SEQ Instructions

Contents

Introducing SEQ

SEQ lets you make things and control them. You can make buggies, floor turtles, cars, cranes, robots, or any imaginable mecahnical device, and then program it to obey a sequence of instructions. You can change the models behaviour by changing your program.

SEQ is a computer in a small box that can be attached to electric motors, lights and sensors. You can build models using Lego & Lego Technic (Lego U.K.), Teko (Osmiroid), Plawco & Plawcotech (Commotion), Fishertechnik (Fischer U.K.), other construction kits, and junk materials.

You press keys on SEQ to make up a sequence of instructions that it remembers. When you tell SEQ to obey these instructions it will carry them out one after the other, controlling whatever is connected.

The best way to find out how SEQ works is to play with it. You can investigate what each key does without anything connected, but it is more fun if you connect motors, lights and switches to the outputs and inputs.

It is not possible to damage SEQ by incorrect connection (but you may exhaust your batteries). You cannot harm SEQ by pressing the wrong keys either.

This manual describes each of SEQ's instructions, together with some technical details. It does not describe how SEQ can be used in the classroom: educational users should also read "Teacher's Guide to SEQ".

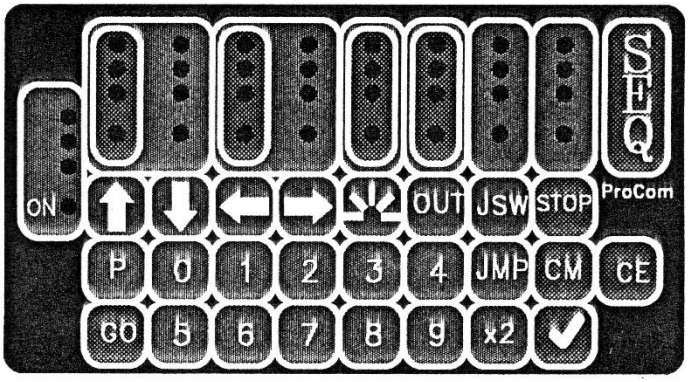
Keyboard

Pressing keys allows you to program SEQ to obey a sequence of instructions.

There are three kinds of keys:

> Command keys are red, and tell SEQ to do something immediately.
>
> Instruction keys are green, and let you program actions into SEQ's memory.
>
> Number keys are also green; every instruction needs at least one number.

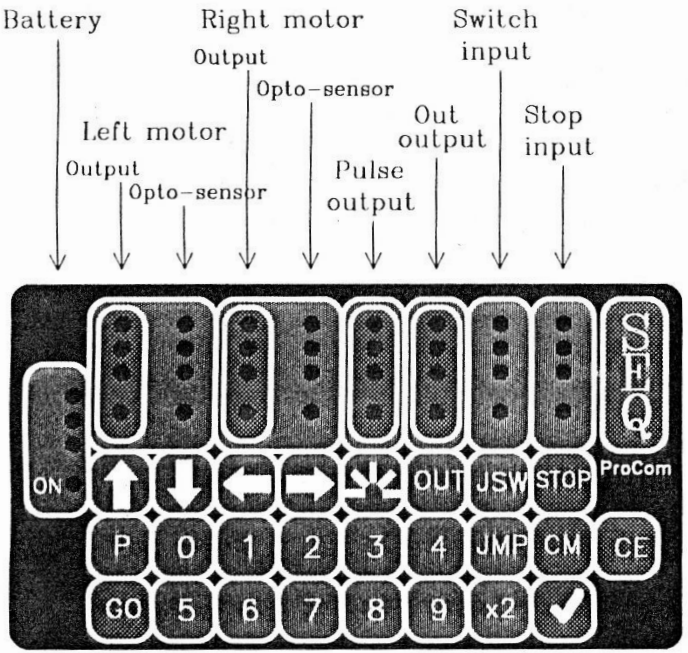
Each time you press a key that SEQ accepts, it will beep. If SEQ doesn't beep when you press a key, you are probably pressing the wrong key. Then you should read the instructions, or try another key.



Sockets

There are 9 pairs of sockets; each one has a small indicator light so you can see when the socket is on.

