MicroTech

MT Servo Sequencers

<Please also study SC.pdf>

Version 3.00

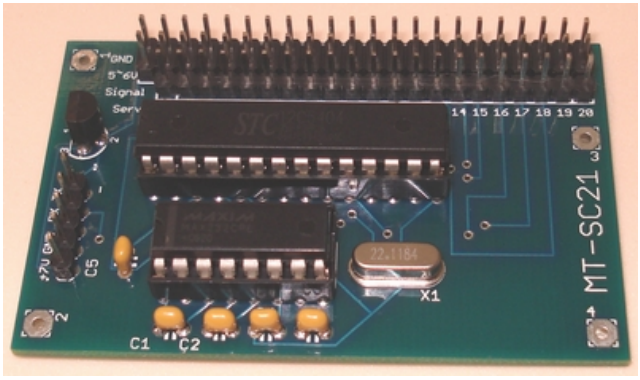
www.mcu.hk

Warning:
Incorrect power connection to any electronic and electrical equipment may seriously damage them or even cause a fire hazard or explosion. Users must take care to identify the correct pins and supply an acceptable voltage to operate them safely.

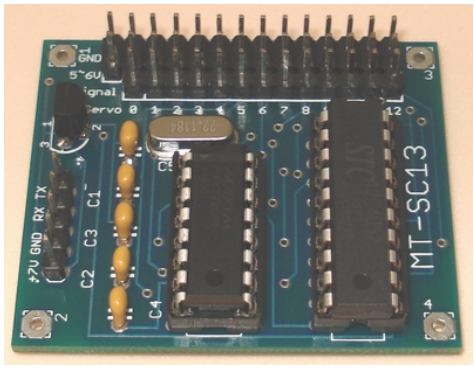
October, 2010

Important Safety Warning

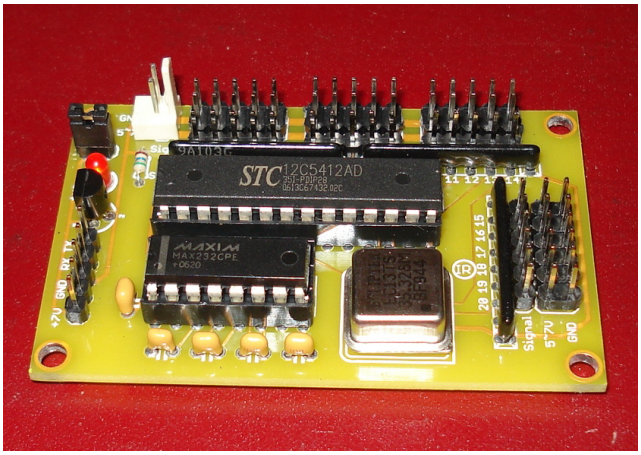
This kit is not intended for young children! Assembly of this kit requires high-temperature soldering and the use of sharp cutting tools. Some included components may become hot, leak, or explode if used improperly. MicroTech strongly recommends that you wear safety glasses when building or working with any electronic equipment. Children should use this kit only under adult supervision. By using this product, you agree not to hold MicroTech liable for any injury or damage related to the use or to the performance of this product. This product is not designed for, and should not be used in, applications where the malfunction of the product could cause injury or damage.



MT-SC21SQ



MT-SC13SQ



MT-SC21SQ-35M

Product Features

1. Parallel signal output mode, all the servo signals go High at the same time
2. Independent servo operation, each servo has it's own set of speed and position settings
3. Individual servo speed control, i.e. different servos are capable of moving at different speeds simultaneously
4. Selectable baudrate from 600 to 19200
5. Command to retrieve servo position
6. Control up to 21/13 servos
7. 24 (MT-SC21SQ), 39 (MT-SC13SQ) storages for servo speeds and positions
8. Simple commands to operate
9. Small board size 65x41mm (MT-SC21SQ), 46x38mm (MT-SC13SQ)
10. 5K bytes memory for up to 256 sequences data storage
11. Integrated with Infra-Red commands

Introduction

The MT Servo Sequencers are the upgrades of MT Servo Controllers with servo commands storage and executing functionality. These servo moving commands are called sequences. These Servo Sequencers share the same PCBs and components with their Servo Controllers counterparts, but equipped with more advanced different microcontrollers. Since the Sequencers are the supersets of the Controllers, users are advised to study the manual of MT Servo Controllers (SC.pdf), and this document focused on those parts that are specific to Sequencers only.

In order to do the programming task easier, a software called MT Servo Controller/Sequencer Console (SQ.exe or SQ2.exe) is given, which is a text editor as well as a downloader to program those sequencing commands for MT-SC21SQ and MT-SC13SQ Servo Sequencers to execute commands in situ, without the need to receive servo moving commands directly from a host PC or microcontroller.

