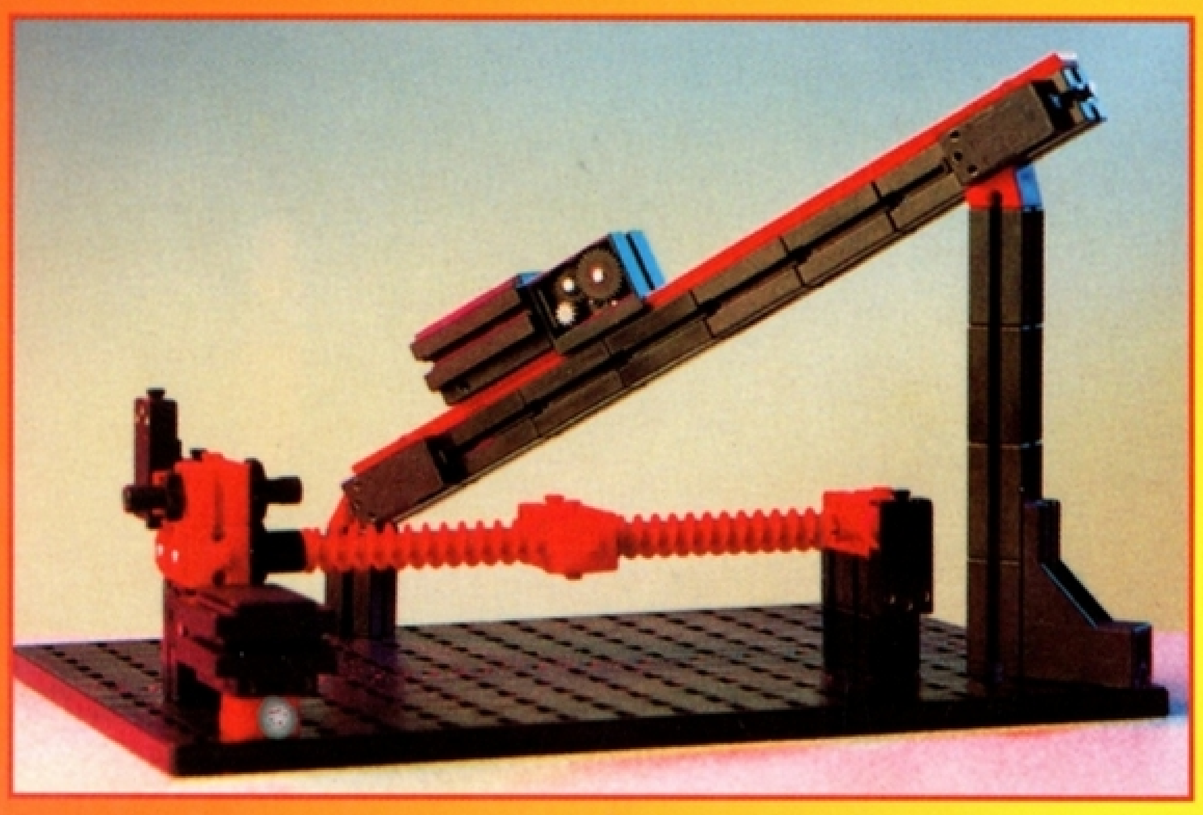
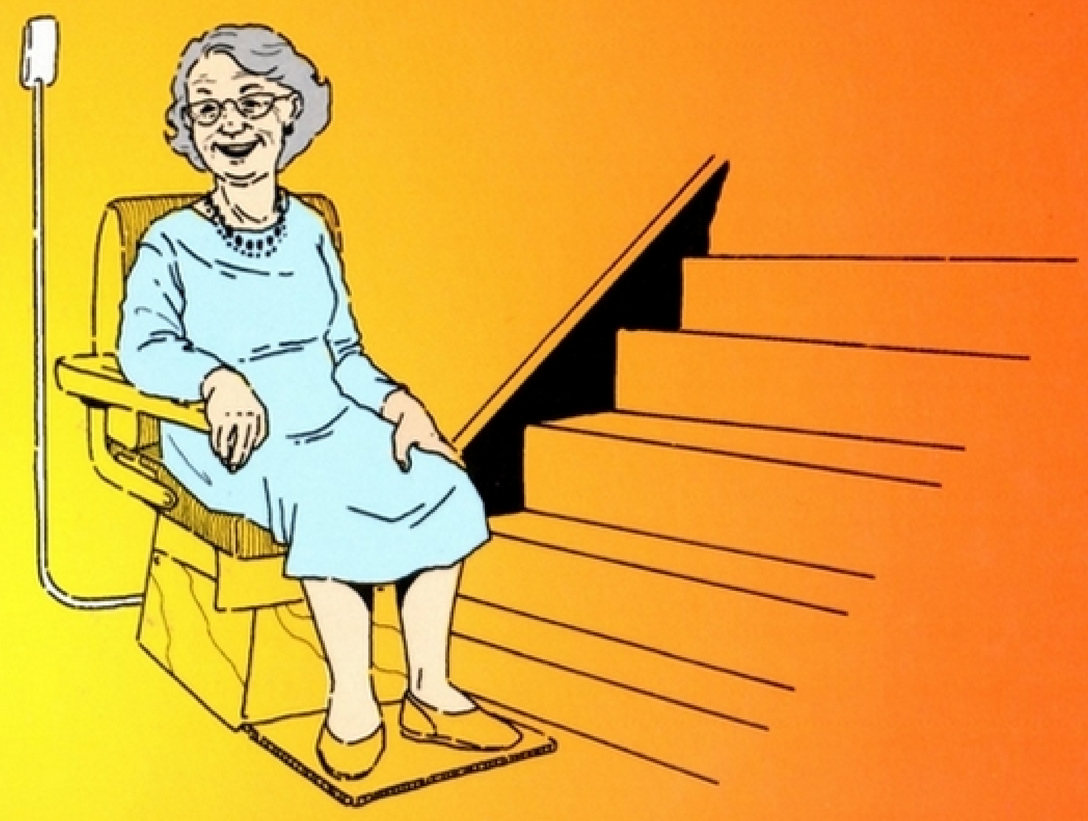
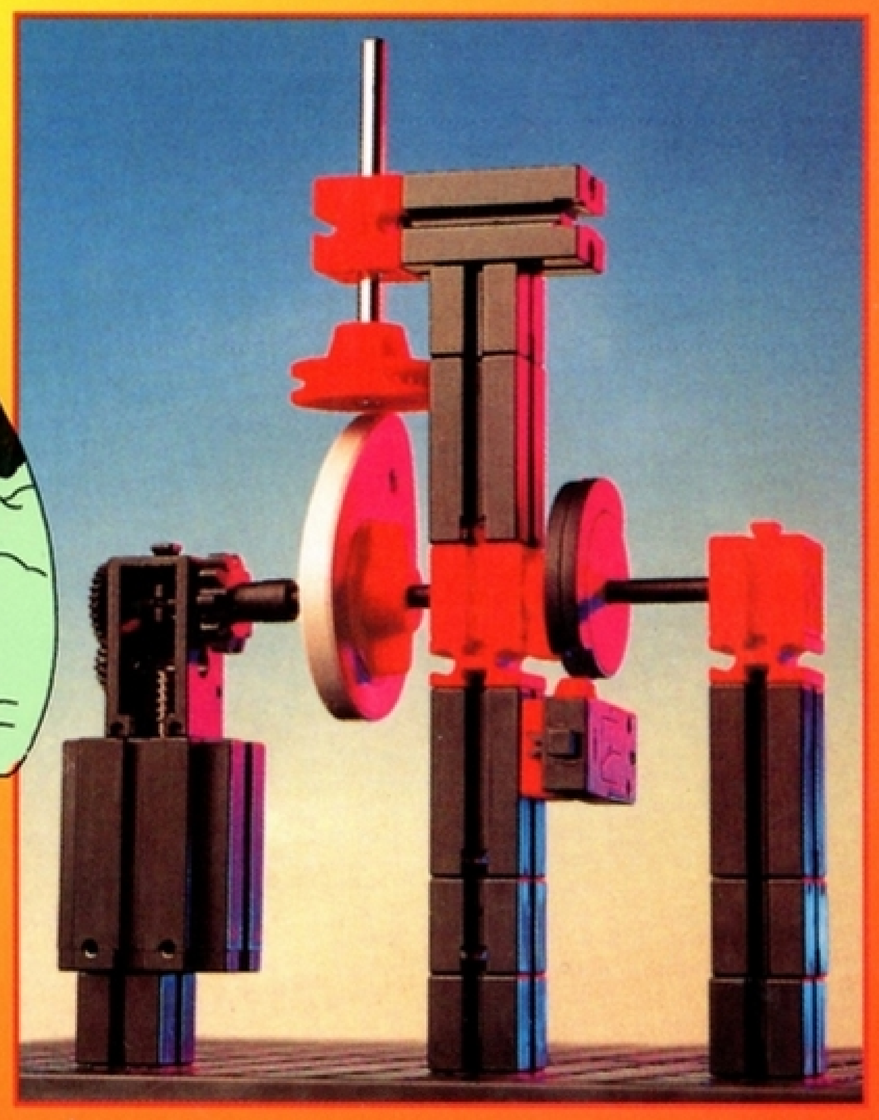
fischertechnik

Designer KIT







ECONOMATICS

fischertechnik®

Designer KIT

ECONOMATICS

Contents

Teacher's Introduction

Logicator and Smart Move

The Start Up and Controlling Movement sections of the book are presented in two versions: one using Logicator software; the other using Smart Move software.

Help Sheets

As there are differences in the way in which procedures are built, run and edited in the two different versions of Logicator and the three different versions of Smart Move, no specific instructions for doing this are included on the worksheets in the Start Up and Controlling Movement sections.

A photocopiable Help Sheet showing the basics of building, running and editing procedures, is provided for each version.

It is important that students have a copy of the appropriate Help Sheet to use with the worksheets.

Start Up

This section is designed for students who have little or no knowledge and experience of using a control interface box and control software. It aims to teach beginners how to use the software to create, run and edit control procedures so that they can:

- Switch output devices on and off in predetermined sequences.
- Switch motors forward and reverse and control their speed.
- Build and use a system that responds to data from sensors (microswitch and digital light sensor).

It also aims to teach a structured approach to designing control systems, including understanding that:

- Systems have input, process and output.
- That efficiency and economy can be achieved in a software system by designing it as a set of linked procedures.

Controlling Movement

This section is designed for students who have the knowledge and understanding of using a control interface box and control software, as covered in the Start Up section.

It aims to teach the following skills and knowledge:

- The use of a wider range of commands and control techniques available through software.
- Reinforcement and extension of the structured approach to designing control systems as input, process and output.
- Development of the use of linked procedures for economy and efficiency.
- The use of a range of mechanical systems to achieve different kinds of movement. This includes: rack and pinion, leadscrew, worm and wheel, crank and slider, cam and follower, gears.
- The use of a wider range of sensors to monitor external events and conditions to feed back data to a control system. This includes digital sensors (limit switch, rotation counter, cam-indexed switch, light sensor), and analogue sensors (switching at a threshold and varying output in relation to varying input).

Note that the Designer Kit does not include analogue sensors. The final part of this section makes use of Smart Light, Temperature and Position sensors in conjunction with the kit.

Project File

This section of the book contains six different projects that provide extension work and assessment opportunities for students who have completed the Controlling Movement section of the book. The projects require that they design and build systems of their own, applying their knowledge and understanding of control to a range of new contexts.

The Designer Kit enables rapid realisation of the essential hardware elements of a system so that software procedures can be tested and modified, and so that designs for construction can be modelled.

Construction Sheets

This section contains a guide to the Fischertechnik construction system and step by step building instructions for the simple models used in the Start Up and Controlling Movement sections of the book. It also provides building instructions for models that can be used in conjunction with the Project File.

Suggested Solutions

This section includes suggested solutions to all the design problems posed in the Start Up, Controlling Movement and Project File sections of the book. It is presented in two versions: one using Logicator software; the other using Smart Move software.

The solutions suggested are not necessarily the only solutions to the problems posed. Students may work out equally good, but different solutions of their own. The only test is: "does it meet the specification?". However some solutions will be more elegant than others.

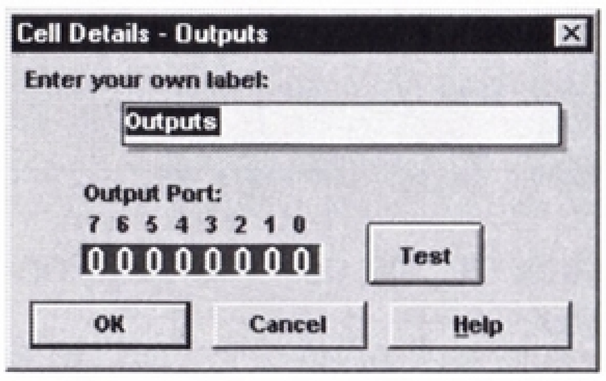
The suggested solutions need not be the end of the activity. Students may go on to develop and improve the system beyond the basic requirements stated in the problem.

Logicator

Help Sheet Logicator for Windows

Outputs

Drag an Outputs command onto the flowsheet. Double click on it to open its Cell Details box:



Each one of the digits in the Output Port represents one of the Digital Outputs sockets on the Smart Box. You can click each digit to set it to switch an output device on or off.

0 This means: switch this output off.

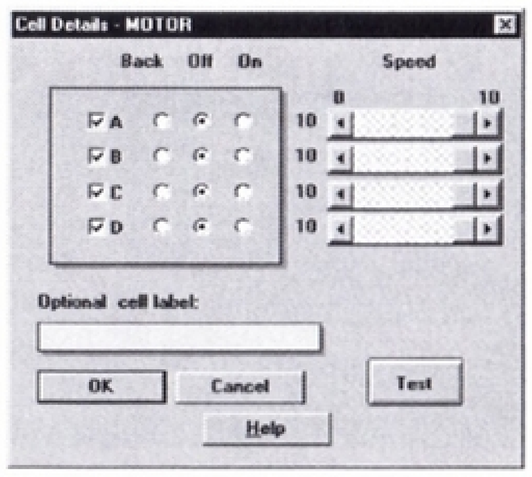
1 This means: switch this output on.

When you have set the digits as you want them, click on the Test button and hold it down to check that the command will do what you want it to do.

Click OK.

MOTOR

Drag a MOTOR command onto the flowsheet. Double click on it to open its Cell Details box:



Each Motor output on the Smart Box is represented by three buttons: On, Off, Back.

Click the On button for your motor. Then click the Test button and hold it down to check which way your motor moves. If you want the motor to move in the opposite direction, then click the Back button.

If you want the command to switch the motor off, click the Off button.

You can use the slider beside the buttons to set the speed of the motor. Try altering the speed and check the result by using the Test button.

When you have set the command to do what you want it to do, then click OK.

WAIT

Drag a WAIT command onto the flowsheet. Double click on it and type the number of seconds you want the flowsheet to wait. If you like, you can select the number from the drop-down list instead.

Click OK.

Drawing routes

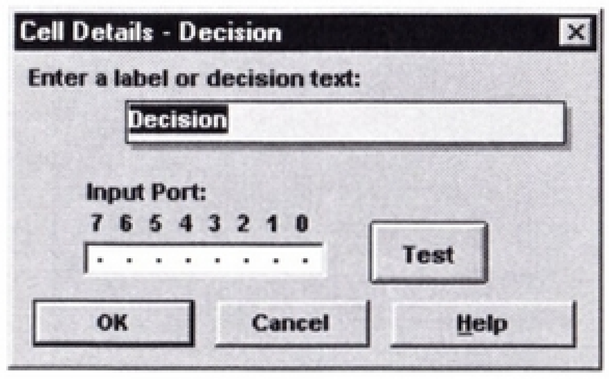
Click with the right mouse button on the command at the start of the route. The cursor will change to a pencil.

Hold down the right mouse button and draw the route in the direction that you want flow to take when the flowsheet runs.

Release the mouse button when you have completed the route.

How to set the details of a Decision command to check if a switch is pressed.

Drag a Decision command onto the flowsheet. Double click on it to open its Cell Details box:



Each one of the digits in the Input Port represents one of the Digital Sensors sockets on the Smart Box.

Press the switch and check the number of the yellow light that lights up on the Smart Box. This will tell you which digit to use.

Click the digit to set it to 1.

Click OK.

Use the same method to set the command to check if a digital light sensor is on.

Drawing routes from a Decision command

The first line that you draw from a Decision command is the "Y" direction, and the second line is the "N" direction.

You can swap the "Y" and "N" routes by double-clicking on the Decision command with the right mouse button.

Click on Help on the menu bar for more information about using Logicator.

Deleting routes

Click at the beginning of the route to be deleted, and press the Delete key.

When you draw a new route from a command, the existing route from the command will automatically be deleted.

To delete a route without deleting the command in which it starts: first click on the command to select it. Then hold down the Ctrl key as you press the Delete key.

Labelling a command

Double click on the command to open its Cell Details box.

The text in the yellow "label" box will be highlighted, so just type your label and click OK.

Deleting a command

Click on the command to select it. Selected commands are coloured red.

Press the Delete key to delete the selected command.

To delete a block of commands, select the block and press the Delete key.

Selecting a block of commands

Hold down the Ctrl key as you click at the top left hand corner of the block and drag down to the right to enclose the required commands and routes in the selection frame.

Selected commands are coloured red. To deselect commands, click on another part of the flowsheet.

Moving commands.

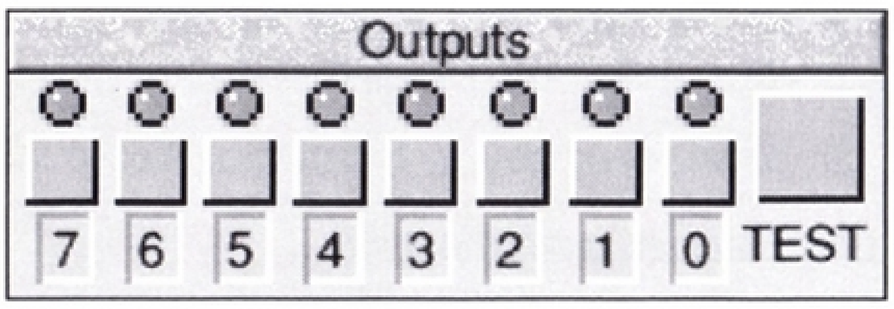
To move a single command, drag it to its new position.

