Name	Code	Direction	Length	Data	Notes
Play Sound	0x06	->MIP	1	BYTE 1 : Sound file index (1~106) or send 0xF7-0xFE for volume	Send 105 to stop playing
			2	BYTE 2 : Delay in intervals of 30ms (0~255)	
				… (repeat sound file index then delay for as many files as you want to play)	
			17	BYTE 17 : Number of times to repeat (0-255)	Only need to send if you want the sequence to repeat
Set Mip Position	0x08	-> MIP	1	BYTE 1 : On back: 0x00, Face down: 0x01	
Distance Drive	0x70	->MIP	5	BYTE 1 : Forward: 0X00 or Backward: 0X01	20 commands are queued
				BYTE 2 : Distance (cm): 0x00-0xFF	No speed control
				BYTE 3 : Turn Clockwise: 0X00 or Anti-clockwise: 0X01	
				BYTE 4 : Turn Angle(High byte): 0x00~0x01	
				BYTE 5 : Turn Angle(Low byte): 0x00~0xFF	Note:0x0000(0)~0x0168(360)
Drive forward with Time	0x71	->MIP	2	BYTE 1 : Speed (0~30)	
				BYTE 2 : Time in 7ms intervals (0~255)	35ms
Drive backward with Time	0x72	->MIP	2	BYTE 1 : Speed (0~30)	
				BYTE 2 : Time in 7ms intervals (0~255)	Time = Byte2 Value * 7ms
Turn left by Angle	0x73	->MIP	2	BYTE 1 : Angle in intervals of 5 degrees (0~255)	Angle = Byte1 Value * 5
				BYTE 2 : Speed (0~24)	
Turn right by Angle	0x74	->MIP	2	BYTE 1 : Angle in intervals of 5 degrees (0~255)	Angle = Byte1 Value * 5
				BYTE 2 : Speed (0~24)	
Continuous Drive	0x78	->MIP	2	Fw:0x01(slow)~-0x20(fast)	Buffer = 0
				OR Bw:0x21(slow)~0x40(fast)	This command is for single drive or turn
				right spin:0x41(slow)~0x60(fast)	Note:Sending per 50ms if held
				OR Left spin:0x61(slow)~0x80(fast)	

				Carzy Fw:0x81(slow)~-0xA0(fast)	
				OR Carzy Bw:0x81(slow)~0xC0(fast)	
				Carzy right spin:0xC1(slow)~0xE0(fast)	
				OR Carzy Left spin:0xE1(slow)~0xFF(fast)	
Set Game Mode	0x76	->MIP	1	BYTE 1 : 0x01 – App	The same as cancel Gesture and Radar
				0x02 – Cage Play back	
				0x03 – Tracking	The same as enable Radar
				0x04 – Dance Play back	
				0x05 – Default Mip Mode	The same as enable Gesture(0x0A)
				0x06 – Stack Play back	
				0x07 – Trick programming and playback	
				0x08 – Roam Mode Play back	
Get current MIP Game Mode	0x82	-> MIP	-		
Current MIP Game Mode	0x82	iOS <-	1	BYTE 1 : 0x01 – App	
				0x02 – Cage	
				0x03 – Tracking	
				0x04 – Dance	
				0x05 – Default Mip Mode	
				0x06 – Stack	
				0x07 – Trick programming and playback	
				0x08 – Roam Mode	
Stop	0x77	->MIP	-	-	
Request MIP status	0x79	-> MIP	-		
MIP status	0x79	iOS <-	1	BTYE 1 : Battery Level :0x4D(4.0V)-0x7C(6.4V)	
				BYTE 2 : On back: 0x00	
				Face down 0x01	
				Upright: 0x02	Note:Send one time per 30 seconds
				Picked up: 0x03	Note:it will be sent after(connecting,falldown,pickup….)
				Hand stand: 0x04	

