Shoot1

						Sub-Address 0-FF	Command 0-FF	Data Length 0-FF	Data							
		3			0-FF		Data	011								
	Notes	Action	Sub-Address	s Command	Command (HEX)	Data Length	Byte:1	2	3	4	5 6	7	8	9	10	11 12
Main		NOOP	0	0	0	0						++	$\rightarrow$	_		
	Starts planned move	Reserve for core protocol Start	0	2	2	0						+	$\rightarrow$	$\rightarrow$		
		Pause	0	3	3	0										
	Stops planned move. Must be executed before controller will accept other commands.	Stop	0	4	4	0										
	Toggles on/off state of debug LED	Debug LED	0	5	5	0						+	-	_		
		Timing Master	0	6	6	0							=			
		Set Stored Name Set Device Address	0	7	7	1-10	2-255 [Byte]	String [1	-10 Characters, Null-terminated, N	ull padded]	_					
		Set Common Line for Step Pulsing	0	9	9	1	0,1,2 [Byte]				_	++	$\rightarrow$	-		$\overline{}$
		Return Home All Motors	0	10	A	0										
		Motors Max Step Rate	0	11	В	2	Steps/S	econd [Int]					-			
1		Alt Input Edge (RISING, FALLING, or CHANGE) Alt I/O Mode	0	12 13	C	1 2	0,1,2 [Byte] Ring (0-255) [Byte]	Tip (0-255) [Byte]				+	=			
		Set Joystick Watchdog	0	14	E	1	True/False (1,0) [Byte]									
		Alt Output Before Shot Delay Time	0	15	F	2		ms) [int]				$\perp$	-	$\rightarrow$		
		Alt Output After Shot Delay Time Alt Output Before Shot Time	0	16	10	2	Time i	ms) [Int] ms) [Int]				+++	$\rightarrow$	-		
		Alt Output After Shot Time	0	18	12	2	Time	ms) [Int]					-	-		
		Alt Output Trigger level	0	19	13	1	HIGH/LOW (1,0) [Byte]									
		Max Program Run Time	0	20	14 15	4		Max Run Time (mS) [I Start Time Delay (second	Jiong]		_		$\rightarrow$	$\rightarrow$		
l		Start Program Delay	0	21	16	1	0 (SMS), 1 (Time Lapse Cont.)	Start Time Delay (second	J, [O.Orig]		+	+	+	$\rightarrow$		+
Į.		Set SMS / Continuous Program Mode Set Joystick Mode	0	22	17	1	2 (Video Cont.) [Byte] True/False (1,0) [Byte]					++	$\rightarrow$	-		
ļ	Causes the motors to go back and forth Between the start and stop positions	Set Ping-Pong Flag	0	24	18	1	True/False (1,0) [Byte]									
		Status Request	0	100			<status type=""></status>		<returns> with hea</returns>	der and master and	race in front	(00 00 00	) 00 00 F	00.00.0	1 Length F	lata)
		Firmware Version	0	100	64	0	Value Type [Byte]	Version #	⇒returns≥ with nea	de and master add	LOS EI HUNT	JU 00 0L	30 00 FF	00 00 0	. cerigin L	u.u.)
