic field is very susceptible to spatial interference, extra care is required when using 9-axis mode. The table below lists recommended use cases and operating conditions.

Mode	Applicable environment	Typical applications	Advantages	Disadvantages	Notes
6-axis mode	All environments	Low-dynamic attitude sensing such as gimbals; indoor robots	1. Stable attitude angle output 2. Completely immune to magnetic interference	Heading slowly drifts over time	Heading drifts slowly over time and cannot be compensated
9-axis mode	Magnetically clean environments	1. Compass, north-finding systems 2. Spacious indoor areas with little magnetic interference where the module does not move widely indoors (typical: motion capture in a studio where the subject does not walk around extensively)	1. Heading does not drift over time 2. Heading can be quickly corrected to north once the geomagnetic field is detected	Any magnetic interference reduces heading accuracy. With severe indoor interference, the heading may not point in the correct direction. In addition, the metal structure of mobile robots and motor operation generate very strong magnetic interference, so mobile robot platforms are not suitable for 9-axis mode.	Magnetometer calibration is required before first use

The module's automatic magnetometer calibration system can only handle fixed magnetic interference sources that are installed together with the module. If there is magnetic interference in the installation environment, the interference must be fixed, and the distance between the interference source and the module must not change after installation (e.g., when the module is installed on iron material that produces magnetic interference, you must rotate and calibrate the iron together with the module, and the iron must not be separated from the compass during use (no relative displacement). Once separated, recalibration is required. If the size of the iron or its distance to the compass is not fixed, the interference cannot be calibrated out; even if calibration succeeds, accuracy will be poor, and you should avoid such installation. Keep a safe distance of 40cm or more).

7.2.3. Mobile Robot Case Study

Assume the customer wants to use 9-axis mode on a mobile robot to obtain accurate, non-drifting heading, and the module is installed on the robot (treated as a rigid body):

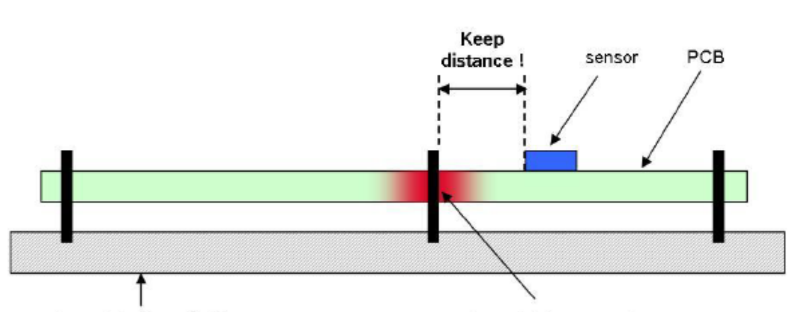
- The robot itself has significant hard-iron interference due to metal structures (parts, circuitry), which corresponds to the "sensor-frame distortion" described above. This part can be calibrated out.
- Due to motor start/stop and the robot passing through different rooms (changing spatial magnetic interference), the spatial magnetic field changes, producing "spatial magnetic distortion" described above. This part cannot be calibrated out.

Both distortions can exist simultaneously and can both be large, posing a big challenge for 9-axis mode. In this case, 6-axis mode is recommended. If 9-axis mode must be used, the following should be satisfied:

1. Calibration: you must calibrate together with the robot (this is feasible only if the robot is small; large robots are difficult to calibrate together). It is incorrect to remove the module, calibrate it alone, and then reinstall it. The robot and module must be treated as one rigid body for calibration to obtain correct results. See the procedure above. After calibration succeeds, power-cycle (reset) the module for the calibration to take effect.
2. Due to complex indoor magnetic environments, even with correct calibration, large heading errors may still occur, especially when motors start/stop or power changes, which can significantly affect the magnetic field.
3. Urban buildings, underground structures, bridges, and other steel structures can cause local magnetic anomalies. In particular, in areas where vehicles are driven by DC motors, 9-axis mode may be unusable. When a DC-driven vehicle passes by, nearby vehicles can become magnetized; all previous calibration parameters become invalid, and the vehicle body may not return to its original magnetic structure.