				Face down on tray: 0x05	
				On back with kickstand: 0x06	
Mip Get Up	0x23	-> MIP	1	BTYE 1 : 0x00 – Get up when mip has fallen front	Mip will attempt to get up from front if angle is correct
				0x01 – Get up when mip has fallen back	Mip will attempt to get up from back if angle is correct
				0x02 – Get up when mip has fallen back or front	
Request weight update	0x81	-> MIP			
Weight update	0x81	iOS <-	1	BYTE 1 : 0xD3(-45 degree) - 0x2D(+45 degree)	
				0xD3 (211) (max)~0xFF(min) (255) is holding the weight on the front	
				0x00(min)~0x2D(max) is holding the weight on the back	
Request Chest LED	0x83	->MIP	-	-	
Chest LED	0x83	iOS <-	3	BYTE 1 : Red (0~255)	
				BYTE 2 : Green (0~255)	
				BYTE 3 : Blue (0~255)	
				BYTE 4 : if flashing then, TIME ON in 10ms intervals (0~255) else Fade in time in 10ms intervals (0~255)	
				BYTE 5 : if flashing then, TIME OFF in 10ms intervals (0~255) else will only be 4 bytes	
Set Chest LED	0x84	->MIP	4	BYTE 1 : Red (0~255)	
				BYTE 2 : Green (0~255)	
				BYTE 3 : Blue (0~255)	
Flash Chest LED	0x89	->MIP	4	BYTE 1 : Red (0~255)	Value of 0 means LED color will be changed immediately
				BYTE 2 : Green (0~255)	
				BYTE 3 : Blue (0~255)	
				BYTE 4 : TIME ON in 20ms intervals (0~255)	
				BYTE 5 : TIME OFF in 20ms intervals (0~255)	
					Time on = Byte4 Value * 10ms

Set Head LED	0x8A	->MIP	4	BYTE 1 : LIGHT 1 (0=off, 1=on, 2=blink slow, 3=blink fast)	Time off = Byte5 Value * 10ms
				BYTE 2 : LIGHT2 (0=off, 1=on, 2=blink slow, 3=blink fast)	
				BYTE 3 : LIGHT3 (0=off, 1=on, 2=blink slow, 3=blink fast)	
				BYTE 4 : LIGHT4 (0=off, 1=on, 2=blink slow, 3=blink fast)	
Request Head LED	0x8B	->MIP	-		
Head LED	0x8B	iOS <-	4	BYTE 1 : LIGHT 1 (0=off, 1=on, 2=blink slow, 3=blink fast)	
				BYTE 2 : LIGHT2 (0=off, 1=on, 2=blink slow, 3=blink fast)	
				BYTE 3 : LIGHT3 (0=off, 1=on, 2=blink slow, 3=blink fast)	
				BYTE 4 : LIGHT4 (0=off, 1=on, 2=blink slow, 3=blink fast)	
Read Odometer	0x85	->MIP	-		
Odometer reading	0x85	iOS <-	4	BYTE 1 & 2 & 3 & 4 : Distance ((0~4294967296)/48.5) cm	
				1cm=48.5，0xFFFFFFFF=4294967295=88556026.7cm	
				BYTE 1 & 2 & 3 & 4 :Btye1 is highest byte	
Rest Odometer	0x86	->MIP	-		
					Sent only when requested unless it is over 4294967296 then it sends to app and resets
Gesture Detect	0x0A	IOS<-	1	BYTE 1 : Left: 0x0A	
				Right: 0x0B	
				Center Sweep Left: 0x0C	
				Center Sweep Right: 0x0D	
				Center Hold: 0x0E	
				Forward: 0x0F	
				Back: 0x10	(700ms hold)
Set Gesture Or Radar Mode	0x0C	-> MIP	-	BYTE 1 :	