The socket on the left with an indicator light marked **ON** is the power socket, and should be connected to a battery. If the green **ON** light doesn't come on, try plugging it in the other way around. If it still doesn't come on, your batteries may be flat.



There are two other types of sockets:

> Output sockets are red. They are usually connected to motors or lights, and have red and/or green indicator lights. There are 4 output sockets on **SEQ**.
>
> Input sockets are green. They are usually connected to switches or opto-sensors, and have yellow indicator lights. There are 4 input sockets on **SEQ**.

The two motor sockets are bidirectional, allowing motors to go both forwards and backwards. The Pulse and Out sockets are unidirectional: motors can go in one direction only.

For instructions about connecting things to these sockets, see the sections on **Output Sockets** and **Input Sockets** on pages 19-21.

Clear Memory command

The key marked **CM** is the **Clear Memory** command.

Pressing this key will remove all instructions from **SEQ**'s memory. Remember to use **CM** whenever you want to start a new program.

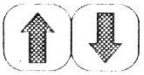
If you do not press **CM** old instructions will be remembered by **SEQ**, and your new instructions will be added after these.

Go command

The **Go** command key tells **SEQ** to obey the instructions you have entered into its memory. Each instruction will be obeyed in turn.

When you first connect **SEQ** there are no instructions stored, so pressing it has no effect.

You can stop **SEQ** at any time by pressing the **Stop** key.



Moving Forward and Backward

Use these keys to program **SEQ** to move a vehicle forward or backward.

Attach the left motor to the left motor output socket above the **Forward** key. Then connect the right motor to the right motor output socket above the **Left** key.

Press the **Forward** key. You must then tell **SEQ** how far to move forward. Do this using the number keys. You can press any number from 1 to 99.

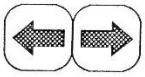
Try this sequence:

Press CM	Press ↑ then 2	Press GO

If the vehicle turns instead of moving forward, turn one of the motor plugs around. If it goes backwards, turn both plugs around. Press **GO** to repeat the program.

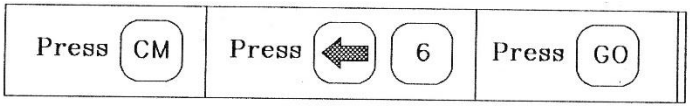
Try programming **SEQ** to move the vehicle fowards and then backwards. Try larger and smaller numbers.

If your vehicle has opto-sensors attached to the motors (or drive shafts), **SEQ** can control the vehicle more accurately. Connect the left opto-sensor to the left motor input socket, above the **Backward** key, and the right opto-sensor to the right motor input socket, above the **Right** key. Repeat the program using **GO**. How far does the vehicle move now?

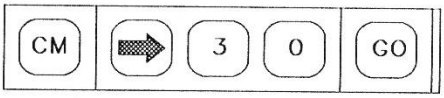


Turning Left and Right

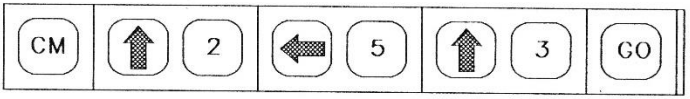
Use these keys to program **SEQ** to turn the vehicle. Press the **Left** or **Right** key. You must then tell **SEQ** how far to turn. Do this using the number keys, entering a number from 1 to 99. Try this sequence:



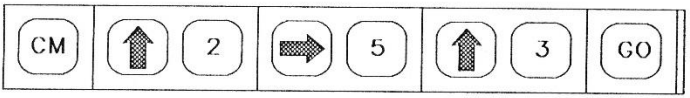
The vehicle will turn to the left. Press **GO** again to turn more. If the amount of turn is small, try larger numbers:



The vehicle will turn to the right by a larger amount. You can combine **Forward, Backward, Left** and **Right**:



Where does the vehicle end up? Move it back to the same start position and enter:

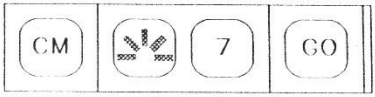


Try to predict what will happen before you press **GO**. You can enter more instructions to make the vehicle follow a more complicated path. **SEQ** can remember up to 40 instructions.

