Multiwii Serial Protocol

From MultiWii

A discussion on the MSP is in this thread: http://www.multiwii.com/forum/viewtopic.php?f=8&t=1516

The general format of an MSP message is:

<preamble>,<direction>,<size>,<command>,,<crc>

Where:

preamble = the ASCII characters '$M'

direction = the ASCII character '<' if to the MWC or '>' if from the MWC

size = number of data bytes, binary. Can be zero as in the case of a data request to the MWC

command = message_id as per the table below

data = as per the table below. UINT16 values are LSB first.

crc = XOR of <size>, <command> and each data byte into a zero'ed sum

command	message_id	direction	data	type	comment
MSP_IDENT	100	FC →	VERSION	UINT 8	version of MultiWii
			MULTITYPE	UINT 8	type of multi: TRI/QUADP,QUADX,BI,GIMBAL,Y6,HEX6,FLYING_WING,Y4,HEX6X,OCTOX8, OCTOFLATP,OCTOFLATX,AIRPLANE,HELI_120,HELI_90,VTAIL4,HEX6H,SINGLECOPTER,DUALCOPTER
			MSP_VERSION	UINT 8	not used currently
			capability	UINT 32	A 32 bit variable to indicate capability of FC board. Currently, BIND button is used on first bit, DYNBAL on second, FLAP on third
MSP_STATUS	101	FC →	cycleTime	UINT 16	unit: microseconds
			i2c_errors_count	UINT 16	
			sensor	UINT 16	BARO<<1|MAG<<2|GPS<<3|SONAR<<4
			flag	UINT 32	a bit variable to indicate which BOX are active, the bit position depends on the BOX which are configured
			global_conf.currentSet	UINT 8	to indicate the current configuration setting
MSP_RAW_IMU	102	FC →	accx	INT 16	unit: it depends on ACC sensor and is based on ACC_1G definition MMA7455 64 / MMA8451Q 512 / ADXL345 265 / BMA180 255 / BMA020 63 / NUNCHUCK 200 / LIS3LV02 256 / LSM303DLx_ACC 256 / MPU6050 512 / LSM330 256
			accy	INT 16	
			accz	INT 16	
			gyrx	INT 16	unit: it depends on GYRO sensor. For MPU6050, 1 unit = 1/4.096 deg/s
			gyry	INT 16	
			gyrz	INT 16	
			magx	INT 16	unit: it depends on MAG sensor.
			magy	INT 16	
			magz	INT 16	
MSP_SERVO	103	FC →	Servo*8	16 x UINT 16	Range [1000;2000] The servo order depends on multi type
MSP_MOTOR	104	FC →	Motor*8	16 x UINT 16	Range [1000;2000] The motor order depends on multi type
MSP_SET_MOTOR	214	→ FC	Motor*8	16 x UINT 16	use to set individual motor value (to be used only with DYNBALANCE config)
MSP_RC	105	FC →	rcData[RC_CHANS]	16 x UINT 16	Range [1000;2000] ROLL/PITCH/YAW/THROTTLE/AUX1/AUX2/AUX3AUX4
MSP_SET_RAW_RC	200	→ FC	rcData[RC_CHANS]	16 x UINT 16	Range [1000;2000] ROLL/PITCH/YAW/THROTTLE/AUX1/AUX2/AUX3AUX4 This request is used to inject RC channel via MSP. Each chan overrides legacy RX as long as it is refreshed at least every second. See UART radio projects for more details.
MSP_RAW_GPS	106	FC →	GPS_FIX	UINT 8	0 or 1
			GPS_numSat	UINT 8	
			GPS_coord[LAT]	UINT 32	1 / 10 000 000 deg
			GPS_coord[LON]	UINT 32	1 / 10 000 000 deg
			GPS_altitude	UINT 16	meter
			GPS_speed	UINT 16	cm/s
			GPS_ground_course	UINT 16	unit: degree*10
MSP_SET_RAW_GPS	201	→ FC	GPS_FIX	UINT 8	this request is used to inject GPS data (annex GPS device or simulation purpose)
			GPS_numSat	UINT 8	
			GPS_coord[LAT]	UINT 32	1 / 10 000 000 deg
			GPS_coord[LON]	UINT 32	1 / 10 000 000 deg
			GPS_altitude	UINT 16	meter
			GPS_speed	UINT 16	cm/s
MSP_COMP_GPS	107	FC →	GPS_distanceToHome	UINT 16	unit: meter
			GPS_directionToHome	UINT 16	unit: degree (range [-180;+180])
			GPS_update	UINT 8	a flag to indicate when a new GPS frame is received (the GPS fix is not dependent of this)
MSP_ATTITUDE	108	FC →	angx	INT 16	Range [-1800;1800] (unit: 1/10 degree)
			angy	INT 16	Range [-900;900] (unit: 1/10 degree)
			heading	INT 16	Range [-180;180]
MSP_ALTITUDE	109	FC →	EstAlt	INT 32	cm
			vario	INT 16	cm/s
MSP_ANALOG	110	FC →	vbat	UINT 8	unit: 1/10 volt
			intPowerMeterSum	UINT 16	
			rssi	UINT 16	range: [0;1023]
			amperage	UINT 16	
MSP_RC_TUNING	111	FC →	byteRC_RATE	UINT 8	range [0;100]
			byteRC_EXPO	UINT 8	range [0;100]
			byteRollPitchRate	UINT 8	range [0;100]
			byteYawRate	UINT 8	range [0;100]
			byteDynThrPID	UINT 8	range [0;100]
			byteThrottle_MID	UINT 8	range [0;100]
			byteThrottle_EXPO	UINT 8	range [0;100]
MSP_SET_RC_TUNING	204	→ FC	byteRC_RATE	UINT 8	
			byteRC_EXPO	UINT 8	

