

# MSP Protocol

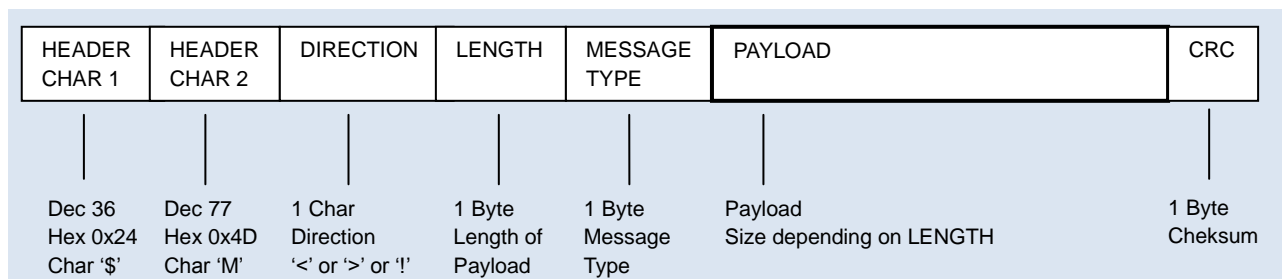
## MSP Protocol Key Features

MultiWii Serial Protocol has the following key features:

- **Compact** - uses 8 Bit Binary Data.
- **Generic** - it can be used transparently by a GUI, OSD, telemetry or homemade configuration tools, i.e. no more specific OSD code should be coded in MultiWii
- **Checksum Protected** - data are sent with a checksum, preventing corrupted configuration to be injected.
- **Header Sensitive** - as it is designed with a specific header, it can be mixed with other frame, like GPS frame, i.e. it will be possible to connect either a GUI or a GPS on the same serial port without changing the configuration.

## MSP Packet Structure

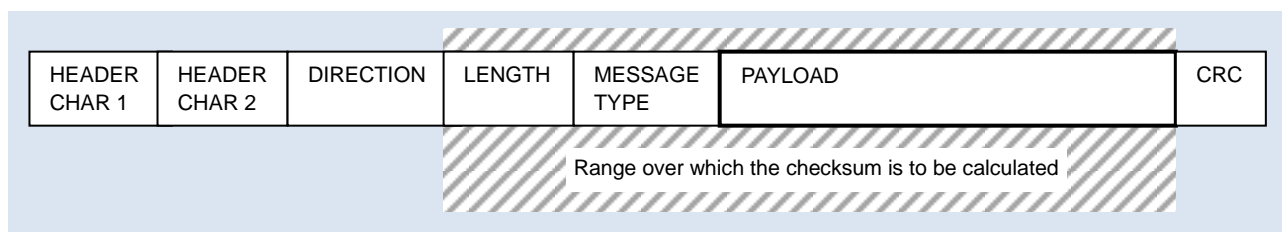
A basic MSP Packet looks as follows:



- Every Message starts with 2 Chars: '\$' and 'M'
- A 1 Char Directions Field follows. This field defines the direction of the message.
- A 1 Byte Length Field is following. Length is defined as being the length of the payload, only. It does not include Header Chars, Direction, Length, Message Type and CRC fields. May be 0 to 255. **The MultiWii software version 2.1 cannot process more than 64 bytes.**
- The Payload is a variable length field, may be empty, if the Length field is zero.
- CRC is a 8 Bit checksum whose calculation is defined below.

## MSP Checksum

The checksum is calculated over the packet, starting and including the LENGTH field, up until, but excluding, the Checksum Field:



The checksum calculated used byte to byte XOR function.

## MSP Message Flow

The main rule remains: MultiWii never sends something on its own. A request must be done in each case to retrieve or set data. Each messages received are acknowledged even if there is no data inside.

Even in case of CRC error or bad message format, MultiWii do not sends something to back.

## Request message to MultiWii

To request simple data without parameters / send a specific command / inject new parameters in MultiWii you must send a messages with Direction Field contained ASCII char '<' (Dec 60, Hex 0x3C).

## MultiWii output message

This message sent as a result of a Request Message. When MultiWii was received correct Request Message, it will be processed and MultiWii send back a messages with Direction Field contained ASCII char '>' (Dec 62, Hex 0x3E), same Message Type field as in Request Message, zero Length field and no Payload.