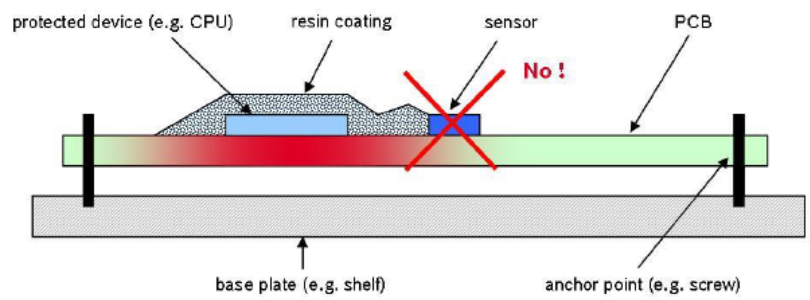
8. Installation and Soldering

1. Keep the installation location away from PCB areas that are prone to deformation, and as far as possible from PCB edges (>30mm) and PCB mounting screw holes (>10mm), etc.

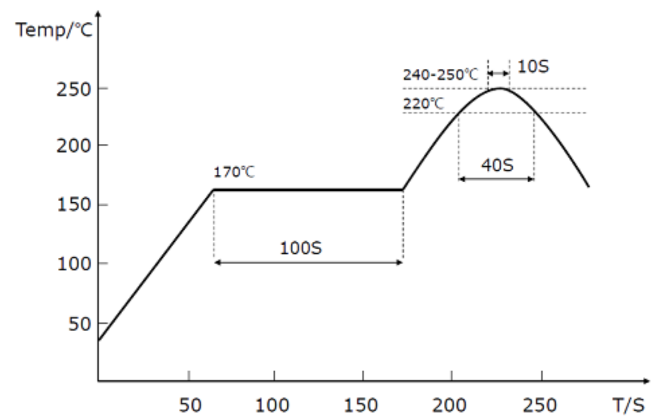


2. Keep the installation location away from strong magnetic devices such as motors and speakers.

3. Do not clean the assembled PCB with an ultrasonic cleaner.
4. Do not pot the product with plastic encapsulation or spray conformal coating. Painting/encapsulation can change the stress on the sensor and therefore affect performance.



5. The recommended reflow soldering temperature profile is shown below:



Note

In the final stage of reflow soldering, natural cooling is required. Do not force air-cool by opening the oven, otherwise product performance may be severely affected.

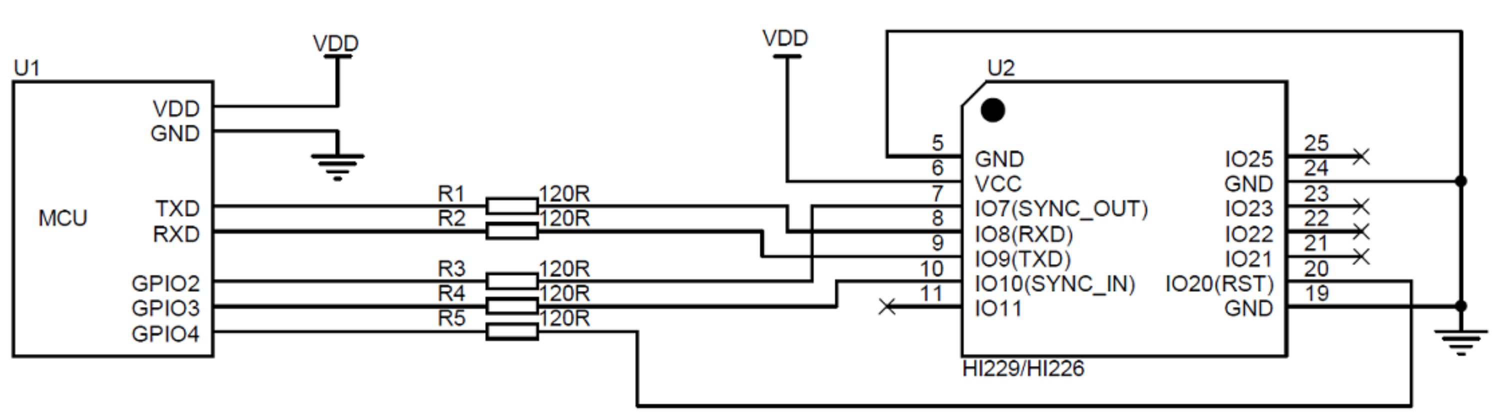
9. Application Guide

9.1. Connecting the Module to a PC

It is recommended to connect to a PC using the evaluation board. The evaluation board provides USB power and USB-to-UART, making it easy to perform performance tests with the PC evaluation software. See the “Evaluation Board” section in the appendix.