To move a block of commands, select them and drag them to their new position.

Help Sheet Acorn Logicator

Outputs

Drag an Outputs command onto the flowsheet. Click on it to open its details box:



Each one of the buttons in this box represents one of the Digital Outputs sockets on the Smart Box. You can click each button to set it to switch an output device on or off.

This means: switch this output off.

This means: switch this output on.

When you have set the buttons as you want them, click on the Test button and hold it down to check that the command will do what you want it to do.

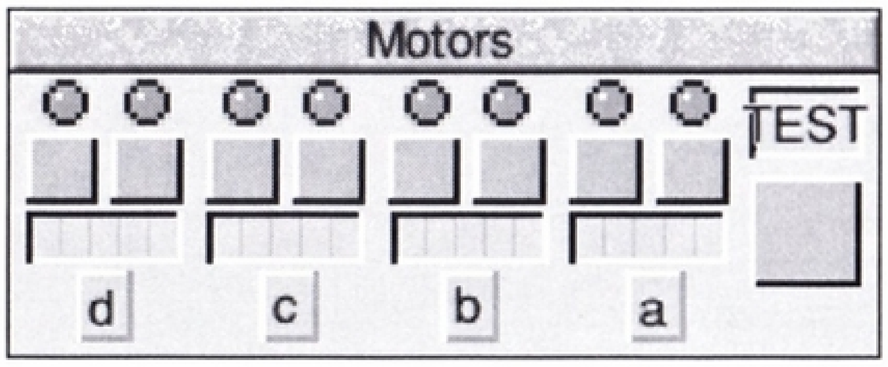
Press Return.

WAIT

Drag a WAIT command onto the flowsheet. Click on it and type the number of seconds you want the flowsheet to wait. Press Return.

Motors

Drag a Motors command onto the flowsheet. Click on it to open its details box:



Each Motor output on the Smart Box is represented by a pair of buttons which you can use to set the motor:

This means: switch this motor off.

This means: switch this motor on in one direction.

This means: switch this motor on in the opposite direction.

Click one of the buttons for your motor. Then click the Test button and hold it down to check which way your motor moves. If you want the motor to move in the opposite direction, then click the other button.

You can reduce the speed of the motor by clicking on sections of the red line below the buttons. Try altering the speed and check the result by using the Test button.

When you have set the command to do what you want it to do, then press Return.

Drawing routes

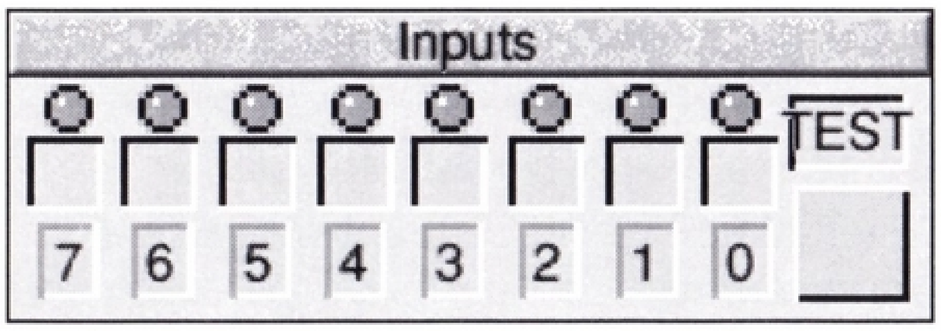
Click with the right mouse button on the command at the start of the route. The cursor will change to a pencil.

Hold down the right mouse button and draw the route in the direction that you want flow to take when the flowsheet runs.

Release the mouse button when you have completed the route.

How to set the details of a Decision command to check if a switch is pressed.

Drag a Decision command onto the flowsheet. Click on it and select "input bits" from the "Type" box.



Each one of the buttons in this box represents one of the Digital Sensors sockets on the Smart Box.

Press the switch and check the number of the yellow light that lights up on the Smart Box. This will tell you which button to use.

Click the button twice to set it to a yellow square.

Click on another part of the flowsheet to close the box.

Use the same method to set the command to check if a digital light sensor is on.

Changing the routes from a Decision command

Open the command and click the "swap" button in the Route box.

Deleting routes

Click at the beginning of the route to be deleted, and press the Delete key.

When you draw a new route from a command, the existing route from the command will automatically be deleted.

Labelling a command

Click on the command to get the text cursor. Type your label and press Return.

Deleting a command

Make sure that it is deselected (coloured blue), then drag it back into the Commands window.

Selecting a block of commands

Click at the top left hand corner of the block to get the cross icon. Then drag down to the right to enclose the required commands and routes in the selection frame.

Selected commands are coloured red. To deselect commands, click on another part of the flowsheet.

Moving commands

To move a single command, drag it to its new position.

To move a block of commands, select them and drag them to their new position.

The Software Guide book gives more information about using Logicator.

Introduction to computer control

Many of the machines that are used in the home and in industry are controlled by a microprocessor that has been programmed to operate the various electrical devices that do the work in the machine.

For example, an automatic washing machine might contain:

an electric motor to rotate the tub,

electrically-operated valves to let water in and out,

a heater to heat the water.

These are called **output** devices.

The washing machine also contains switches and sensors. For example, there might be:

an on/off switch,

a key-pad to select different washing cycles,

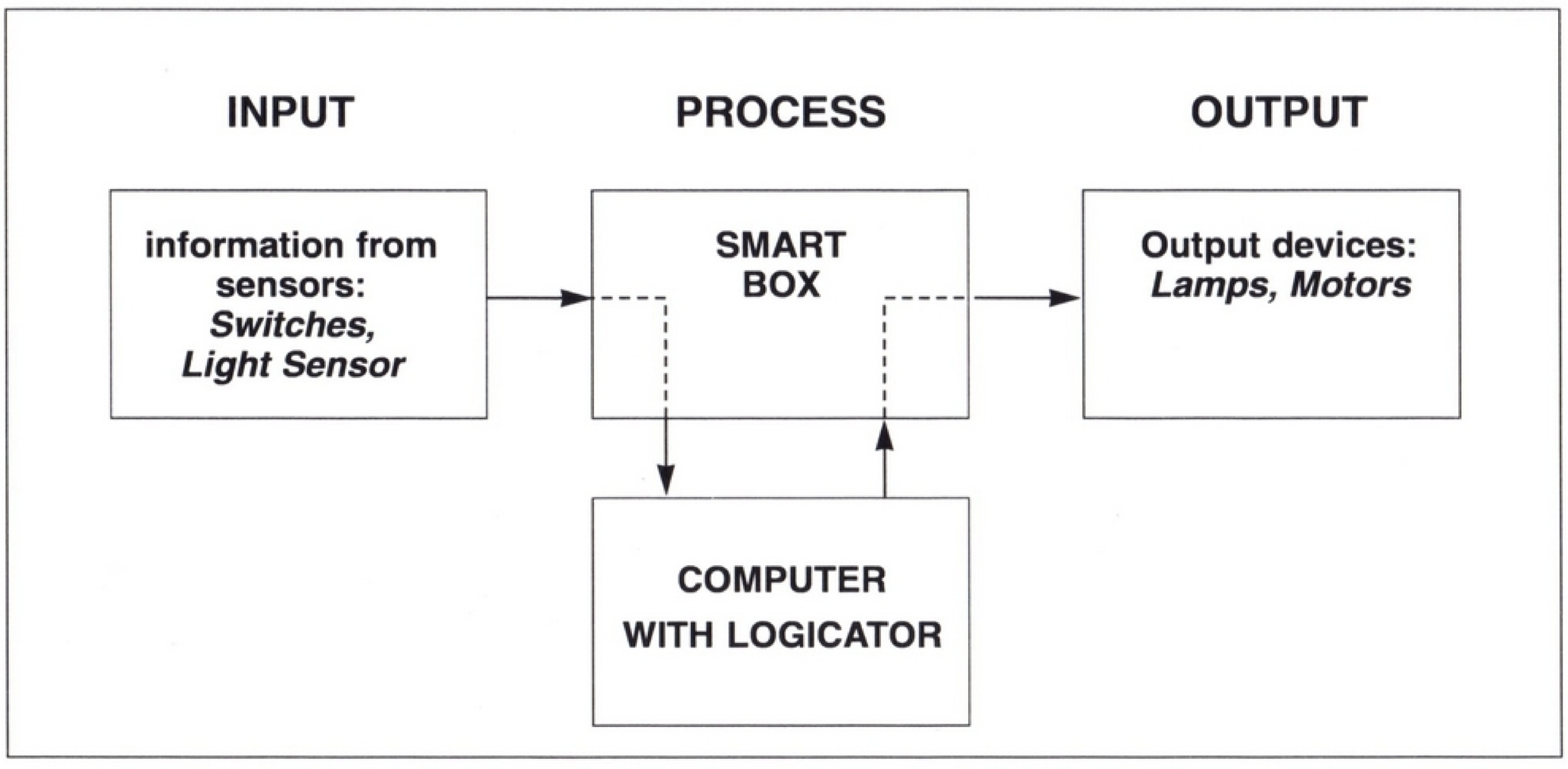
a sensor to check the temperature of the water.

These are called **input** devices.

The job of the microprocessor is to check the state of the input signals from these switches and sensors, and to switch the output devices on and off as they are needed.

This is called the **process**.

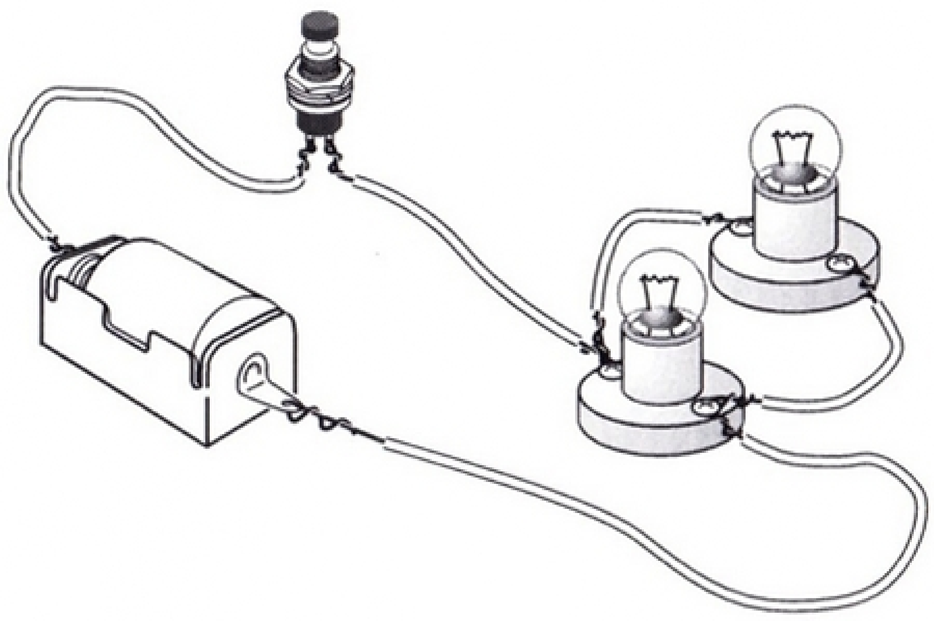
All control systems have input, process and output. The diagram below shows how the Smart Box and computer make up the process stage of the control systems that you will build. You will use Logicator software to design and build the programs that control these systems.



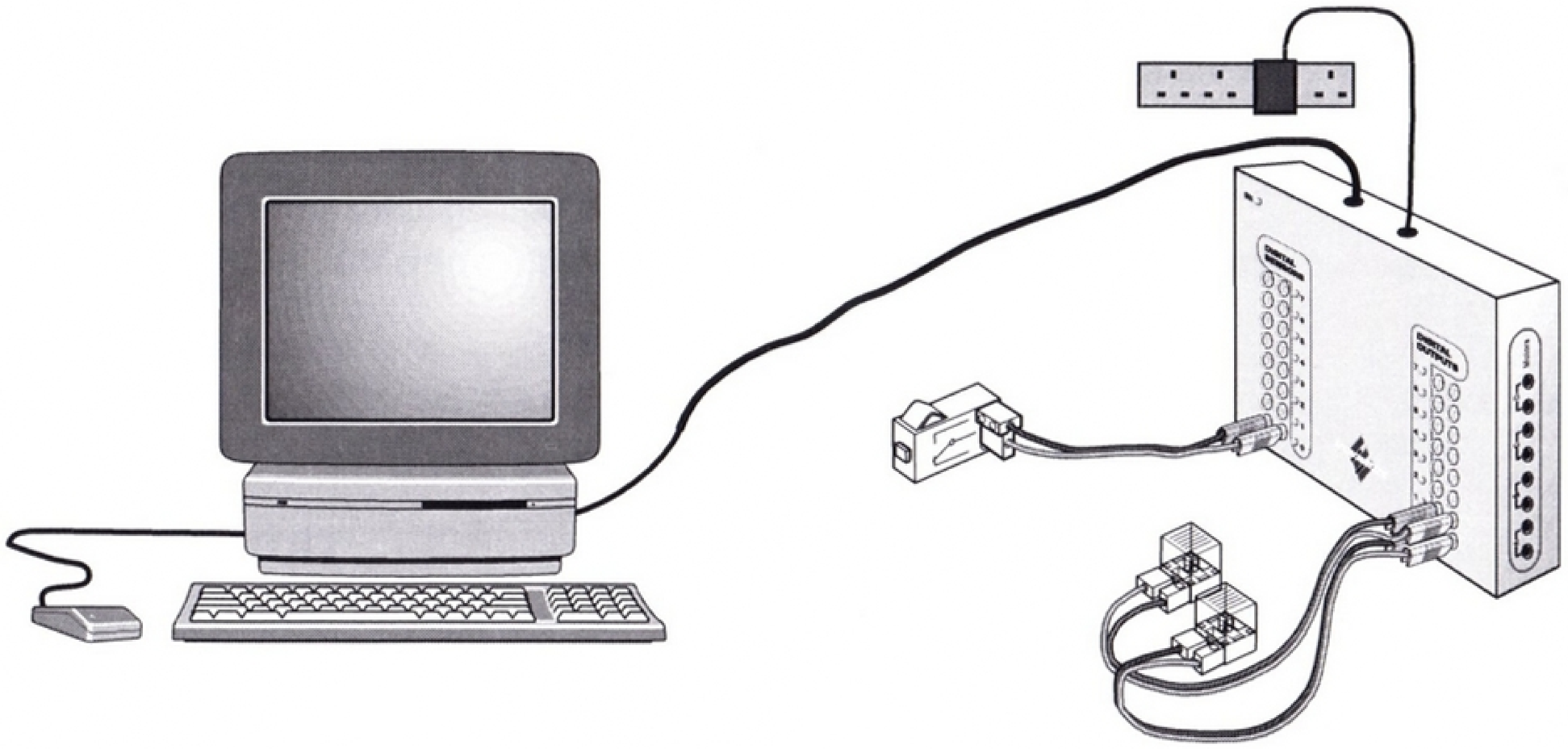
The Smart Box is a control interface. It has two main functions:

1. It switches output devices such as motors and lamps on and off in response to signals from the computer. It also provides the power to drive these devices. The Smart Box contains a transformer which converts the mains voltage into the low voltage power needed for the motors and lamps on the models.
2. It processes information from sensors into a form that the computer can deal with.

This diagram shows the advantages of using a computer-based control system. Can you think of any disadvantages?



There are only two control possibilities with this circuit - the bulbs are either on or off, depending whether the switch is pressed. If you want to change the way the circuit behaves, you have to reconstruct it.



Using the computer provides multiple control possibilities for the same components - the bulbs can be switched on and off in any combination and sequence for any length of time. The sequence can be made to repeat itself or change automatically after a particular time. The sequence can be programmed to start immediately a switch is pressed, or after a short time delay. The bulbs can be programmed to switch on automatically under particular conditions, e. g. when it gets dark.

One of the biggest advantages of computer control is that software is used to create the instructions for all these control possibilities, so they can be changed easily and quickly.

START UP 1 : Control sequence for lamps

Build the model shown on the Hand Dryer construction sheet. There are two lamps on the model (see fig. 1). They simulate the heater.

The lamps are output devices. They should be connected to Digital Outputs sockets on the Smart Box.

