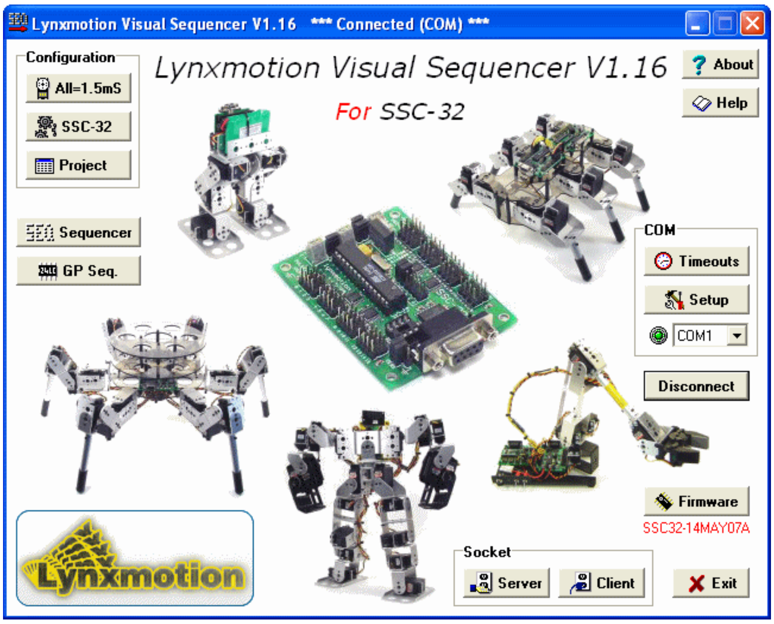
Lynxmotion Visual Sequencer V1.16

For SSC-32

Manual



Author and programmer : Laurent Gay - lynxrios@yahoo.fr

Lynxmotion Visual Sequencer V1.16 Manual

Safety First!

Read and understand the documentation associated with any of the tools used in the assembly of these kits. Work in a clean, well lit environment. Work slowly taking breaks often. Plan your work with plenty of extra time to avoid cramming to complete the project at the last minute. Lynxmotion, Inc. has taken every step to ensure the products sold are safe when used in a responsible manner. Therefore, Lynxmotion, Inc. can not be held accountable for irresponsible, careless or reckless behavior of the builder.

These kits are purely educational. Items sold by Lynxmotion, Inc. are not authorized for use in human contact, medical, life-saving, life-support, industrial or light industrial applications. Do not under any circumstances use these robots to move, touch, or handle dangerous or hazardous materials. Doing this could result in injury or death to the user, or damage to property.

Robots move without warning, wear eye protection at all times!

Lynxmotion, Inc. is not responsible for any special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs of recovering, reprogramming, or reproducing of data associated with the use of the mechanics, hardware or software it sells.

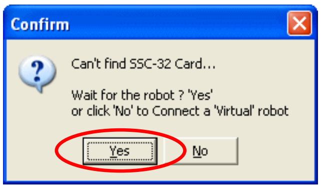
Table of contents

Lynxmotion Visual Sequencer V1.16 Manual

1 - Configuration of the SSC-32 card

Step 1. Install the Visual Sequencer program. Insert the CD into your CD drive and follow the instructions.

The controller card is NOT powered from the Serial or the Serial to USB cable. It will be necessary to apply servo power to complete the setup. Connect the Serial or the Serial to USB cable, wait for the system to recognize the card and run Visual Sequencer. If you run Visual Sequencer before the system recognizes the card, you will see the message box in Figure 1. Click 'Yes', and the program will enable the grayed buttons when the card is ready.



Confirm

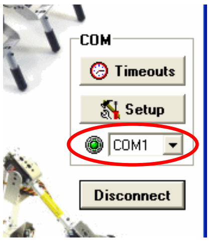
Can't find SSC-32 Card...

Wait for the robot ? 'Yes'
or click 'No' to Connect a 'Virtual' robot

If the card is not detected, select the right COM port number in the list box.

If you don't know the COM port number you are using :
- Just try from COM1, COM2 etc…
  when you find the right one, the card will auto-connect.

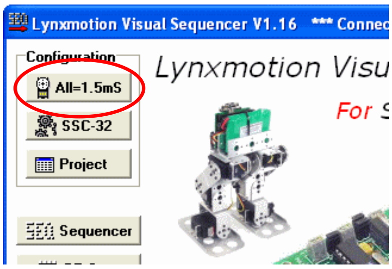
If the COM port you are using is not listed (USB to Serial cable issue) :
- Make sure your USB to Serial cable is connected.
- Wait for Windows to recognize it.
- Click on the 'Setup' button and select your COM port in the Setup form.



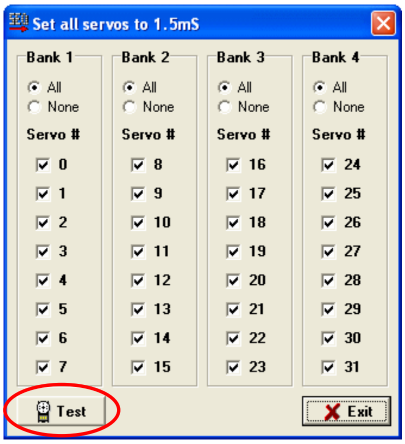
Check that the SSC-32's firmware is set to SSC32X-ER V1.16 or later.

Step 2. Test servos

Generate a 1.5mS pulse for all servos (neutral) :
- Click on the 'All=1.5mS' button.



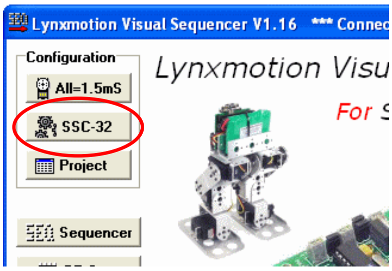
- Select the servos you want to test
- Click on the 'Test' button then 'Yes' on the warning message.
- Click on the 'Stop' button to return to the previous robot state.



Set all servos to 1.5mS

Bank 1	Bank 2	Bank 3	Bank 4
● All	● All	● All	● All
○ None	○ None	○ None	○ None
Servo #	Servo #	Servo #	Servo #
☑ 0	☑ 8	☑ 16	☑ 24
☑ 1	☑ 9	☑ 17	☑ 25
☑ 2	☑ 10	☑ 18	☑ 26
☑ 3	☑ 11	☑ 19	☑ 27
☑ 4	☑ 12	☑ 20	☑ 28
☑ 5	☑ 13	☑ 21	☑ 29
☑ 6	☑ 14	☑ 22	☑ 30
☑ 7	☑ 15	☑ 23	☑ 31

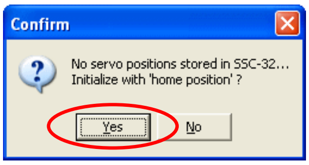
Step 3. To enter the SSC-32 configuration screen, click on the 'SSC-32' button in the upper left corner.



Lynxmotion Visual Sequencer V1.16 Manual

1 - Configuration of the SSC-32 card

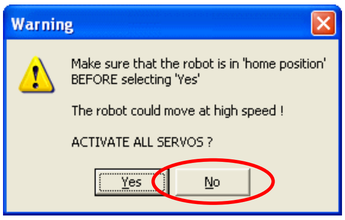
Step 4. Click 'Yes' on this window.



Confirm

No servo positions stored in SSC-32...
Initialize with 'home position' ?

Step 5. Click 'No' on this window.



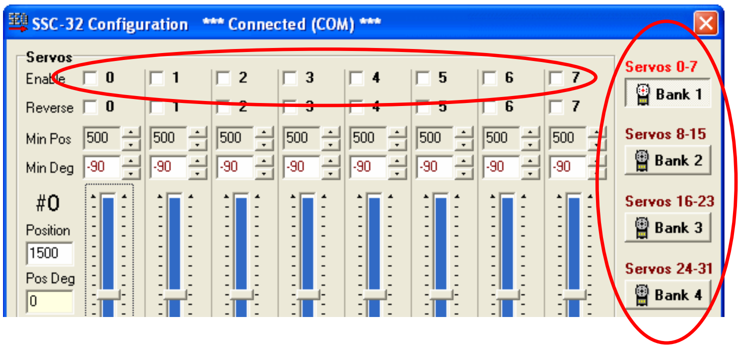
Warning

Make sure that the robot is in 'home position'
BEFORE selecting 'Yes'

The robot could move at high speed !

ACTIVATE ALL SERVOS ?

Step 6. Select each 'bank' and make sure the boxes in the top row are NOT CHECKED, then switch the robot power ON.



SSC-32 Configuration *** Connected (COM) ***

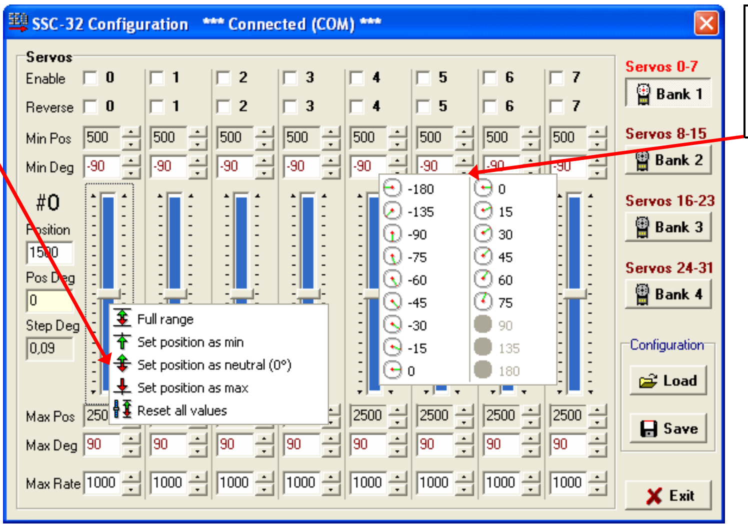
Servos
Enable □ 0 □ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Reverse □ 0 □ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Min Pos 500 500 500 500 500 500 500 500
Min Deg -90 -90 -90 -90 -90 -90 -90 -90
#0
Position
1500
Pos Deg
0
Servos 0-7
Bank 1
Servos 8-15
Bank 2
Servos 16-23
Bank 3
Servos 24-31
Bank 4

Step 7

Step 7.1 Overview.

Before continuing, let's take a look at the SSC-32 configuration form.

To quickly adjust the 'Min Pos' and the 'Max Pos', move the slider to the position and right click to show the popup menu.



SSC-32 Configuration *** Connected (COM) ***

Servos
Enable □ 0 □ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Reverse □ 0 □ 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7
Min Pos 500 500 500 500 500 500 500 500
Min Deg -90 -90 -90 -90 -90 -90 -90 -90
#0
Position
1500
Pos Deg
0
Step Deg
0,09
Full range
Set position as min
Set position as neutral (0°)
Set position as max
Reset all values
-180
-135
-90
-75
-60
-45
-30
-15
0
0
15
30
45
60
75
90
135
180
Max Pos 250 2500 2500 2500 2500
Max Deg 90 90 90 90 90 90 90 90
Max Rate 1000 1000 1000 1000 1000 1000 1000 1000
Servos 0-7
Bank 1
Servos 8-15
Bank 2
Servos 16-23
Bank 3
Servos 24-31
Bank 4
Configuration
Load
Save
Exit

To quickly adjust the 'Min Deg' and the 'Max Deg', right click on the "Min/Max Deg" edit box to show the popup menu.