9.2. Connecting the Module to an MCU

Connect the module to the MCU via a TTL-level UART. It is recommended to connect the module’s RST pin to an MCU GPIO to allow the MCU to forcibly reset the module.



Note

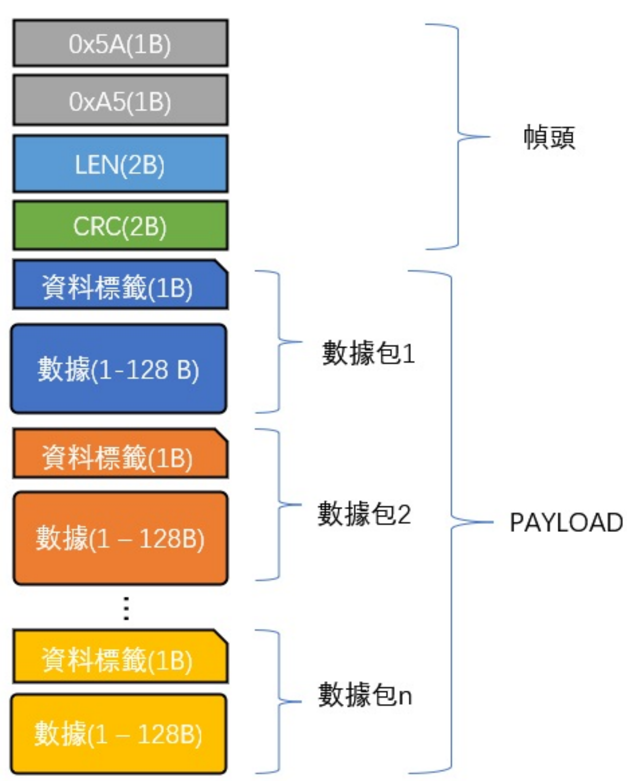
1. If you do not use sync input (SYNC_IN) and sync output (SYNC_OUT), you may leave SYNC_IN and SYNC_OUT unconnected.
2. The 120-ohm resistor is for easier debugging and to prevent level mismatch between the MCU and module. It can be removed, but keeping it is recommended.
3. For the VCC voltage range, refer to the manual.
4. The module includes an internal power-on reset circuit, so RST can be left unconnected, but it is recommended to connect it to a host GPIO to enable software reset.

10. UART Communication Protocol

After power-up, the module outputs data frames at the factory-default output data rate (ODR, typically 100). The frame format is:

```
UART frame structure:
<Header (0x5A)><Frame type (0xA5)><Length><CRC><Payload>
```

Field	Value	Length (bytes)	Description
Header	0x5A	1	Fixed value 0x5A
Frame type	0xA5	1	Fixed value 0xA5
Length	1-512	2	Length of the payload in the frame, LSB first. Length indicates the payload length (excluding Header, Frame type, Length, CRC).
CRC	-	2	16-bit CRC over all fields except CRC itself (Header, Frame type, Length, Payload). LSB first.
Payload	-	1-512	Data carried in one frame. It consists of multiple **sub-packets**. Each sub-packet includes a tag byte and the corresponding data. The tag determines the data type and length.



CRC implementation function:

```c
/*
    currectCrc: previous crc value, set 0 if it's first section
    src: source stream data
    lengthInBytes: length
*/
static void crc16_update(uint16_t *currectCrc, const uint8_t *src, uint32_t lengthInBytes)
{
    uint32_t crc = *currectCrc;
    uint32_t j;
    for (j=0; j < lengthInBytes; ++j)
    {
        uint32_t i;
        uint32_t byte = src[j];
        crc ^= byte << 8;
        for (i = 0; i < 8; ++i)
        {
            uint32_t temp = crc << 1;
            if (crc & 0x8000)
```

```
            {
                temp ^= 0x1021;
            }
            crc = temp;
        }
    }
    *currectCrc = crc;
}
```

11. UART Data Packets

11.0.1. Packet Summary

Packet tag	Packet length (including 1-byte tag)	Name	Notes
0x90	2	User ID	
0xA0	7	Factory-calibrated acceleration	
0xB0	7	Factory-calibrated angular rate	
0xC0	7	Magnetic field strength	
0xD0	7	Euler angles	
0xD1	17	Quaternion	
0xF0	5	Barometric pressure	Outputs 0
0x91	76	IMUSOL (IMU data set)	Recommended

11.0.2. Supported Packet List

11.0.3. 0x90 (User ID)

2 bytes. The user-configured ID.