		Run Status	0	101	65	0	Value Type [Byte] Value Type [Byte]	True/False								
Į.		Run Time	0	102	66	0	Value Type [Ulong] Value Type [Byte]	True/False	Time (ms)			+F	-			-
1		Currently Exposing Timing Master Value	0	103	68	0	Value Type [Byte]	True/False					-+	-		
		Name	0	105	69	0	Value Type [String]		String [1-10 Charac	ters, Null-terminate	d, Null padde	d]				
		Motors Max Step Rate	0	106	6A	0	Value Type [Uint]		Second				$\rightarrow$			
		Voltage Reading Current to Motors	0	107	6B 6C	0	Value Type [Fixed] Value Type [Fixed]	Voltage (V) (Fixed	point – must divide by 100 on mas	ter side)		++	$\rightarrow$	_		
		Alt Input Edge (RISING, FALLING, or CHANGE)	0	109	6E	0	Value Type [Byte]	0,1,2	d point – must divide by 100 on m.	Jotel Giocy			-	-		
		Alt I/O Mode	0	110	6F	0	Value Type [Byte] Value Type [Int]	[Byte 0] Ring (0-255)	[Byte 1] Tip (0-255) [Byte 1] Tip, High/Low(1,0)				=			
		Limit Switch High/Low Status Alt Output Before Shot Delay Time	0	111 112	70 70	0	Value Type [Int] Value Type [Uint]	[Byte 0] Ring, High/Low (1,0)	(ms) [Byte 1] Tip, High/Low(1,0)				$\rightarrow$	_		
		Alt Output After Shot Delay Time	0	113	71	0	Value Type [Uint]	Time	(ms)			+	$\rightarrow$	$\rightarrow$		
		Alt Output Before Shot Time	0	114	72	0	Value Type [Uint]	Time	(ms)							
		Alt Output After Shot Time Alt Output Trigger level	0	115 116	73 74	0	Value Type [Uint]	Time HIGH/LOW (1,0)	(ms)			+	$\rightarrow$	$\rightarrow$		
		Start Program Delay	0	116	75	0	Value Type [Byte] Value Type [Ulong] Value Type [Byte]	HIGH/LOW (1,0)	tart Time Delay (seconds)			+	$\rightarrow$	-	_	
		SMS / Continuous Program Mode	0	118	76	0	Value Type [Byte]	0 (SMS), 1 (Cont.)	(4555.135)				-			
		Controller Power Cycle	0	119	77	0	Value Type [Byte]	True/False (1,0)			_		$\rightarrow$	$\rightarrow$		
		Joystick Mode Ping-Pong Flag	0	120	78 79	0	Value Type [Byte] Value Type [Byte]	True/False (1,0) True/False (1,0)								
		Joystick Watchdog Mode Status	0	122	7A	0	Value Type [Byte]	True/False (1,0)								
	Reports the percentage complete of the current program as a whole number	Program % Complete	0	123	7B	0	Value Type [Byte]	0-100								
	The three least significant bits of the byte represent the motor attach states for each of the motors. Motor 1 = bit 0, motor 2 = bit 1, motor 3 = bit 2.	Check Motor Attachment	0	124	7C	0	Value Type [Byte]	0-7								
Matara	I, HIOLOI 3 - DR Z.	NOOP	1-3													
Motors	<u> </u>	Reserve for core protocol	1-3	1	1	0				+		$\pm \pm$	=+	_+	+	
	Cuts power to motor when not executing a	Motor Sleep	1-3	2	2	1	True/False (1,0) [Byte]									
	move. True by default.  Must enabled before executing a move.	Motor Enable	1-3	3	3	1	True/False (1,0) [Byte]				_	-	$\rightarrow$			
	Stops motor, even if a planned move is in	Stop Motor Now	1-3	4	4	0	Tracit also (1,0) [byte]				+-	+	-	-		
I	progress.	Stop MOTOL NOW	1-0		,	0					_	+	$\rightarrow$	$\rightarrow$		
Į	Number of steps the motor should move in addition to the commanded distance when reversing direction.	Set Backlash Steps	1-3	5	5	2	Step	s [Uint]								
General Motor Commands	Number of microsteps per full motor step.											П				
	There are 200 full steps per rotation of the motor (and ~3800 full steps per gearbox output shaft rotation)	Set Microstep Value	1-3	6	6	1	1, 2, 4, 8, 16 [Byte]									
1	There are 200 full steps per rotation of the motor (and ~3800 full steps per gearbox output shaft rotation)	Set Microstep Value Set Motor Max Step Speed	1-3	6	6	1 2		cond [Uint]				Н	$\pm$			
ĺ	There are 200 full steps per rotation of the motor (and ~3800 full steps per gearbox output shaft rotation)  Flips motor direction, regardless of current							cond [Uint]					$\dashv$	-		