## Unknown message type

This message sent as a result of a Request Message with unknown Message Type. When MultiWii was received correct Request Message but with unknown Message Type field, MultiWii send back a messages with Direction Field contained ASCII char '!' (Dec 33, Hex 0x21). same Message Type field as in Request Message, zero Length field and no Payload.

## MSP Messages Overview

Message Type	Code	Type	Description
<a href="#">MSP_IDENT</a>	100	GET	MultiWii version, multitype, protocol version, capability variable
<a href="#">MSP_STATUS</a>	101	GET	cycletime, errors_count, sensor present, box activation, current setting number
<a href="#">MSP_RAW_IMU</a>	102	GET	9 DOF
MSP_SERVO	103	GET	8 servos
MSP_MOTOR	104	GET	8 motors
MSP_RC	105	GET	8 rc chan
MSP_RAW_GPS	106	GET	fix, numsat, lat, lon, alt, speed, ground course
MSP_COMP_GPS	107	GET	distance home, direction home, GPS_update
MSP_ATTITUDE	108	GET	2 angles, 1 heading, headFreeModeHold
MSP_ALTITUDE	109	GET	altitude, variometer
<a href="#">MSP_BAT</a>	110	GET	vbat, powermetersum
MSP_RC_TUNING	111	GET	rc rate, rc expo, rollpitch rate, yaw rate, dyn throttle PID, throttle mid, throttle expo
MSP_PID	112	GET	up to 16 P I D (8 are used)
MSP_BOX	113	GET	up to 16 checkbox (11 are used)
MSP_MISC	114	GET	powermeter trig + 8 free for future use
MSP_MOTOR_PINS	115	GET	which pins are in use for motors & servos, for GUI
MSP_BOXNAMES	116	GET	the aux switch names (ANGLE;HORIZON;BARO;....)
MSP_PIDNAMES	117	GET	the PID names (ROLL;PITCH;YAW;....)
MSP_WP	118	GET	get a WP, WP# is in the payload, returns (WP#, lat, lon, alt, flags) WP#0-home, WP#16-poshold
MSP_SET_RAW_RC	200	SET	8 rc chan (8x2)
MSP_SET_RAW_GPS	201	SET	fix, numsat, lat, lon, alt, speed
MSP_SET_PID	202	SET	up to 16 P I D (8 are used)
MSP_SET_BOX	203	SET	up to 16 checkbox (11 are used)
MSP_SET_RC_TUNING	204	SET	rc rate, rc expo, rollpitch rate, yaw rate, dyn throttle PID
MSP_ACC_CALIBRATION	205	SET	no param
MSP_MAG_CALIBRATION	206	SET	no param
MSP_SET_MISC	207	SET	powermeter trig + 8 free for future use
MSP_RESET_CONF	208	SET	no param
MSP_WP_SET	209	SET	sets a given WP (WP#,lat, lon, alt, flags)
MSP_SELECT_SETTING	210	SET	Select Setting Number (0-2)
MSP_SPEK_BIND	240	SET	
MSP_EEPROM_WRITE	250	SET	
MSP_DEBUGMSG	253	GET	debug string buffer
MSP_DEBUG	254	GET	debug1,debug2,debug3,debug4

## MSP\_IDENT (100)

<b>Message</b>	MSP_IDENT			
<b>Description</b>	Request MultiWii SW version, multirotor type, protocol version, capability variable			
<b>Type</b>	Get			
<b>Comment</b>	No payload			
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'<'	Direction		Dec 60, Hex 0x3C
3	0	Length		No payload
4	100	Message Code		MSP_IDENT, Dec 100, Hex 0x64
5	100	CRC		Dec 100, Hex 0x64

<b>Message</b>	MSP_IDENT			
<b>Description</b>	Reply MultiWii SW version, multirotor type, protocol version, capability variable			
<b>Type</b>	Get			
<b>Comment</b>				
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'>'	Direction		Dec 62, Hex 0x3E
3	7	Length		Length of payload
4	100	Message Code		MSP_IDENT
5	Byte	VERSION		MultiWii software version
6	Byte	MULTIYPE		Multirotor type. See Multitype definition table
7	Byte	MSP_VERSION		MultiWii serial protocol version
8	DWord	Capability		Capability
12	Byte	CRC		Depends on the data marked in blue