Full range :
Min Pos = 500
MaxPos = 2500

Set Position as min
Min Pos = slider position

Set Position as max
Max Pos = slider position

Reset All values :
Min Pos = 500
MaxPos = 2500
Min Deg = -90
Max Deg = 90
Max Rate = 1000

'Min Deg' and 'Max Deg' values can be entered directly by typing in the white boxes

The 'Max Rate' value
is the servo' speed limitation.
If you try to move faster
while building a project,
the servo's box will turn yellow
(Sequencer form)

You can directly enter the
'Max Rate' Value by
typing in the white boxes.
Range is 0 - 8000 :
- 0 = full speed
- 1 = very slow
- 8000 = very fast

Set Position as neutral (0)) : assign the slider position to the 0°value.

Lynxmotion Visual Sequencer V1.16 Manual

1 - Configuration of the SSC-32 card

Step 7

Step 7.2 Enable the servos you need one by one

For each servo you're using :

- Move the slider to the middle.
- Click on the 'Enable' checkbox to activate the servo. (Warning: It can move fast!)

- Move the slider to the Robot's neutral position for this servo. Normally this will be very close to the 1500 position.
- Right click on the current slider to show the popup menu and click 'Set position as Neutral (0°)'.

- Move the slider carefully to the minimum position that the robot can handle… Don't allow the servo to crash!
- Right click on the current slider to show the popup menu and click 'Set position as min'.
- Use a protractor to measure the angle from the Neutral position and enter this value into the corresponding Min Deg box using the the up/down arrows (use a negative value).

- Move the slider carefully to the maximum position that the robot can handle… Don't allow the servo to crash!
- Right click on the current slider to show the popup menu and click 'Set position as max'.
- Use a protractor to measure the angle from the Neutral position and enter this value into the corresponding Max Deg box using the the up/down arrows (use a positive value).

- Move the slider again to the position you want to set as 'home position'.
  (Most of the time it will be 0°, but you can use any position you want.)

Step 8. Save your progress

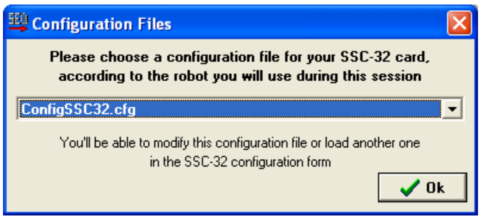
- Click on the 'Save' button and select a file name (ConfigSSC32.cfg is default).

- The sliders' current positions will be saved as the robot's 'home position'.

- The 'ConfigSSC32.cfg' file is loaded by default each time you run Visual Sequencer.
  If you will only have one configuration file (one robot), you can use this filename.

- The 'ConfigSSC32.cfg' file must be saved to the installation directory in order to be found by the software.

- If you have saved several "yourconfigfilename.cfg" files (in the installation directory),
  the software will show this window at startup :



Configuration Files

Please choose a configuration file for your SSC-32 card,
according to the robot you will use during this session

ConfigSSC32.cfg

You'll be able to modify this configuration file or load another one
in the SSC-32 configuration form

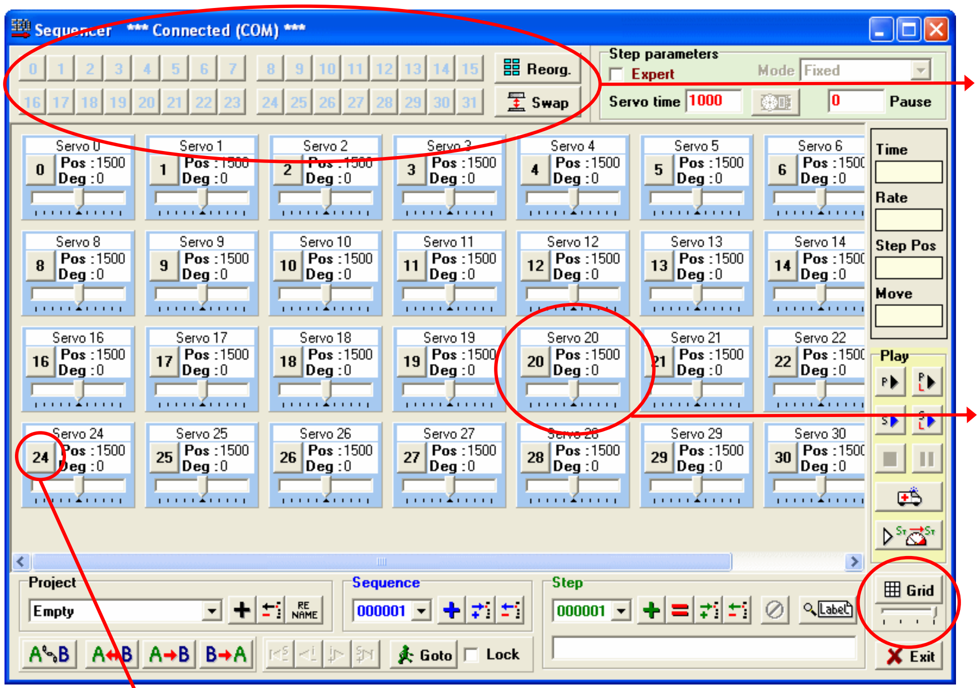
- If you have saved some "yourconfigfilename.cfg" files in another location than the installation directory,
  they won't be listed in this window, you'll have to load them manually in the SSC-32 config form module.

- Configuration files contain every SSC-32 configuration form's parameters, even the trackbars position (home position),
  so, use as many cfg files as you have robots.

- Configuration files prevents your robot from moving to an unsafe position
  and give the angle ratio by joint, allowing to share csv exported programs with other users.

2 - The 'Sequencer' module

2.1 Servo boxes.

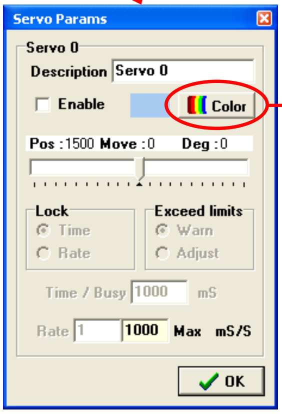


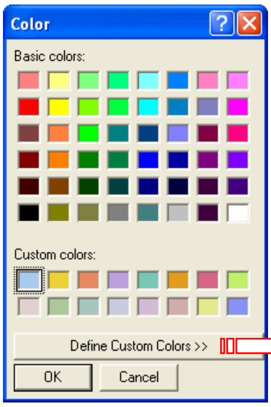
- Hide/Show servo boxes with the buttons.
- Swap between hidden/shown boxes with the 'Swap' button.
- Use the 'Reorg.' button to reorganize all the boxes to default positions and to return to default color and name.

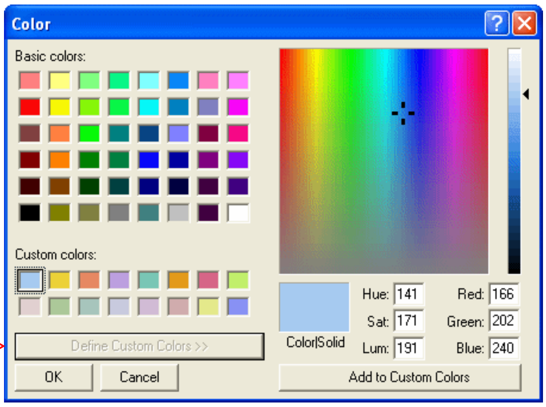
- Click a box and keep pressed to move and build your robot shape.

- Click on the 'Grid button' to display an 8X8 dot grid for easily align boxes.
- Use the slider to choose the : 1X1, 2X2, 4X4 or 8X8 positioning grid. (The displayed grid will stay 8X8)

- Click the button on a box to show its extended menu.



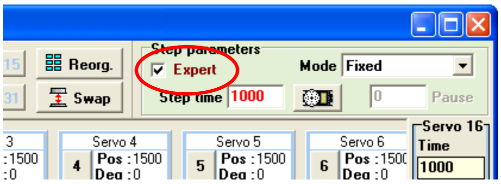


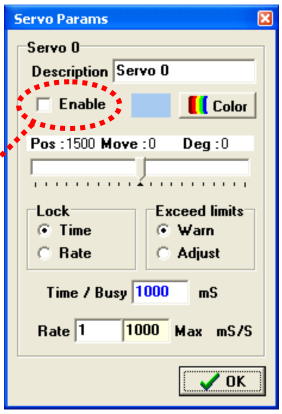


Hue: 141 Sat: 171 Lum: 191 Red: 166 Green: 202 Blue: 240

You can :
- Change the servo description.
- Enable or disable servo.
- Change the box color.
- Move the servo with a bigger slider.

If you have enabled the 'Expert' mode, the 'servo params' menu will allow to :
- Use a different servo time.
- Use a different servo rate
- Choose to lock the time or the rate
- Choose to warn or auto adjust, when rate/time limits are exceed



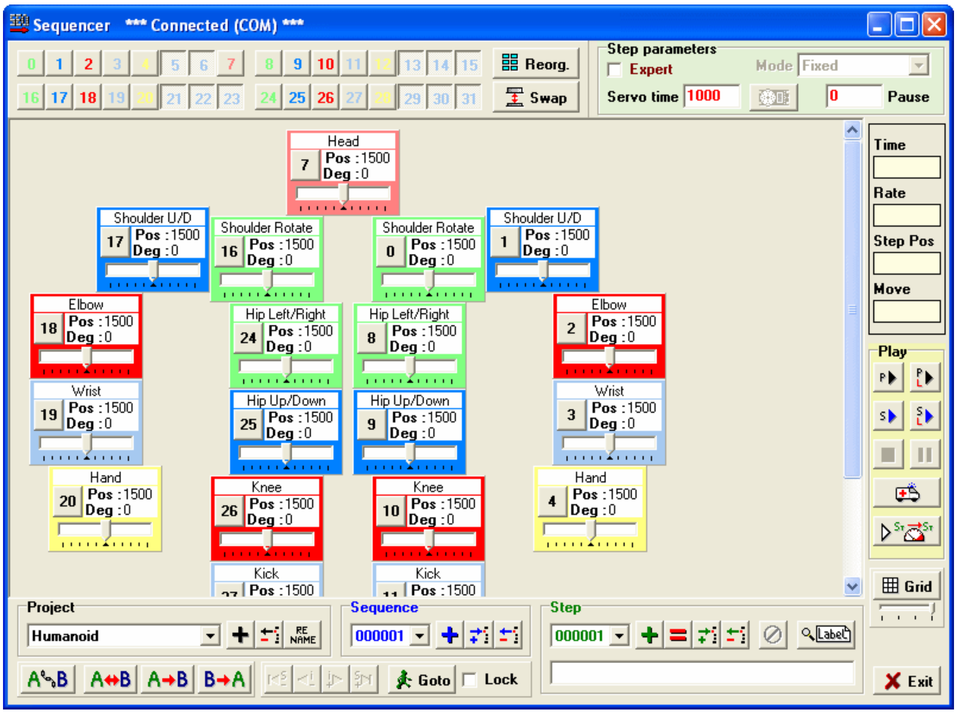


Don't forget to enable the servos you want to use in the 'Servo Params' menu

2 - The 'Sequencer' module

2.1 Servo boxes.

Here is an example :