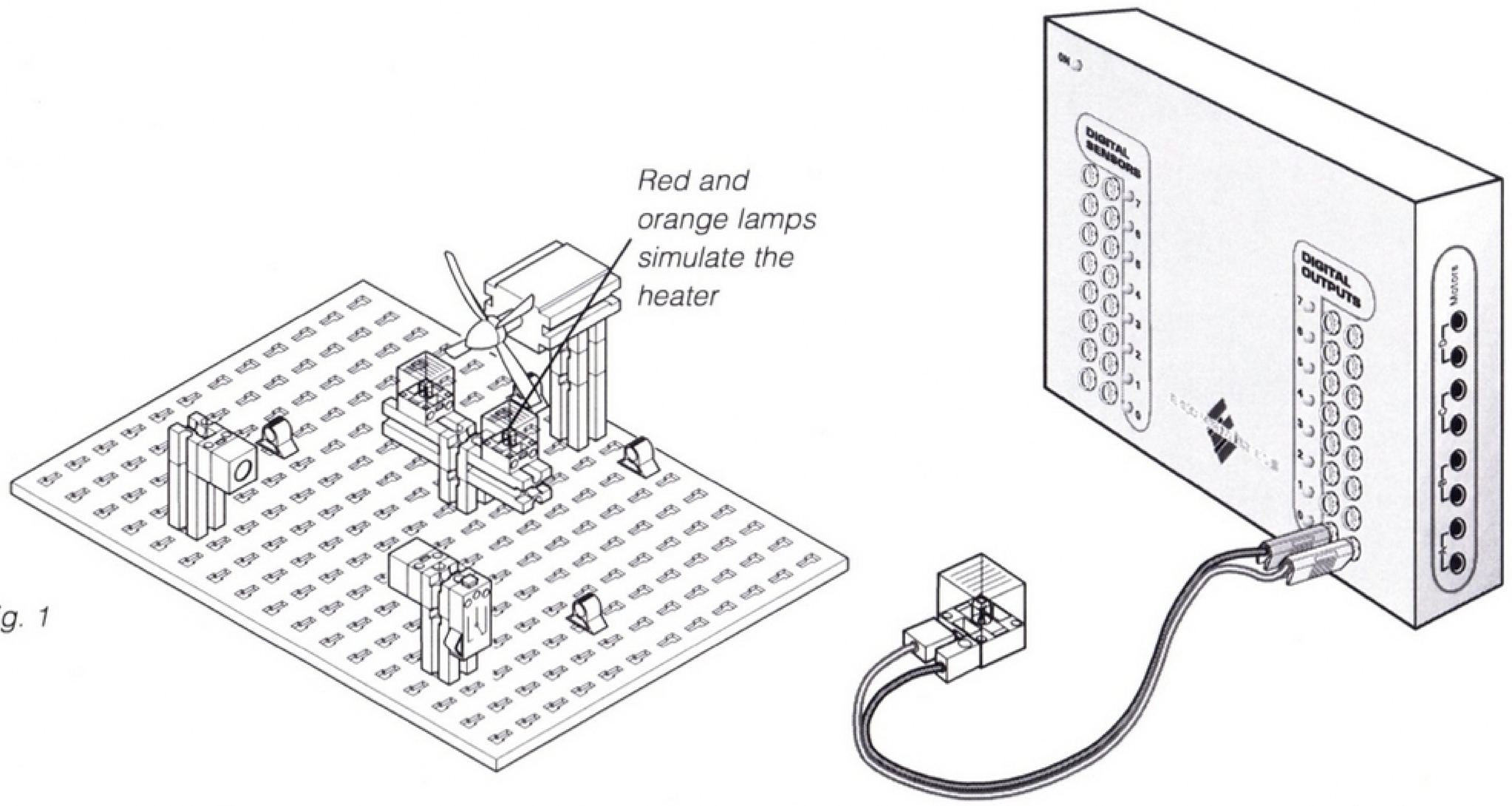


Fig. 1

Task One

You use an Outputs command to switch on or off any of the output devices connected to Digital Outputs sockets on the Smart Box.

You use a WAIT command to keep the output devices switched on or off for any length of time you like.

The Help Sheet shows you how to use these commands. Use it to help you to build the routine shown in fig. 2

Set the Outputs and WAIT commands so that the flowsheet:

switches on both of the lamps,

waits for 2 seconds,

then switches them off.

Simply drag the STOP command onto the flowsheet.

When you have built the routine, click the green button on the screen to run your flowsheet.

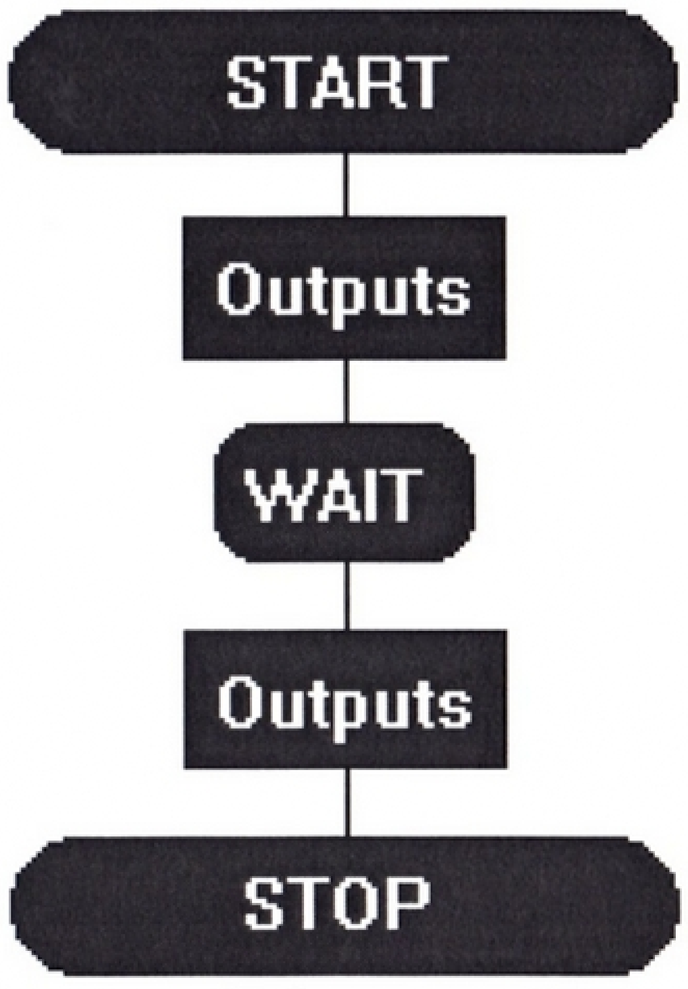


Fig. 2

Task Two

Add two more commands to the flowsheet so that it looks like fig.3.

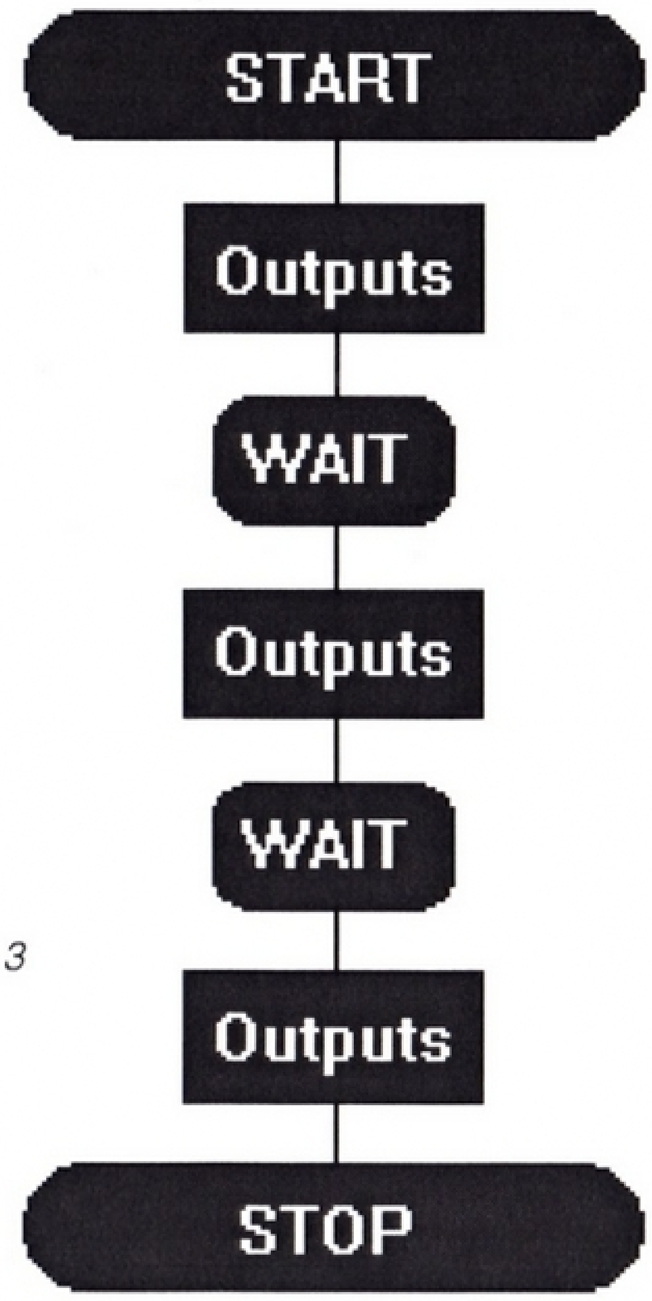


Fig. 3

Then set the commands so that the flowsheet:

switches on the red lamp,

waits 1 second,

switches on the orange lamp, so both lamps are on,

waits 1 second,

switches off both lamps.

Run the flowsheet.

Task Three

Each time you want to start the light sequence, you have to re-run the flowsheet.

It is more efficient to add a repeat loop as shown in fig. 4, so that the flowsheet repeats the sequence automatically.

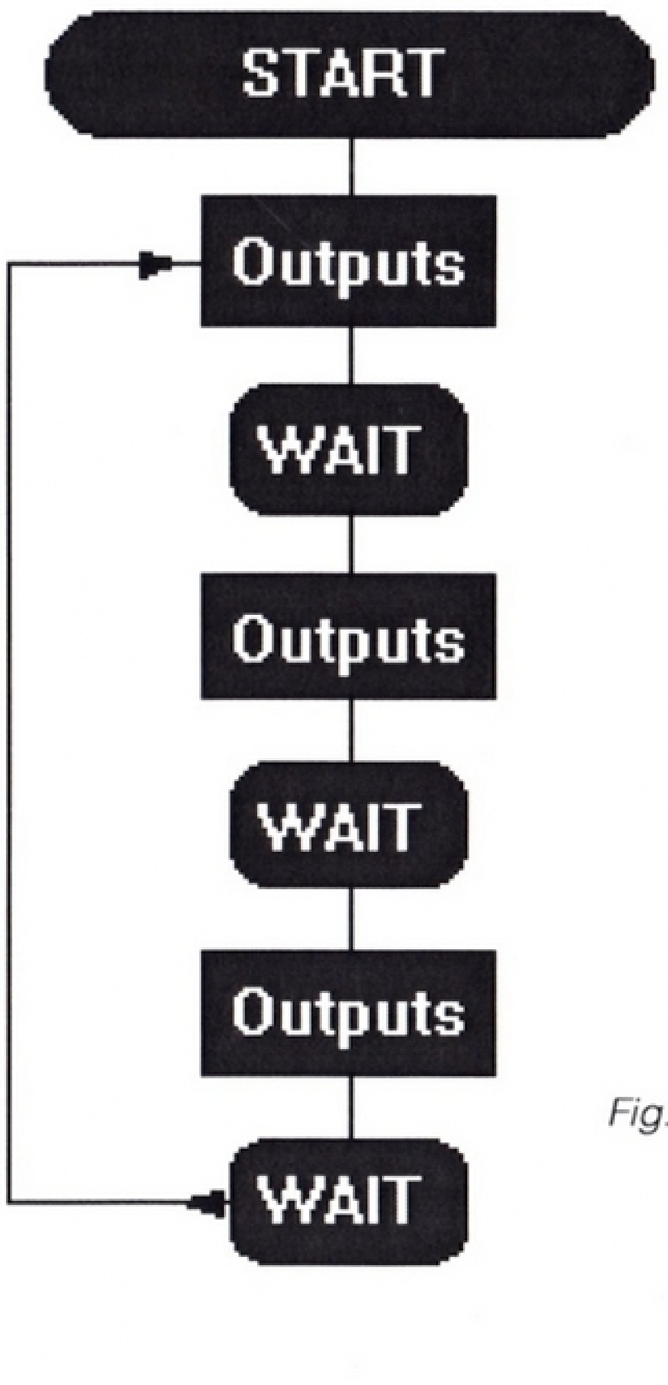


Fig. 4

Follow these instructions to edit your flowsheet to look like fig. 4:

1. Delete the STOP command. The Help Sheet tells you how to do this.

2. Add a WAIT command in its place. Set it to wait for 1 second. This gives you time to see that the lamps have switched off.

3. Use the right mouse button to draw the route from this WAIT 1 command back up to the beginning of the flowsheet.

Run the flowsheet to test it.

Click the red button on the screen to stop the flowsheet running.

START UP 2 : Controlling a motor

The electric motor on the Hand Dryer model drives a fan that blows air over the heater and then onto the user's hands (see fig. 5).

The motor should be connected to a Motors socket on the side of the Smart Box.

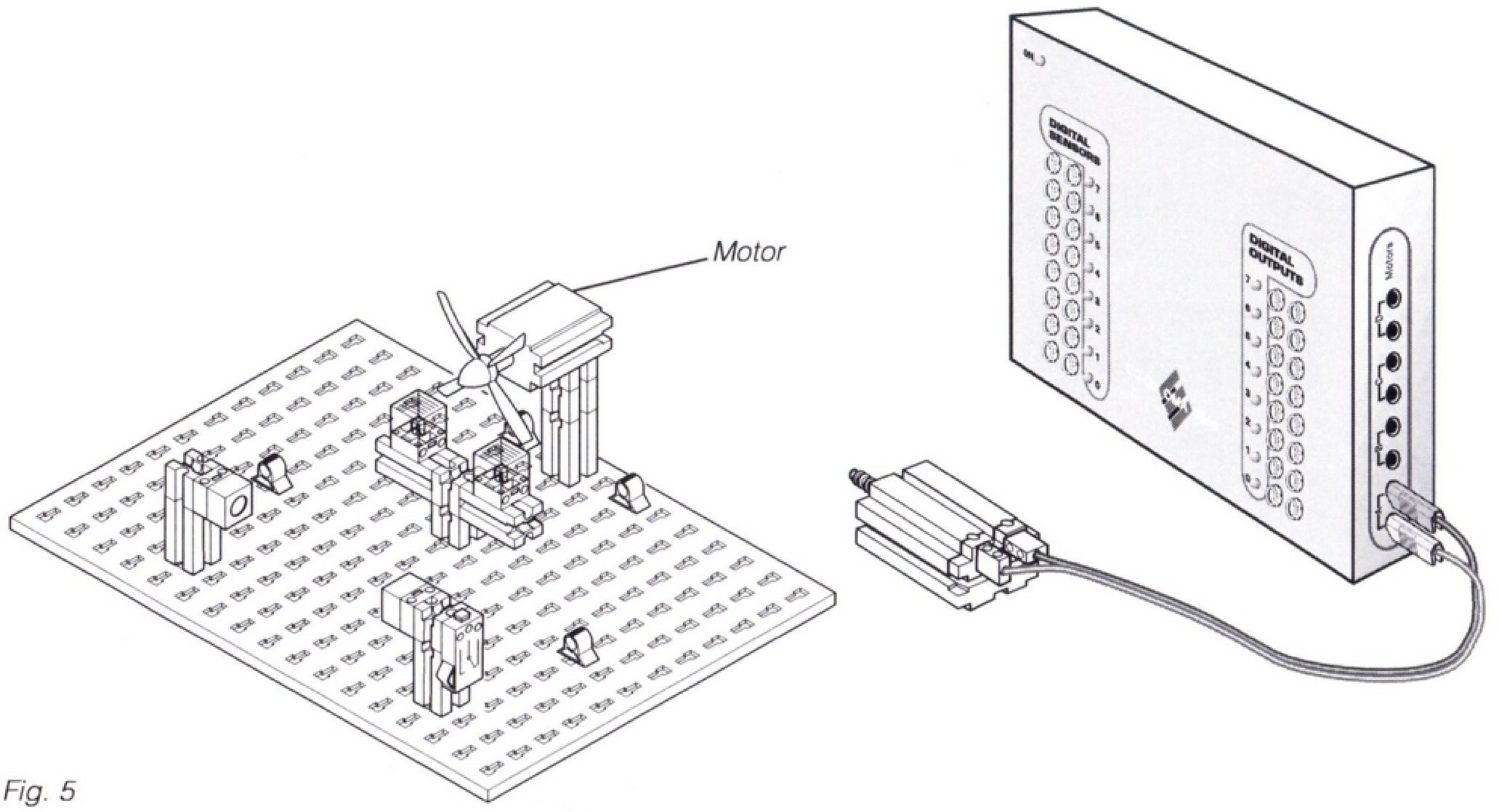


Fig. 5

Task Four

Select File_New from the Menu to begin a new flowsheet.

You use a MOTOR command to switch on or off any of the motors connected to Motors sockets on the side of the Smart Box.

Use the Help Sheet to help you to build the routine shown in fig. 6. Set the commands so that the flowsheet:

switches on the motor,

waits for two seconds,

then switches the motor off.

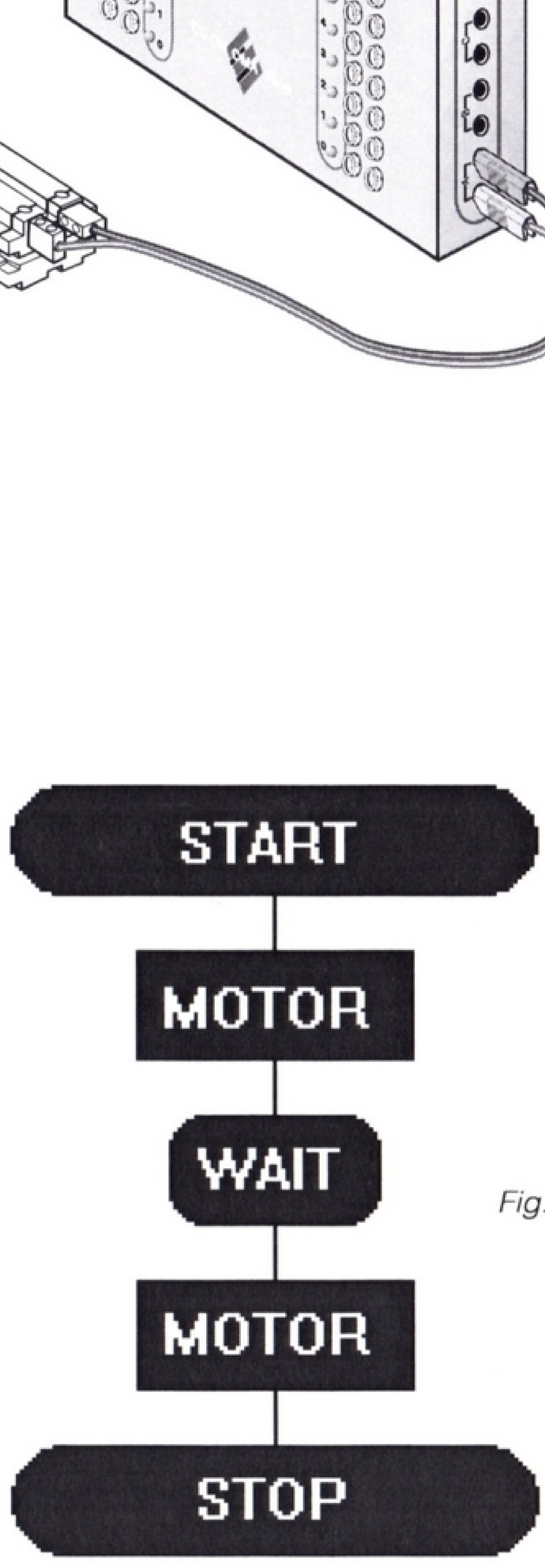


Fig. 6

Task Five

Edit your flowsheet so that:

The motor moves clockwise for 3 seconds,

It then switches off for 1 second.