## Multitype definition table

Multitype	Symbolic name	Description
1	TRI	Tricopter
2	QUADP	Quadrocopter +
3	QUADX	Quadrocopter X
4	BI	
5	GIMBAL	
6	Y6	
7	HEX6	
8	FLYING_WING	
9	Y4	
10	HEX6X	
11	OCTOX8	Java GUI is same for the next 3 configs
12	OCTOFLATP	MultiWinGui shows this differently
13	OCTOFLATX	MultiWinGui shows this differently
14	AIRPLANE SINGLECOPTER DUALCOPTER	airplane / singlecopter / dualcopter (not yet properly supported)
15	HELI_120_CCPM	
16	HELI_90_DEG	
17	VTAIL4	
18	HEX6H	

## MSP\_STATUS (101)

<b>Message</b>	MSP_STATUS			
<b>Description</b>	Request cyclotime, errors_count, sensor present, box activation, current setting number			
<b>Type</b>	Get			
<b>Comment</b>	No payload			
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'<'	Direction		Dec 60, Hex 0x3C
3	0	Length		No payload
4	101	Message Code		MSP_STATUS, Dec 101, Hex 0x65
5	101	CRC		Dec 101, Hex 0x65

<b>Message</b>	MSP_STATUS			
<b>Description</b>	Reply cyclotime, errors_count, sensor present, box activation, current setting number			
<b>Type</b>	Get			
<b>Comment</b>				
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'>'	Direction		Dec 62, Hex 0x3E
3	11	Length		Length of payload
4	101	Message Code		MSP_STATUS, Dec 101, Hex 0x65
5	Word	cycleTime	microsecond	IMU loop spent time in microseconds
7	Word	i2c_errors_count		I2C bus error counter
9	Word	sensors		Sensors presents on the bord
11	DWord	active_box		Active box
15	Byte	currentSet		Current setting profile number
16	Byte	CRC		Depends on the data marked in blue

## MSP\_RAW\_IMU (102)

<b>Message</b>	MSP_RAW_IMU			
<b>Description</b>	Request raw sensors data			
<b>Type</b>	Get			
<b>Comment</b>	No payload			
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'<'	Direction		Dec 60, Hex 0x3C
3	0	Length		No payload
4	102	Message Code		MSP_RAW_IMU, Dec 102, Hex 0x66
5	102	CRC		Dec 102, Hex 0x66

<b>Message</b>	MSP_RAW_IMU			
<b>Description</b>	Reply raw sensors data			
<b>Type</b>	Get			
<b>Comment</b>				
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'>'	Direction		Dec 62, Hex 0x3E

3	18	Length		Length of payload
4	102	Message Code		MSP_RAW_IMU, Dec 102, Hex 0x66
5	Word	accSmooth[X]		Accelerometer data X axis (Roll)
7	Word	accSmooth [Y]		Accelerometer data Y axis (Pitch)
9	Word	accSmooth [Z]		Accelerometer data Z axis (Yaw)
11	Word	gyroData[X]		Gyroscope data X axis (Roll)
13	Word	gyroData[Y]		Gyroscope data X axis (Pitch)
15	Word	gyroData[Z]		Gyroscope data X axis (Yaw)
17	Word	magADC [X]		Magnetometer data X axis (Roll)
19	Word	magADC [Y]		Magnetometer data X axis (Pitch)
21	Word	magADC [Z]		Magnetometer data X axis (Yaw)
23	Byte	CRC		Depends on the data marked in blue

## MSP\_BAT (110)

<b>Message</b>		MSP_BAT		
<b>Description</b>		Request vbat, powermetersum		
<b>Type</b>		Get		
<b>Comment</b>		No payload		
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'<'	Direction		Dec 60, Hex 0x3C
3	0	Length		No payload
4	110	Message Code		MSP_BAT
5	110	CRC		Dec 110, Hex 0x6E

<b>Message</b>		MSP_BAT		
<b>Description</b>		Reply vbat, powermetersum		
<b>Type</b>		Get		
<b>Comment</b>				
Offset	Data	Name	Unit	Description
0	'\$'	Header char 1		Dec 36, Hex 0x24
1	'M'	Header char 2		Dec 77, Hex 0x4D
2	'>'	Direction		Dec 62, Hex 0x3E
3	3	Length		Length of payload
4	110	Message Code		MSP_BAT
5	Byte	Vbat	0.1 V	result is Vbatt in 0.1V steps
6	Word	Powermetersum		power meter trash
8	Byte	CRC		Depends on the data marked in blue