

# RIT SPEX Hyperion Protocol

Dylan R. Wagner, Alec Herbert, Chris Johnson

February 1, 2018

## 1 Abstract

The Hyperion Protocol defines data packing techniques and configurations in order to achieve communication between endpoints. A universal header will be defined which allows senders the freedom to include a wide range of information from individual sensors or information groups. In addition, multiple other data configurations will be defined. These additional data configurations will store information regarding data groups from sensors without any notion of time captured.

## 2 Header

The header will be required for any outgoing and incoming data within the realm of the Hyperion protocol. This header will include information regarding the which data group (Sensor values) is contained in a message, flags and time sent. Any data wished to be sent with the Hyperion Protocol will be tacked on behind the header. The Hyperion Protocol header will be **24bits** long or **3 bytes**.

### 2.1 Header Data

Data contained in the header will have a maximum size, this size will be defined in bits and largest value stored.

**Data Group (4 bits)** The Data Group section will contain information to distinguish which data is contained after the header. A size of 4 bits will be allocated to this section, this allows for 16 separate values. This will be plenty for the number of data groups sent within the Hyperion system.

**Flags (4 bits)** The Flags section will be used to define how the data will be parsed or handled. A size of 4 bits will allow for 4 flags to be set at a time.

**Sent Time (16 bits)** The Sent Time section will be used to determine the time sent in seconds from the Hyperion system. This will be used for timestamps/data evolution analysis. This section will need to be 16 bits. This allows for a value of maximum value of 65535. This allows us to send data time stamped from startup for 18.2 hours.

## 2.2 Visualization

Table 1: Hyperion Header

1	2	Sent Time (16 bits)	1	Data Group (4 bits)
		Data	2	Flags (4 bits)

## 2.3 Header Values

This section displays mappings between header values and their meanings. In the header only two sections will need further definitions. These sections are the Data Group and Flags sections.

### 2.3.1 Data Group Mapping

Table 2: Data Group Mapping

Value	Mapping
0	No Data
1	LSM9DS1
2	BME280
3	CCS811
4	LIS331
5	PerfectFlite Strato Logger
6	Orientation
7	Commands
8	
9	
10	
11	
12	
13	
14	
15	

## 3 Data Groups

### 3.1 LSM9DS1

This section will define the data frame for the LSM9DS1 data group. The LSM9DS1 data group contains information regarding acceleration, gyroscope and magnetometer sensors in the X, Y and Z axis. In total, this data frame will be 36 bytes in size.

#### 3.1.1 LSM9DS1 Data

**Acceleration X, Y, Z (12 bytes - 96 bits)** The acceleration section of the LSM9DS1 data frame will store the X, Y, Z values. Each axis is 4 bytes or 32 bits long. The X, Y, Z values will be placed in that order.

**Gyroscope X, Y, Z (12 bytes - 96 bits)** The gyroscope section of the LSM9DS1 data frame will store the X, Y, Z values. Each axis is 4 bytes or 32 bits long. The X, Y, Z values will be placed in that order.

**Magnetometer X, Y, Z (12 bytes - 96 bits)** The magnetometer section of the LSM9DS1 data frame will store the X, Y, Z values. Each axis is 4 bytes or 32 bits long. The X, Y, Z values will be placed in that order.

#### 3.1.2 Visualization

Table 3: LSM9DS1 Data frame

ax (4 bytes)	ay (4 bytes)	az (4 bytes)
gx (4 bytes)	gy (4 bytes)	gz (4 bytes)
mx (4 bytes)	my (4 bytes)	mz (4 bytes)

### 3.2 BME280

This section will define the data frame for the BME280 data group. The BME280 data group contains information regarding atmospheric temperature, pressure and humidity. In total, this data frame will be 8 bytes in size.

#### 3.2.1 BME280 Data

**Temperature (2 bytes)** The temperature section of the BME280 data frame will store the atmospheric temperature in degrees Celsius. In total, the temperature section will be 2 bytes.

**Pressure (2 bytes)** The pressure section of the BME280 data frame will store the atmospheric pressure in kPa. In total, the pressure section will be 2 bytes.

