

# Receiving Data

표준 UART 프로토콜은 모든  $\mu$ Landing 제품의 기본 설정이지만 CAN 및 I<sup>2</sup>C 프로파일도 제공합니다. 시스템에 대한 프로토콜 또는 기타 요구 사항의 변경을 요청하려면 당사에 문의하십시오.

# UART Data Format

이것은 uLanding's UART protocol의 최신 버전 입니다.

우리의 직렬 시리얼 통신 인터페이스는 표준 UART 프로토콜을 사용합니다.

사용자의 특정요구사항에 따라 protocol을 수정할 수 있습니다.

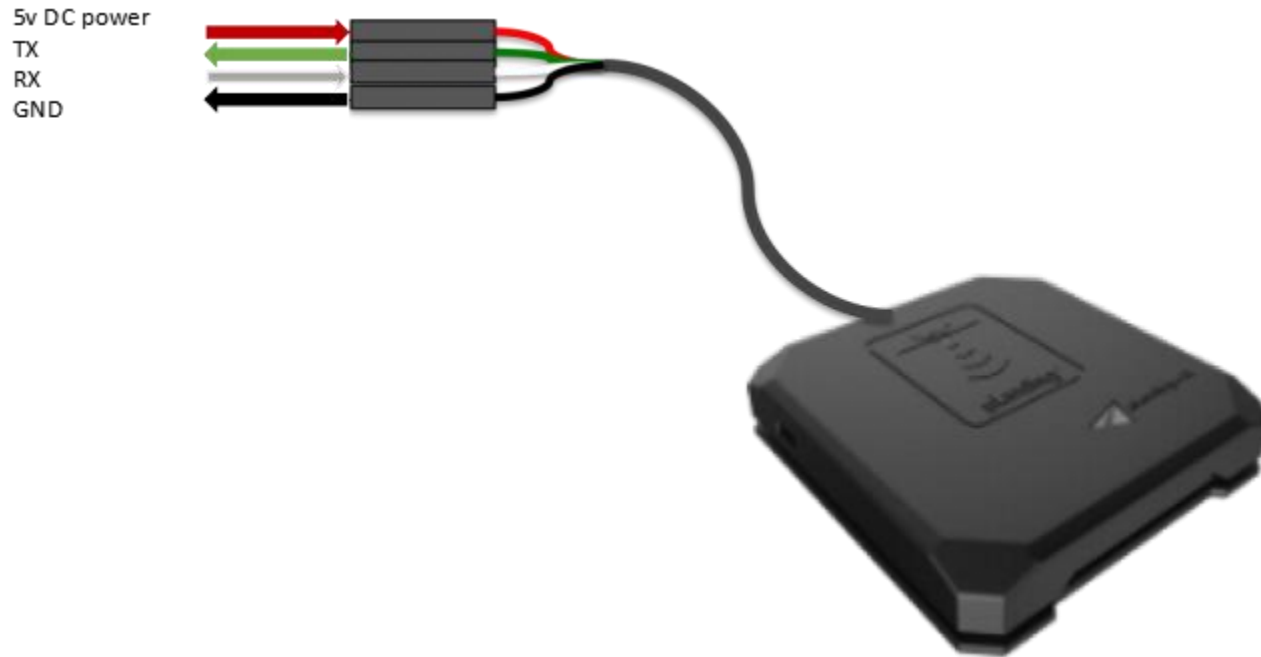
프로토콜 또는 시스템에 관한 기타 요구 사항의 변경을 요청하려면 aerotenna에 문의하십시오.

## 통신 프로토콜

- 표준 UART
- Baud rate 는 115200 b/s
- 데이터 길이는 8bit에다가 Start bit 1개 Stop bit 1개 그리고 parity bit는 없습니다.
- I/O 표준은 3.3v 레벨 TTL 입니다.

# Connection 인터페이스

UART 데이터 포맷에서 초록색 단자는 TX핀 하얀색 단자는 RX핀 빨간색 단자는 Power핀 검정색 단자는 GND핀 입니다.



# 직렬 전송 인터페이스의 데이터 형식

출력 데이터는 6바이트로 구성됩니다. 바이트 구조는 아래에 설명되어 있습니다. 'x'는 동적 데이터를 포함하는 가변비트입니다.

Byte 1	0xFE	Packet Header
Byte 2	0x01	Version ID
Byte 3	0bxxxxxxx	Altitude (Least Significant 8 Bits)
Byte 4	0bxxxxxxx	Altitude (Most Significant 8 Bits)
Byte 5	0bxxxxxxx	SNR (signal to noise ratio) this is helpful in analyzing data.
Byte 6	0bxxxxxxx	Checksum (see formula below)

# 설명

## Altitude

고도 바이트는 결합되어 (총 16 비트) 고도 정보를 센티미터로 나타낼 수 있습니다. 구조는 0x [MSB] [LSB]입니다. 여기서 MSB와 LSB는 각각 16 진수 (8 비트)입니다.

