

SSC32 Registers					
Number	Name	Minimum	Maximum	Default	Description
0	Enable	0	65535	0	A bit field (16 bits) that enables various features of the SSC32.
1	Transmit Delay	0	65535	600	The delay, in microseconds, before transmitting the first byte of a response from the SSC32.
2	Transmit Pacing	0	65535	70	The delay, in microseconds, between bytes in a response from the SSC32.
3-31	(Reserved)				
32-63	Initial Pulse Offset	-100	100	0	The initial value of the pulse offset (PO) for each servo. Register 32 corresponds to servo #0, register 33 to servo #1, etc.
64-95	Initial Pulse Width	0	65535	1500	The initial value of the pulse width for each servo. Register 64 corresponds to servo #0, register 65 to servo #1, etc. A value of 0 leaves the servo output at a continuous logic '0'; a value of 65535 leaves the servo output at a continuous logic '1'. All other values are clipped to the range 500 - 2500 microseconds.
96-255	(Reserved)				
<p>Note: Registers 0-15 are intended for global use, affecting all operation of the SSC32; registers 32-255 are intended for individual servo channel configuration, and so are in groups of 32 registers.</p>					

Enable Register (R0) Bit Definitions		
Bit	Name	Definition
15 (msb)	Global Disable	If '1', disables all of the features controlled by the Enable register; If '0', the individual bit values will be used to enable or disable the features.
14-4	(Reserved)	
3	Initial Pulse Width Enable	If '1', enables the Initial Pulse Width register values at startup; If '0', the default value of 0 will be used.
2	Initial Pulse Offset Enable	If '1', enables the Initial Pulse Offset register values at startup; If '0', the default value of 0 will be used.
1	TX Delay/Pacing Enable	If '1', enables the Transmit Delay and Transmit Pacing values; If '0', the default values of 600 us and 70 us will be used.
0 (lsb)	Startup String Enable	If '1', enables execution of the startup string when power is applied to the SSC32; If '0', the startup string will not be executed.

Register Read/Write			
Command	Argument	Description	Examples
Register write: R <r> = <n> <cr>	r = reg number, 0-255 n = value	Programs the value of a register. Spaces are optional around the register number and value.	R0=1023 <cr> R32 = -50 <cr>
Register read: R <r> <cr>	r = reg number, 0-255	Displays the value of a register, followed by a carriage return. The displayed value is in ASCII format, and is terminated with a carriage return.	R0 <cr> result: 1023<cr> R32 <cr> result: -50<cr>
Set to defaults: RDFLT <cr>	none	Sets all of the registers to their default values. When the command is complete the SSC32 will transmit the string OK<cr>.	RDFLT <cr> result: OK<cr>
Note: The RDFLT command may take over a second to execute. It should not be invoked while a timed move or sequence player is active. No register writes should be performed until the RDFLT is complete (as indicated by the OK response).			
Note: If multiple R= commands are being sent by software, it is recommended that the software read the value of each register after it is written. This will ensure that each register write has completed before the next is started.			
Note: If an RDFLT or R= command is executing, do not power down the SSC32 until the command has completed. To determine whether the command has completed, read a register value.			
Note: Each time a register is written, the EEPROM location(s) used to store the value experience a small amount of wearout. The typical maximum number of writes is 100,000. Do not write your software to rapidly change the register values, or you might cause permanent wearout of the EEPROM in the ATmega168 processor.			

Misc Commands			
Command	Argument	Description	Examples
STOP <n> <cr>	0-31	Immediately stops the specified servo at its current position. A space is optional before the servo number.	STOP0 <cr> STOP 31 <cr>
Note: If the servo is part of a timed move, the other servos will continue moving and a query command will indicate that the move continues until the total time for the original move has elapsed. This is true even if <i>all</i> of the servos in the original move are stopped.			
Note: The EER and EEW commands no longer work to access internal EEPROM. They are replaced by Register Read/Write and Startup String commands. EER and EEW continue to function for external EEPROM.			

Startup Strings			
Command	Argument	Description	Examples
Delete characters: SSDEL <n> <cr>	0-255	Deletes <n> characters from the end of the startup string. If <n> is greater than the length of the startup string, then SSDEL deletes the entire string.	SSDEL 5 <cr> - Deletes the last 5 characters of the startup string SSDEL 255 <cr> - Deletes the entire startup string
Concatenate: SSCAT <string> <cr>	Up to 100 ASCII characters	Concatenates <string> to the current startup string. The blank space immediately following "SSCAT" is ignored, but all other spaces are part of the string. The string is terminated by a carriage return, and may not contain embedded carriage returns. Commands in the startup string are terminated with a semicolon (including the last command).	SSCAT #0P1000#1P2000T3000;<cr> SSCAT PL0 SQ5 SM50;<cr>
Display startup string: SS <cr>	none	Displays the entire startup string, surrounded by quotation marks and followed by a carriage return.	SS <cr> result: "#0P1000#1P2000T3000;PL0 SQ5 SM50;"<cr>
Note: The programmed startup string is executed at powerup of the SSC32, if the Startup String Enable bit is set in the Enable register. The startup string is executed after any register values are applied (e.g. initial pulse width).			
Note: The maximum total length of the startup string is 100 ASCII characters. Any additional characters will be ignored.			
Note: The following commands should not be used in a startup string: EER, EEW, R=, SSCAT, SSDEL			
Note: The SS command may take hundreds of milliseconds to execute, depending on Baud rate. It should not be invoked while a timed move or sequence player is active.			
Note: The SSCAT command may take hundreds of milliseconds to execute. It should not be invoked while a timed move or sequence player is active.			
Note: If an SSDEL or SSCAT command is executing, do not power down the SSC32 until the command has completed. To determine whether the command has completed, send an SS command and wait for the response.			
Note: Each time the startup string is changed the EEPROM locations used to store the value experience a small amount of wearout. The typical maximum number of writes is 100,000. Do not write your software to rapidly change the startup string, or you might cause permanent wearout of the EEPROM in the ATmega168 processor.			

Examples	
Command	Result
SSDEL 255 <cr>	
SS <cr>	""<cr>
SSCAT #0P2000T5000;<cr>	
SS <cr>	"#0P2000T5000;"<cr>
SSCAT XXXX<cr>	
SS <cr>	"#0P2000T5000;XXXX"<cr>
SSDEL 4 <cr>	
SS <cr>	"#0P2000T5000;"<cr>
SSDEL 6 <cr>	
SS <cr>	"#0P2000"<cr>
SSCAT #1P1000T4000;PL0SQ5;<cr>	
SS <cr>	"#0P2000#1P1000T4000;PL0SQ5;"<cr>

**Additional Examples**

<b>Command</b>	<b>Description</b>
RDFLT	Set all registers to default values
SSDEL 255	Erase the startup string
R0	Display register 0
R0=2 R1=2000 R2=1000	Set TX delay to 2000us and TX pacing to 1000us. ( R0=2Bit 1 of R0 enables TX delay/pacing.)
R0=12 R32=50 R64=1000	Set the pulse offset for servo 0 to 50 and the initial pulse width to 1000. (Bits 2&3 of R0 enable pulse offset and pulse width.)
R0=13 SSDEL 255 SSCAT #0P1500T5000;	Move R0 slowly to a pulse width of 1500 at startup. Assume the initial pulse width is set as in the previous example. (Bit 0 of R0 enables the startup string.)
SS	Display the current startup string.