**Humidity (2 bytes)** The humidity section of the BME280 data frame will store the atmospheric humidity by percentage. In total, the humidity section will be 4 bytes.

**BME\_Altitude (2 bytes)** The Altitude section of the BME280 data frame will store the altitude derived by pressure. In total, the BME\_Altitude section will be 2 bytes.

#### 3.2.2 Visualization

Table 4: BME280 Data Group

Temperature (2 bytes)	Pressure (2 bytes)
Humidity (2 bytes)	BME_Altitude (2 bytes)

### 3.3 CCS811

This section will define the data frame for the the CCS811 data group. The CCS811 data group contains information regarding atmospheric composition. In, total, this data frame will be 4 bytes in size.

#### 3.3.1 CCS811 Data

**Total Volatile Organic Compounds (2 Bytes)** The Total Volatile Organic Compounds section of the CCS811 data frame will store the total volatile organic compounds in ppm. In total, this section will be 2 bytes.

**Equivalent Carbon Dioxide (2 Bytes)** The Equivalent Carbon Dioxide section of the CCS811 data frame will store the equivalent carbon dioxide in ppm. In total, this section will be 2 bytes in size.

#### 3.3.2 Visualization

Table 5: CCS811 Data Frame

VOC (2 bytes)	ECD (2 bytes)
---------------	---------------

### 3.4 LIS331

This section will define the data frame for the the LIS331 data group. The LIS331 data group contains information acceleration in Gees. In, total, this data frame will be 12 bytes in size.

#### 3.4.1 LIS331 Data

**X, Y, Z Acceleration (12 bytes)** The Acceleration section of the LIS331 data frame will store the acceleration sensor readings in Gees. The X, Y, Z parts of this section will be 4 bytes each. In total, this data section will be 12 bytes.

#### 3.4.2 Visualization

Table 6: LIS331 Data Frame

Acceleration (X)	Acceleration (Y)	Acceleration (Z)
------------------	------------------	------------------

### 3.5 PerfectFlite Strato Logger

This section will define the data frame for the Strato Logger data group. The Strato Logger data group provides accurate information regarding altitude. In total, this data group will be 4 bytes in size.

#### 3.5.1 Strato Logger Data

**Altitude (4 bytes)** The Altitude section of the Strato Logger data group will store the altitude in meters. In total, this data section will be 4 bytes in size.

#### 3.5.2 Visualization

Table 7: PerfectFlite Strato Logger Data Group

Altitude (2 bytes)
--------------------

### 3.6 Orientation

This section will define the data frame for the Orientation data group. The Orientation data group will contain calculations for Roll, Pitch and Yaw. In total, this data group will be 4 bytes in size. Values will be 9 bits in length with the signed bits being stored within the first byte.

#### 3.6.1 Orientation Data

**Roll (9 bits)** The Roll section of the Orientation data group will store the Roll from range -180 to 180 (0 to 360 degrees).

**Pitch (9 bits)** The Pitch section of the Orientation data group will store the Pitch from range -90 to 90 (0 to 180 degrees).

**Yaw (9 bits)** The Yaw section of the Orientation data group will store the Yaw from range -180 to 180 (0 to 360 degrees).

#### 3.6.2 Visualization

Table 8: Orientation Data Group

Signed(1 byte)	Roll(1 byte)	Pitch(1 byte)	Yaw(1 byte)
----------------	--------------	---------------	-------------



### 3.7 Commands/Command Info

This section will outline the header configurations for separate commands to and from the payload. This Group will hold no data, thus this data frame is 0 bytes in size. Header flags will be used in place of a data frame.

#### 3.7.1 Flags

Table 9: Flag Meanings

Bit Pos(Most significant)	Meaning
0	Arm Payload
1	Disarm Payload
2	Payload Armed
3	Payload Disarmed

### **3.8 Info Messages**

This section will outline the anatomy of a info message data frame. This data frame has no defined length, only a length of greater than 3 bytes.

#### **3.8.1 Data**

Data attached to this frame will be stored in the form of a string. This string has no defined length. However it will be less than the maximum length of transmitted packets defined by the communication device.