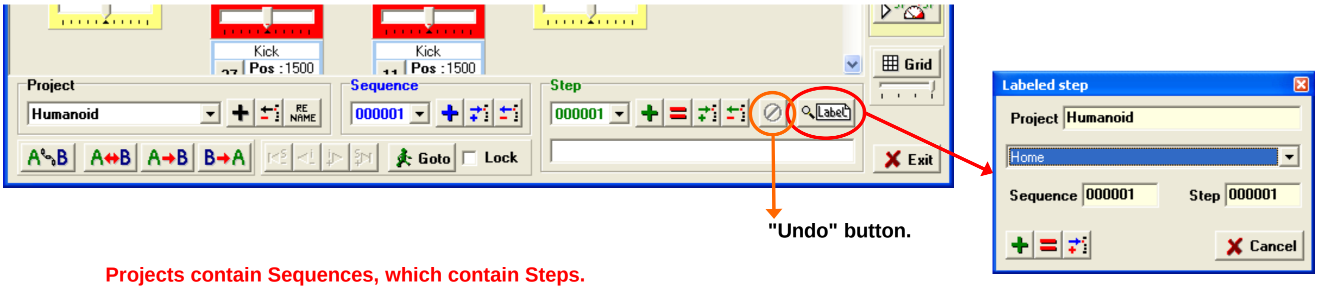
Sequencer *** Connected (COM) ***

Step parameters: □ Expert; Mode Fixed; Servo time 1000; Pause 0

Project Humanoid; Sequence 000001; Step 000001

2.2 Storing data.

2.2.1 Overview.



"Undo" button.

Projects contain Sequences, which contain Steps.

You can store the current robot position as a 'Step' using the 'plus sign' icons :
- The Project black 'plus sign' icon will append this step in a new Project.
- The Sequence blue 'plus sign' icon will append this step in a new Sequence of the current Project.
- The Step green 'plus sign' icon will append this step in the current Sequence of the current Project.

Projects :
- You can 'Append', 'Delete' or 'Rename' Projects.
 For more about Projects, see the 'Project module'.

Sequences :
- You can 'Append', 'Insert' or 'Delete' Sequences.

Steps :
- You can 'Append', 'Overwrite', 'Insert' or 'Delete' Steps.
- You can add a Label to a Step, it allow to recall it by its name, for append, overwrite or insert.

Use the combo-boxes to select a Project/Sequence/Step or use the four S-, I-, I+ and S+ icons to navigate through Sequences and Steps.

Click 'Goto' to move the robot to the selected Step.
Check 'Lock' to 'auto move' the robot to the selected step while browsing them.

2 - The 'Sequencer' module

2.2 Storing data.

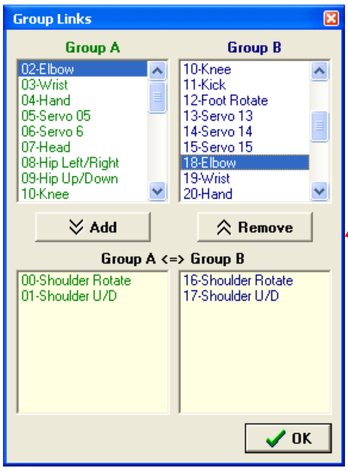
2.2.2 Advanced.

If you want to create a new project with the same Block boxes arrangement (robot shape) as another project :
- Select the 'template' project you want to use.
- Click the black 'plus' sign in the Project box to add as a new project.
- Enter the new project name in the popup window and click 'OK'.

To use the 'Label' system :
- Select a Step to Label.
- Type the Step name into the Label edit box.

To add another instance of a Labeled Step :
- Select the Step where you want to use the Labeled one.
- Click on the Label button and select the Labeled Step you want to use.
- Click on the Labeled step window Add, Overwrite or Insert button.

2.2.3 Group.

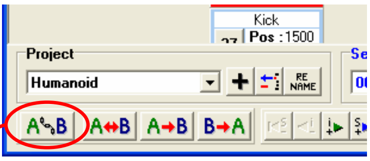


Group Links

Group A
02-Elbow
03-Wrist
04-Hand
05-Servo 05
06-Servo 6
07-Head
08-Hip Left/Right
09-Hip Up/Down
10-Knee

Group B
10-Knee
11-Kick
12-Foot Rotate
13-Servo 13
14-Servo 14
15-Servo 15
18-Elbow
19-Wrist
20-Hand

Group A <=> Group B
00-Shoulder Rotate | 16-Shoulder Rotate
01-Shoulder U/D | 17-Shoulder U/D



The benefit of Group system is the ability to apply effect to it
Group is defined by project

To create a Group link:
- Select a servo in the 'Group A' box.
- Select the servo to link in the 'Group B' box
- Click 'Add'.

…and so on.

To delete a link :
- Select a Link in the 'Group A <=> group B' box.
- Click 'Remove'.

2.2.4 Group and Effect.

When a group is defined, you can apply an effect to it.
- Select Step you need to modify.
- Click on the effect button you need.
- **Select the target of the resulting effect :**
- Selected step => overwrite the current step.
- Robot position => move the robot to new position.
- Click 'Apply'.

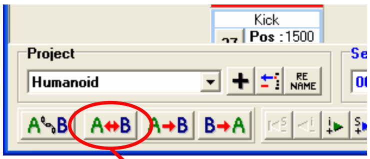
In any case, the data source for the effect is the Selected Step.
The result is stored in the same Step or just apply to Robot, or both.

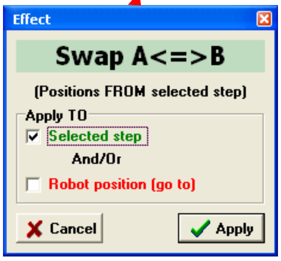
you can use the 'Labeled step' system to add an existing step to the project, and then, modify it with an effect.

For now, 3 effects can be used, Swap A<=>B, Copy A=>B, Copy B=>A.
Use 'Swap A<=>B' to build a mirror effect (bipeds legs).
Use 'Copy A=>B' or 'Copy B=>A' to build a 'Clone' effect (bipeds legs or arms)

So, 'Swap A<=>B' exchange the positions of the 'Group A' servos with the positions of the 'Group B' servos.

'Copy A=>B' or 'Copy B=>A' copy the positions of one Group to the other one.





Effect

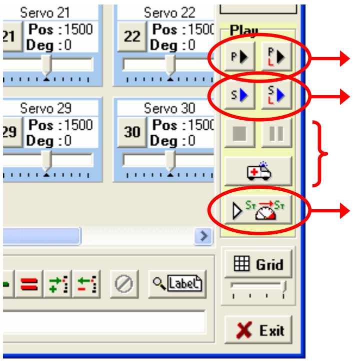
Swap A<=>B

(Positions FROM selected step)

Apply TO
☑ Selected step
And/Or
□ Robot position (go to)

2 - The 'Sequencer' module

2.3 Play.



- Play the Project selected 'once' or with 'auto loop'.

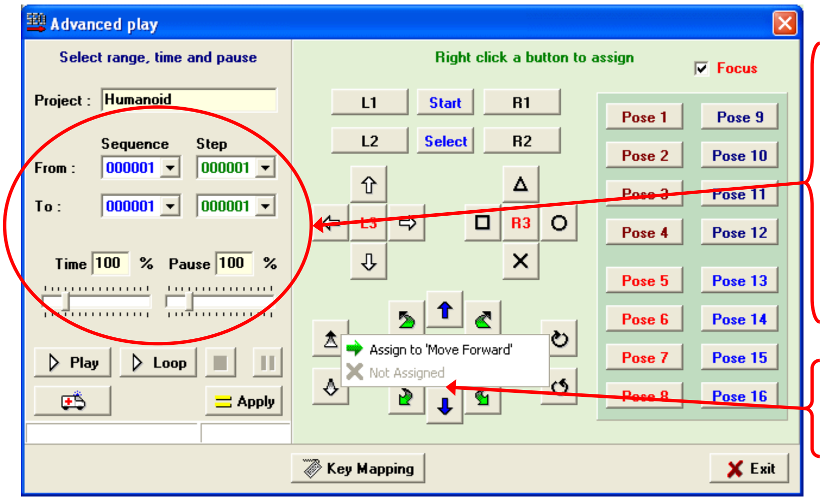
- Play the Sequence selected 'once' or with 'auto loop'.

While playing, you can 'Stop' or 'Pause/Play'.
Emergency button stops the robot and disables all servos torque.
(Warning, digital servos can't be disabled!)

- Open the 'Advanced play' Module (see next chapter).

3 - The 'Advanced play' module

3.1 Range.



Project : Humanoid

From : 000001 000001

To : 000001 000001

Time 100 % Pause 100 %

- Select the start sequence and step.
- Select the end sequence and step.
- Make sure the sliders are set to 100%.

- Click on the 'Play' button.

- While Playing, you can adjust the 'Time' slider, it modifies the servos 'moving time' (speed) in real time.
- You can adjust the 'pause' slider too, it modifies the pause (if there is one) between steps.

- The two sliders can be adjusted from 25% to 400%.

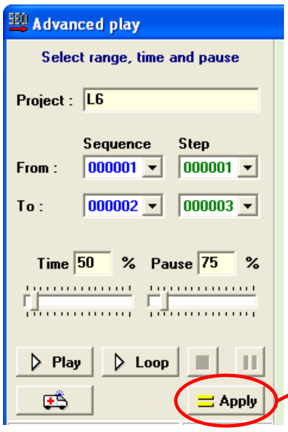
- Right click on a button to assign these settings to it.
- Then left click on it to reload settings and auto-play.

Button assignment is auto saved.
Each project can have its own button assignment.

- If you want to use the 'Loop' button, make sure that you have selected a range that can loop!
If the end sequence/step left the robot in a very different position than the start sequence/step, it could damage the robot.

3.2 Apply changes.

You can apply the new Time and Pause ratio to the Range selected.



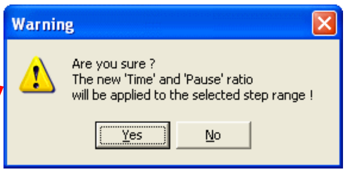
Project : L6

From : 000001 000001

To : 000002 000003

Time 50 % Pause 75 %

In this example:
- The servos Time will be set to 50% for all steps within the range selected.
  So, it multiply speed by 2 in this case…less time to go from a point to another = more speed.
  (a 1000 ms Time became a 500 ms one for example)
- The Pause between steps (if there is one) will be set to 75% for all steps within the range selected.
  (a 1000 ms Pause became a 750 ms one for example).