Pulse

This instruction makes **SEQ** pulse something. Connect a light to the **Pulse** socket directly above the **Pulse** key.

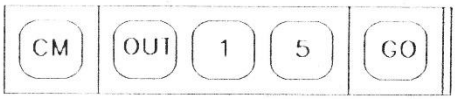
Press the **Pulse** key. You must then tell **SEQ** how many times to pulse the light. Press a number, from 1 to 99. How would you make the light pulse 7 times? One sequence could be:



Out

Connect a light to the **Out** socket, directly above the **Out** key. This instruction makes **SEQ** turn on whatever is connected to the **Out** socket.

Press the **Out** key. You must then tell **SEQ** how long to turn on for. Type a number from 1 to 99. The numbers are tenths of a second. How would you turn **Out** on for one and a half seconds? One sequence could be:

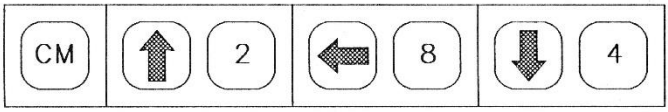


The **Out** output can be used for many purposes. It could be connected to another motor on a crane to raise and lower the jib. Or attached to a light to flash morse code.

CE Clear Error command

This key lets you remove the last instruction from SEQ's memory. So if you make a mistake, or SEQ doesn't do what you expect, you can change your mind.

Each time you press CE the last instruction is removed from SEQ's memory. Enter this sequence:

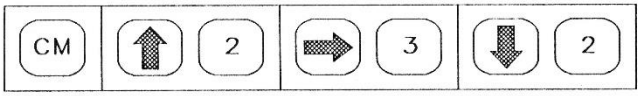


To remove the last 2 instructions, press CE twice. The program now consists of one step, **Forward 2**, and new instructions can be entered to replace the others.

✓ Test command

The **Test** command can be used to try out the last instruction entered. Pressing the **Test** key makes SEQ carry out the last instruction in its memory.

The **Test** key can be pressed as often as you like. Used with the **Clear Error** command the **Test** key lets you test and edit each instruction as it is entered:



Pressing **Test** will make SEQ carry out the instruction **Backward 2**. Pressing CE will remove this instruction.

Stop command

Once the **Go** command key has been pressed **SEQ** will carry out the instructions in its memory until it has done the last one you entered.

If something goes wrong, you can stop **SEQ** immediately by pressing the **Stop** key. Next time **Go** is pressed **SEQ** will start at the beginning again.

You can also attach a switch, opto-sensor, or other input to the **Stop** input immediately above the **Stop** key.

When this switch is closed **SEQ** will stop obeying instructions immediately. The switch acts as an emergency stop.

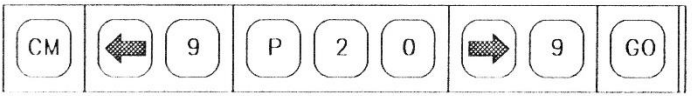
It will not be possible to enter any instructions or commands until the emergency stop switch is opened again.

Pause

The **Pause** instruction tells **SEQ** to wait for a while before carrying out the next instruction.

Press **P** if you want to program a delay into a sequence of instructions. You must also tell **SEQ** how long to pause for. Enter a number from 1 to 99. The time delay is in tenths of a second.

This example will turn a vehicle to the left, waits for 2 seconds (20 tenths of a second), then turns it back again:

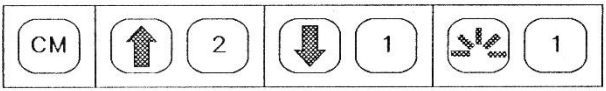


x2 **Repeat**

The **Repeat** key allows you to instruct **SEQ** to repeat a sequence of instructions, saving you from entering them twice.

To repeat some instructions, press the **Repeat** key. You must also tell **SEQ** how many of the previous instructions you want to repeat. Enter a number from 1 to the number of instructions in **SEQ**'s memory.

Try this sequence of instructions:



To make the vehicle return to its starting point, you could enter **x2 2**. This will make **SEQ** repeat the last 2 instructions, so the obeyed sequence will be:

Forward 2
Backward 1
Pulse 1
Backward 1
Pulse 1

What would happen if you tried **x2 3** instead? What if you tried **x2 1**?