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0x90
1	uint8_t	1	-	User ID

11.0.4. 0xA0 (Acceleration)

7 bytes total, LSB. Outputs the sensor's raw acceleration.

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0xA0
1	int16_t	2	0.001G (1G = 1g)	X-axis acceleration
3	int16_t	2	0.001G	Y-axis acceleration
5	int16_t	2	0.001G	Z-axis acceleration

11.0.5. 0xB0 (Angular Rate)

7 bytes total, LSB. Outputs the sensor's raw angular rate.

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0xB0
1	int16_t	2	0.1°/s	X-axis angular rate
3	int16_t	2	0.1°/s	Y-axis angular rate
5	int16_t	2	0.1°/s	Z-axis angular rate

11.0.6. 0xC0 (Magnetic Field Strength)

7 bytes total, LSB. Outputs the sensor's raw magnetic field strength.

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0xC0
1	int16_t	2	0.001Gauss	X-axis magnetic field
3	int16_t	2	0.001Gauss	Y-axis magnetic field
5	int16_t	2	0.001Gauss	Z-axis magnetic field

11.0.7. 0xD0 (Euler Angles)

7 bytes total, LSB. int16 format. There are three axes, each taking 2 bytes, ordered as Pitch/Roll/Yaw. The received Roll and Pitch values are the physical values multiplied by 100; Yaw is multiplied by 10.

Example: if the received Yaw = 100, it indicates a heading of 10°.

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0xD0
1	int16_t	2	0.01°	Pitch
3	int16_t	2	0.01°	Roll
5	int16_t	2	0.1°	Yaw

11.0.8. 0xD1 (Quaternion)

17 bytes total. float format. Four values in order: W X Y Z. Each value is 4 bytes (float), LSB.

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0xD1
1	float	4	-	W
5	float	4	-	X
9	float	4	-	Y
13	float	4	-	Z

11.0.9. 0xF0 (Barometric Pressure)

5 bytes total. float format. (Only for products that include a barometric pressure sensor.)

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0xF0
1	float	4	Pa	Atmospheric pressure

11.0.10. 0x91 (IMUSOL)

76 bytes total. This is a new packet introduced to replace older packets such as A0, B0, C0, D0, D1, etc. It integrates raw IMU sensor output and attitude-solution data. All data fields are LSB.

Byte offset	Type	Size	Unit	Description
0	uint8_t	1	-	Packet tag: 0x91
1	uint8_t	1	-	ID
2	-	2	-	Reserved
4	float	4	Pa	Barometric pressure (supported on some variants)
8	uint32_t	4	ms	Timestamp since system power-up; increments by 1 every millisecond
12	float	12	1G (1G = 1g)	Factory-calibrated acceleration, XYZ axes, float per axis
24	float	12	deg/s	Factory-calibrated angular rate, XYZ axes, float per axis
36	float	12	uT	Magnetic field strength, XYZ axes, float per axis
48	float	12	deg	Node Euler angles in order: Roll, Pitch, Yaw; float per angle
60	float	16	-	Node quaternion set in order WXYZ; float per component

11.1. Factory Default Packet

The factory default packets included in one frame are defined as follows:

Product	Default output packet
HI226	91
Hi229	91
CH100	91
CH110	91

11.2. Frame Structure Examples

11.2.1. Frame configured with packets `0x90,0xA0,0xB0,0xC0,0xD0,0xF0`

Capture one frame using a serial assistant. The frame is 41 bytes total. The first 6 bytes are the header, length, and CRC. The remaining 35 bytes are the payload. Assume the received bytes are stored in a C array `buf`:

5A A5 23 00 FD 61 **90** 00 **A0** 55 02 3D 01 E2 02 **B0** FE FF 17 00 44 00 **C0** 80 FF 60 FF 32 FF **D0** 64 F2 6C 0E BB 01 **F0** 00 00 00 00

- Step 1: Check the header and obtain the payload length and frame CRC:

Header: `5A A5`

Payload length: `23 00`: (0x00<<8) + 0x23 = 35

Frame CRC: `FD 61`: (0x61<<8) + 0xFD = 0x61FD

- Step 2: Verify CRC

```
uint16_t payload_len;
uint16_t crc;

crc = 0;
payload_len = buf[2] + (buf[3] << 8);

/* calulate 5A A5 and LEN filed crc */
crc16_update(&crc, buf, 4);

/* calulate payload crc */
crc16_update(&crc, buf + 6, payload_len);
```

The computed CRC is 0x61FD, which matches the CRC carried in the frame, so the CRC check passes.