It then moves anticlockwise for 3 seconds,

The motor then stops.

Run your flowsheet to test it

START UP 3 : Using input signals from micro-switches

Fig. 7 shows the micro-switch on the model. It can be used to switch the dryer on.

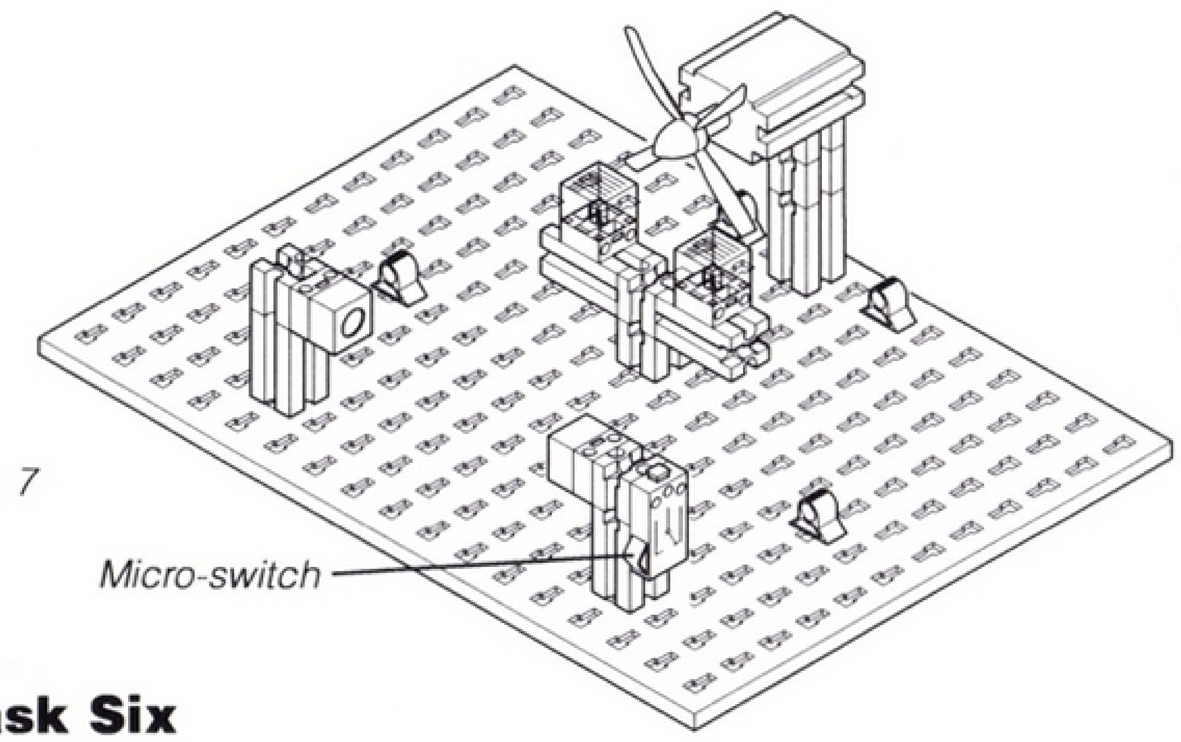
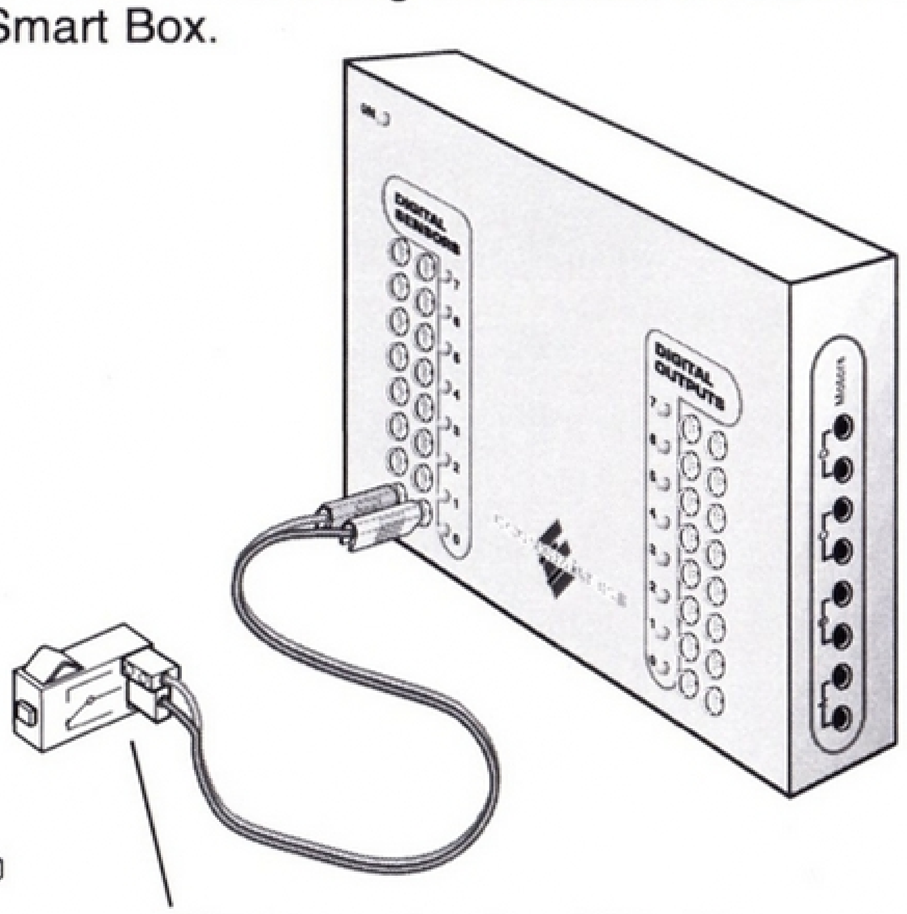


Fig. 7

The micro-switch is an input device. It should be connected to the Digital Sensors sockets on the Smart Box.



Plug into sockets 1 and 3 on the switch

Task Six

Select File_New from the Menu to begin a new flowsheet.

This task shows how you can use the input signal from a switch to start a sequence. You use a Decision command to check the input signal from a switch connected to Digital Sensors sockets.

Use the Help Sheet to help you to build the routine shown in fig. 8.

Set the Decision command to check if the switch is pressed. Set the other commands so that the two lamps switch on for 2 seconds and then switch off.

Run the flowsheet.

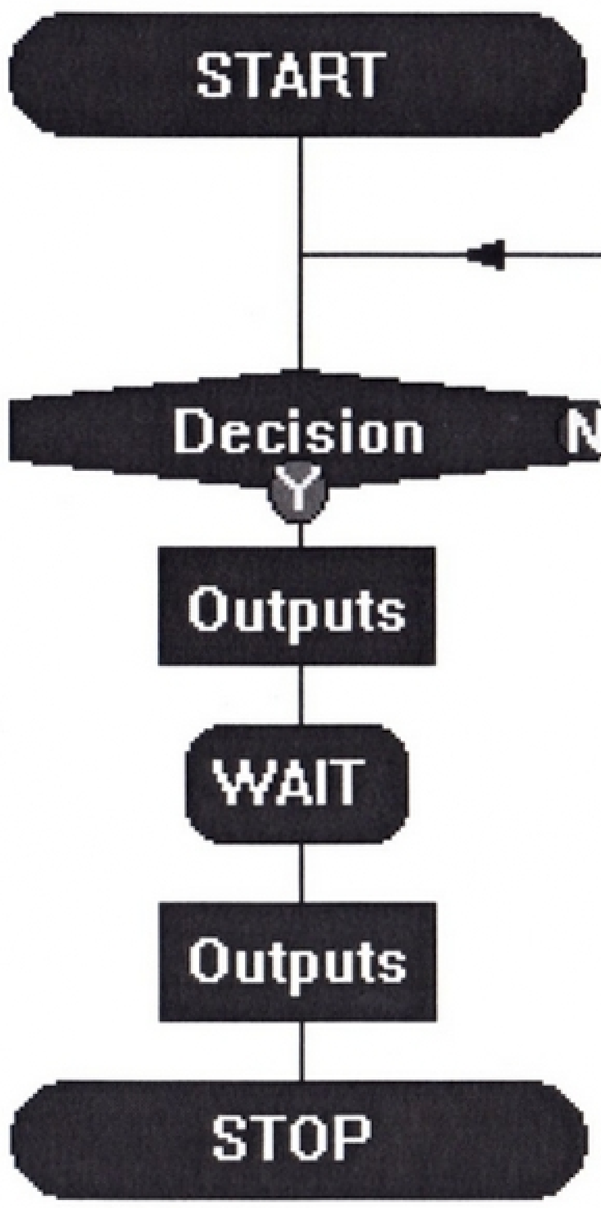


Fig. 8

Task Seven

Each time you want to use the switch to start the light sequence, you have to re-run the flowsheet. It would be more efficient to add a repeat loop so that the flowsheet returns automatically to check the switch again when the sequence has stopped.

Edit your flowsheet so that it does this.

Run the flowsheet to test it.

START UP 4 : Using input signals for motor control

Task Eight

This task shows how you can use the input signal from a switch to start or stop a motor.

Select File_New from the Menu to begin a new flowsheet.

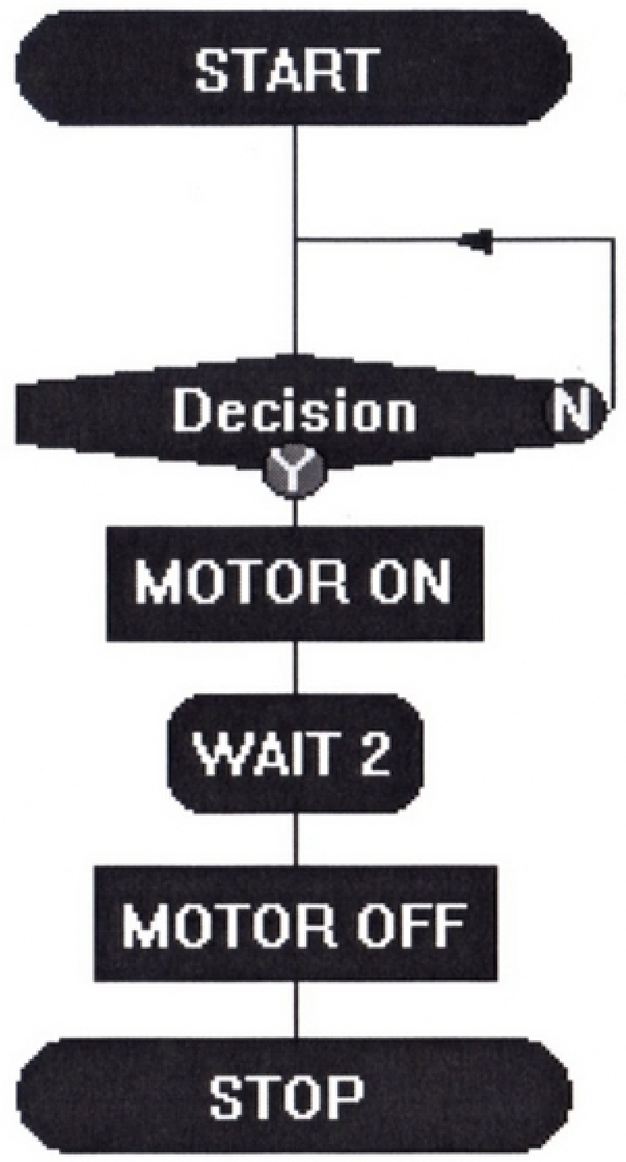
Build the routine shown in fig. 9. This flowsheet will start the motor when you press the switch.

The MOTOR commands have been given labels to show what they do, but don't forget to set the buttons in the details box as well.

Set the Decision command to check if the switch is pressed.

Run the flowsheet to test it.

Fig. 9



Before you do Task Nine, follow these instructions to set the motor so that it provides the correct output for a Hand Dryer:

The fan can turn in two different directions. In one direction, it will blow air. In the other direction, it will draw air in. The hand dryer needs to blow air out over the lamps.

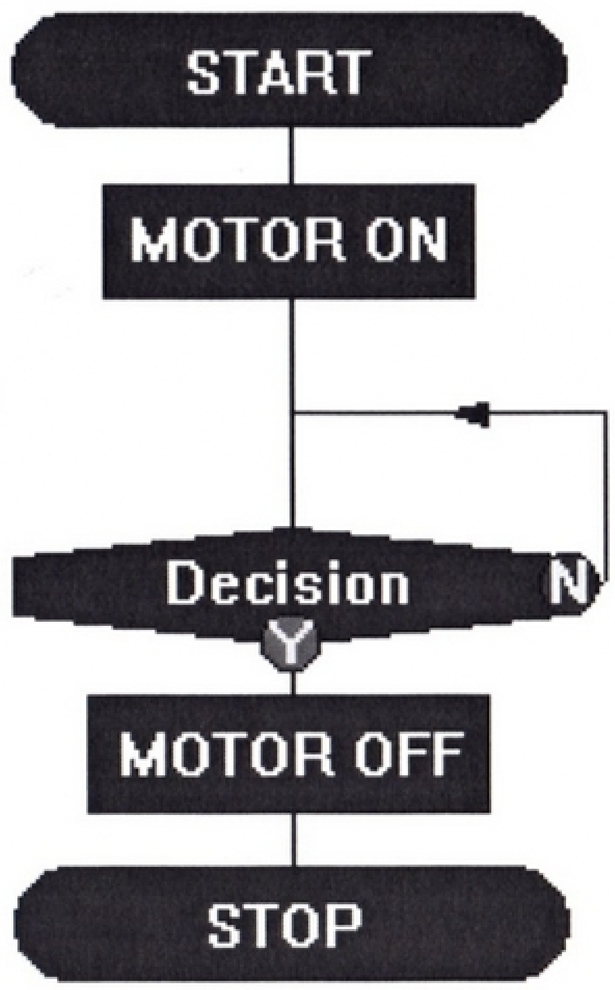
Open the MOTOR ON command that switches on the fan, and use the Test button to check which way the fan is turning. Set the command so that the fan blows air out over the lamps, and turns at a medium speed.

Task Nine

Now edit your flowsheet to look like fig. 10. This flowsheet will start the motor immediately and stop it when you press the switch.

Run the flowsheet to test it.

Fig. 10



START UP 5 : Design your own system

Now, use what you have learned about Logicator to design and test a control system for the Hand Dryer that meets the following specification:

Specification

When you press the switch on the model, the fan and heater (red and orange lamps) must switch on.

They must stay on for 5 seconds to allow the user to dry his or her hands.

The fan and lamps must then switch off.

Run your flowsheet to check that it meets the specification.

Extensions

Edit your flowsheet so that it meets one or both of the following additions to the Specification.

Addition 1 to Specification

When the fan and lamps have switched off, you must be able to switch them on again by pressing the switch on the model.

Add a green lamp to the model, and use it to indicate that the hand dryer is available for use. It must be switched on when the dryer is not on, and switched off when the dryer is on.

Addition 2 to Specification

The hand dryer must switch on when you press the switch. But instead of staying on for 5 seconds and then switching off, it must stay on until you press the switch again. It must then switch off.

Help

The software runs very quickly so the switch might be checked for the second time while your finger is still on it. You can overcome this problem by including a WAIT command (e.g. WAIT 0.5) before the Decision command that checks the switch for the second time.

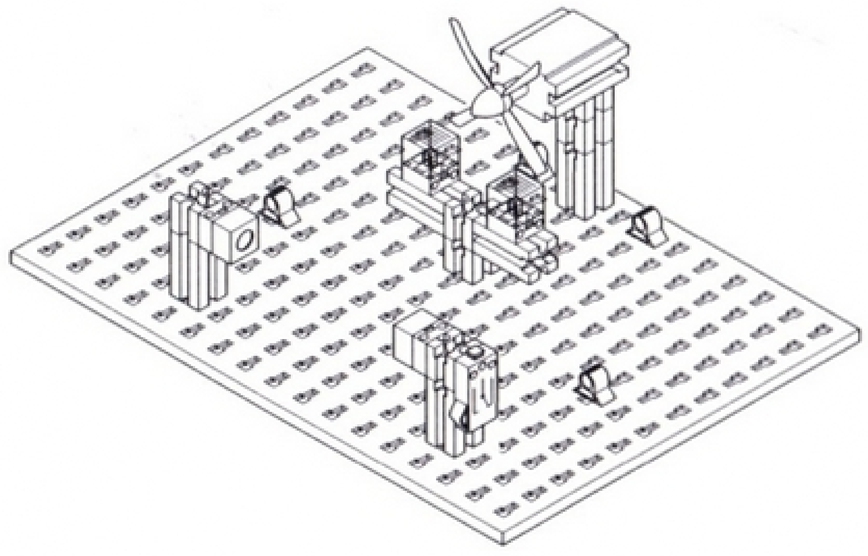
START UP 6 : Building a complex system

The control system that you have built for the dryer is already getting quite long and complicated. If it stays in the form of one long routine, it will become more and more difficult to follow as you add more features to it.

The most efficient and economical way to develop a more complex system is to build each section separately. In Logicator, these separate sections of a system are called Macros. An advantage of building the system using macros is that it keeps the main routine short, simple and easy to follow.

This sheet shows how you can use this method to develop a flowsheet to meet the following new specification:

New Specification



When you press the switch on the model, the fan and heater (red and orange lamps) must switch on.

As soon as the dryer starts, a 3-note jingle sounds and an advertising message is displayed on the screen. The message is displayed for 5 seconds, and then disappears.

The fan and lamps must then switch off.

When the fan and lamps have switched off, you must be able to switch them on again by pressing the switch on the model.

1. The flowsheet shown in fig. 11 contains the macro for operating the dryer. Notice that it is separate from the main routine that will begin with the START command.