Warning

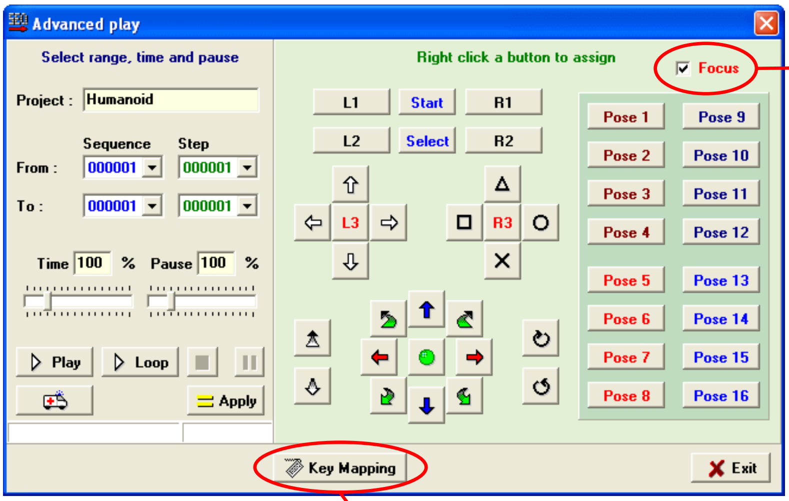
Are you sure ?
The new 'Time' and 'Pause' ratio
will be applied to the selected step range !

Lynxmotion Visual Sequencer V1.16 Manual

3 - The 'Advanced play' module

3.3 Key mapping.

You can use a keyboard instead of a mouse to press buttons on the green panel.



Project : Humanoid

	Sequence	Step
From :	000001	000001
To :	000001	000001

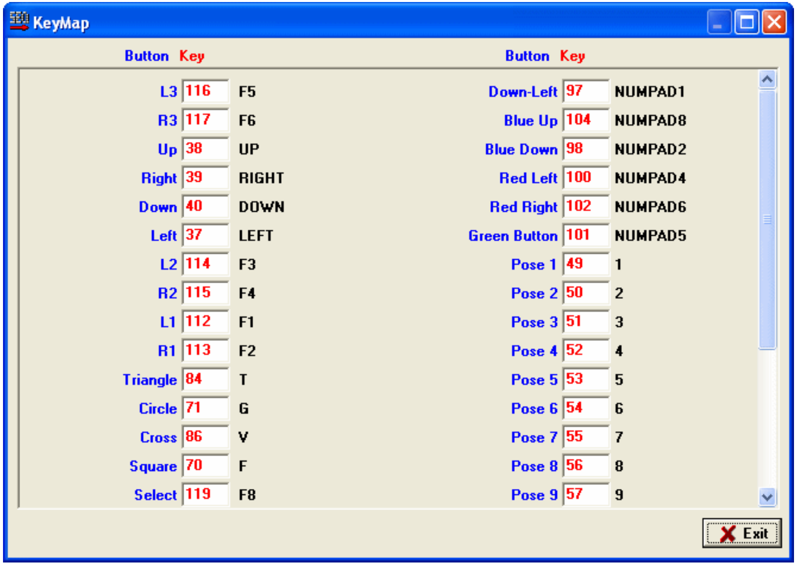
Time 100 % Pause 100 %

The "Focus" check-box sets the focus on the green panel for all keys. This is useful because the "Arrow" and "page up/down" keys can be used on the left panel as well as the green panel with the key mapping feature.

- The "Focus" box is automatically uncheked when you click on a slider or a list-box so you can use any keys without launching a sequence by mistake.

- To set the focus on the green panel you can click on the "Focus" check-box or anywhere on a green surface (except buttons themselves).

If "Focus" is unchecked, the Key mapping feature is disabled.



Button	Key		Button	Key	
L3	116	F5	Down-Left	97	NUMPAD1
R3	117	F6	Blue Up	104	NUMPAD8
Up	38	UP	Blue Down	98	NUMPAD2
Right	39	RIGHT	Red Left	100	NUMPAD4
Down	40	DOWN	Red Right	102	NUMPAD6
Left	37	LEFT	Green Button	101	NUMPAD5
L2	114	F3	Pose 1	49	1
R2	115	F4	Pose 2	50	2
L1	112	F1	Pose 3	51	3
R1	113	F2	Pose 4	52	4
Triangle	84	T	Pose 5	53	5
Circle	71	G	Pose 6	54	6
Cross	86	V	Pose 7	55	7
Square	70	F	Pose 8	56	8
Select	119	F8	Pose 9	57	9

By default no keys are assigned to buttons.

To assign a key to a button :
- Click on a white box.
- Press the key to assign.

Settings are automatically saved when this window is closed.

In this example, the 4 white arrow buttons are assigned to the 4 arrow keys.
- "Right" button => "RIGHT" key.
etc…

Key mapping is the same for all projects.

Button synthesis.

- Selecting a step range, a speed and a pause value on the left panel THEN right-clicking on a button on the right panel to assign this settings to it, will make this button able to recall these settings and play the step range selected automatically.

- Assigning keys to buttons with the key mapping feature allows using the keyboard instead of the mouse to press buttons.

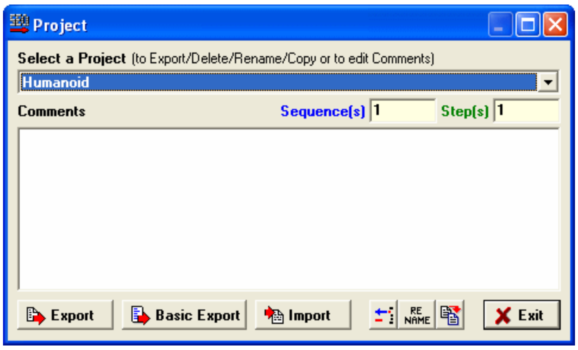
- Buttons can be pressed with the mouse, but you can also use the Socket Client Demo program (see chapter 9).

- Key mapping assignment are the same for all projects (the same key will always press the same button, no matter the project).
- Button assignment are different for each project. (Each project can have its own step range and speed/pause settings assigned to its buttons.)
For example if the white 'up' arrow button on the green panel is "key-mapped" with the keyboard 'up' arrow:
=> Pressing the keyboard 'up' arrow could launch the range "from sequence 1 / step 1 to sequence 2 / step 3" on the project "MyProject 01" and it could launch the range "from sequence 4 / step 1 to sequence 4 / step 5" on the project "MyProject 02".

Lynxmotion Visual Sequencer V1.16 Manual

4 - The 'Project' module

Here you can export, import, rename, delete or copy a Project and add comments.

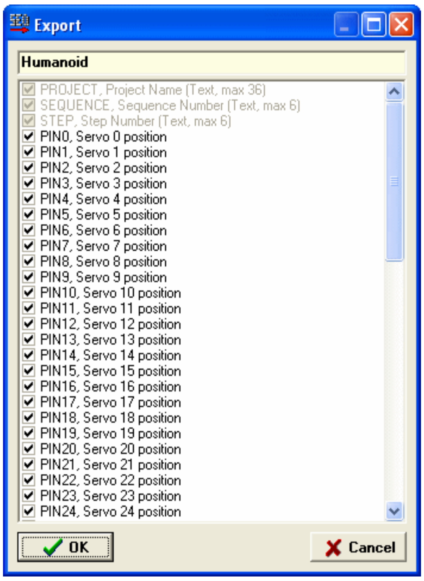


Select a Project (to Export/Delete/Rename/Copy or to edit Comments)

Humanoid

Comments Sequence(s) 1 Step(s) 1

4.1 Export.



Humanoid

☑ PROJECT, Project Name (Text, max 36)
☑ SEQUENCE, Sequence Number (Text, max 6)
☑ STEP, Step Number (Text, max 6)
☑ PIN0, Servo 0 position
☑ PIN1, Servo 1 position
☑ PIN2, Servo 2 position
☑ PIN3, Servo 3 position
☑ PIN4, Servo 4 position
☑ PIN5, Servo 5 position
☑ PIN6, Servo 6 position
☑ PIN7, Servo 7 position
☑ PIN8, Servo 8 position
☑ PIN9, Servo 9 position
☑ PIN10, Servo 10 position
☑ PIN11, Servo 11 position
☑ PIN12, Servo 12 position
☑ PIN13, Servo 13 position
☑ PIN14, Servo 14 position
☑ PIN15, Servo 15 position
☑ PIN16, Servo 16 position
☑ PIN17, Servo 17 position
☑ PIN18, Servo 18 position
☑ PIN19, Servo 19 position
☑ PIN20, Servo 20 position
☑ PIN21, Servo 21 position
☑ PIN22, Servo 22 position
☑ PIN23, Servo 23 position
☑ PIN24, Servo 24 position

Select the Project to export.
Select the fields (all selected by default).

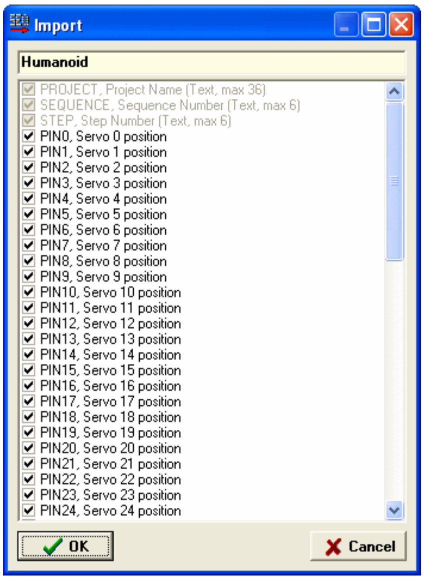
Click on the 'Export' button, select a destination directory / file name and click save.

File format is csv, it's a text file with a semicolon separator for fields.
Notepad, Excel...etc, are able to open and edit this.

Use Export to make a backup of your work or to share projects with friends.

When you export a project it creates a 'shp' file too (with the same filename), it contains the robot shape and params from the sequencer module.
You'll have to provide both csv and shp files in order to share a project!

4.2 Import.



Humanoid

☑ PROJECT, Project Name (Text, max 36)
☑ SEQUENCE, Sequence Number (Text, max 6)
☑ STEP, Step Number (Text, max 6)
☑ PIN0, Servo 0 position
☑ PIN1, Servo 1 position
☑ PIN2, Servo 2 position
☑ PIN3, Servo 3 position
☑ PIN4, Servo 4 position
☑ PIN5, Servo 5 position
☑ PIN6, Servo 6 position
☑ PIN7, Servo 7 position
☑ PIN8, Servo 8 position
☑ PIN9, Servo 9 position
☑ PIN10, Servo 10 position
☑ PIN11, Servo 11 position
☑ PIN12, Servo 12 position
☑ PIN13, Servo 13 position
☑ PIN14, Servo 14 position
☑ PIN15, Servo 15 position
☑ PIN16, Servo 16 position
☑ PIN17, Servo 17 position
☑ PIN18, Servo 18 position
☑ PIN19, Servo 19 position
☑ PIN20, Servo 20 position
☑ PIN21, Servo 21 position
☑ PIN22, Servo 22 position
☑ PIN23, Servo 23 position
☑ PIN24, Servo 24 position

Import a previously exported file
Select the csv file to import (it will import the corresponding shp file too)
Select the fields (all by default)

Click on the 'Import' button, select a project name and click 'Ok'.