- Step 3: Decode the payload

`90 00`: ID packet; 0x90 is the tag; ID = 0x00.

`A0 55 02 3D 01 E2 02`: acceleration packet; 0xA0 is the tag. The 3-axis acceleration is:

X-axis = (int16_t)((0x02<<8) + 0x55) = 597 (unit: mG)

Y-axis = (int16_t)((0x01<<8) + 0x3D) = 317

Z-axis = (int16_t)((0x02<<8) + 0xE2) = 738

`B0 FE FF 17 00 44 00`: angular-rate packet; 0xB0 is the tag. The 3-axis angular rate is:

X-axis = (int16_t)((0xFF<<8) + 0xFE) = -2 (unit: 0.1°/s)

Y-axis = (int16_t)((0x00<<8) + 0x17) = 23

Z-axis = (int16_t)((0x00<<8) + 0x44) = 68

`C0 80 FF 60 FF 32 FF`: magnetic-field packet; 0xC0 is the tag. The 3-axis magnetic field is:

X-axis = (int16_t)((0xFF<<8) + 0x80) = -128 (unit: 0.001Gauss)

Y-axis = (int16_t)((0xFF<<8) + 0x60) = -160

Z-axis = (int16_t)((0xFF<<8) + 0x32) = -206

`D0 64 F2 6C 0E BB 01`: Euler-angle packet; 0xD0 is the tag.

Pitch = (int16_t)((0xF2<<8) + 0x64) / 100 = -3484 / 100 = -34.84°

Roll = (int16_t)((0x0E<<8) + 0x6C) / 100 = 3692 / 100 = 36.92°

Yaw = (int16_t)((0x01<<8) + 0xBB) / 10 = 443 / 10 = 44.3°

`F0 00 00 00 00`: barometric pressure packet; 0xF0 is the tag.

```
float prs;
prs = memcpy(&prs, &buf[37], 4);
```

Final result:

```
id              : 0
acc(G)          :    0.597    0.317    0.738
gyr(deg/s)      :   -0.200    2.300    6.800
mag(uT)         :  -12.800  -16.000  -20.600
eul(R/P/Y)      :   36.920  -34.840   44.300
```

11.2.2. Frame configured with packet `0x91`

Capture one frame using a serial assistant. The frame is 82 bytes total. The first 6 bytes are the header, length, and CRC. The remaining 76 bytes are the payload. Assume the received bytes are stored in a C array `buf`:

5A A5 4C 00 6C 51 **91** 00 A0 3B 01 A8 02 97 BD BB 04 00 9C A0 65 3E A2 26 45 3F 5C E7 30 3F E2 D4 5A C2 E5 9D A0 C1 EB 23 EE C2 78 77 99 41 AB AA D1 C1 AB 2A 0A C2 8D E1 42 42 8F 1D A8 C1 1E 0C 36 C2 E6 E5 5A 3F C1 94 9E 3E B8 C0 9E BE BE DF 8D BE

- Step 1: Check the header and obtain the payload length and frame CRC:

Header: `5A A5`

Payload length: `4C 00`: (0x00<<8) + 0x4C = 76

Frame CRC: `6C 51`: (0x51<<8) + 0x6C = 0x516C

- Step 2: Verify CRC

```
    uint16_t payload_len;
    uint16_t crc;

    crc = 0;
    payload_len = buf[2] + (buf[3] << 8);

    /* calulate 5A A5 and LEN filed crc */
    crc16_update(&crc, buf, 4);

    /* calulate payload crc */
    crc16_update(&crc, buf + 6, payload_len);
```

The computed CRC is 0x516C, so the CRC check passes.

- Step 3: Decode the payload

The payload starts at `0x91`. In C, you can define a struct to parse the data conveniently.