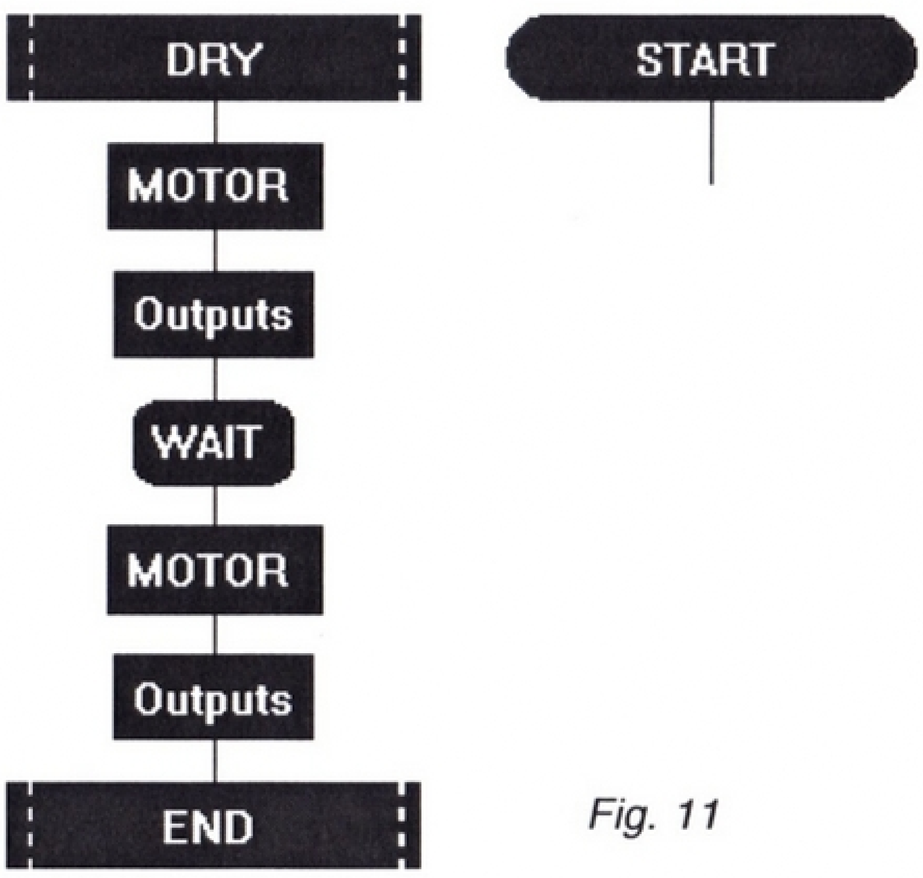


Fig. 11

Follow these instructions to build this flowsheet:

Drag a Macro command onto the flowsheet. Give it the label: DRY. The Help Sheet tells you how to label commands.

Enter the commands under the Macro command in the normal way, so that the fan and lamps are switched on for 5 seconds.

Drag an END command onto the flowsheet.

When you have finished the macro, test it by clicking on the Macro command to select it (selected commands are coloured red), and clicking the green button on the screen.

2. Fig. 12 shows the JINGLE macro for sounding the 3-note jingle. Drag a SOUND command onto the flowsheet. Open it and enter the number of the note.

Test the macro.

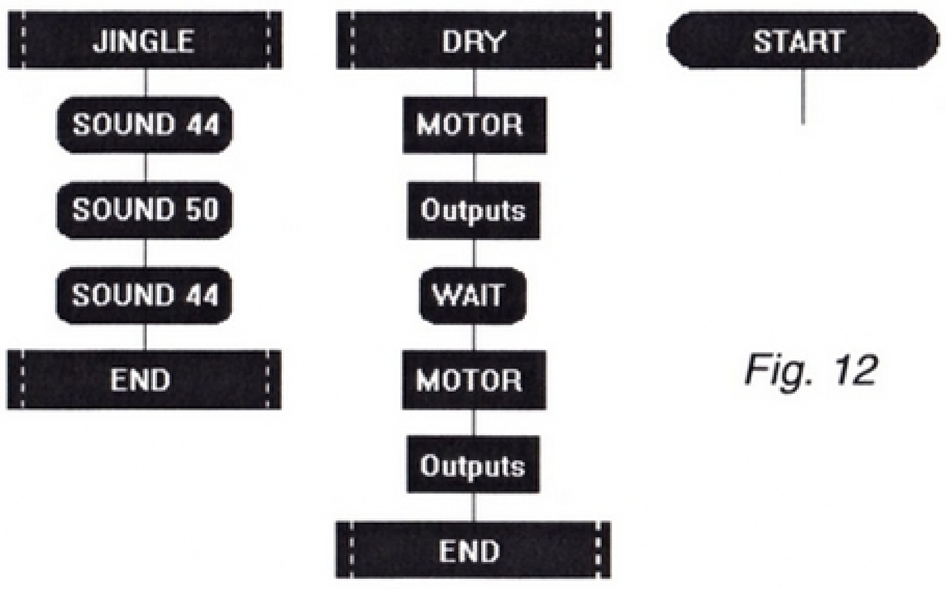


Fig. 12

3. Fig. 13 shows the ADVERT macro for displaying the advertising message. Use the information in the Help panel on this page to help you to add this macro to your flowsheet.

Test the macro.

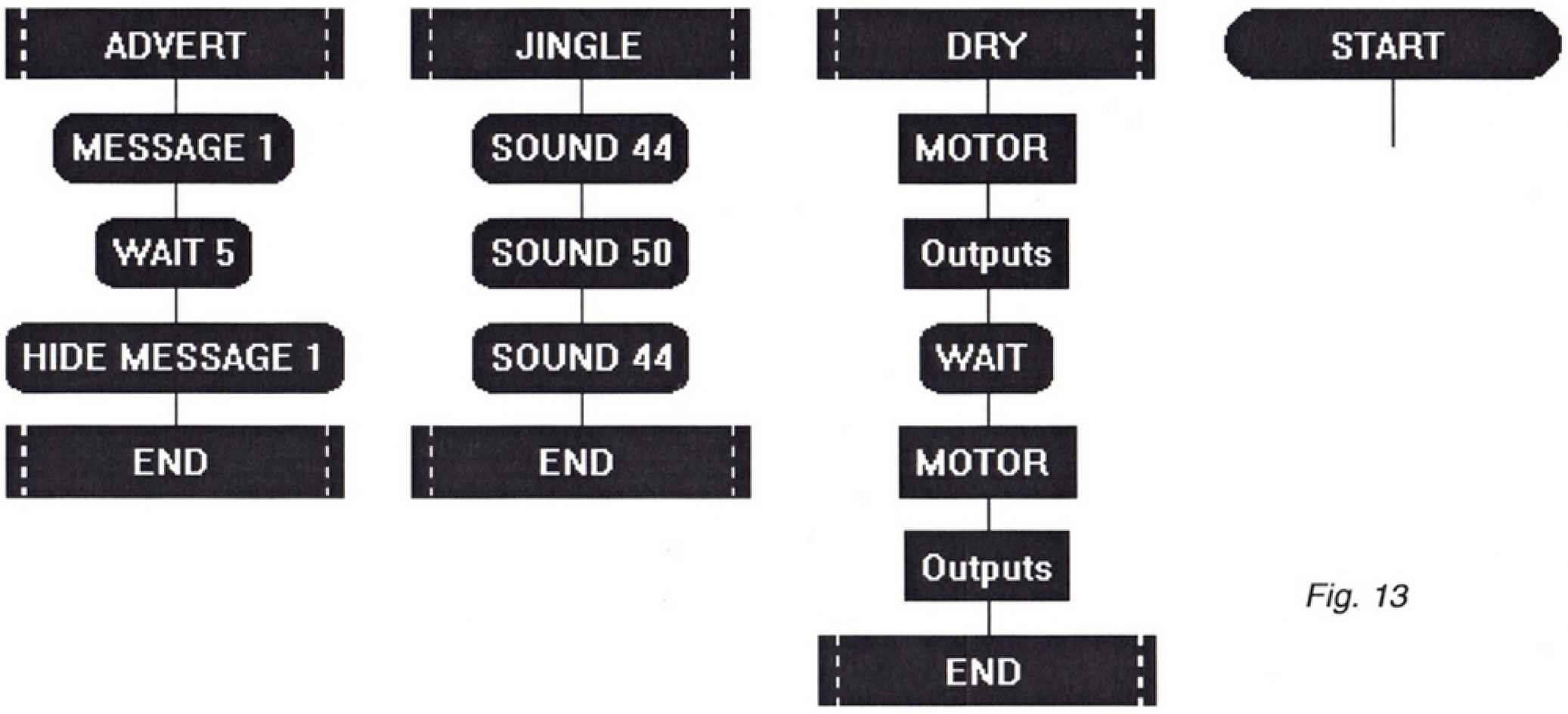


Fig. 13

Help

Logicator for Windows

Drag a MESSAGE command onto the screen. Open it. Type the text of your message. Click OK.

You use a HIDE MESSAGE command to remove the message from the screen. Drag a HIDE MESSAGE command onto the screen. Double click on it to open its Cell Details box. Type the number of the message that you want to remove. Click OK.

Acorn Logicator

Drag a Message command onto the screen. Open it and click in the Message box to get the text cursor. Type the text of your message. Press Return.

In Acorn Logicator, messages are not numbered. Your message command will appear on the flowsheet as Message On.

You use a Message Off command to remove the message from the screen. Drag a Message command onto the flowsheet. Click on the command to get the text cursor. Edit the label to "Message Off".

4. Fig. 14 shows the complete system. Follow these instructions to help you to build it:

a. The main routine (beginning with the START command) is simply made up of the Decision command that checks the switch, and a Do Macro command.

Drag the Do Macro command onto the flowsheet. Give it the label: DRY.

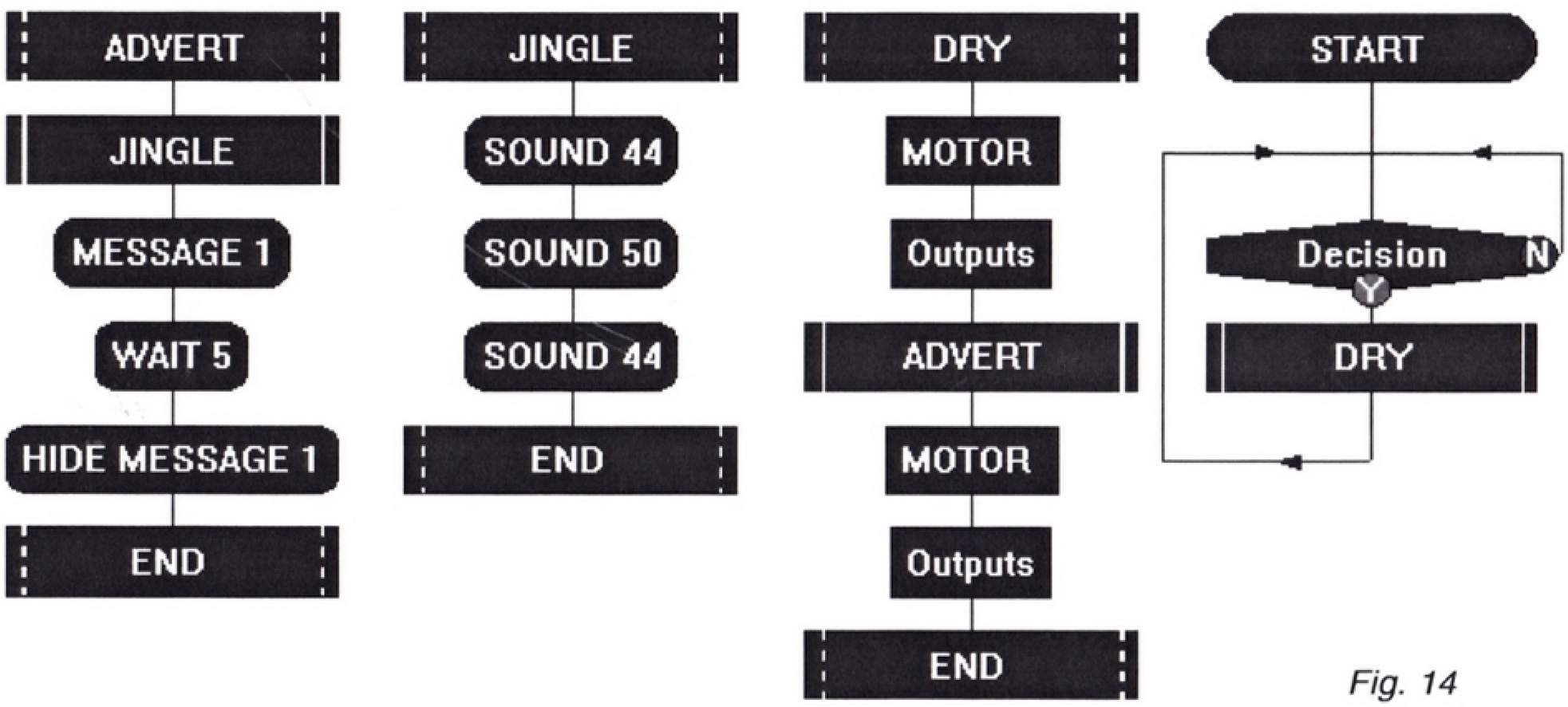


Fig. 14

b. Delete the WAIT 5 command from the DRY macro and replace it by a Do Macro labelled ADVERT as shown in fig. 14.

c. Add a Do Macro labelled JINGLE into the ADVERT macro as shown in fig. 14.

When you want to run the flowsheet, click the START command to select it before you click the green button.

Using macros is an efficient way to build control systems because it makes them much simpler to edit. Try editing your flowsheet to meet the following addition to the specification:

Addition to Specification

During the 5 seconds that the dryer is on, two different advertising messages must be displayed. The first must appear for 2 seconds, then disappear. The second must appear for 3 seconds, then disappear.

The jingle must sound each time a message appears or disappears.

START UP 7 : Responding to a sensor

The Hand Dryer model includes a light beam and digital light sensor (see fig. 15) which can be used to feed back information to the system by signalling that someone has put his or her hands under the dryer. This makes the system into an automatic hand dryer.

When light from the lamp shines on the digital light sensor, the sensor is on. When somebody blocks the light, the sensor switches off.

The macro shown in fig. 16 switches on the light beam and then checks the signal from the light sensor. Use the information below to help you to build it

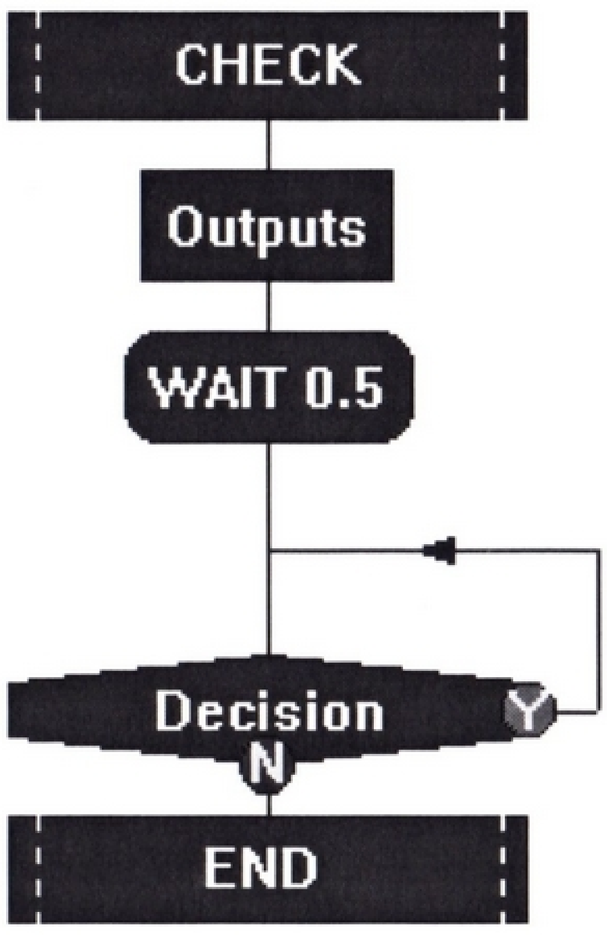


Fig. 16

1. Set the Outputs command to switch on the lamp that provides the light beam. Click and hold down the Test button, and look at the Smart Box. Two indicator lamps should be on - the green indicator light by the socket connected to the lamp, and the yellow indicator light by the socket connected to the light sensor.

Put your hand between the lamp and the sensor. The yellow indicator light should switch off. If the sensor is not responding, try swapping over the two plugs that connect it to the Smart Box.

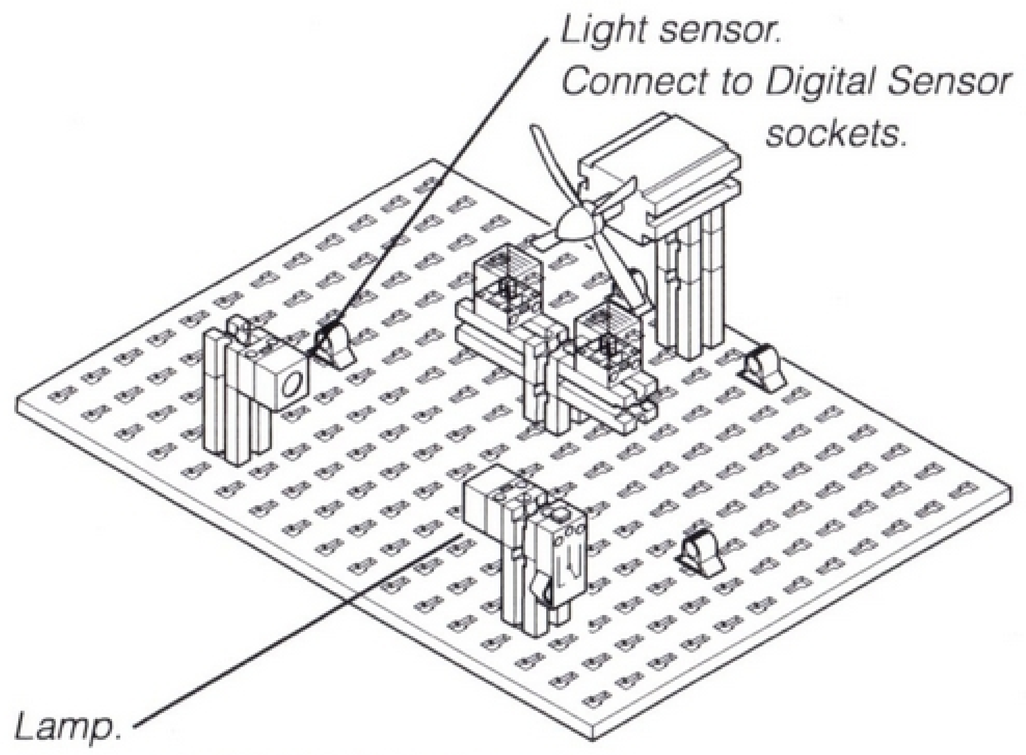


Fig. 15

2. The WAIT command allows time for the lamp to switch on fully before the sensor is checked.

3. Set the Decision command to check if the sensor is on.

Important - Make sure that the Y and N routes are drawn correctly as shown in fig. 16. The Help Sheet tells you how to change them over if you need to.