ĺ	There are 200 full steps per rotation of the motor (and ~3800 full steps per gearbox output shaft rotation)  Flips motor direction, regardless of current program	Set Motor Max Step Speed Set Direction	1-3 1-3 1-3	7	7	2	Steps/Se	cond [Uint]					$\equiv$			
	There are 200 full steps per rotation of the motor (and ~3800 full steps per gearbox output shaft rotation)  Flips motor direction, regardless of current	Set Motor Max Step Speed Set Direction Set Home Limit Here Set End Limit Here	1-3 1-3 1-3 1-3	7 8 9	7 8	2	Steps/Se	cond [Uint]					#			
	There are 200 full steps per rotation of the motor (and ~3800 full steps per gearbox output shaft rotation)  Flips motor direction, regardless of current program  Saves home limit position	Set Motor Max Step Speed Set Direction Set Home Limit Here Set End Limit Here Send Motor to Home Limit	1-3 1-3 1-3 1-3 1-3	7 8 9 10	7 8 9	2	Steps/Se	cond (Uint)								
	There are 200 full steps per rotation of the motor (and -3800 full steps per gearbox output shaft rotation) Filips motor direction, regardless of current program. Saves from limit position Saves and limit position	Set Motor Max Step Speed  Set Direction  Set Home Limit Here  Set End Limit Here  Send Motor to Home Limit  Send Motor to End Limit	1-3 1-3 1-3 1-3	7 8 9	7 8 9	2	Steps/Se									
	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shart rotation)  Flips motor direction, regardless of current program  Saves home limit position  Saves end limit position  Does not apply to finite manual moves	Set Motor Max Step Speed  Set Direction  Set Home Limit Here  Set End Limit Here  Send Motor to Home Limit Send Motor to Find Limit Set Ordinuous Speed	1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12	7 8 9 A B C	2 1 0 0 0 0	Steps/Se	Steps/Second [flor								
	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shaft rotation)  Flips motor direction, regardless of current program  Saves home limit position  Saves end limit position  Does not apply to finite manual moves	Set Motor Max Step Speed  Set Direction  Set Home Limit Here Set End Limit Here Send Motor to Home Limit Send Motor to End Limit Set Continuous Speed Set Motor Continuous Speed	1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12	7 8 9 A B C	2 1 0 0 0	Steps/St 0, 1 [Byte]		oat]							
Manual Move	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shall rotation). Figs motor direction, regardless of current sogram. Saves home intil position.  Saves end limit position.  Does not apply to finite manual moves.  Does not apply to finite manual moves.	Set Motor Max Step Speed  Set Direction  Set Home Limit Here Set End Limit Here Set End Limit Here Send Motor to Home Limit Send Motor to Home Limit Send Motor to End Limit Set Continuous Speed Set Motor Continuous Motion Accel/Decel Rate Execute Simple Motor Move	1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12 13 14 15	7 8 9 A B C	2 1 0 0 0 0 0	Steps/Se	Steps/Second [floi Steps/Second*2 [fix	steps [Ulong]							
Manual Move	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shaft rotation)  Flips motor direction, regardless of current program  Saves home limit position  Saves end limit position  Does not apply to finite manual moves	Set Motor Max Step Speed  Set Direction  Set Home Limit Here  Set Ernd Limit Here  Send Motor to Home Limit Send Motor to Home Limit Send Motor to Fond Limit  Set Continuous Speed  Set Motor Continuous Motor Accel/Decel Rate  Execute Simple Motor Move  Set Program Start point	1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12 13 14 15	7 8 9 A B C	2 1 0 0 0 0 0 0 4 4 4 5	Steps/St 0, 1 [Byte]	Steps/Second [floid Steps/Second*2 [fix Step position [tony	steps [Ulong]							