Define the 0x91 packet struct:

```c
__packed typedef struct
{
    uint8_t     tag;                    /* tag: 0x91 */
    uint8_t     id;                     /* module ID */
    uint8_t     rev[2];
    float       prs;                    /* pressure */
    uint32_t    ts;                     /* timestamp */
    float       acc[3];                 /* acceleration */
    float       gyr[3];                 /* angular rate */
    float       mag[3];                 /* magnetometer */
    float       eul[3];                 /* Euler angles: Roll,Pitch,Yaw */
    float       quat[4];                /* quaternion */
}id0x91_t;
```

`__packed` is a compiler keyword (Keil) indicating byte-packed alignment, so each struct member maps 1:1 to the 0x91 packet layout. During reception, you can memcpy the received array directly into the struct. (Note: when defining the struct, it must be 4-byte aligned.) Here `buf` points to the frame header, and `buf[6]` points to the payload.

```c
    /* Receive and interpret data using the 0x91 packet definition */
    __align(4) id0x91_t dat;     /* struct must be 4 byte aligned */
    memcpy(&dat, &buf[6], sizeof(id0x91_t));
```

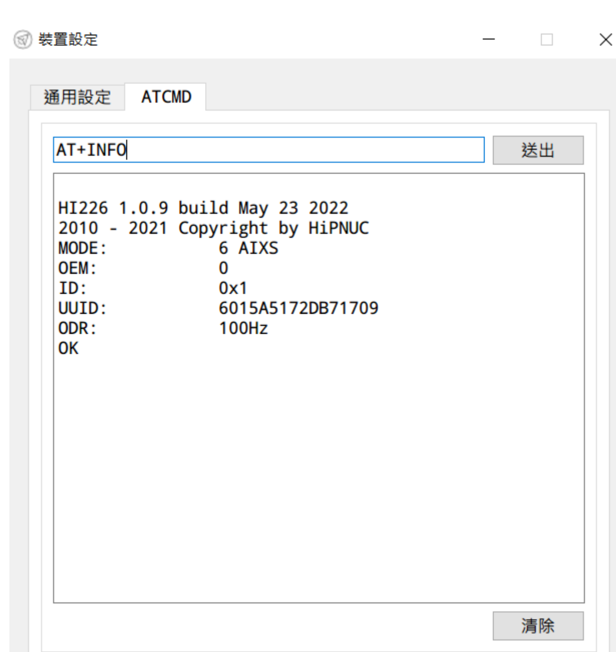
The decoded `dat` result:

```
id              : 0
timestamp       : 310205
acc             :    0.224    0.770    0.691
gyr             :  -54.708  -20.077 -119.070
mag             :   19.183  -26.208  -34.542
eul(R/P/Y)      :   48.720  -21.014  -45.512
quat            :    0.855    0.310   -0.310   -0.277
```