## SNR

SNR의 정의는 ' $10 * \log_{10} (\text{Power of Signal} / \text{Noise Floor})$ '이며 그 범위는 0 dB ~ 60 dB입니다. 고도 판독 값은 반사 정도를 나타내기 위해 SNR과 연관됩니다.

## Checksum

checksum 바이트는 다음과 같이 사용될 수 있습니다.

$\text{checksum} = (\text{Version\_ID} + \text{Altitude\_H} + \text{Altitude\_L} + \text{SNR}) \& 0xFF$

만약 checksum = 1,이면 check 성공

만약 checksum = 0,이면 check 실패

# Python 2.7 에서 데이터 수신/표시 예제코드

이 예제 스크립트를 실행하기 전에 필요한 Python 패키지를 다운로드하십시오.

```
1 from serial import Serial
2 import struct
3
4 serial = Serial('COM7', 115200, 8, 'N', 1)  ### This serial port name may be various on different machines
5 serial.flushInput()
6 serial.flushOutput()
7
8 while(1):
9     byte = struct.unpack('B', serial.read(1))[0]
10    if(byte == 0xFE):
11        byte2 = struct.unpack('B', serial.read(1))[0]
12        byte3 = struct.unpack('B', serial.read(1))[0]
13        byte4 = struct.unpack('B', serial.read(1))[0]
14        byte5 = struct.unpack('B', serial.read(1))[0]
15        byte6 = struct.unpack('B', serial.read(1))[0]
16        header = byte
17        version_id = byte2
18        lsb_byte = byte3
19        msb_byte = byte4
20        snr = byte5
21        check_sum_transmitter = byte6
22        check_sum_receiver = (byte2 + byte3 + byte4 + byte5) & 0xFF
23        ALTITUDE = byte3 + (byte4<<8)
24
25    ### Print received data
26    if (check_sum_receiver == check_sum_transmitter):
27        print "checksum passed, ALTITUDE: " + str(ALTITUDE) + ", SNR:" + str(snr) + "\r\n"
28    else:
29        print "checksum failed!!!" + "\r\n"
30    else:
31        continue
```



# Arduino 에서 데이터 수신 / 표시 예제코드

예제 스크립트를 실행하기 전에 Arduino IDE를 설치하십시오.

```
Serial Port Data

// This function is called only once, at reset.
void setup() {
  Serial.begin(115200);
  Serial.println("uLanding Readout");
}

// This function is called continuously, after setup() completes.
void loop() {
  int altitude = 0;
  int data[64];
  int data_packet[6];
  int idx;
  int receiver_checksum;
  int SNR;

  /// Buffer size = default 64 Bytes, this depends on setting from Arduino board
  while (Serial.available() == 63) {
    delay(20);
    for(int i=0;i<64;i++){
      data[i]=Serial.read();
    }

    for (int j=0;j<12;j++){
      if (data[10+j] == 0xFE && data[10+j+1] == 0x01){
        idx = j;
        break;
      }
    }
    data_packet[0] = data[10+idx];
    data_packet[1] = data[11+idx];
    data_packet[2] = data[12+idx];
    data_packet[3] = data[13+idx];
    data_packet[4] = data[14+idx];
    data_packet[5] = data[15+idx];

    altitude = data_packet[2] + (data_packet[3] << 8);
    SNR = data_packet[4];
    receiver_checksum = (data_packet[1] + data_packet[2] + data_packet[3] + data

    /// Print received data
    if (receiver_checksum == data_packet[5]){
      Serial.print("Checksum passed, ALTITUDE: ");
      Serial.print(altitude);
      Serial.print(", SNR: ");
      Serial.print(SNR);
      Serial.print("\n\r");
    }else{
      Serial.print("Checksum failed!!!");
      Serial.print("\n\r");
    }
  }
}
```