Import your home-made csv file
A csv file must contain only ONE project.
The first line is the fields' names separated with semicolons.
All following lines must contain at least:
- the project name
- a Sequence number from 1 to 999999 with no missing number (1 and 3 without 2 is not valid)
- a Step number from 1 to 999999 with no missing number (1 and 3 without 2 is not valid)

and the other fields to import.

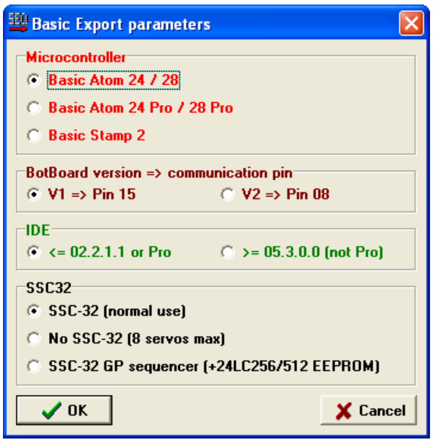
If your field names differ from the internal field names, you'll be asked to reassign them.

Importing your home-made csv is a risk to crash the internal database!
To prevent this, close the software, go to installation directory and backup the two files 'MovesSeq.dbf' and 'MovesSeq.mdx' in a secure place. If you crash the database, you'll just have to close the software and restore these two files.

4 - The 'Project' module

4.3 Export Basic code for Basic Atom or Basic Stamp 2

In the 'Project' module, select 'Export' then 'Export Basic' button, the 'Basic Export parameters' will appear.



Basic Export parameters

Microcontroller
- ● Basic Atom 24 / 28
- ○ Basic Atom 24 Pro / 28 Pro
- ○ Basic Stamp 2

BotBoard version => communication pin
- ● V1 => Pin 15
- ○ V2 => Pin 08

IDE
- ● <= 02.2.1.1 or Pro
- ○ >= 05.3.0.0 (not Pro)

SSC32
- ● SSC-32 (normal use)
- ○ No SSC-32 (8 servos max)
- ○ SSC-32 GP sequencer (+24LC256/512 EEPROM)

Select the kind of microcontroller.

Select your BotBoard Version => communication pin.

If you choose Basic Atom 24 / 28, select the IDE you use.

Select SSC-32 mode (see below)
Click 'OK' and select a destination file/folder.

Open the exported file with the 'Basic Micro Atom IDE' (for Basic Atom *.bas files)
or the 'Basic Stamp Editor v2.2' (for Basic Stamp 2 *.bs2 files)
(don't forget to connect the microcontroller to the PC)
and download the program to the microcontroller.

Choice : 'Basic Atom 24 / 28' or 'Basic Atom 24 Pro / 28 Pro' or 'Basic Stamp 2' and 'SSC-32 (normal use)' :
- Connect the microcontroller to the SSC-32 (don't forget to remove some jumpers).

Choice : 'Basic Atom 24 / 28' or 'Basic Atom 24 Pro / 28 Pro' and 'No SSC-32 (8 Servos Max)'
- Connect the servos directly to the Bot Board (pin 0 to 7) in the same order they were on the SSC-32 card.
In other words, if you were using SSC-32 Pins #12, #17, #18, #24, #25 etc… within the exported project,
use Bot Board' Pin #0 for the servo #12, the BB pin #1 for the servo #17, the BB Pin #2 for the servo #18 and so on…
You can change the pins assignement editing the Basic program, but only Bot Board' Pins #0 to #7 are allowed as destination.

Choice : 'Basic Atom 24 / 28' or 'Basic Atom 24 Pro / 28 Pro' and 'SSC-32 GP sequencer (+24LC256/512 EEPROM)'
- Connect the microcontroller to the SSC-32 (don't forget to remove some jumpers).
The SSC-32 must use the GP sequencer firmware (see SSC-32 page at lynxmotion.com)
The SSC-32 must have a 24LC256 or 24LC512 EEPROM in its socket (see SSC-32 manual page 1)
For more information about GP sequencer, see this manual page 14 chapter "7. GP Sequencer."

***** Make sure to select the correct Microcontroller using this choice ! *****
The 'PULSOUT' command does not work the same way on Basic Atom and Basic Atom PRO.
If you try to run a Basic exported program for a kind of Basic Atom on another kind,
YOU MAY DAMAGE YOUR SERVOS/ROBOT!
This concerns only programs exported for BA ONLY (servos connected directly on a Bot Board not using a SSC-32 card).

The Basic exported file contains position data, durations, pause data,
and the complete program to read data and perform movements.

Basic Stamp 2 programs require the version 2.2 of the BASIC Stamp Editor (with Pbasic 2.5 included).

Exported servos positions are coded in the basic program, in order to save memory :
Positions and servo numbers are save on a Word value as follow : bit 0 to 10 = (servo position - 500), bit 11 to 15 = servo number.
The servos which do not change position since the previous step are not saved.
The first Step of each Sequence contain all the servos positions even if they did not change since the last Step,
this way, you can call a full sequence from any robot position, making sure all servos will be initialized.

To end the group order with the Time value, the previous servo number is stored again in a word bit 11 to 15,
and the Time value is stored in bit 0 to 10 but divided by 32.
A step can contain more than one group of servos with their different Time values (Expert mode).

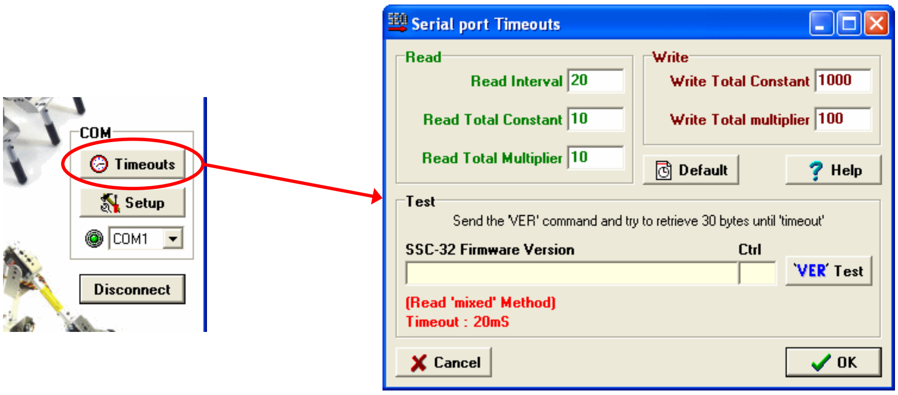
To end the Step with the Pause value, the previous servo number is stored a second time in a word bit 11 to 15,
and the Pause value is stored in bit 0 to 10 but divided by 32.
The Pause value can be different than the Time value(s)…(Expert mode).

The exported program contains a table with the sequences start index, in order to be able to find them in the unique position data table.

If you need the real data flow (not coded) sent to the SSC-32 :
- Play the full project in the Sequencer software and Edit the file 'Playlog.txt' in the installation directory. It will contain the pause values too.

5 - The 'Timeouts' module

Here, you can change some advanced COM port communication values.
** Don't change anything if there's no communication problem ! **



Read
Read Interval 20
Read Total Constant 10
Read Total Multiplier 10

Write
Write Total Constant 1000
Write Total multiplier 100

Test
Send the 'VER' command and try to retrieve 30 bytes until 'timeout'
SSC-32 Firmware Version
Ctrl

(Read 'mixed' Method)
Timeout : 20mS

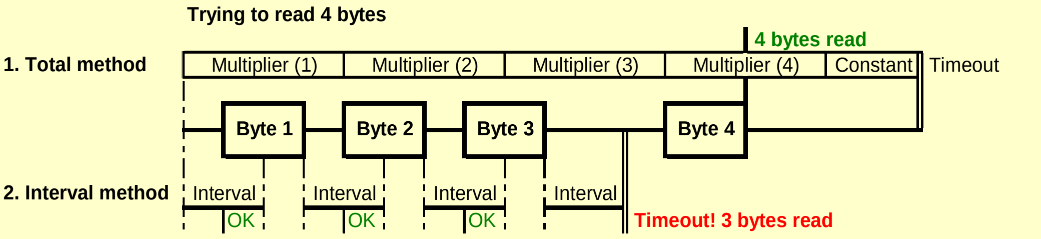
If you're using a 'Wiport' and have experienced some communication problems, try to set the 'Read interval' = 30 (no change on other values)
Some USB to serial cable needs a 'Read interval' values >= 30 too.

The 'Write' values are not critical, so no need to change them.

Before clicking 'Test', check the 'Timeout : xxmS' value. It's the time you will wait in the worst case.
If this value is too big, it could freeze the window for a long time, be careful !

More information with the 'Help' button :

You normally have to specify the number of bytes when reading bytes on the Com port. However, when you don't know how many bytes to read, the timeout value will let you know when the Com port has finished sending data.



1. Read Total method :
Activated if the "Read Interval" value = 0
- The "Read Total Multiplier" is the value multiplied by the number of bytes requested. The "Read Total Constant" value is the maximum time to wait for all the requested bytes before generating a timeout event.

Timeout value depends on the number of bytes requested

2. Read Interval method :
Activated if the "Read Interval" value is > 0 and Both "Read Total Multiplier" and "Read Total Constant" = 0
- The "Read Interval" value is the maximum time to wait between bytes before generating a timeout event.

Timeout value is the "Read Interval"

3. Read 'mixed' method :
Activated if the "Read Interval" value is > 0 and "Read Total Multiplier" and/or "Read Total Constant" > 0
- The first Timeout value elapsed will end the function.

Timeout value depends on the number of bytes requested and the "Read Interval"

Read Interval method ends faster when you don't know how many bytes to ask for.
For example, when asking for the SSC-32's version string, you don't know how many bytes the SSC-32 card will return, but you know you don't want more than 30 bytes to display.
Sequencer asks for 30 bytes, and the timeout occurs after the last byte is sent (it could be 14, 15, etc…)
So 10mS (default) after the last byte is sent by the SSC-32 card, the Read buffer is cleared to remove unrequested bytes.

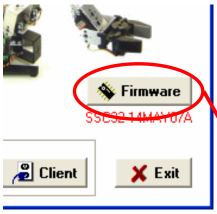
If there are some missing bytes, try increasing the "Read Interval" value.
If it doesn't help, try increase the "Read Total Multiplier" then "Read Total Constant" slowly.

6 - SSC-32 Firmware update.

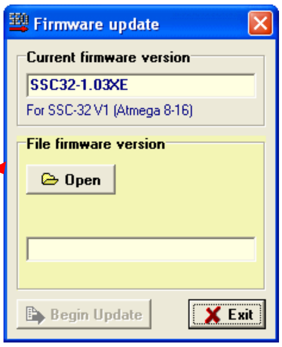
Don't try to update the SSC-32 firmware if all is working properly and if you don't need to.
Sequencer works fine with the "SSC-32-1.06XE" firmware.
Be warned that some "specific" firmware may not work with Sequencer.