Manual Move	There are 200 full steps per rotation of the motor (and ~300 full steps per gearbox output shall rotation). The step per gearbox output shall rotation in the step shall rotation in the step shall rotation in the step shall rotation. Saves shome innit position.  Saves shome innit position.  Does not apply to finite manual moves. Dies not apply to finite manual moves use of "start" and "stop" commands.	Set Motor Max Step Speed  Set Direction  Set Home Limit Here Set End Limit Here Set End Limit Here Send Motor to Home Limit Send Motor to Home Limit Set Continuous Speed Set Motor Continuous Motion Accel/Decel Rate Execute Simple Motor Move  Set Program Start point Set Program Start point	1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12 13 14 15	7 8 9 A B C C D E	2 1 0 0 0 0 0 0 4 4 4 5	Stepu/Se 0, 1 [Byte]  Dir (0, 1) [Byte]	Steps/Second [floi Steps/Second*2 [fix	steps [Ulong]							
Manual Move Commands	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shall rotation). Dut steps per gearbox output shall rotation?  Filips motor direction, regardless of current program.  Saves home imit position.  Does not apply to finite manual moves.  Does not apply to finite manual moves.  Direct move command, does not require use of "start" and "stop" commands.  I = Linear, 2 = Quadratic, 3 = Inverted Quadratic.	Set Motor Max Step Speed  Set Direction  Set Home Limit Here  Set Ernd Limit Here  Send Motor to Home Limit Send Motor to Home Limit Send Motor to Fond Limit  Set Continuous Speed  Set Motor Continuous Motor Accel/Decel Rate  Execute Simple Motor Move  Set Program Start point	1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12 13 14 15	7 8 9 A B C	2 1 0 0 0 0 0 0 4 4 4 5	Steps/St 0, 1 [Byte]	Steps/Second [floid Steps/Second*2 [fix Step position [tony	steps [Ulong]							
Manual Move Commands	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shart rotation)  Figure 1 full steps per gearbox output shart rotation)  Figure 1 full steps per gearbox output shart rotation)  Saves a full mit position  Saves end mit position  Does not apply to finite manual moves pose not apply to finite manual moves Direct move command, does not require use of 'start' and 'stop' commands.  I = Linear, Z = Quadratic, 3 = Inverted Quadratic  I = Linear, Z = Quadratic, 3 = Inverted Quadratic	Set Motor Max Step Speed  Set Direction  Set Home Limit Here Set End Limit Here Set End Limit Here Send Motor to Home Limit Send Motor to Home Limit Set Continuous Speed Set Motor Continuous Motion Accel/Decel Rate Execute Simple Motor Move  Set Program Start point Set Program Start point	1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12 13 14 15	7 8 9 A B C C	2 1 0 0 0 0 0 4 4 5	Stepu/Se 0, 1 [Byte]  Dir (0, 1) [Byte]	Steps/Second [floid Steps/Second*2 [fix Step position [tony	steps [Ulong]							
Manual Move Commands Programmed Travel	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shall rotation). Dut steps per gearbox output shall rotation?  Filips motor direction, regardless of current program.  Saves home imit position.  Does not apply to finite manual moves.  Does not apply to finite manual moves.  Direct move command, does not require use of "start" and "stop" commands.  I = Linear, 2 = Quadratic, 3 = Inverted Quadratic.	Set Motor Max Step Speed  Set Direction  Set Home Limit Here Set End Limit Here Send Motor to Home Limit Send Motor to Home Limit Send Motor to Foot Limit Send Motor to Foot Limit Send Motor to Gred Limit Set Confinious Speed Set Motor Continuous Motion Accel/Decel Rate Execute Simple Motor Move  Set Program Start point Set Program Stop point Set Easing (Ramping) Mode  Set Lead-In Shots	1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12 12 13 14 15	7 8 9 A B B C C	2 1 0 0 0 0 0 0 4 4 4 5	Stepa/Se 0.1 [Byte]  Dr (0.1) [Byte]  Dr (0.1) [Byte]	Steps/Second (Bold Steps/Second / 2 (%) Steps/Second / 2 (%) Step position (long Step	Steps [Ulong]							