Run the macro to test it. The macro should end when you break the light beam.

Now edit your Hand Dryer control system so that it meets the following specification:

Specification

When you break the light beam on the model, the fan and heater (red and orange lamps) must switch on.

As soon as the dryer starts, a 3-note jingle sounds and an advertising message is displayed on the screen. The message is displayed for 5 seconds, and then disappears. The fan and lamps must then switch off.

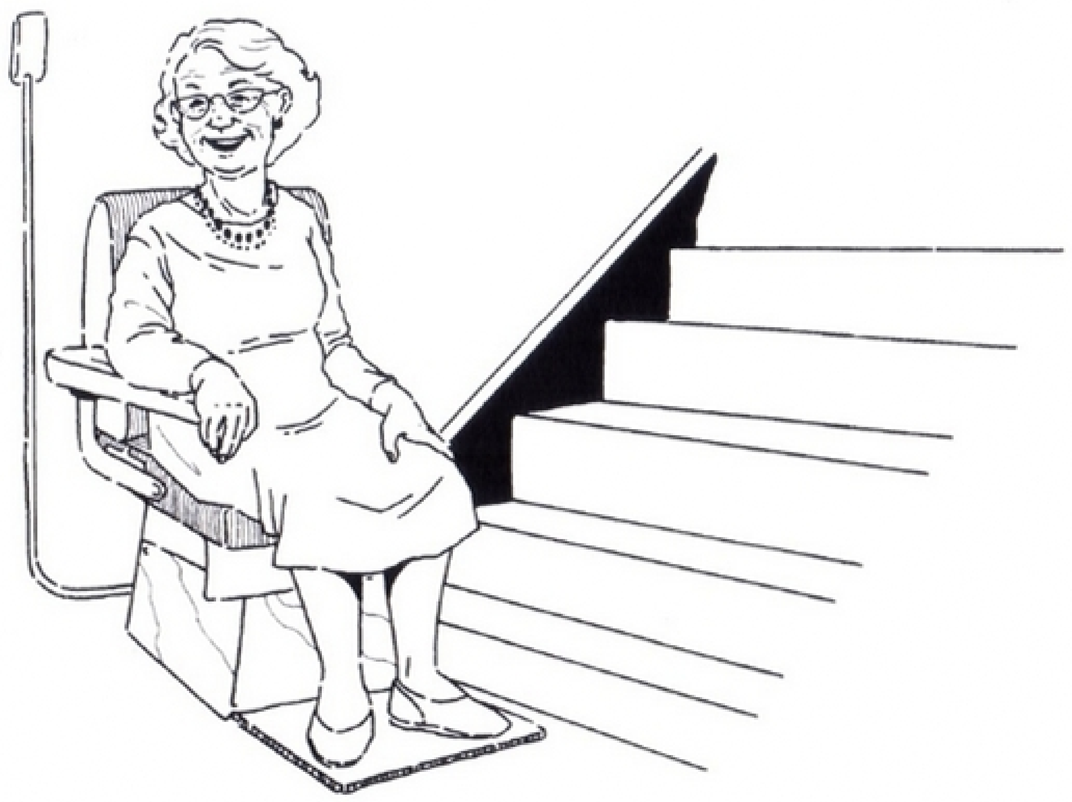
When the fan and lamps have switched off, you must be able to switch them on again by breaking the light beam on the model.

CONTROLLING MOVEMENT 1 : Stairlift

A motorised stairlift makes life much easier for people who have difficulty walking up and down stairs.

The output of the stairlift system is the movement of the chair. The input is provided by the push buttons fixed on the arm of the chair. The passenger uses these buttons to select whether the chair moves up or down the staircase.

Fig 1 shows how the control system for the stairlift could be sketched out as a flow diagram.



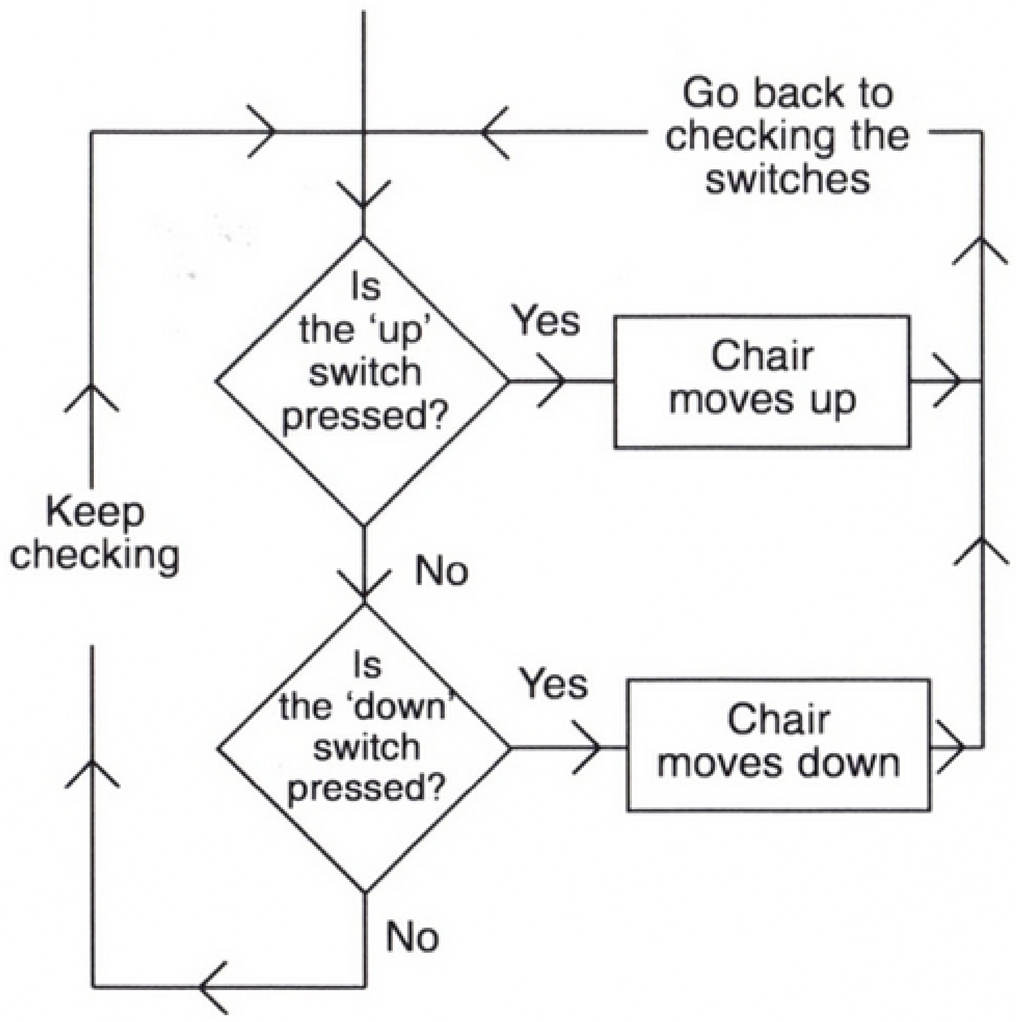
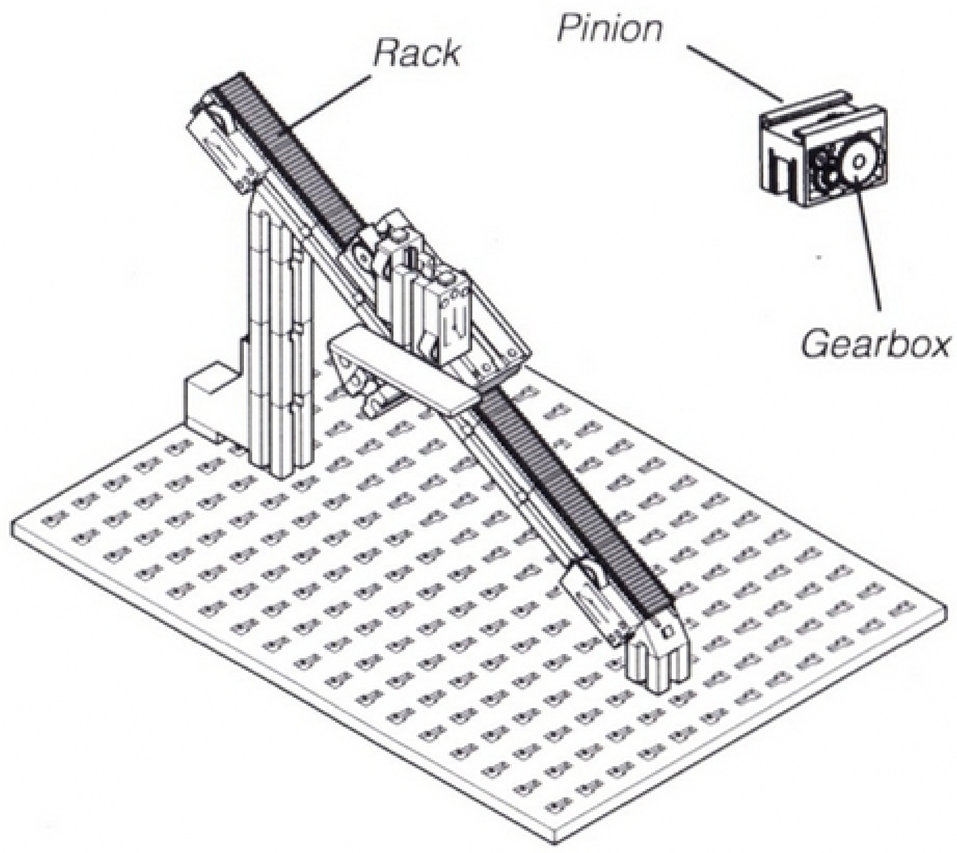


Fig 1

Build the model of the stairlift shown on the Rack and Pinion 1 construction sheet. Connect the motor and switches to the Smart Box as shown on the sheet.

Rack and Pinion 1

The mechanism of the stairlift model is made up of two mechanical systems : the gearbox and the rack and pinion.



Rack and Pinion 1

The input of the gearbox system is the fast rotary movement of the worm drive on the motor. The output is the much slower rotary movement of the gear that meshes with the rack. This gear is called a pinion.

The input of the rack and pinion system is the rotary movement of the pinion. The output is the linear movement of the motor and gearbox up and down the rack.

Limit Switches

Switches 1 and 2 on the model are the limit switches (see fig 2). They feedback information to the system about the position of the chair. For example, when the gearbox presses switch 2, the chair is at the top of the staircase.

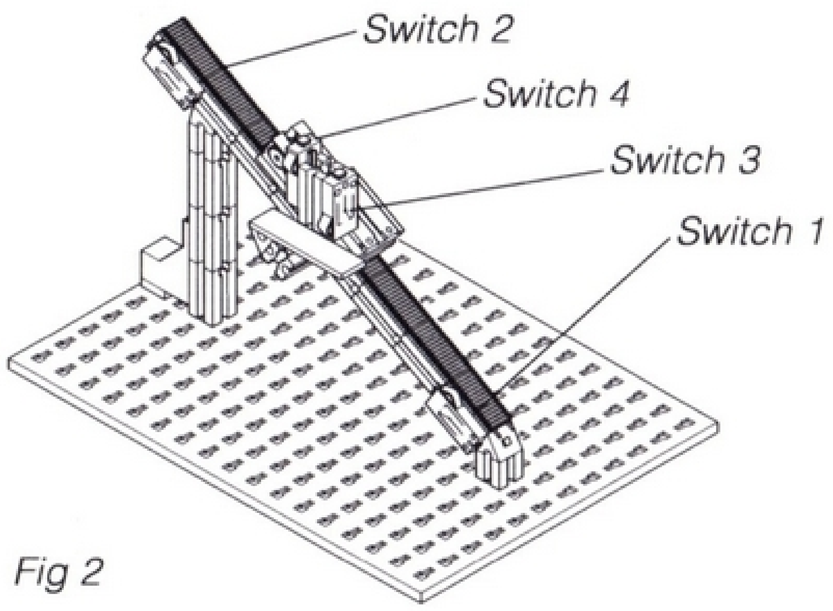


Fig 2

So the way to control the movement of the stairlift is to switch on the motor and use a Decision command to check if the limit switch is pressed. When it is pressed, then switch off the motor.

Build the macro shown in fig 3 and test it to check that it moves the chair to the top of the staircase and then stops.

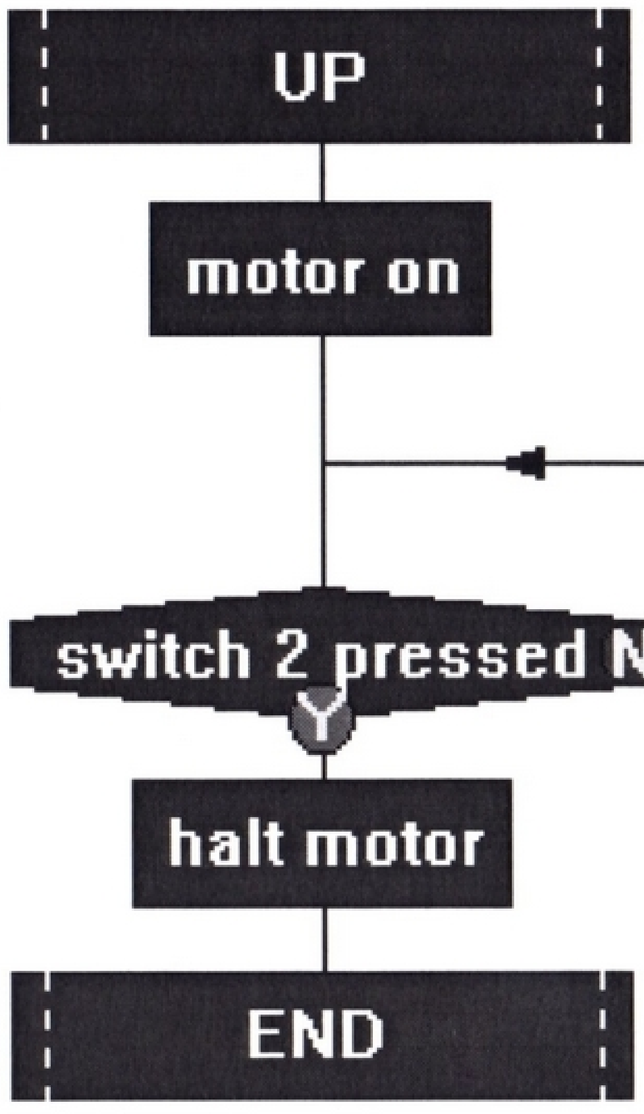
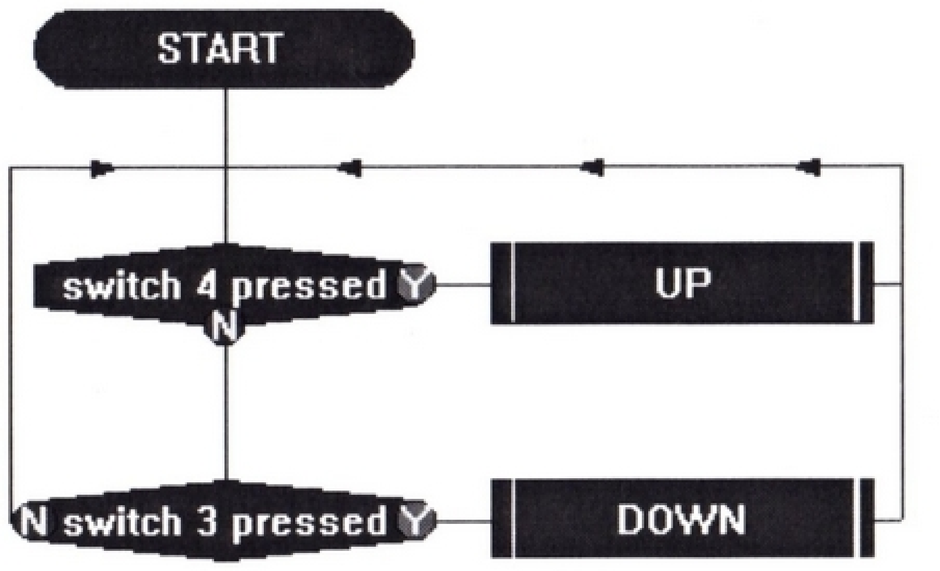


Fig 3.

The Complete System

The two switches fixed to the chair (switches 3 and 4) can be used to select which direction the lift moves in. The complete system needs to check these two switches and carry out the appropriate UP or DOWN macro if one of them is pressed. Fig 4 shows how this can be done.