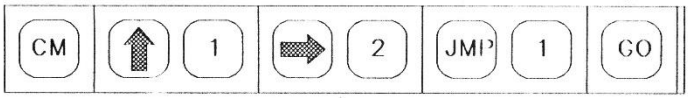
SEQ will not allow you to repeat instructions outside the range of those entered. So in this example the only **Repeat** numbers possible are 1, 2 or 3.

JMP Jump

The **Jump** instruction gives you more control over the program sequence than the **Repeat** instruction. When **SEQ** obeys a **Jump** instruction the next instruction to be obeyed can be anywhere in **SEQ**'s memory.

You can jump backward to an earlier instruction (the first instruction is number 1). You can also jump forward to a later instruction, one that you haven't entered yet!

To enter a **Jump** instruction, press the **Jump** key. You must give **SEQ** the number of the instruction to jump to. Enter a number from 1 to 40. For example:



SEQ will carry out instruction 1 (**Forward 1**), then instruction 2 (**Right 2**). It will then obey instruction 3 and jump to instruction 1. The vehicle will go **Forward 1** and **Right 2** forever (what shape will it draw?).

This could be halted by the **Stop** key, or a switch attached to the **Stop** input, or even by the batteries running out!

Complicated **Jump** intruction sequences are best planned on paper.

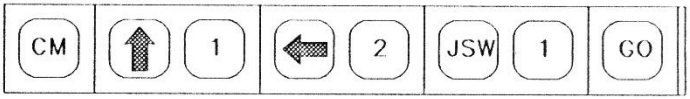
Jump on switch

This key lets you enter instructions that are similar to the **Jump** instruction, except that an input is used to control whether **SEQ** will jump or not.

Connect a switch, opto-sensor or other input to the **Switch** input socket, immediately above the **JSW** key.

To enter a **Jump on switch** instruction, press the **JSW** key. You must then tell **SEQ** which instruction to jump to if the switch is closed. Enter a number from 1 to 40.

When **SEQ** obeys a **Jump on switch** instruction, it will look to see if the switch is closed (switch light on). If it is closed, **SEQ** will jump to the new instruction. Otherwise **SEQ** will continue with the next instruction (the one after **Jump on switch**):



SEQ will obey the first 2 instructions. When it reaches the third instruction **SEQ** will test the switch input. If the switch is closed it will jump to instruction 1, otherwise it will finish. While the switch is closed **SEQ** will continue to move the vehicle forwards and left. Note that this is opposite to the example for **Jump**.

Like **Jump**, **Jump on switch** can jump both backwards to an earlier instruction, or forwards to a later one.

Stop mode commands

Stop can be used with other keys to alter **SEQ**'s behaviour.

Normally **SEQ** controls the two motors on a vehicle together. Some models may require independent control of each motor. **Stop-Left** and **Stop-Forward** lets you change **SEQ**'s motor contol mode.

When using opto-sensors for better control of each motor you may want to count several pulses for each movement instruction. **Stop-number** changes the pulse count.

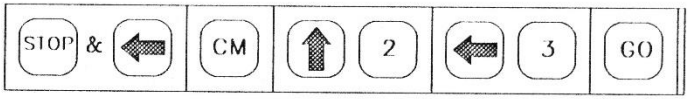
Normally **SEQ** beeps when you press keys to enter instructions or commands, and during the **Pulse** instruction. **Stop-Pulse** turns beeping off, or on again.

Normally **SEQ** jumps on a **JSW** instruction when the switch is closed. **Stop-JSW** inverts this switch mode.

Independent motor control

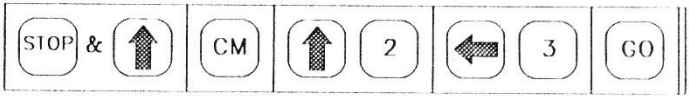
To select independent motor control, press the **Stop** key and hold it down, then press the **Left** key. A double beep indicates that **SEQ** is now in independent motor control mode. Remember to clear memory with the **CM** key as earlier movement instructions will not be in this new mode.