			byteRollPitchRate	UINT 8	
			byteYawRate	UINT 8	
			byteDynThrPID	UINT 8	
			byteThrottle_MID	UINT 8	
			byteThrottle_EXPO	UINT 8	
MSP_PID	112	FC →	PIDITEMS x conf.pid[]	3*PIDITEMS x UINT 8	Currently, PIDITEMS is constant = 10 Order : ROLL / PITCH / YAW / ALT / POS / POSR / NAVR / LEVEL /MAG / VEL VEL is not used
MSP_SET_PID	202	→ FC	PIDITEMS x conf.pid[]	3*PIDITEMS x UINT 8	
MSP_BOX	113	FC →	BOXITEMS x conf.activate[]	BOXITEMS x UINT 16	BOXITEMS number is dependant of multiwii configuration The size of the message is enough to know the number of BOX For each BOX, there is a 16 bit variable which indicates the AUX1->AUX4 activation switch. Bit 1: AUX1 LOW state / bit 2: AUX1 MID state / bit 3: AUX1 HIGH state / bit 4: AUX2 LOW state ….. bit 13: AUX 4 HIGH state
MSP_SET_BOX	203	→ FC	BOXITEMS x conf.activate[]	BOXITEMS x UINT 16	
MSP_MISC	114	FC →	intPowerTrigger1	UNIT 16	
			conf.minthrottle	UNIT 16	minimum throttle to run motor in idle state (range [1000;2000])
			MAXTHROTTLE	UNIT 16	maximum throttle (range [1000;2000])
			MINCOMMAND	UNIT 16	throttle at the lowest position (range [1000;2000] , could be occasionally a little bit less than 1000 depending on ESCs)
			conf.failsafe_throttle	UNIT 16	should be set less than hover state (range [1000;2000])
			plog.arm	UNIT 16	counter
			plog.lifetime	UNIT 32	
			conf.mag_declination	UNIT 16	magnetic declination (unit:1/10 degree)
			conf.vbatscale	UNIT 8	
			conf.vbatlevel_warn1	UNIT 8	unit: 1/10 volt
			conf.vbatlevel_warn2	UNIT 8	unit: 1/10 volt
			conf.vbatlevel_crit	UNIT 8	unit: 1/10 volt
MSP_SET_MISC	207	→ FC	intPowerTrigger1	UNIT 16	
			conf.minthrottle	UNIT 16	
			MAXTHROTTLE	UNIT 16	not used currently as a set variable
			MINCOMMAND	UNIT 16	not used currently as a set variable
			conf.failsafe_throttle	UNIT 16	
			plog.arm	UNIT 16	not used, it's here to have the same struct as get
			plog.lifetime	UNIT 32	not used, it's here to have the same struct as get
			conf.mag_declination	UNIT 16	magnetic declination (unit:1/10 degree)
			conf.vbatscale	UNIT 8	
			conf.vbatlevel_warn1	UNIT 8	unit: 1/10 volt
			conf.vbatlevel_warn2	UNIT 8	unit: 1/10 volt
			conf.vbatlevel_crit	UNIT 8	unit: 1/10 volt