Check "SSC-32 Servo Controller" page at www.lynxmotion.com for Firmware update

First connect to the SSC-32, then click on the "Firmware" button.



Click on "Open" and browse for the new firmware file (*.abl).



Firmware update

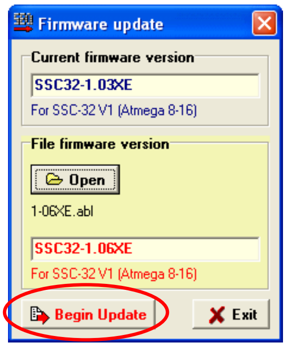
Current firmware version

SSC32-1.03XE

For SSC-32 V1 (Atmega 8-16)

File firmware version

Check the firmware file version and click "Begin Update".



Firmware update

Current firmware version

SSC32-1.03XE

For SSC-32 V1 (Atmega 8-16)

File firmware version

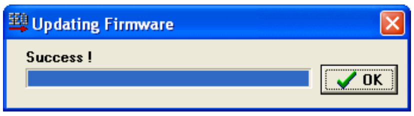
1-06XE.abl

SSC32-1.06XE

For SSC-32 V1 (Atmega 8-16)

Don't turn off the SSC-32 card during the firmware update process!

Click "OK" then "Exit".



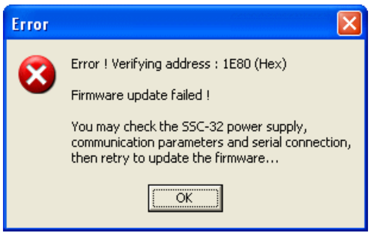
Updating Firmware

Success !

If you get errors during the firmware update process,
you must try to update the firmware again or the SSC-32 card won't work correctly.

Don't close the "Firmware Update" form and don't disconnect (COM Port) from the SSC-32, or you may not be able to reconnect Sequencer to the SSC-32 as the Sequencer program checks to see if the SSC-32 is ok before allowing connections to it.

If you can't connect to the SSC-32 because Sequencer is no longer allowing it, go to www.lynxmotion.com and download the free "SSC-32 Lynx terminal" program to update your firmware. (It will connect no matter what).

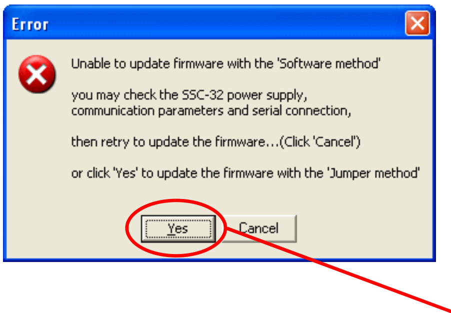


Error

Error ! Verifying address : 1E80 (Hex)

Firmware update failed !

You may check the SSC-32 power supply, communication parameters and serial connection, then retry to update the firmware...



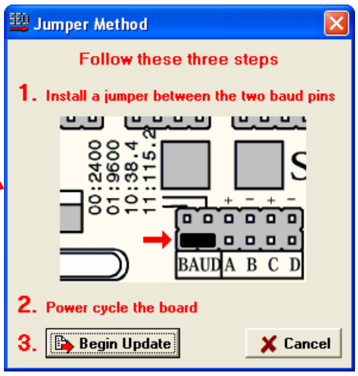
Error

Unable to update firmware with the 'Software method'

you may check the SSC-32 power supply, communication parameters and serial connection,

then retry to update the firmware...(Click 'Cancel')

or click 'Yes' to update the firmware with the 'Jumper method'



Jumper Method

Follow these three steps

1. Install a jumper between the two baud pins

2. Power cycle the board

3.

If the card is no longer able to update its firmware using the "Software method", then you can use the "Firmware method", which will force the card to accept the firmware update. Follow the 3 steps described.
Don't forget Step 2 : Power cycle the board (off then on) or it won't work.

7 - The GP Sequencer.

7.1 Overview.

By default, the SSC-32 firmware includes an onboard sequencer for Hexapod 2DoF robots (H2),
it allows to control easily an H2 robot using simple commands (see SSC-32 manual page 8, 9 and 10).
The free software 'Lynxterm' (Lynx SSC-32 terminal) takes advantage of this feature,
giving a complete support to configure and control the SSC-32 H2 sequencer.

Using the alternative GP Sequencer Firmware and adding a 24LC256 or 24LC512 EEPROM into the SSC-32 socket,
the default H2 sequencer is replaced by a General Purpose sequencer, which uses the 24LC256/512 EEPROM
to store sequences and steps information.
So, using simple commands, you can play a sequence previously stored in the EEPROM,
with speed, direction and pause between steps control.
If you use a Bot Board to control the SSC-32, the microcontroller on it is now free to perform other tasks,
and its memory is no longer used to store move data, as it's now stored into the SSC-32 (optional) EEPROM.

2 independent players, up to 128 sequences, 256 steps with 32 servos max by sequences.
The total number of steps depends on servos used and EEPROM memory (24LC256 = 32k bytes, 24LC512 = 64k bytes).

This new Visual Sequencer version gives complete support for GP Sequencer :
- You can export any Visual Sequencer project to a memory dump file (*.EEP) for 24LC256/512 EEPROM.
- Load this file with the Hexadecimal/ASCII editor or the Sequences Viewer, then write it to the EEPROM.
- You can change EEPROM value directly using the Hexadecimal/ASCII editor (for experts only)
or use the Sequences Viewer to modify some Sequences/Steps values using a "human comprehensible" display.
(The Sequences Viewer also allows loading an EEPROM dump and 'adding' its Sequences to the EEPROM instead of 'overwriting'.)
- The Control Panel allows you to test the GP Sequencer's players.

7.2 GP Sequencer support.

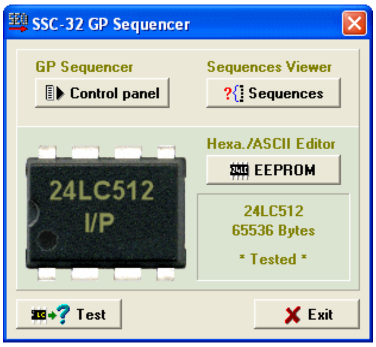
7.2.1 Export a project to GP Sequencer.

From the main window, open the 'Project' form, select 'Basic Export'.
Then choose your microcontroller (so it will export a little demo program to control the GP sequencer from a Bot Board).
Select 'SSC-32 GP Sequencer (+24LC256/512 EEPROM)'.
Click 'Ok', choose a file name and click 'save'.
It saves 2 files :
- "yourfilename.bas" or "yourfilename.bs2" to use with your Basic IDE and to download to the microcontroller on the Bot Board.
- "yourfilename.EEP" to use with the Hexa/ASCII editor or the Sequences Viewer to write the EEPROM.

7.2.2 SSC-32 GP Sequencer Main Menu.

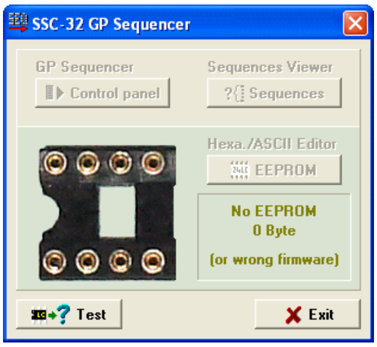
From the main window, open the 'GP Sequencer' form,
It will auto-detect the EEPROM and show the information.

If all is correct.



24LC512
65536 Bytes
* Tested *

If there's no EEPROM or if you're not using a "GP" firmware.

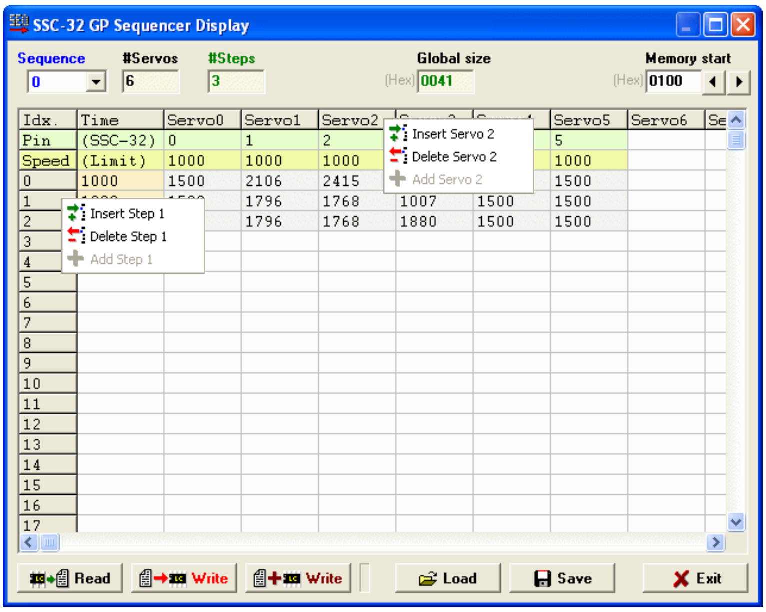


No EEPROM
0 Byte
(or wrong firmware)

Always switch off the SSC-32 before installing or removing an EEPROM!
Align the half circle mark on the EEPROM with the mark on the socket.
Check your Firmware, you must use the GP specific firmware to use this.

7.2.3 Sequences Viewer.

With this form you can see the EEPROM contents in the GP Sequencer format, it's more comprehensible than the Hexa/ASCII editor. This is the place where you can load the previously exported projects (*.EEP) and write it to EEPROM, you can edit some values, add, insert or delete some steps/servos then save the changes to disc or overwrite the EEPROM.