Manual Move Commands	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shart rotation)  Figure 1 full steps per gearbox output shart rotation)  Figure 1 full steps per gearbox output shart rotation)  Saves a full mit position  Saves end mit position  Does not apply to finite manual moves pose not apply to finite manual moves Direct move command, does not require use of 'start' and 'stop' commands.  I = Linear, Z = Quadratic, 3 = Inverted Quadratic  I = Linear, Z = Quadratic, 3 = Inverted Quadratic	Set Motor Max Step Speed  Set Direction  Set Home Limit Here Set End Limit Here Set End Limit Here Send Motor to Home Limit Send Motor to Home Limit Send Motor to Fond Limit Set Confinuous Speed Set Motor Continuous Motion Accel/Decel Rate Execute Simple Motor Move  Set Program Stat point Set Program Stop point Set Easing (Ramping) Mode  Set Lead-in Shots Set Travel Shots(SMS) / Travel Time (Cont.)	1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3 1-3	7 8 9 10 11 12 13 14 15	7 8 9 A B C C	2 1 0 0 0 0 0 4 4 5	Stepu/Se   0.1 [Byte]     Dir (0.1) [Byte]	Steps/Second [floid Steps/Second*2 [fic Step position [lon Step positi	Steps [Ulong]							
Manual Move Commands  Programmed Travel	There are 200 full steps per rotation of the motor (and -300 full steps per gearbox output shart rotation)  Figure 1 full steps per gearbox output shart rotation)  Figure 1 full steps per gearbox output shart rotation)  Saves a full mit position  Saves end mit position  Does not apply to finite manual moves pose not apply to finite manual moves Direct move command, does not require use of 'start' and 'stop' commands.  I = Linear, Z = Quadratic, 3 = Inverted Quadratic  I = Linear, Z = Quadratic, 3 = Inverted Quadratic	Set Motor Max Step Speed  Set Direction  Set Home Limit Here Set End Limit Here Send Motor to Home Limit Send Motor to Home Limit Send Motor to Foot Limit Send Motor to Foot Limit Send Motor to Gred Limit Set Confinious Speed Set Motor Continuous Motion Accel/Decel Rate Execute Simple Motor Move  Set Program Start point Set Program Stop point Set Easing (Ramping) Mode  Set Lead-In Shots	13 13 13 13 13 13 13 13 13 13 13 13 13 1	7 8 9 10 11 11 12 13 14 15 16 17 18	7 8 9 A B C D E F 10 11 12 13	2 1 0 0 0 0 0 0 0 4 4 4 5 5	Stepu/Se   0.1 [Byte]     Dir (0.1) [Byte]	Steps/Second (Bold Steps/Second / 2 (%) Steps/Second / 2 (%) Step position (long Step	Steps [Ulong]							

	Send Motor to Program Stop Point	1-3	24	18	0	

	Manual SMS movement. Not yet									1			1 1			
Stop-Motion i	implemented.	Advance One	SMS Increment	1-3	25	19	0									
ommands i	Manual SMS movement. Not yet implemented.	Go Back One	SMS Increment	1-3	26	1A	0									
F	Sets the current position as home, disables												_			
I	limits, and sets start/stop positions to home position.	Reset Limits and Prog	ram Start/Stop Positions	1-3	27	1B	0									