MSP_MOTOR_PINS	115	FC →	8*PWM_PIN	8 x UNIT 8	motor pin indication
MSP_BOXNAMES	116	FC →	string of BOX items	string	all the configured CHECKBOX name separated by ;
MSP_PIDNAMES	117	FC →	string of PID items	string	all the PID name separated by ;
MSP_WP	118	FC →	wp_no	UINT 8	not fully implemented yet, works partially for HOME POSITION (wp 0) and HOLD position (wp 15)
			lat	UINT 32	
			lon	UINT 32	
			AltHold	UINT 32	
			heading	UINT 16	
			time to stay	UINT 16	
			nav flag	UINT 8	
MSP_SET_WP	209	→ FC	wp_no	UINT 8	
			lat	UINT 32	
			lon	UINT 32	
			AltHold	UINT 32	
			heading	UINT 16	
			time to stay	UINT 16	
			nav flag	UINT 8	
MSP_BOXIDS	119	FC →	ID*CHECKBOXITEMS	CHECKBOXITEMS x UINT 8	each BOX (used or not) have a unique ID. In order to retrieve the number of BOX and which BOX are in used, this request can be used. It is more efficient than retrieving BOX names if you know what BOX function is behing the ID. See enum MultiWii.cpp (0: ARM, 1 ANGLE, 2 HORIZON, …)
MSP_SERVO_CONF	120	FC →	8 x conf.servoConf[]	8 x [UINT 16, UINT 16, UINT 16, UINT 8]	struct servo_conf_ is 7 bytes length: min:2 / max:2 / middle:2 / rate:1 [1000;2000], [1000;2000], [1000;2000], [0;100] Special use: middle normal range is [1000;2000] If middle < RC_CHANS => the relevant rc channel is the middle position of the servo (usefull for gimbal where you wnt to control the middle axis via a rc chan) Depending on the servo use in multiwii type, rate is used to reverse the direction of servo (first bit) or to set a proportional range
MSP_SET_SERVO_CONF	212	→ FC	8 x conf.servoConf[]	8 x [UINT 16, UINT 16, UINT 16, UINT 8]	
MSP_ACC_CALIBRATION	205	→ FC	no param		used to calibrate the ACC
MSP_MAG_CALIBRATION	206	→ FC	no param		used to calibrate the MAG
MSP_RESET_CONF	208	→ FC	no param		reset all params to default
MSP_SELECT_SETTING	210	→ FC	global_conf.currentSet	UINT 8	select the setting configuration (you can set for instance different pid and rate) Range: 0, 1 or 2

MSP_SET_HEAD	211	→ FC	magHold	INT 16	Set a new head lock reference Range [-180;+180]
MSP_BIND	240	→ FC	no param		Currently only uses to bind spektrum sttellites
MSP_EEPROM_WRITE	250	→ FC	no param		write the settings to the eeprom