The Servo Sequencer works in 2 modes, normally it works in the Controller mode, in which it just behaves the same as a normal servo controller. As soon as it receives the 'Play' command or being set to 'AutoPlay' mode at reboot, it then jumps into the Player mode to execute the commanding sequences previously downloaded. The Servo Sequencer will remain in this mode until it encounters the 'STOP' command or has reached the end of playing sequences, then it returns back to the normal Controller mode. When the Servo Sequencer is in the Player mode, the only command user can send to it is the 'Stop Play' command, in which this will interrupt the Servo Sequencer and returns it back to the normal Controller mode.

The reason to have the 'STOP' commands between sequences is that user can send command (233 x) to execute from a particular sequence, stops, and returns to the Controller mode, not necessary from the first sequence and execute all the sequences. This is useful during development and debugging.

Controlling Commands

The followings describe the functions for the command buttons on the Servo Sequencer controlling console.

```
@Seq = X,  indicates the current selected active sequence.
Total Seq, indicates the total number of non-empty sequences.

Clear     = Clear current sequence entry.
Copy      = Copy current sequence to buffer.
Paste     = Paste buffer to current sequence.
Insert    = Insert an empty sequence before the current sequence.
Pack      = Pack and remove all the empty sequences.
Check     = Check and verify all sequences syntax.
Send      = Send sequences to Servo Sequencer's memory.
Get       = Retrieve sequence from Servo Controller's memory.
Play      = Execute sequences on Servo Sequencer.
Stop      = Send command to Servo Sequencer to stop playing sequences.
=>>Set    = Set the activated settings on Servo Console to current sequence.
<<=Set    = Set the current sequence settings to Servo Console.
Open      = Open and load sequences from a data file.
Save      = Save sequences to a data file.
Calc      = Calculate the total of non-empty sequences.
AutoPlay  = Set the Servo Sequencer to automatically play sequences at reboot.
XAutoPlay = Disable the Servo Sequencer's autoplay setting.
Optimize  = This command optimizes codes by removing those servo entries in which
            they didn't changed from previous immediate sequences. Since this
            operation is 'sequence order dependent', be careful!
```

Here are the command byte codes for all the commands for controlling the Servo Sequencer by a microcontroller, as well as by the Servo Sequencer Console:

```
230     = Download sequences to Servo Sequencer, "Send" command.
231     = Retrieve sequences from Servo Sequencer, "Get" command.
232     = Start executing from sequence 0, "Play" command.
233 x   = Start executing from sequence x, x=0..255.
234     = Interrupt and stop playing sequences, "Stop" command.
235 x   = (DLY) Delay for x units, x=0..255.
236 x   = (GOTO) Goto x sequence (or call a set of sub-sequences), x=0..255.
237     = (STOP) Stop command, exit Player mode.
238 1   = Set AutoPlay flag to ON, "AutoPlay" command.
238 0   = Set AutoPlay flag to OFF, "XAutoPlay" command.
239 y   = (SPA) Set all servos to speed y, y=0..5.
250 x y = (SPD) Set servo x to speed y, x=0..32, y=0..5.
229 x y = (IR) Set IR command, x=0..9, y=0..255.
253 x   = (RTV) Retrieve from position storage x, x=0..38.
254     = (RET) Return from sub-sequences.
```

Pseudo codes for downloading sequences into the Servo Sequencer by a microcontroller:

```
Send 230       // command code to initiate downloading
Delay(1000);     // wait for 1 second for servo sequencer to clear it's flash memory
FOR each seq in all the sequences DO
  Send <num>     // num=number of bytes in current sequence, when num=<0>, terminate downloading
  FOR i=1 to num DO
    Send seq[i]; // send the i^th byte
    // servo sequencer will echo back each byte received to sender for verification
  ENDFOR
ENDFOR
```

Sub-Sequences and IR Commands

By using the combination of GOTO, IR and RET commands, users can program the sequences into different sections, therefore, different sets of sequences can be invoked and executed. The last command within these sub-sequences must be the RET command, so that the Player can returns to the point where the sub-sequences is being called. Be aware that, since the Player is not capable to reentrant (i.e. calling a set of sub-sequences within another sub-sequences), users must wait for the called sub-sequences to finish before calling another set of sub-sequences.

The Sequencers will automatically turn on the IR command mode whenever an IR command is present in the playing sequences. In this mode, the IR pin will be set as input for IR beam detection and the PWM output on this pin is disabled. The format for the IR beam is NEC or JVC compatible, so that the numeric value of 1 corresponds to key '1' on a remote controller. For the current implementation, only numeric values of 0 to 9 are handled. If a numeric value is received but not defined, then it will set the Player to jump to sequence 0.