				0x00: Disable Gesture and Radar	
				0x02: Gesture Mode on (Disable Radar)	Gesture mode is tracking hand gestures
				0x04: Radar Mode on (Disable Gesture)	
Get Radar Mode	0x0D	-> MIP	-	-	
Radar Mode Status	0x0D	iOS <-	1	BYTE 1: 0x00: Disable Gesture and Radar	
				0x02: Gesture Mode on (Disable Radar)	
				0x04: Radar Mode on (Disable Gesture)	
Radar Response	0x0C	iOS <-	1	BYTE 1 : 0x01: No object Or object disappear	
				0x02: See object in 10cm~30cm	
				0x03: See object less than 10cm	Used for radar
MIP Detection Mode	0x0E	-> MIP	2	BYTE 1 : Off: 0x00, On: 0x1-255 for ID number	
				BYTE 2 : Set IR Tx power(1~120)(About 1cm~300cm)	
Request MIP Detection Mode	0x0F	-> MIP	-	-	
Mip Detection Status	0x0F	iOS <-	2	BYTE 1 : Off: 0x00, On: 0x1-255 for ID number	This constantly 'pings' other MiPs to check if any are in range. When this is enabled it will automatically disable Radar mode & gesture mode
				BYTE 2 : Set IR Tx power(1~120)(About 1cm~300cm)	
Mip Detected	0x04	iOS <-	1	BYTE 1 : ID number，if got the 0x00 means find one MIP without setting number	
Shake Detected	0x1A	iOS <-	1	-	
IR Remote Control Enabled	0x10	-> MIP	1	BTYE 1 : Off: 0x00, On: 0x01	
Request IR Control Enabled	0x11	-> MIP	-	-	
IR Control Status	0x11	iOS <-	1	BTYE 1 : Off: 0x00, On: 0x01	

Sleep	0xFA	MIP<->IOS	-		Power down bluetooth.
Disconnect App	0xFE	->MIP	-	-	Mip should switch off app mode and return to previous mode
Force BLE disconnect	0xFC	->MIP	-	-	
Set User Data	0x12	-> MIP	2	BYTE 1 : Data address(0x20~0x2F)	
				BYTE 2 : Data	
Get User Or Other Eeprom Data	0x13	-> MIP	1	BYTE 1 : User Data address(0x20~0x2F) and other data is in Eeprom	
MIP User Or Other Eeprom Data	0x13	iOS <-	2	BYTE 1 : User Data address(0x20~0x2F) and other data is in Eeprom	
				BYTE 2 : Data	
Get Mip Software Version	0x14	-> MIP	-	-	
Mip Software Version	0x14	iOS <-	7	BYTE 1: Year (software ver)	
				BYTE 2: Month (software ver)	Gets some software version info
				BYTE 3: Day (software ver)	Date of the software release
				BYTE 4: Unique Version #	
Get Mip Hardware Info	0x19	-> MIP	-	-	this is used if more than one release on one day, normally it is 0x00
Mip Hardware Info	0x19	iOS <-	2	BYTE 1: Voice chip version	
				BYTE 2: Hardware Version	
Set Mip Volume	0x15	-> MIP	1	BYTE 1: Volume level between 0-7	Sets the MIP volume level, Power off save
Get Mip Volume	0x16	-> MIP	-	-	
Mip Volume	0x16	iOS <-	1	BYTE 1: Volume level between 0-7	Reads the current MIP volume level
Send IR Dongle code	0x8C	-> MIP	6	BYTE1:IR data bit31~bit24	

				BYTE2:IR data bit23~bit16	
				BYTE3:IR data bit15~bit8	
				BYTE4:IR data bit7~bit0	
				BYTE5:IR data numbers(1~32):e.g. BYTE5=0x08 means BYTE4 is useful.	
				BYTE6:IR Tx power(1~120)(About 1cm~300cm)	
				Note:It can be useful in shoting game.	
Receive IR Dongle code	0x03	iOS <-	3~5	BYTE 1 : 0x02,0x03,0x04	Receive an IR command
				BYTE2~BYTE5 are the datas of Transmiting.	
				Byte2 is high byte,Byte5 is low byte	
				(0x03,0x02,0xNN,0xNN)	
				(0x03,0x03,0xNN,0xNN,0xNN)	
				(0x03,0x04,0xNN,0xNN,0xNN,0xNN)	
Clap times	0x1D	iOS <-	1	BYTE 1: 0x01 – 0xFF times	
Clap Enabled	0x1E	-> MIP	1	BTYE 1 : Off: 0x00, On: 0x01(Default is disable after App connecting)	
Request Clap Enabled	0x1F	-> MIP	-	-	
Clap Status	0x1F	iOS <-	3	BTYE 1 : Off: 0x00, On: 0x01	
				BYTE2~BYTE3(Delay time by two clap)	
Delay time between two claps	0x20	-> MIP	2	BYTE1(high)~BYTE2(low)(Delay time by two clap)	