In independent motor control mode the **Forward** and **Backward** instructions control the motor attached to the socket immediately above the **Forward** key, and the **Left** and **Right** instructions control the motor attached to the socket above the **Left** key. For exampe:



This selects independent motor control mode. The **Forward 2** instruction will make the motor attached to the **Forward** socket move in one direction. **Left 3** will turn the motor attached to the **Left** socket in the other direction.

To change back to normal control of both motors at the same time, press and hold the **Stop** key, then press the **Forward** key. A double beep indicates normal mode has been selected. Don't forget to press **CM**, or **SEQ** may still remember independent mode instructions. This:



will make both motors move (in opposite directions), so the vehicle will move forward. Then **Left 3** makes both motors move (in the same direction), so it will turn left.

Note that normal motor control makes **SEQ** obey **Forward** and **Backward** instructions by moving the motors 8 times further than **Left** and **Right** (unlike independent motor control). This allows accurate turns to be made.

Inverse switch mode

Normally **SEQ** obeys the **Jump on Switch** instruction (**JSW**) by jumping if the switch is closed (switch light on). Pressing **STOP-JSW** selects inverse switch mode, so **SEQ** obeys all **JSW** instructions by jumping if the switch is open. Pressing **STOP-JSW** again selects normal switch mode.

Changing the pulse count

The number you enter for a **Forward**, **Backward**, **Left** or **Right** instruction controls the number of pulses SEQ counts from each motor's opto-sensor. If the opto-sensors are attached directly to the motor, or you want the vehicle to move further, you can increase this pulse count.

Press and hold the **Stop** key, then press 2. A double beep indicates that the new pulse count has been accepted. SEQ will now count 2 pulses instead of 1. For example, **Left 5** will make SEQ count 10 pulses. The table below shows how many pulses SEQ will count:

Stop-number	Pulse count	
Stop-1	1	(normal)
Stop-2	2	
Stop-3	4	
Stop-4	8	
Stop-5	16	
Stop-6	32	
Stop-7	64	
Stop-8	128	
Stop-9	256	

Left 5 would make SEQ count 1280 pulses if **Stop-9** has been pressed, instead of 5 pulses in normal mode. **Forward 5** would make SEQ count 10,240 pulses if **Stop-9** has been pressed, instead of 40 pulses in normal mode.

Changing the pulse count is like changing gear. You **do not** have to press **Clear Memory** after changing pulse count.

Output sockets

The 4 output sockets are each marked with a red frame. You can connect anything that expects to receive electricity at a similar voltage to the batteries. Usually an electric motor, light bulb or relay will be used.

You will need to put small Lego-type plugs (2.5mm diameter) on the end of the wires from the motor or light. Plug these into one of the pairs of output sockets. The hole in between each pair of sockets is to allow Lego plugs to be connected, and is otherwise not used.

To experiment, connect something to the **Out** socket, directly above the key marked **Out**. Press **Out 10**, then **Go**. The motor or light should work for one second.

Notice that the red indicator light below the **Out** socket also comes on. This show you that power is being sent to the socket. If the motor or light doesn't work, check the wires and the plugs for lose connections. If the motor moves in the wrong direction, plug it in the other way around. It isn't possible to reverse the electricity direction of the **Pulse** and **Out** sockets.

The **Left Motor** socket and indicator is directly above the **Forward** key. The **Right Motor** socket and indicator is directly above the **Left** key. These output sockets can change electricity direction, so you can reverse the motors. When the indicator light is red, the electricity is going in the same direction as the **Pulse** and **Out** sockets. When the indicator light is green, the direction has been reversed.

Input sockets

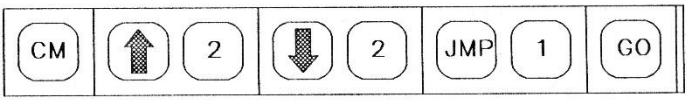
There are four input sockets, and each one has a yellow indicator light. You can connect anything that can change from on to off to the input sockets. Usually switches or opto-sensors are used.

Experiment by connecting a simple, normally open switch to the **Switch** input, immediately above the **JSW** key. When the switch is open, the yellow indicator light will be off, and when you close the switch it will come on. If it doesn't do this, check the wires to the switch and the plugs for lose connections.