The following example (in file "SQ-IR.txt") illustrates the use of IR commands:

```
{0} IR,1,4,IR,2,5,IR,3,6,
{1} S0,0,DLY,10,
{2} S0,180,DLY,10,
{3} GOTO,1,
{4} S0,70,DLY,30,RET,
{5} S0,90,DLY,30,RET,
{6} S0,120,DLY,30,RET,
```

The above program simply defines three IR commands which will jump to sequences, 4, 5 and 6, if the defined numeric values of 1, 2 and 3 are received, respectively. Normally, the 'servo 0' moves to position 0 and 180 back and forth repeatedly, but as soon as an IR command is received, the program jumps to the sub-sequences as specified, which set the servo to move to position 70, 90 or 120. It is important to add the RET commands at the ends to signify the ends of these sub-sequences. Since the Player is NOT capable to reentrant, to avoid

deadlock, do not issue another IR command until the current IR command is finished. However, if in the deadlock situation, simply issue a non-defined numeric number (such as 0) to jump to sequence 0.

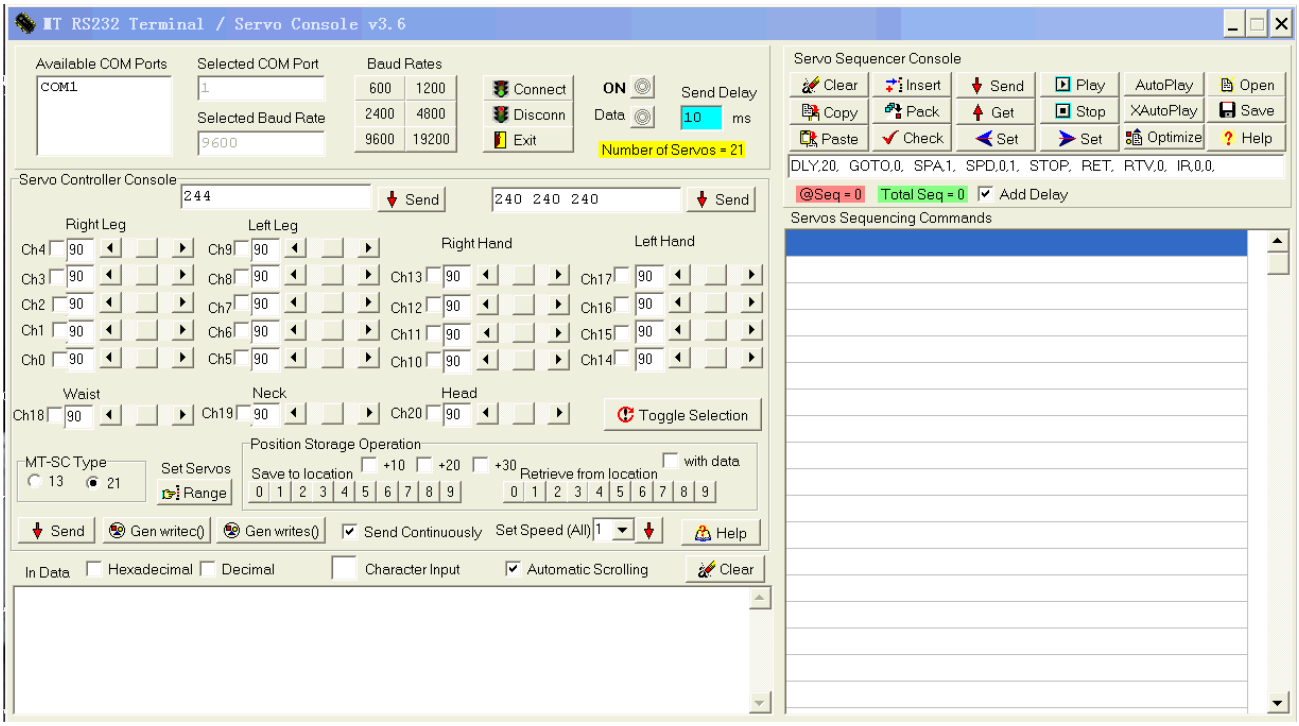
`NOTE: The IR pin for MT-SC21SQ is "Servo 18", whereas for MT-SC13SQ is "Servo 12".`

Please refer to document, "MT chips.pdf" for how to interface with an IR detector.

Testing and Usage

By using the MT Servo Controller/Sequencer Console (SQ.exe or SQ2.exe), user can send test data to the servo sequencer to move the servos. The numbers within the text input boxes are the ASCII codes that will send to the sequencer. Note that, spaces between the numbers are skipped and ignored.

User can also simply use the sliding bars within the "Servo Controller Console" section to move servos interactively to determine the positions of all the required servos contributing to a particular robotic gesture.