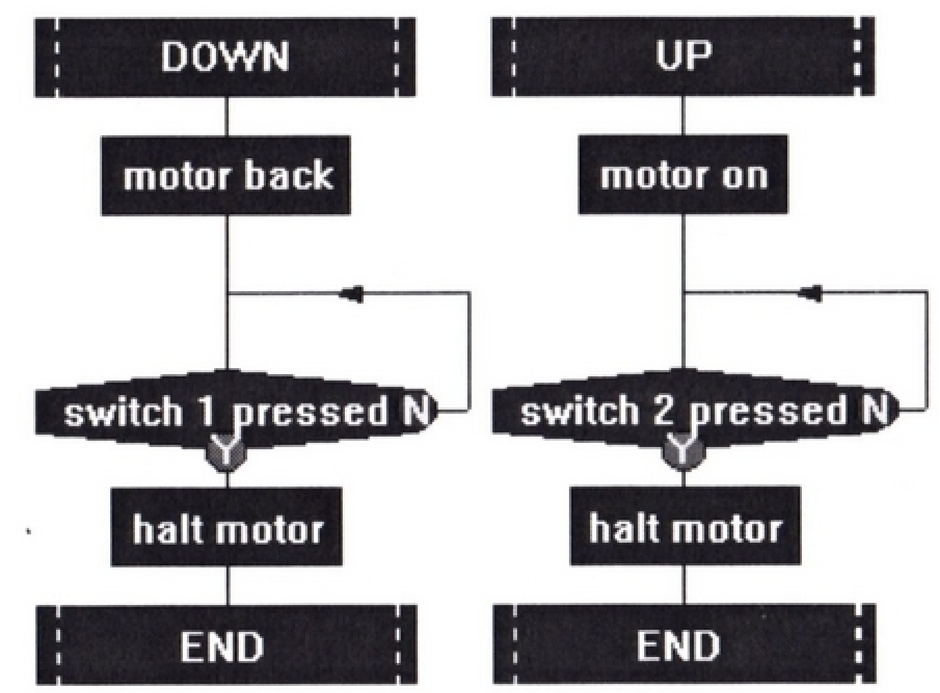


Fig 4.

Notice that repeat loops are used so that:

i) the switches are checked all the time when the chair isn't moving
ii) the system goes back to checking the switches as soon as the chair has reached its destination.

Extending the System

Add a red light and a green light to the model. Develop the control system so that the red light shines while the chair is moving. When the chair stops, the red light should switch off and the green light should switch on to show that it is safe to get off.

You could also use the SOUND command to provide a signal that the chair has stopped moving.

Using the Event Command

The stairlift system that you have built so far is not very convenient for forgetful people. If they realise they have forgotten something when the chair is halfway up or down, they have to complete the journey and then go back.

The system needs to be redesigned so that it meets the following specification:

Specification

At any time in the upward journey, the passenger can press the 'down' switch and go back to the bottom. At any time in the downward journey, they can press the 'up' switch and go back to the top.

You can use Logicator's Event command for this system. An Event command instantly captures the flow of control whenever a pre-set digital input condition occurs to trigger it.

Use the information in the Help panel to edit your flowsheet to look like fig 5. Run the flowsheet to check that it meets the specification.

Help

1:Event

Drag an Event command onto the flowsheet. Double click on it to open its Cell Details box. It has the same Input Port line as a Decision command. Set it to respond when the appropriate switch is pressed. Click OK. The command will automatically label itself 1: Event.

An Event will be triggered only if it has been enabled by an EVENT command.

EVENT

Drag an EVENT command onto the flowsheet. Double click on it to open its Cell Details box.

Select ON from the drop-down list called "State". Click OK.

When an Event is triggered, it must be re-enabled before it can be triggered again. So, flow is routed back through the EVENT ON command to re-enable the Event.

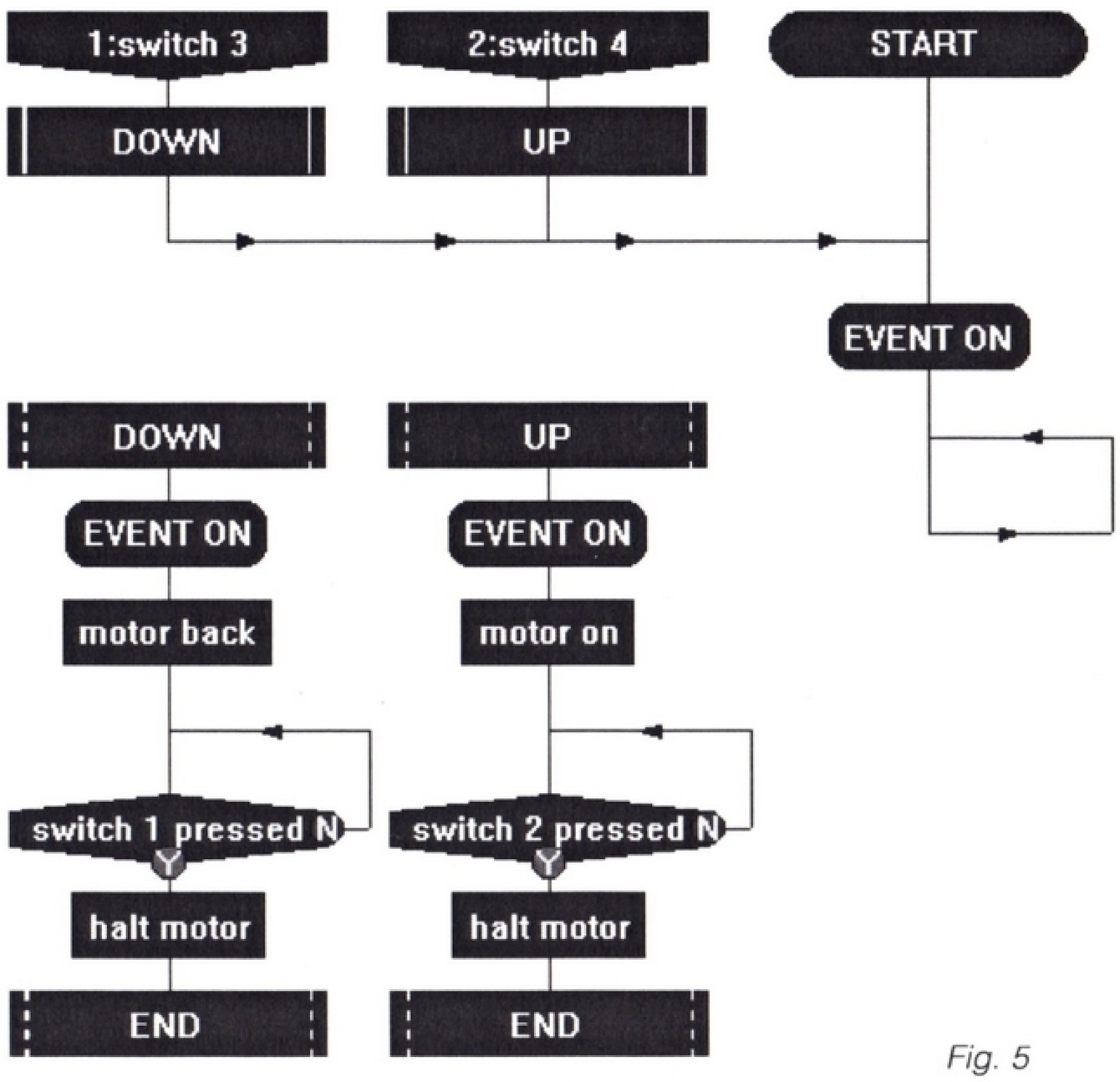


Fig. 5

CONTROLLING MOVEMENT 2 : Packing System

In the packing department of a sweet factory, boxes are sealed and printed with a `best by' date on an indexing table. The table moves round step by step to bring each open box under the sealing machine in turn.

Fig 1 shows the control system required. The most efficient way to build the system is to take each section separately.

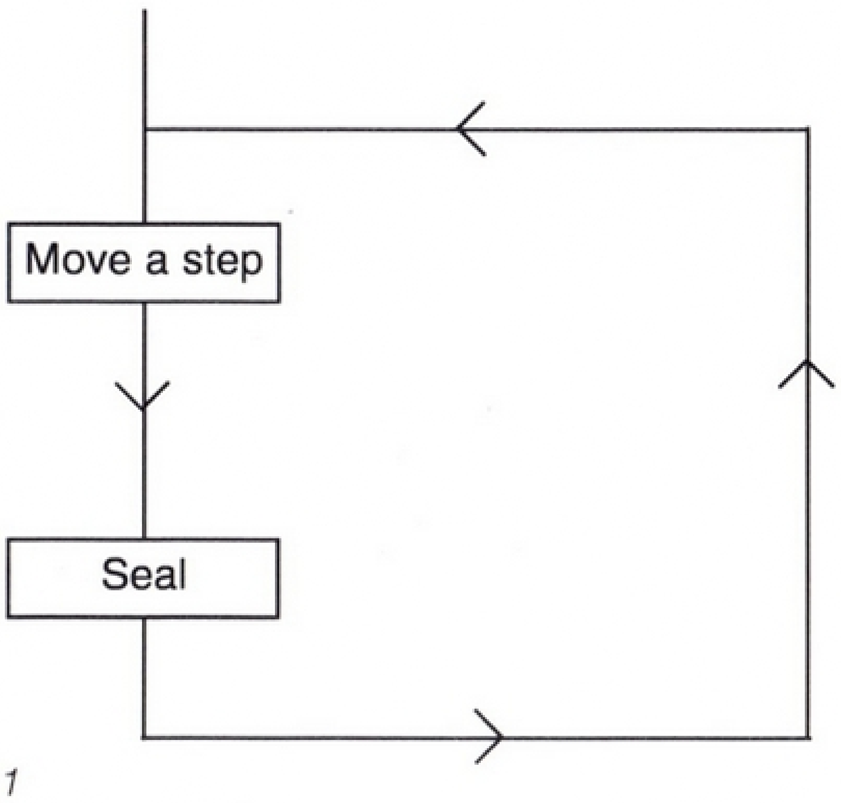
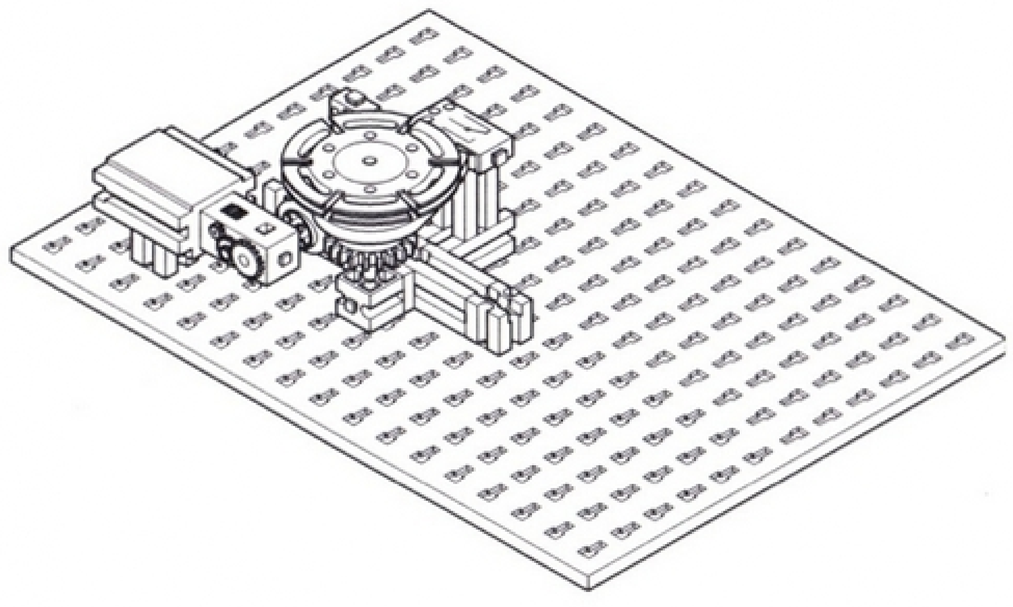


Fig. 1

1. Move a Step

Build the model of the turntable shown on the Worm and Wheel construction sheet. Connect it to the Smart Box as shown on the sheet.

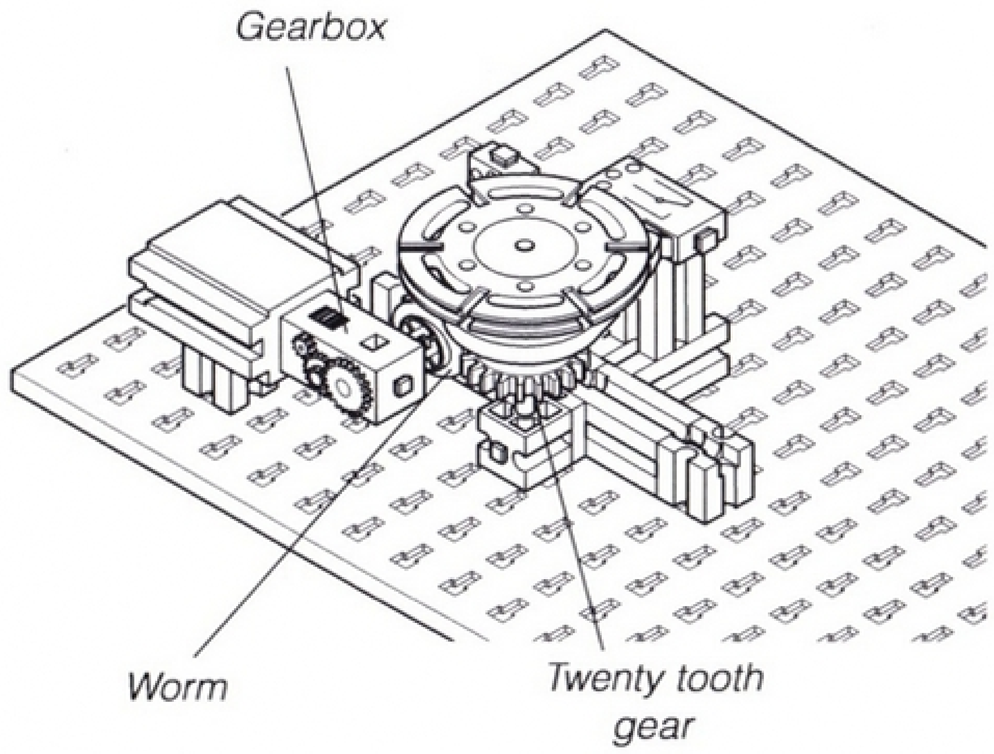


Worm and Wheel

Worm and Wheel

The mechanism of the turntable combines:

i) A gearbox which reduces the speed of movement and increases the torque provided by the motor,



ii) A worm and wheel. The input of this system is the rotary movement from the gearbox. The output is the rotary movement of the turntable. The worm and wheel (the process) changes the direction of the movement through 90°. The worm is really a one-tooth gear so when it is used with a twenty-tooth gear as on this model, speed is reduced by a ratio of 1:20. There is a corresponding increase in torque.

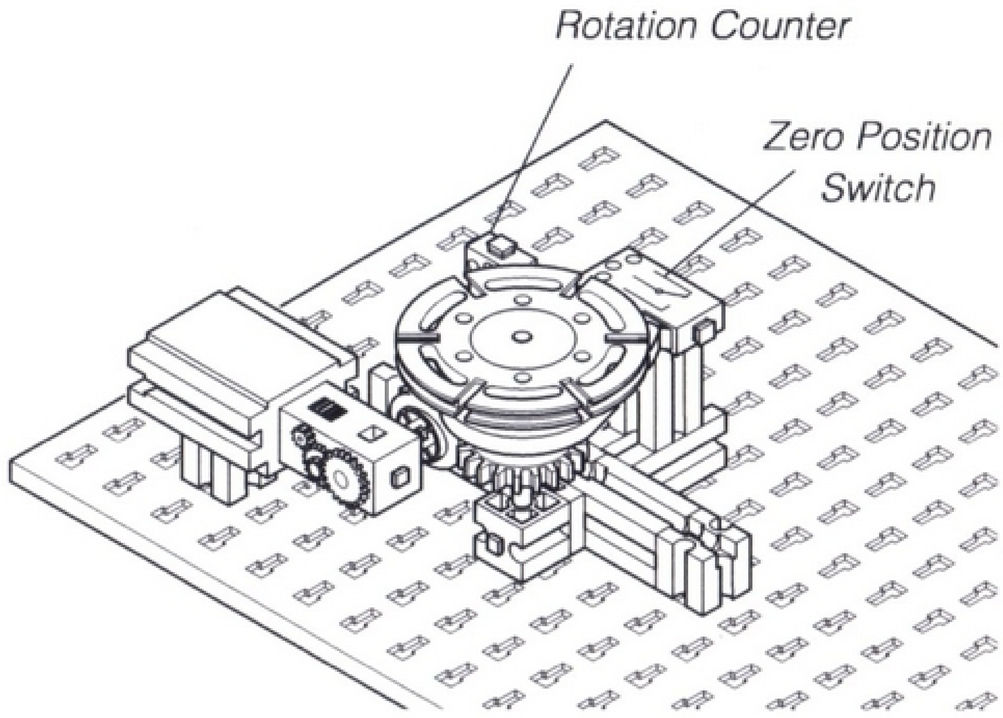
Control System for the Turntable

The turntable has six slots in it. Each box is held in one of these slots. Therefore each step in the system must be the distance between two slots.

The system needs information about the position and movement of the turntable. This information is provided by a rotation counter and a zero position switch.

The rotation counter is made up of a switch and a pulsing wheel. The pulsing wheel is connected to the motor, so that as the motor rotates, the wheel pulses the switch. This sensor provides information about how far the turntable has moved.

A COUNT command is used to count the number of pulses on the switch, so that the motor can be stopped when a set number have been counted.



The zero position switch is used to provide information about the position of the turntable. When the cam presses this switch the system can tell that the turntable is in its start position.

Build the flowsheet shown in fig 2. The ZERO macro sends the turntable to its start position. Use the information in the Help panel to build the STEP macro.

Help

Logicator for Windows

Drag a COUNT command onto the flowsheet and double click on it to open its Cell Details box.

Type 17 into the text box.

The COUNT command has the same Input Port line as a Decision command. The counter switch is connected to input 0, so set its digit to 1. Click OK.

Acorn Logicator

Drag a COUNT command onto the flowsheet and click on it to open it.

The COUNT command has the same Input Port line as a Decision command. The counter switch is connected to input 0, so click twice on its button in the Inputs box to set it to a yellow square.

Type 17 into the “Count - times” box. Press Return.