You can also use opto-sensors; the most convenient ones are Lego opto-sensor bricks. These can be used to provide an input from external or reflected light. They are specially useful in providing feedback from shafts or wheels driven by motors attached to the motor outputs.

Place a Lego opto-sensor brick on each shaft together with a counting disc. Connect the opto-sensor on the shaft driven by the left motor to the socket on **SEQ** next to the **Left Motor** output socket (directly above the **Backward** key). Similarly connect the opto-sensor on the shaft driven by the right motor to the socket on **SEQ** next to the **Right Motor** output socket (directly above the **Right** key).

Test the opto-sensors by entering:



As each disc rotates pulses are sent to **SEQ**, and the yellow indicator lights will flash on and off. Press the **Stop** button to halt this sequence. You may need to adjust the distance between the opto-sensor bricks and each counting disc. Bright lights can confuse the sensors too!

If the amount of rotation is too small, you can increase the pulse count using the Stop-number command.

Beep messages

Each instruction and number key has it's own note. When **SEQ** accepts an instruction or numeric key press, this note will be heard (unless the beeper has been disabled). Other beeper noises may be heard:

I'm awake (3 falling notes): power connected, self test ok.
All clear (2 notes, high-low): **CM** all instructions cleared.
I'm going now (4 notes): **GO** (at start & finish).
I'm full up (3 notes, high-low-high): no more room.
STOP mode (2 notes, both the same): mode changed.
I'm still awake (4 notes, all the same): unused for 5 mins.

Squawks: a **Pulse** instruction has been obeyed.

Technical details

SEQ requires a power supply capable of providing 4.5 to 12 volts at a sufficient current to drive all the motors, lights and sensors that may be connected. In practice this means at least 1.5 amps. The battery voltage should also be suitable for the motors, lights and sensors. For Technical Lego with 4.5 volt motors, four 1.5 volt cells (high power, alkaline or rechargeable) are fine. The first sign of too low a battery voltage is that opto-sensors start behaving erractically.

Reverse battery connection will not damage SEQ, but the green battery light will stay off. With no sensors or output devices, SEQ uses around 0.1 amps. All outputs are individually short circuit protected.

The motor outputs are capable of delivering 1 amp (with a 2 amp peak, allowing for surge) at around 1.5 volts below the battery input voltage.When their indicator lights show red the top socket is positive; when green, the polarity is reversed and the bottom socket is positive.

The Pulse and Out outputs can deliver the same current at around 0.9 volts below battery input voltage. The top socket of each pair is always positive.

Each input can supply up to 50 mA to a sensor. The top socket of each pair is always positive. The inputs work by testing the voltage on the lower socket. If this voltage is near to 0 volts, the input is treated as off, and the indicator light stays off. When the voltage on the lower input socket becomes more positive than a pre-set minimum, the input is treated as on, and the indicator will come on.

Quick reference guide

Output instructions:

(each instruction expects a number from 1 to 99)

Forward amount:

normal mode:	both motors forward
independent mode:	left motor forward

Backward amount:

normal mode:	both motors backward
independent mode:	left motor backward

Left amount:

normal mode:	left motor forward, right motor backward
independent mode:	right motor forward

Right amount

normal mode:	left motor backward, right motor forward
independent mode:	right motor backward

Pulse t:

Pulse t times

Out t:

Output for t tenths of a second

Control instructions:

Pause t:

Wait for t tenths of a second

Repeat n:

Do the last n instructions again

Jump n:

Jump to instruction n

Jump on switch n:

Jump to instruction n if switch is closed

Commands:

Clear Memory:
Forget all instructions

Clear Error:
Forget the last instruction

Go:
Obey the instructions stored in memory, starting at instruction 1, and stopping when:
there are no more instruction to obey
or the **Stop** key is pressed
or the Stop input sensor is activated

Test:
Obey the last instruction in memory

Stop:
Stop obeying instructions

Stop mode commands:

Stop & Left:
Select independent motor mode

Stop & Forward:
Select normal motor mode (both at once)

Stop & n:
Set motor opto-sensor pulse count

Stop & Flash:
Select beeper on or off

Stop & JSW:
Set inverse or normal switch mode