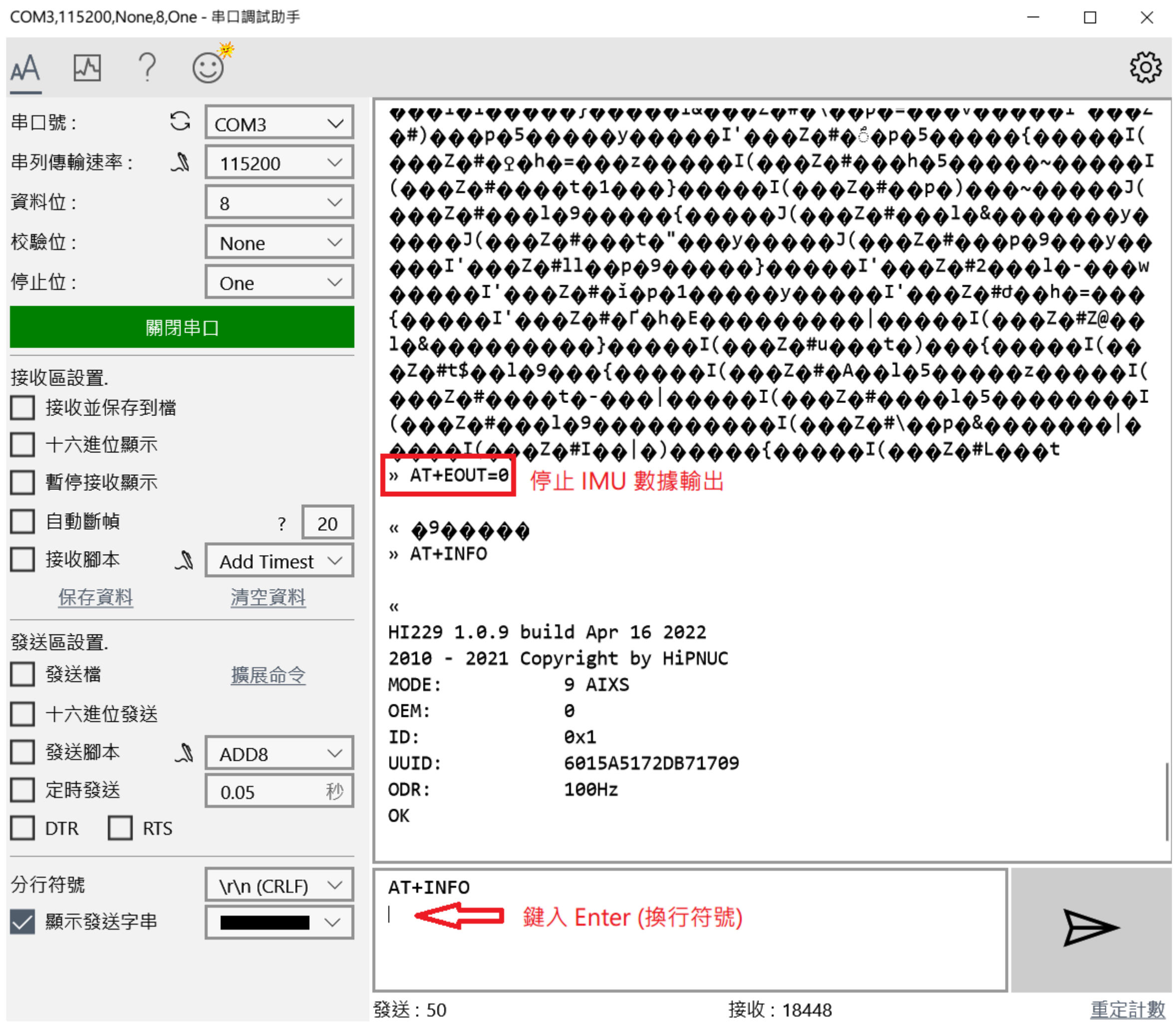
12. AT Commands

When communicating with the module via UART, the module supports an AT command set for configuring/viewing parameters. An AT command always starts with the ASCII string `AT`, followed by control characters, and ends with CRLF `\r\n`.

Entering AT commands from the PC software:



Testing with a serial debugging assistant:



Common AT commands are listed below:

Command	Function	Saved on power-off (Y) / Not saved (N)	Takes effect immediately (Y) / After reset (R)	Notes
AT+ID	Set module user ID	Y	R	
AT+INFO	Print module information	N	Y	
AT+ODR	Set UART output frame rate	Y	R	
AT+BAUD	Set UART baud rate	Y	R	
AT+EOUT	Data output enable/disable	N	Y	
AT+RST	Reset the module	N	Y	
AT+TRG	Single-shot output trigger	N	Y	Supported on some variants
AT+SETPTL	Set output data packets	Y	Y	Supported on some variants
AT+MODE	Set operating mode	Y	R	Supported on some variants
AT+GWID	Set wireless gateway ID	Y	R	Supported on some variants

AT+ID

Set module user ID

Example `AT+ID=1`

AT+INFO

Print module information, including model, version, firmware release date, etc.

AT+ODR

Set UART output data rate. Saved across power cycles; takes effect after module reset.

Example Set output rate to 100Hz: `AT+ODR=100`

Note: When ODR is high (e.g., 200), the default baud rate 115200 may not provide enough bandwidth. In that case, increase the module baud rate (e.g., 921600) so the module can output data frames at the configured ODR. Supported ODR values: 1,2,5,10,20,50,100,200,400Hz.

AT+BAUD

Set UART baud rate. Selectable values: `9600/115200/460800/921600`

Example `AT+BAUD=115200`

Note

- Use this command with extra care. An incorrect baud rate may prevent communication with the module.
- The baud rate is saved across power cycles and takes effect after module reset. The host baud rate must also be updated accordingly.
- During firmware upgrade, switch back to 115200 baud.

AT+EOUT

UART output enable/disable

Example Enable output: `AT+EOUT=1` Disable output: `AT+EOUT=0`

AT+RST

Reset the module

Example `AT+RST`

AT+SETYAW

Set heading (Yaw). Format: `AT+SETYAW=<MODE>,<VAL>`

- MODE=0 Absolute mode: set heading directly to VAL. Example: `AT+SETYAW=0,90` sets heading to 90°.
- MODE=1 Relative mode: add VAL to the current heading. Example: `AT+SETYAW=1,-10.5` decreases heading by 10.5°. If the original heading is 30°, it becomes 19.5° after the command.