F t t	The controller will automatically select the highest resolution microstepping (up to 1/4 stepping) that can be used to achieve the program parameters. It will also report be the setting it uses. O will be reported when the command is called at an illegal time (i.e. when the motor is in motion). 255 will be reported when the speed required by the current plain parameters exceeds the controller's top speed.	Auto Set Proç	gram Microsteps	1-3	28	1C	0	Value Type [Byte]	0, 4, 8, 16, 255							
٦		Status	Request	1-3	100			<status type=""></status>		<returns> with hear</returns>	der and master	address in front (00 0	0 00 00 00 0	F 00 00 01	ength Data)	
		Motor	r Enable	1-3	100	64	0	Value Type [Byte]	True/False (1,0)							
			ish Steps	1-3	101	65	0	Value Type [Uint]	Ste	eps						
neral Motor			tep Value ection	1-3 1-3	102 103	66 67	0	Value Type [Byte] Value Type [Byte]	1, 2, 4, 8, 16 0, 1					_	_	
Query ommands		Motor Max	Step Speed	1-3	104	68	0	Value Type [Int]	Steps/S	Second						
ommanas			nit Position	1-3	105	69	0	Value Type [Long]		Position						
1		Current M	otor Position	1-3	106	6A	0	Value Type [Long]		Position						
		Motor	Running	1-3	107	6B	0	Value Type [Byte]	True/False (1,0)							
anual Move Query		Continue	ous Speed	1-3	108	6C	0	Value Type [Psudo-floatFixed point – must divide by 100 on master side]		Steps/Second						
ommands		Motor Continuous M	otion Accel/Decel Rate	1-3	109	6D	0	Value Type [Psudo-floatFixed point – must divide by 100 on master side]		Steps/Second^2						
		Easing (Ra	imping) Mode	1-3	110	6E	0	Value Type [Byte]	1, 2, 3							
[		Program	Start point	1-3	111	6F	0	Value Type [Long]		Position						
		Program	Stop point	1-3	112	70	0	Value Type [Long]		Position						
grammed			/ Travel Time (Cont.)	1-3 1-3	113 114	71	0	Value Type [Ulong]		S) or Total Travel Time (ms) (cont.)					_	
vel Query			In Shots am Accel	1-3	115	72 73	0	Value Type [Int] Value Type [Ulong]	Sh Accel Period	ots – Shots (SMS) or Time in ms (Cor	w )				_	
ommands		Progra	im Decel	1-3	116	74	0	Value Type [Ulong]	Decel Period	<ul> <li>Shots (SMS) or Time in ms (Cor</li> <li>Shots (SMS) or Time in ms (Cor</li> </ul>	nt.)					
			Cont. Program Move	1-3	117	75	0	Value Type [Psudo-floatFixed point – must divide by 100 on master side]		Steps/Second	,					
Cameras		N	OOP	4	0	0	0						1			
		Reserve for	core protocol	4	1	1	0									
		Camer	ra Enable	4	2	2	1	True/False (1,0) [Byte]						_	_	
	Triggers exposure with length set by "Exposure Time" command.		se Now er Time	4	3 4	3	0		Function Time (mC) II	Hand						
ŀ			er rime is Time	4	5	5	2	Focus Time	Exposure Time (mS) [I (mS) [Uint]	Jiongj				_	_	
r	The system will stop a move once it reaches the max number of camera exposures.	Max	Shots	4	6	6	2	Coun	[Uint]							
		Exposi	ure Delay w Shutter	4	7 8	7 8	2	Delay (n True/False (1,0) [Byte]	nS) [Uint]					_	_	_
t	This causes two trigger signals to be sent to the camera in the event that the user has		Repeat Shot)	4	9	9	1	True/False (1,0) [Byte]								
	the camera in "mirror up" mode. Length of SMS interval	Int	erval	4	10	A	4		Interval Time (mS) [U	long]					_	
I																
			Request	4	100	64		<status type=""></status>	T(T-I (4.0)	<returns> with hear</returns>	der and master	address in front (00 0	00 00 00 0	F 00 00 01	ength Data)	
}			ra Enable	4	100	64 65	0	Value Type [Byte] Value Type [Byte]	True/False (1,0) True/False (1,0)			+	1	-	_	_
l.							U			5 Time (0)			_		_	
ſ			ing now? er Time	4	102	66	0	Value Type [Ulong]								
-		Trigg	er Time us Time		102 103	66 67	0	Value Type [Ulong] Value Type [Uint]	Focus Ti	Exposure Time (mS) me (mS)						
		Trigg Focu Max	er Time us Time s Shots	4 4	102 103 104	67 68	0	Value Type [Uint] Value Type [Ulong]		me (mS) Count						
		Trigg Focu Max Exposi	er Time is Time : Shots ure Delay	4 4 4	102 103 104 105	67 68 69	0	Value Type [Uint] Value Type [Uint] Value Type [Uint]	Delay	me (mS) Count						