SSC-32 GP Sequencer Display

Sequence: 0 | #Servos: 6 | #Steps: 3 | Global size (Hex) 0041 | Memory start (Hex) 0100

Idx.	Time	Servo0	Servo1	Servo2			Servo5	Servo6	Se
Pin	(SSC-32)	0	1	2			5		
Speed	(Limit)	1000	1000	1000			1000		
0	1000	1500	2106	2415			1500		
1			1796	1768	1007	1500	1500		
2			1796	1768	1880	1500	1500		
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									

**Select the sequence to display.
Right click on a column to insert/delete a servo.
Right click on the first free column
to add a servo (here column "servo6").**

**Right click on a row to insert/delete a step.
Right click on the first free row
to add a step (here row "3").**

Click on a cell to edit it :

- **Pin : 0 to 31, no duplicate pins are allowed.**
- **Speed : 0 to 65535 ms, is the speed limit, no matter the synchronization.**
- **Time : 0 to 65535 ms, step time, the SSC-32 will adjust all servo's speed to synchronize them so they will finish their moves all together. (but will not go pass the speed limit value)**
- **Servo(x)/Idx(y) : 500 to 2500, these blue cells contains servo's pulses to reach.**

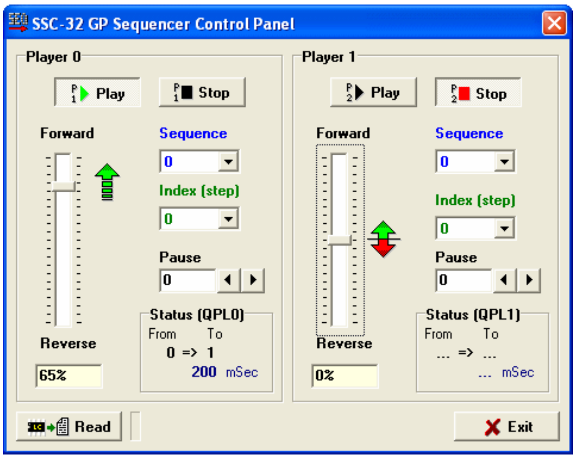
**The "Read" button, reads the EEPROM contents and display the result in the window.
The red "Write" button with the red arrow, overwrites the EEPROM contents with all the sequences present in the window.
The brown "Write" button with the brown 'plus' sign, adds to the EEPROM all the sequences present in the window.
The "Load" button loads an EEPROM dump file (*.EEP) in the window.
The "Save" button saves all the sequences present in the window to an EEPROM dump file (*.EEP).**

**When you add the window content to the EEPROM, the window contains now all the sequences (the new EEPROM contents).
To write an exported project (from Visual Sequencer), just click "Load", select the right yourfilename.EEP file, click "Open" then click the red "Write" button to overwrite the EEPROM with the new project or the brown one to add it.**

The "Memory start" selector is the lower memory address to use to store sequences, no need to change this to add some sequences to another place, the only reason to change this is to preserve a part of the memory for your own purpose :
- **memory from hex 0000 to hex 00FF is reserved to store the sequences Table Of Contents.**
- **selecting a "Memory start" to hex 0200 for example will preserve memory from hex 0100 to hex 01FF for your own purpose.**

7.2.4 Control Panel.

**The GP Sequencer Control Panel lets you test the 2 players with real time control on them, a cool way to see if all is ok.
Select the Sequence to test, click "Play", move the slider to increase/decrease or reverse speed,
use the "Pause" selector to add some pause between steps.**



SSC-32 GP Sequencer Control Panel

Player 0: Sequence 0 | Index (step) 0 | Pause 0 | Status (QPL0) From 0 => To 1, 200 mSec | 65%

Player 1: Sequence 0 | Index (step) 0 | Pause 0 | Status (QPL1) From ... => To ..., ... mSec | 0%

Don't run the 2 players on the same sequence.

Don't run the 2 players on sequences which control the same servos.

In both cases, the result is a strange mix.

The GP Sequencer's players are autonomous and completly handled by the SSC-32. while they are playing (even playing at 0% speed) you could experience some weird behavior trying to move servos with the SSC-32 config form or the "normal" sequencer form.

The players are not automatically stopped when you exit the window...or the software itself. They will continue to play until you stop them.

7 - The GP Sequencer.

7.2.5 Hexa/ASCII Editor.

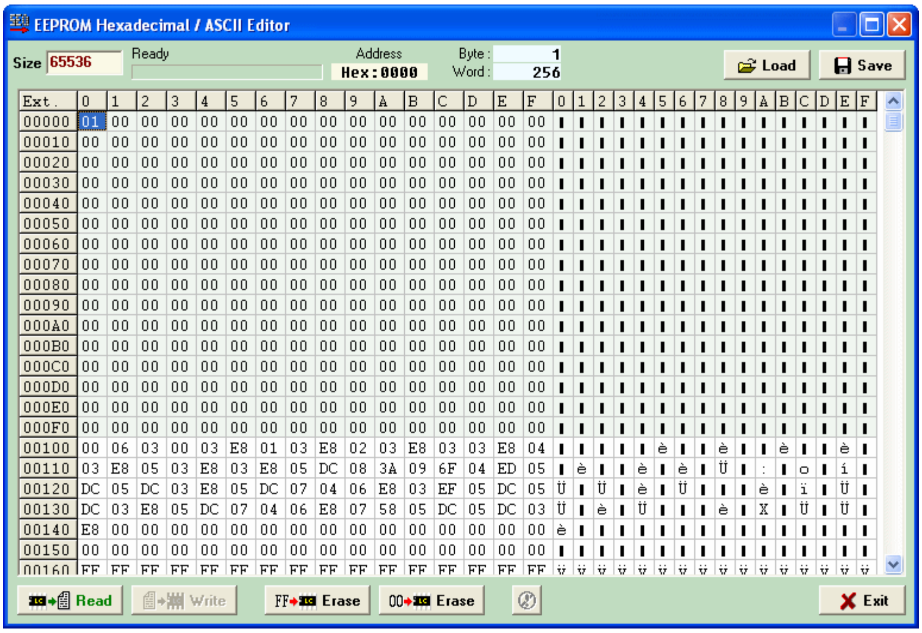
**The hexadecimal/ASCII editor is reserved for experts, because you must know the GP sequencer format exactly.
See "7.3 More information" and learn how the Sequences are stored and handled by the GP sequencer first.**

To be short, the blue lines are the Sequences Table Of Contents, for example the bytes 0 and 1 are an address (pointer) to the sequence #0, in the image below, a pointer to address hex 0100, at address hex 0100 we can see the sequence #0 contents.

Size is the current EEPROM size, Address is the one of the blue cursor, Byte is the decimal value of the hex one under blue cursor, Word is the decimal value of the two bytes : the one pointed by the blue cursor and the next one.

**To enter an hexadecimal value (first 16 columns) use one or two characters from 0 to 9 and A to F.
To enter an ASCII value (last 16 columns) use any displayable ASCII characters.
the first 16 columns are the same memory bytes as the last 16 columns, but using a different format to display.**

For example, at address hex 013A (Line 00130, left column A) you can see "58" (hexadecimal value, decimal value is 88), and looking at the same line but the right column A you can see "X" which is the ASCII character for value hex 58 (dec 88)



EEPROM Hexadecimal / ASCII Editor

Size 65536 Ready Address Hex:0000 Byte : 1 Word : 256

```
Ext.  0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
00000 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
000A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
000B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
000D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
000E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00100 00 06 03 00 03 E8 01 03 E8 02 03 E8 03 03 E8 04 ▮▮▮▮▮è▮▮è▮▮è▮▮è▮
00110 03 E8 05 03 E8 03 E8 05 DC 08 3A 09 6F 04 ED 05 ▮è▮▮è▮è▮Ü▮:▮o▮í▮
00120 DC 05 DC 03 E8 05 DC 07 04 06 E8 03 EF 05 DC 05 Ü▮Ü▮è▮Ü▮▮▮è▮ï▮Ü▮
00130 DC 03 E8 05 DC 07 04 06 E8 07 58 05 DC 05 DC 03 Ü▮è▮Ü▮▮▮è▮X▮Ü▮Ü▮
00140 E8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 è▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
00160 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

**Load an *.EEP file.
Save an *.EEP file.
EEP files are EEPROM dump.**

**"Read" button reads the complete EEPROM, you can stop reading by clicking on the "!" small button.
"Write" button writes the complete EEPROM, you can stop writing by clicking on the "!" small button.
"FF Erase" and "00 Erase" buttons writes the complete EEPROM with hex FF or hex 00 bytes, clicking on the "!" small button will stop the process.**

Read/Write and Load/Save with the Hexadecimal/ASCII Editor are different than the Sequences Viewer :

**- Here, all bytes are read/written (while you don't stop the process) or loaded/saved.
- With the Sequences Viewer, only the first hex 0100 bytes + the sequences data pointed by the Table Of Contents (by 32 bytes packet) are read/written or loaded/saved.
(In the example above, bytes from hex 0000 to hex 015F...the sequence #0 ends at hex 0141 but a full 32 bytes packet ends at hex 015F)**

That means two things : the Sequences Viewer Read/Write process is faster than the Hexadecimal/ASCII editor, but the Sequences Viewer does not Read/Write or Load/Save any other bytes than the ones concerning Table Of Contents and Sequences data.

7.3 More information.

**Take a look into the installation directory and open the "GP Sequencer" folder.
(Typically : "C:\Program Files\Sequencer_SSC-32\GP Sequencer")**

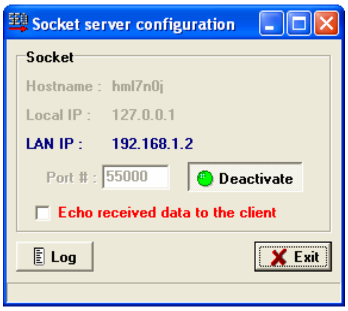
**Here you can find :
 - The GP Sequencer Firmware : 1-1gbeta.abl
(but you should check at Lynxmotion.com for update)**

**- The GP Sequencer Documentation : doc.pdf
- A GP Sequencer example : xls.pdf
- The same as above but in xls format : example.xls**

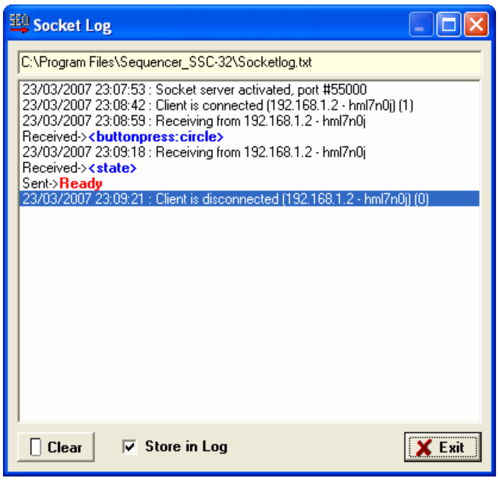
8 - Sockets.

8.1 Socket Server.

The socket server allows an external application to control some feature in Visual Sequencer,
the LAN IP is the address that the external application have to connect, the Port # is the one to use to communicate,
(IP address + Port # is called a socket), click "Activate" to enable the socket server, the log file will help you to see the communications.



Socket
Hostname : hml7n0j
Local IP : 127.0.0.1
LAN IP : 192.168.1.2
Port # : 55000
☐ Echo received data to the client



C:\Program Files\Sequencer_SSC-32\Socketlog.txt