Note: Do select the correct servo sequencer type before issuing any command. Do tick the 'Add Delay' box to append a delay of 'DLY,20,' to the end of servo commands, when the '=>>Set' command is clicked.

Example commands

Initialize all the position storage and sequence data memory
240 240 240

Retrieve the sequencer's firmware information.
244

Move servos 0, 1 and 2 to position 30
0 30 1 30 2 30

Move servos 0, 1 and 2 to position 150
0 150 1 150 2 150

Set servos 0 and 1 to moving speed 1 (slowest)
250 0 1 250 1 1

Save current settings to memory storage location 4
252 4

Retrieve settings from memory storage location 4
253 4

Execute from sequence 3
233 3

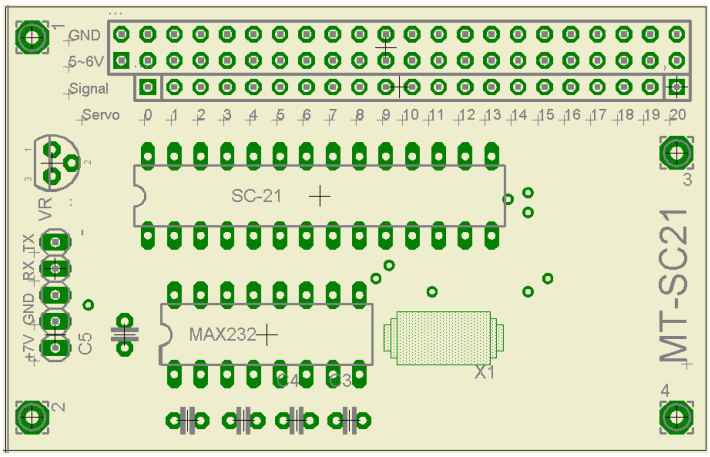
Set baudrate to 1200,8N1
246 12

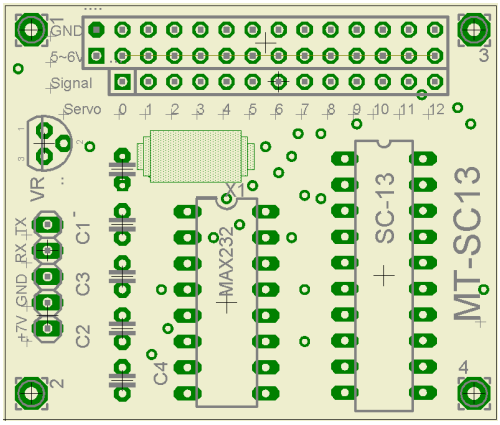
After issuing this command, the sequencer starts operating in this new baudrate setting, you need to re-select the Terminal's baudrate to 1200 (don't forget to press the "Connect" button as well), in order be able to communicate with the sequencer again.

Adding Comments to Sequences

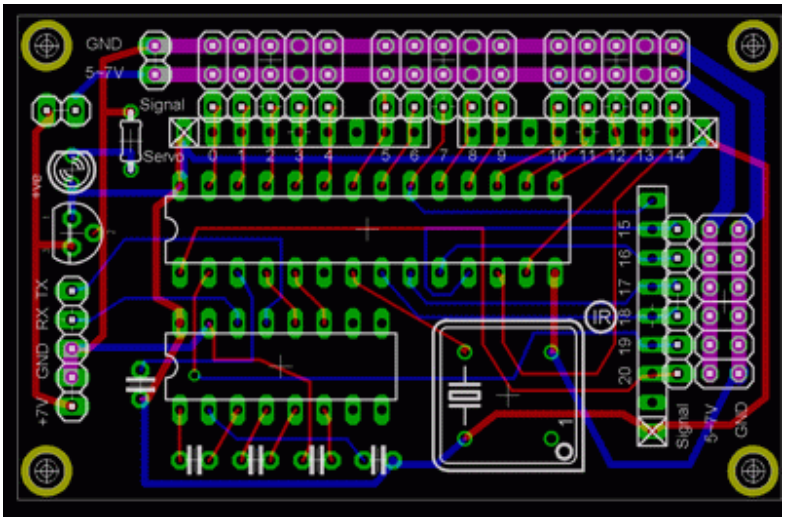
Texts within the { } brackets are treated as comments and not considered parts of the sequences, however, these comments will NOT be saved into the Servo Sequencers to save precious memory. Therefore, these comments will be lost if the sequences are being read back from the Sequencer. Users are advised to keep the originals of sequences files. Besides, the editing buffer for each sequence can takes up to a maximum of 255 characters only.

Pinouts and Board Layout





NOTE: The IR pin for MT-SC21SQ is "Servo 18", whereas for MT-SC13SQ is "Servo 12".



Optionally connect the jumper to power the board ICs from the power supply to servos.

Please refer to document, “MT chips.pdf” for how to interface with an IR detector.