-		Trigg Focu Max Exposi Focus	er Time is Time : Shots ure Delay w Shutter	4 4 4 4	102 103 104 105 106	67 68 69 6A	0 0 0 0	Value Type [Uint] Value Type [Ulong] Value Type [Uint] Value Type [Byte]	Delay 1, 0	me (mS) Count						
		Trigg Focu Max Exposi Focus Mirror Up (	er Time is Time : Shots ure Delay	4 4 4	102 103 104 105	67 68 69	0	Value Type [Uint] Value Type [Uint] Value Type [Uint]	Delay	me (mS) Count						
	Number of shots that have been taken so far fusing the current program.	Trigg Focu Max Exposi Focus Mirror Up ( Interv	er Time ss Time s Shots ure Delay w Shutter Repeat Shot)	4 4 4 4 4 4	102 103 104 105 106 107	67 68 69 6A 6B	0 0 0 0	Value Type [Uint] Value Type [Ulong] Value Type [Uint] Value Type [Byte] Value Type [Byte]	Delay 1, 0	me (mS) Count (mS)						
f	Number of shots that have been taken so far during the current program.	Trigg Focu Max Exposi Focus Mirror Up ( Interv	er Time ss Time ss Time (Shots ure Delay w Shutter Repeat Shot) ral Time nt Shots	4 4 4 4 4 4 4 4	102 103 104 105 106 107 108	67 68 69 6A 6B 6C	0 0 0 0 0 0	Value Type [Uint] Value Type [Ulong] Value Type [Uint] Value Type [Byte] Value Type [Byte] Value Type [Ulong]	Delay 1, 0	me (mS) Count (mS)						
Į.	Number of shots that have been taken so far during the current program.	Trigg Focu Max Exposi Focus Mirror Up ( Interv	er Time us Time Address	4 4 4 4 4 4 4 4 4 5 8 8 8 9 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9	102 103 104 105 106 107 108 109	67 68 69 6A 6B 6C 6D	0 0 0 0 0	Value Type [Uint] Value Type [Ulong] Value Type [Uint] Value Type [Byte] Value Type [Byte] Value Type [Ulong]	Delay 1, 0	me (mS) Count (mS)						
roadcasts	far during the current program.  These function the same as the start, stop, and pause commands above, but can be	Triggg Focus Max Exposos Focus Mirror Up ( Interv	er Time ss Time ss Time s Shots ure Delay w Shutter Repeal Shot) val Time nt Shots  Address	4 4 4 4 4 4 4 4 Sub-Address	102 103 104 105 106 107 108 109 <b>S Command</b>	67 68 69 6A 6B 6C 6D Data Length	0 0 0 0 0 0	Value Type [Uint] Value Type [Ulong] Value Type [Uint] Value Type [Byte] Value Type [Byte] Value Type [Ulong]	Delay 1, 0	me (mS) Count (mS)						
broadcasts	far during the current program.	Trigg Focu Maxax Expose Focus Focus Mirror Up ( Inter- Curre Start Stop	er Time is Time is Time Shots Shots wer Delay w Shutter Repeat Shot) val Time nt Shots  Address  1	4 4 4 4 4 4 4 4 5 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	102 103 104 105 106 107 108 109 5 Command 1	67 68 69 6A 6B 6C 6D Data Length	0 0 0 0 0 0	Value Type [Uint] Value Type [Ulong] Value Type [Uint] Value Type [Byte] Value Type [Byte] Value Type [Ulong]	Delay 1, 0	me (mS) Count (mS)						
oadcasts	far during the current program.  These function the same as the start, stop, and pause commands above, but can be used to synchronize movement of multiple	Triggg Focus Max Exposos Focus Mirror Up ( Interv	er Time ss Time ss Time s Shots ure Delay w Shutter Repeal Shot) val Time nt Shots  Address	4 4 4 4 4 4 4 4 Sub-Address	102 103 104 105 106 107 108 109 <b>S Command</b>	67 68 69 6A 6B 6C 6D Data Length	0 0 0 0 0 0	Value Type [Uint] Value Type [Ulong] Value Type [Uint] Value Type [Byte] Value Type [Byte] Value Type [Ulong]	Delay 1, 0	me (mS) Count (mS)						

Query Value Types	
0	Byte
1	Unsigned Int
2	Int
3	Long
4	Unsigned Long
5	Float
6	String

The floats are actually fixed points. They are multiplied by 100 and transmitted as longs, so they need to be divided by 100 on the master side to resolve the true value.