```
23/03/2007 23:07:53 : Socket server activated, port #55000
23/03/2007 23:08:42 : Client is connected (192.168.1.2 - hml7n0j) (1)
23/03/2007 23:08:59 : Receiving from 192.168.1.2 - hml7n0j
Received-><buttonpress:circle>
23/03/2007 23:09:18 : Receiving from 192.168.1.2 - hml7n0j
Received-><state>
Sent->Ready
23/03/2007 23:09:21 : Client is disconnected (192.168.1.2 - hml7n0j) (0)
```

☑ Store in Log

Commands list :

- "<ssc-32:ver>" will return the SSC-32 firmware version.

- "<buttonpress:buttonId>"
ButtonId is one of the following:
L1, R1, L2, R2, L3, R3, Start, Select, Green Button,
Up, Down, Right, Left, Triangle, Circle, Cross, Square,
Move Forward, Move Backward, Turn Clockwise, Turn Anticlockwise,
Blue Up, Blue Down, Red Right, Red Left, Up-Right, Down-Right, Down-Left, Up-Left, Pose 1 to Pose 16.

- "<state>" will return "Busy" if a sequence is currently playing or "Ready" if none is playing,
"Paused" if paused or "Error" if Visual Sequencer is not in the "Advanced Play" form.

- use "<seq:stop>" to stop playing a sequence, "<seq:pause>" to pause and "<seq:resume>" to resume.

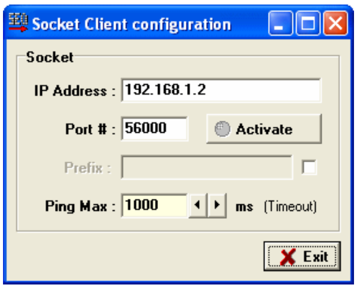
<buttonpress:buttonId>, <seq:xxx> and <state> works if Visual Sequencer is in the "Advanced play" window.

8.2 Socket Client.

The socket client allows connecting to a robot using TCP/IP instead of COM port, but you need some hardware on the robot side
to receive TCP/IP flow (via WIFI for example) and convert it to the SSC-32 COM port.

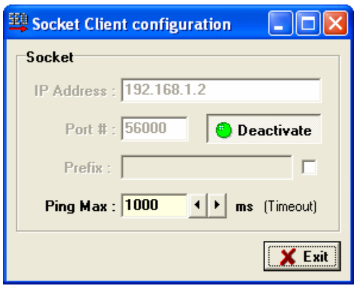
You can already use two WIFI COM ports to control a robot, one on the PC side, one on the robot side.
But the Socket system allows the PC to communicate directly through the network, so if you're using an external WIFI modem (www box),
you can use it to communicate with the robot, saving one of the two WIFI COM port adapter,
now you only need the one on the robot side, which must be compatible with your WIFI modem.
Note : if you have an internal WIFI modem that you can configure to act as a WIFI COM port on the PC side,
don't bother using the socket client method, connect directly to it using its emulated COM port #.

Enter the "robot side WIFI COM port" IP address
Enter the "robot side WIFI COM port" port # (socket)



Socket
IP Address : 192.168.1.2
Port # : 56000
Prefix : ☐
Ping Max : 1000 ms (Timeout)

Click "Activate"
If the "robot side WIFI COM port" is found it will stay activated.



Socket
IP Address : 192.168.1.2
Port # : 56000
Prefix : ☐
Ping Max : 1000 ms (Timeout)

Ping Max is used as a
Timeout here,
adjust from 100 to 2000.

You can only connect to COM port OR Socket client, not both at the same time, "activating" the Socket client doesn't mean "connect".
To finish connecting the software to the robot, exit this window and click "connect" on the Main form.
If the SSC-32 card is found through the network, it will connect and show the firmware version.

If a SSC-32 card is found on a COM port, it will auto disable the Socket client and it will connect to COM port.
If the "robot side WIFI COM port" is not found through the network it will auto disable the Socket client.

"Prefix" allows to add a string before each "normal" data flow to the SSC-32, don't use prefix if you don't know or don't need to.

9 - Socket Client Demo.

**This little C++ Demo program was made with Borland C++ Builder 6.0.
Source code is given "as is", it shows how to communicate with Visual Sequencer.**

**The Socket Client Demo program can connect to Visual Sequencer Socket Server
in order to command the Advanced Play buttons through a network.**

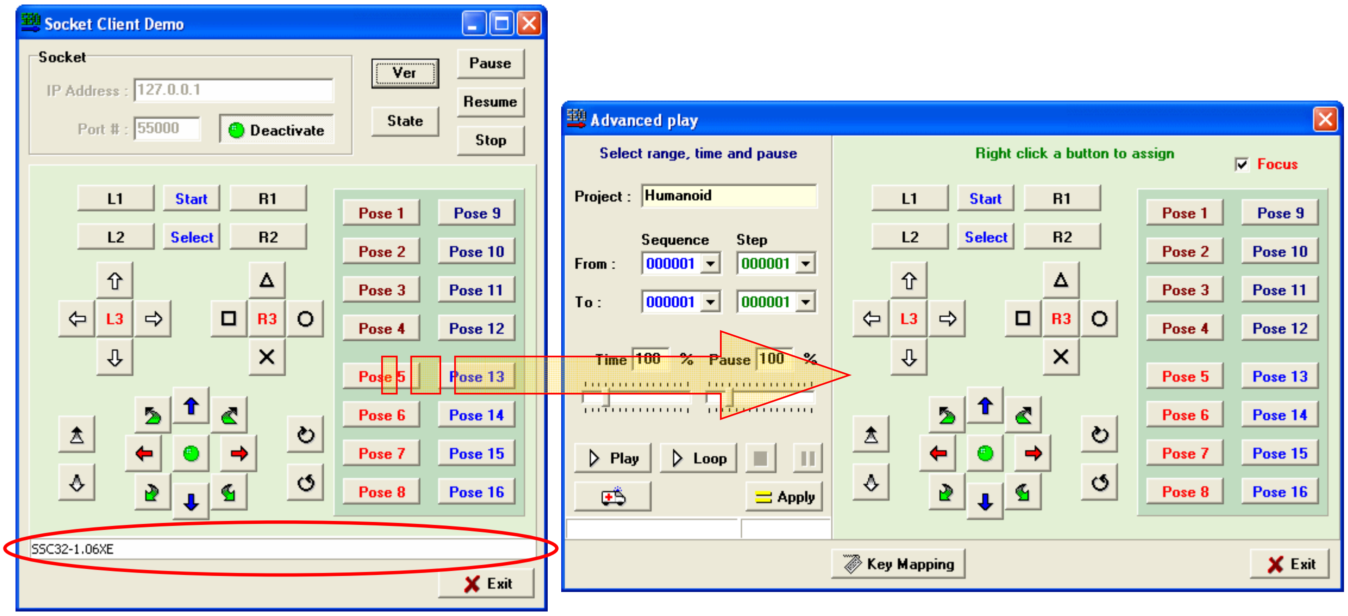
How it woks :
- At first, run Visual Sequencer and open the Socket Server form.
- Note the Local IP if you plan to test the Demo program on the same PC or note the LAN IP if you want to test through a network, then note the port #.
- Activate the Socket Server, your Firewall may warn you about Visual Sequencer trying to communicate, you'll have to allow this.
- Close the Socket Server form and go to the Sequencer form and select a project.
- Go to the Advanced Play form and assign some sequences to buttons.
- Run the Socket Client Demo program, enter the right IP previously noted and so on the Port #.
- Activate the Socket Client.

On the example below, I'm using the local IP (127.0.0.1 for all PC), so it's a test with both programs running on the same PC.

Test 1 :
**- Click on the 'Ver' button, it asks Visual Sequencer the SSC-32 firmware version.
If all is ok, it will display the SSC-32 firmware version in the white status panel.
("Virtual" appears if you are connected to a virtual robot.)**

Test 2 :
**- Click on the 'State' button, it asks Visual Sequencer the Advanced Play state.
If all is ok, it will display "Ready" in the white status panel.**

Test 3 :
**- Click on a control button (one of the ones you have assigned in Visual Sequencer).
If all is ok, it will send the order to run the according sequence(s) with Visual Sequencer.**



Socket Client Demo

Socket
IP Address : 127.0.0.1
Port # : 55000

SSC32-1.06XE

Advanced play

Select range, time and pause
Project : Humanoid
From : Sequence 000001 Step 000001
To : Sequence 000001 Step 000001
Time 100 % Pause 100 %

Right click a button to assign
☑ Focus

Source code files description.

**SocketClientDemo.exe is the compiled program, it can't run if Visual Sequencer is not installed (it uses Borland dll).
SocketClientDemo.bpr is the Borland C++ Builder project file.
SocketClientDemo.res is the resource file (application icon is in here).
SocketClientDemo.cpp is the application entry.
MainForm.dfm is the main form data (Buttons, ComboBox, CheckBox,Socket Client etc...).
MainForm.h is the main form header.
MainForm.cpp is the main form code.**

Lynxmotion Visual Sequencer V1.16 Manual

10 - Command line.

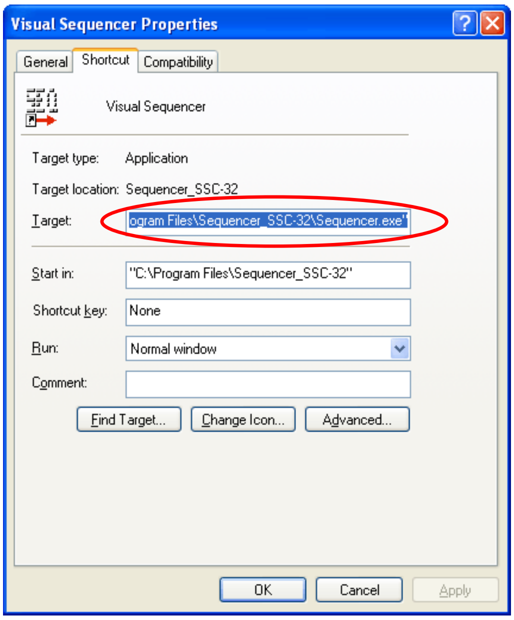
Using command line, you can start and make Visual Sequencer :
- going directly to the 'Sequencer' form with the right project selected.
- doing the same as above, bypassing initialization warning message and play a selected project automatically.

First, you need to create a new Visual Sequencer shortcut :

1. Right-click an open area on the desktop, point to New, and then click Shortcut.
2. Click Browse.
3. Locate the **C:\Program Files\Sequencer_SSC-32\Sequencer.exe** file and click on it to select, click Open or Ok, and then click Next.
4. Type a name for the shortcut and then click Finish.

Now, you need to edit this shortcut to add some command line parameters :

1. Right-click on your new Visual Sequencer shortcut, select the 'shortcut' tab.
2. Click on the 'Target' Box and go to the end of the line **"C:\Program Files\Sequencer_SSC-32\Sequencer.exe"**
3. Here you can add 'sequencer' to open automatically the Sequencer form.
4. Then you can add "Your project name", it will select this project automatically.
5. At last you can add a third parameter to automatically play the selected project :
   - 'play' to play the complete project
   - 'playloop' to play the complete project and play it again until you stop it.



Target type: Application

Target location: Sequencer_SSC-32

Target: ogram Files\Sequencer_SSC-32\Sequencer.exe"

Start in: "C:\Program Files\Sequencer_SSC-32"

Shortcut key: None

Run: Normal window

Comment:

For the (third parameter) 'play' command to work properly, the robot must be 'on' before running Visual Sequencer with the shortcut!

Warning! When using the (second parameter) 'play' or 'playloop' command, there's not a message asking to set the robot in default position before activating all servos… It will activate them automatically!

Examples :

To Open the 'Sequencer' form,
"C:\Program Files\Sequencer_SSC-32\Sequencer.exe" sequencer

To Open the 'Sequencer' form and select the "brat" project,
"C:\Program Files\Sequencer_SSC-32\Sequencer.exe" sequencer "brat"

To Open the 'Sequencer' form and select the "brat" project and to play it automatically,
"C:\Program Files\Sequencer_SSC-32\Sequencer.exe" sequencer "brat" play

To Open the 'Sequencer' form and select the "brat" project and to play it automatically then loop until you stop it,
"C:\Program Files\Sequencer_SSC-32\Sequencer.exe" sequencer "brat" playloop