# CAN bus Data Format

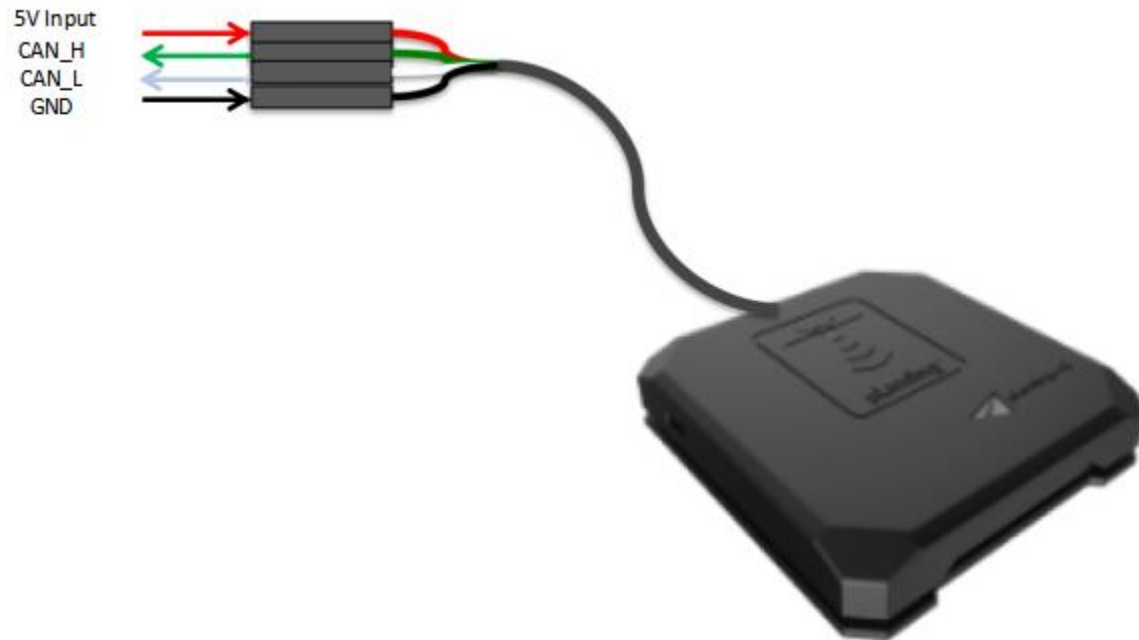
μLanding ™는 또한 95Hz의 속도로 업데이트된 CAN 버스 데이터 형식 버전이 있습니다. CAN 버스 프로토콜에 대한 문의사항은 Aerotenna에 문의하십시오.

## 통신 프로토콜

- 확장 데이터 프레임 형식
- Baud rate 는 250 Kb/s
- 데이터 길이는 8바이트 입니다.
- CAN ID 프레임은 **0x00090002**입니다.

# Connection 인터페이스

초록색 포트는 CAN\_H 이고 하얀색 포트는 CAN\_L이고 빨간색은 전원핀 검정색은 GND핀 입니다.



# Data Format

이 섹션에서는 CAN의 데이터 필드에 데이터 형식을 소개합니다. 데이터 필드는 8 바이트 (64 비트)로 구성됩니다. 'x'는 가변 비트를 나타냅니다.

Bytes No.	bits details	Description
1	0bxxxxxxxx	Altitude (Most Significant 8 bits)
2	0bxxxxxxxx	Altitude (Least Significant 8 bits)
3	0bxxxxxxxx	SNR (Most Significant 8 bits)
4	0bxxxxxxxx	SNR (Least Significant 8 bits)
5-8	0b00000000	N/A (Reserved)

# UART 프로토콜 데이터

UART 프로토콜에서 데이터 패키지가 수신되면 고도 데이터 메시지는 Byte 3 & Byte 4에서 얻을 수 있습니다. 참고로, 이는 센티미터로 표시됩니다.

# CAN 프로토콜 데이터

CAN 프로토콜에서 데이터 패키지가 수신되면 고도 데이터 메시지가 데이터 필드의 Byte 1 & Byte 2에서 얻어집니다. 센티미터 단위로 표시됩니다.



## Altimeter Tracker

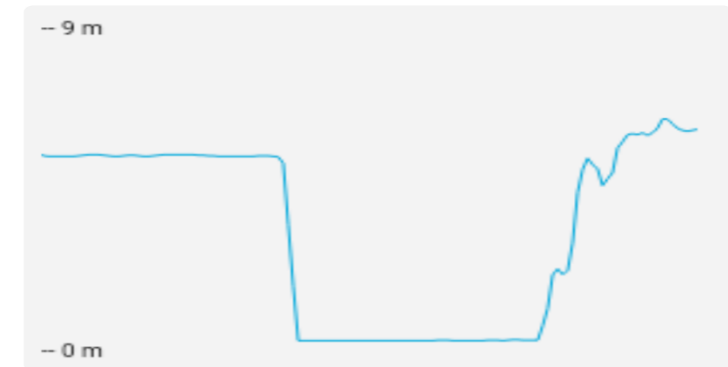
**SCAN**

## Altimeter Tracker

Altitude (meters)

**5.98**

Altitude Tracker



**DISCONNECT**

µLanding을 설정하기위한 추가 참고 자료가 필요하다면 다음 섹션을 확인하십시오.