AT+MODE

Set module operating mode

Examples

- Set 6-axis mode (no magnetometer correction): `AT+MODE=0`
- Set 9-axis mode (magnetometer participates in heading correction): `AT+MODE=1`

AT+SETPTL

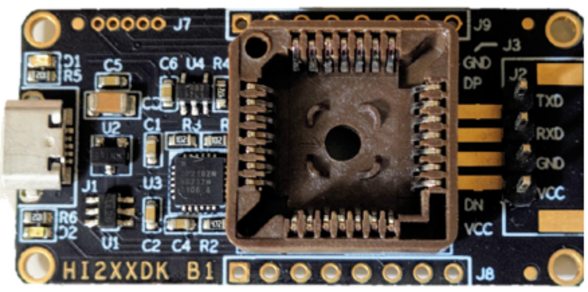
Set output protocol:

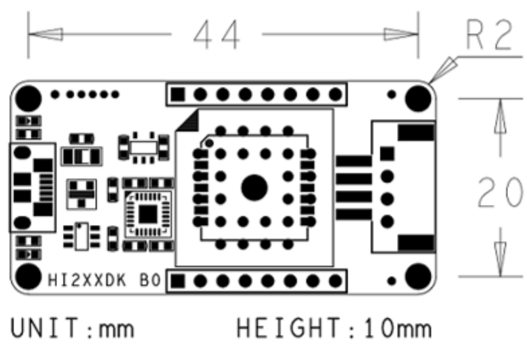
Configure which packets are included in a frame. Format: `AT+SETPTL=<ITEM_ID>,<ITEM_ID>...`

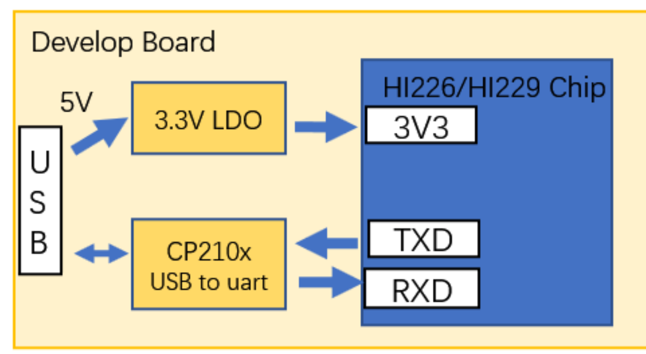
Examples

- Configure the module to output packet 91 (IMUSOL): `AT+SETPTL=91`
- Configure the module to output acceleration (A0), angular rate (B0), integer-format Euler angles (D0), and quaternion (D1): `AT+SETPTL=A0,B0,D0,D1`

13. Appendix A - Evaluation Board







13.1. Overview

Provides power supply and USB-to-UART, enabling customers to quickly evaluate the product.

The package includes the CP2104 USB-UART driver. Connect the USB cable between the PC and the module, open the PC software in the package, connect to the serial port, and in the default state the module outputs the factory default packet(s) at 115200 baud.

13.2. Removing the Module from the Evaluation Board

By default, the module is inserted in the evaluation board's PLCC28 socket. To remove the module, follow these steps:

- Power off and prepare a small screwdriver or tweezers.
- Pry the module out from the PLCC socket, or push it out from the round opening on the back side.

Note

- The evaluation board is intended for quick verification and evaluation of module performance. It does not provide any additional computing functions.
- The USB interface is not suitable for industrial-grade applications or high-motion scenarios. If your application involves high motion (motion capture, etc.), **it is not recommended to directly use the evaluation board in your product.**

14. Appendix B - Firmware Upgrade and Restore Factory Settings

This product supports firmware upgrades. Under normal usage, customers do not need to upgrade firmware themselves.
Firmware upgrade procedure:

- Connect the module, open the PC software, set both the module baud rate and the PC baud rate to 115200, and open the firmware update window.
- Click the connect button. If module connection information is displayed, the upgrade system is ready. Click "Open" to select a firmware file with the `.hex` extension, then click "Write".
- After completion, a success message will appear. Close the serial port and power-cycle the module. Firmware upgrade is complete.