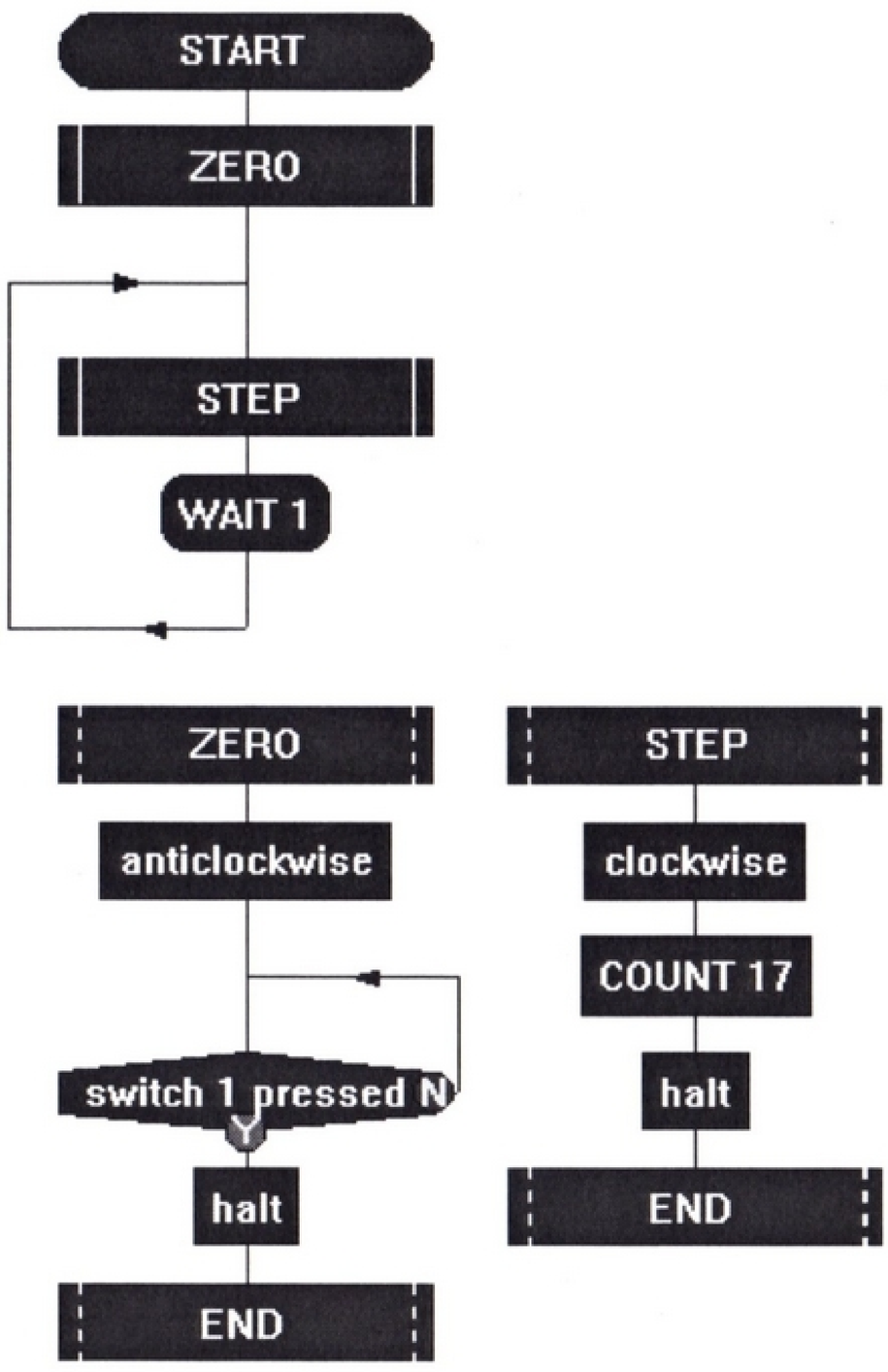


Fig 2

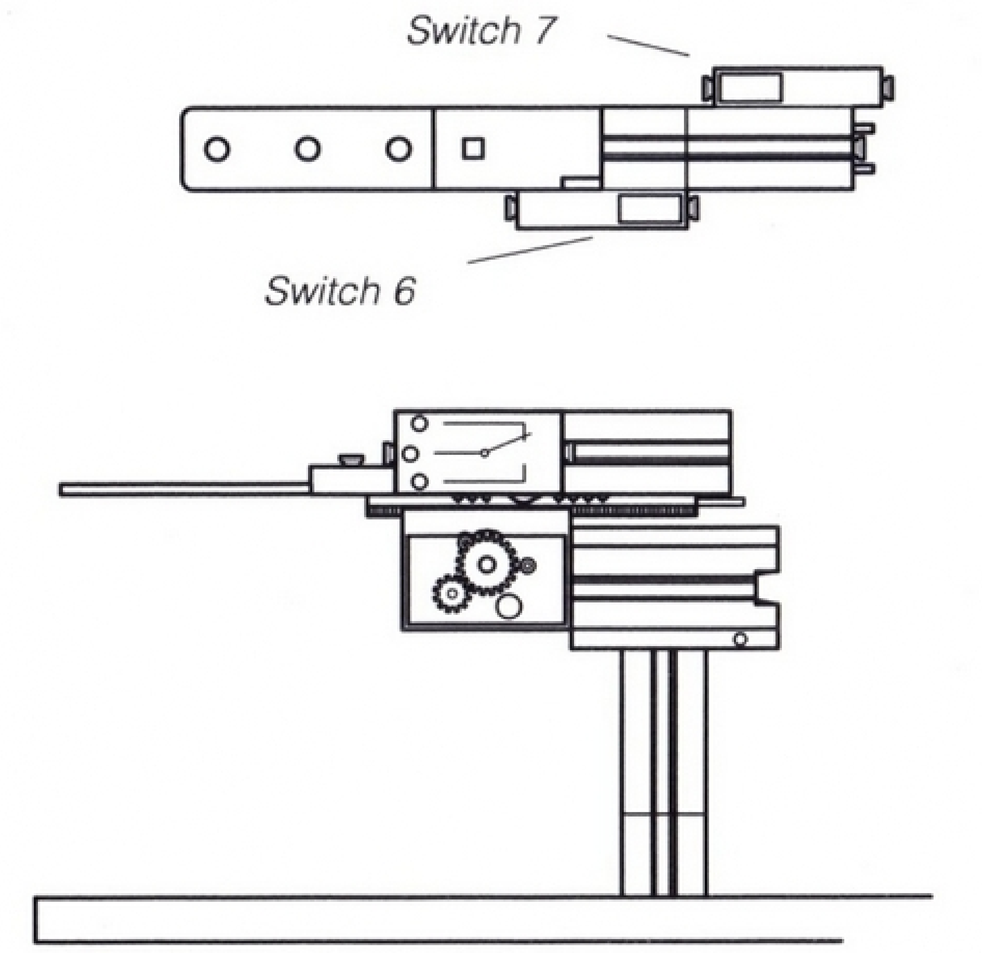
2. Seal

Build the model shown on the Rack and Pinion 2 construction sheet. Fix it on to the baseplate by the turntable and connect it to the Smart Box as shown on the sheet.

Arrange the model so the slider will move out and back over a slot in the turntable to simulate the sealing machine sealing a box held in the slot.

Although it uses the same mechanism as the stairlift (Rack and Pinion 1), the Rack and Pinion 2 model is different because the motor and gearbox are fixed and the rack moves. Limit switches are also used differently.

The motor moves the rack forward until switch 6 is no longer pressed. It moves it backward until switch 7 is no longer pressed. You can increase or decrease the distance the slider moves by moving the switches. If you need to move the rack by hand, ease the gearbox up to disengage it from the worm drive on the motor.



Rack and Pinion 2

Add the SEAL macro shown in fig 3 to your flowsheet. Run the macro to test that this part of the system works correctly. Then click on the START command and then on Run! to run the complete packing machine system.

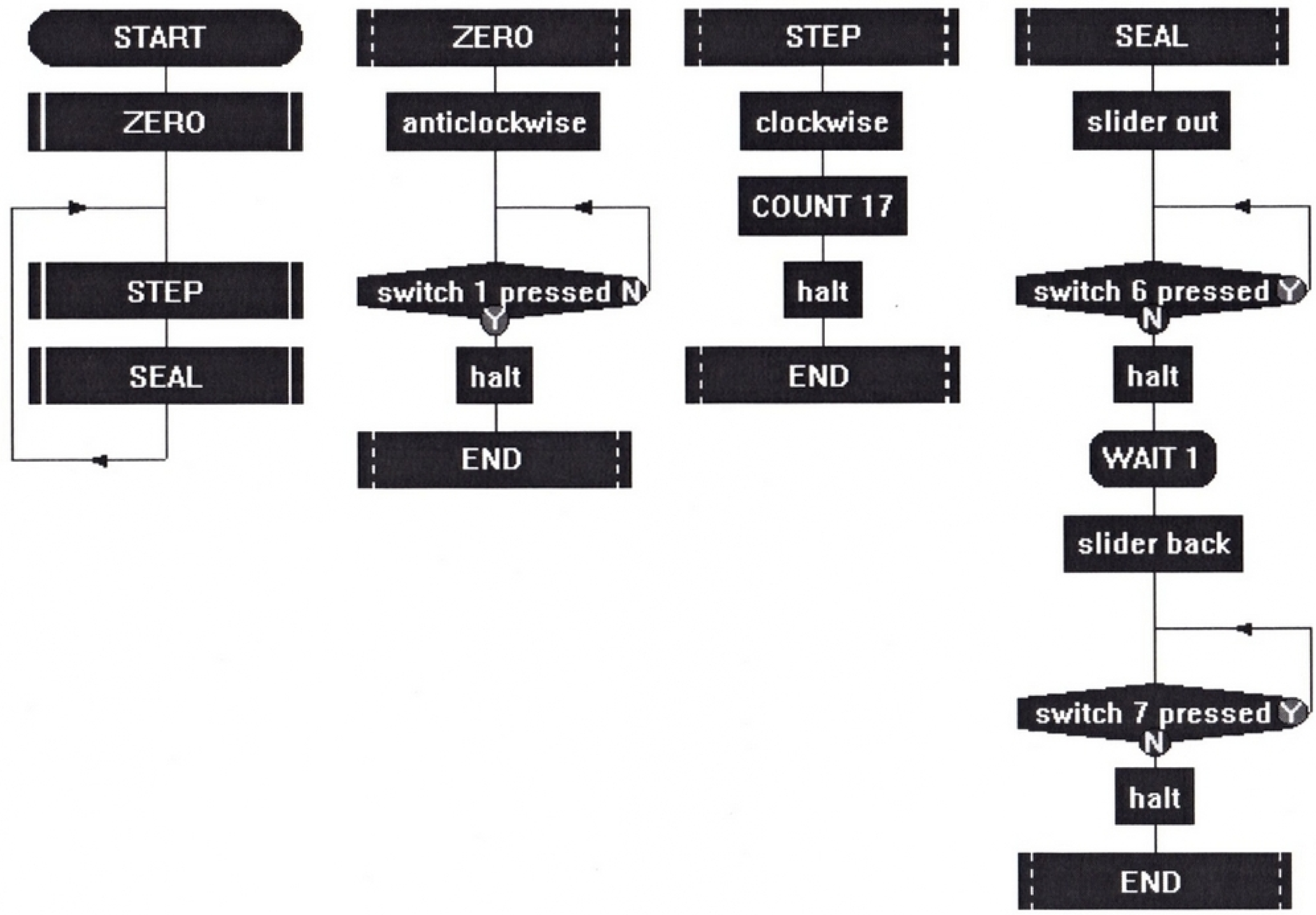


Fig 3. Complete packing system. Make sure that you set the Y and N directions correctly on the Decision commands in the SEAL macro.

Extending the System

1. Add red and green lamps to the model. Develop the system to use them as indicators. The red light shows that the machine is switched on so operators must stay clear of moving parts. The green light shows that the machine is switched off.

2. Develop the system so it keeps a count of how many boxes have been sealed, and displays this information on the screen.

Fig 4 shows the macro for signalling and displaying this count. The command "INC A" acts as a counter. Each time flow passes through it, one is added to the value of A.

Use the information in the Help panel to help you to build this macro. When you have built and tested it, add it to your flowsheet. Remember also to add a Do Macro command called RECORD into the main routine.

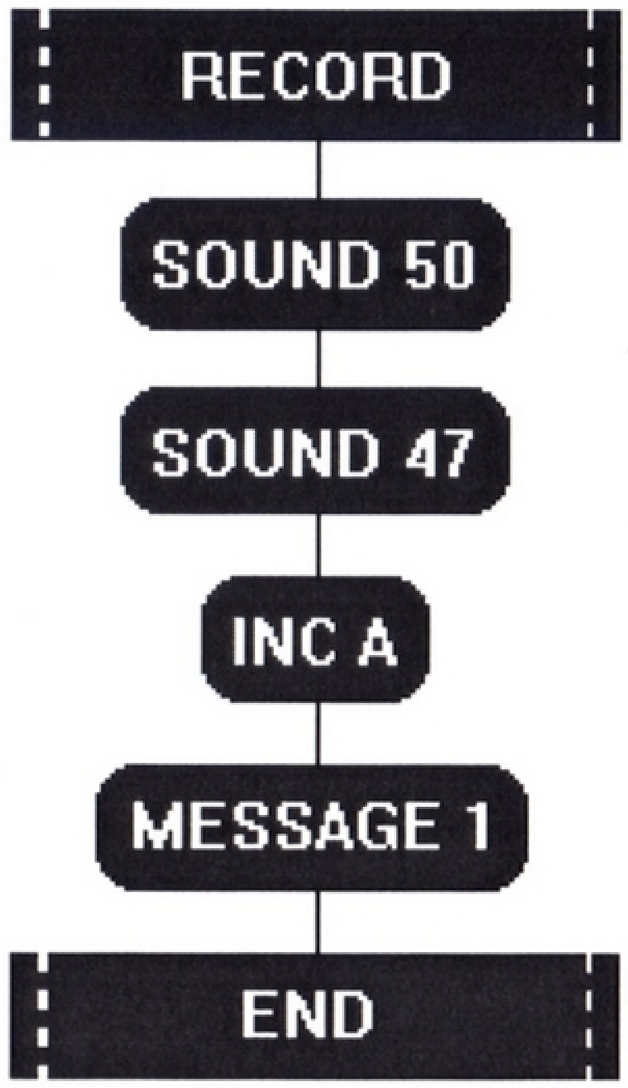


Fig. 4

Help

Drag a SOUND command onto the flowsheet. Open it and enter the number of the note.

Logicator for Windows
Drag an INC command onto the flowsheet, and double click on it to open its Cell Details box. Select "A" from the drop-down list. Click OK. This command acts as a counter. Each time flow passes through it, one is added to the value of A.

Drag a MESSAGE command onto the screen. Double click on it to open its Cell Details box. Type the following text:

Process total = [A]

Putting A into square brackets means that as its value changes, the message will be updated. Click OK.

Acorn Logicator
Drag an INC command onto the flowsheet. Click on it to get the text cursor. Type: a. Press Return. This command acts as a counter. Each time flow passes through it, one is added to the value of A.

Drag a Message command onto the screen. Open it and click in the Message box to get the text cursor. Type the following text:

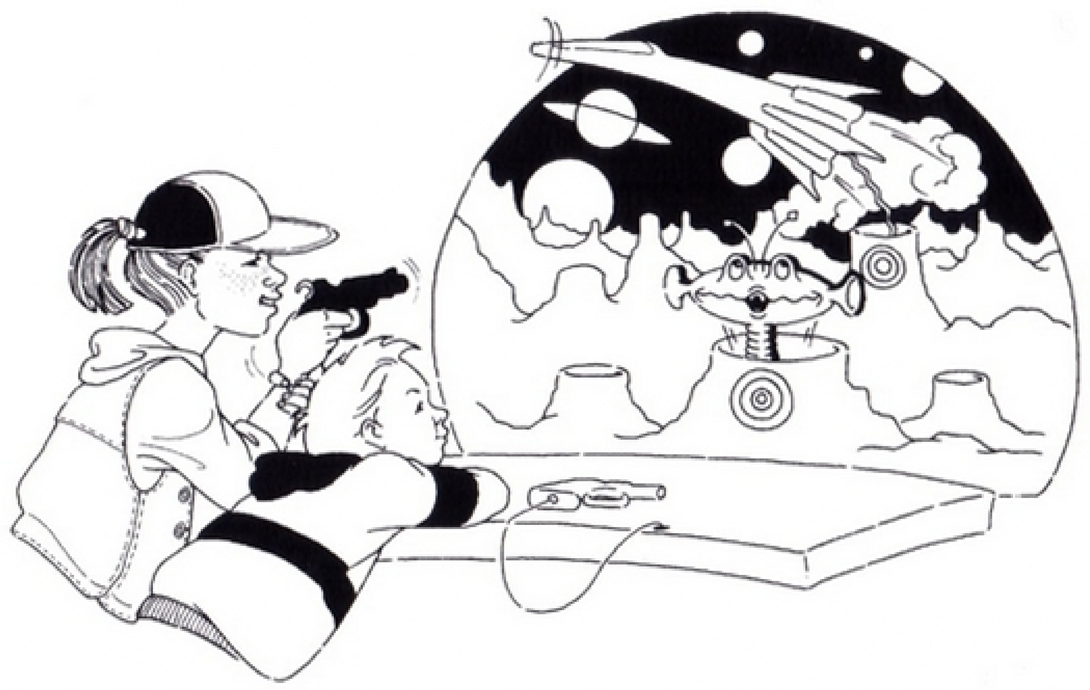
Process total =
<a>

Putting A into this kind of brackets means that as its value changes, the message will be updated. Press Return.
In Acorn Logicator, messages are not numbered. Your message command will appear on the flowsheet as Message ON.

CONTROLLING MOVEMENT 3 : Shooting Gallery

The novelty shooting gallery shown here has several targets. Each one contains a light sensor. The gun fires a light beam when the trigger is pressed. If the gun is aimed straight at a target, it switches the sensor on and a figure or object pops out from behind a part of the scenery.

Fig 1 shows a flow diagram sketch of the control system needed for the shooting gallery.



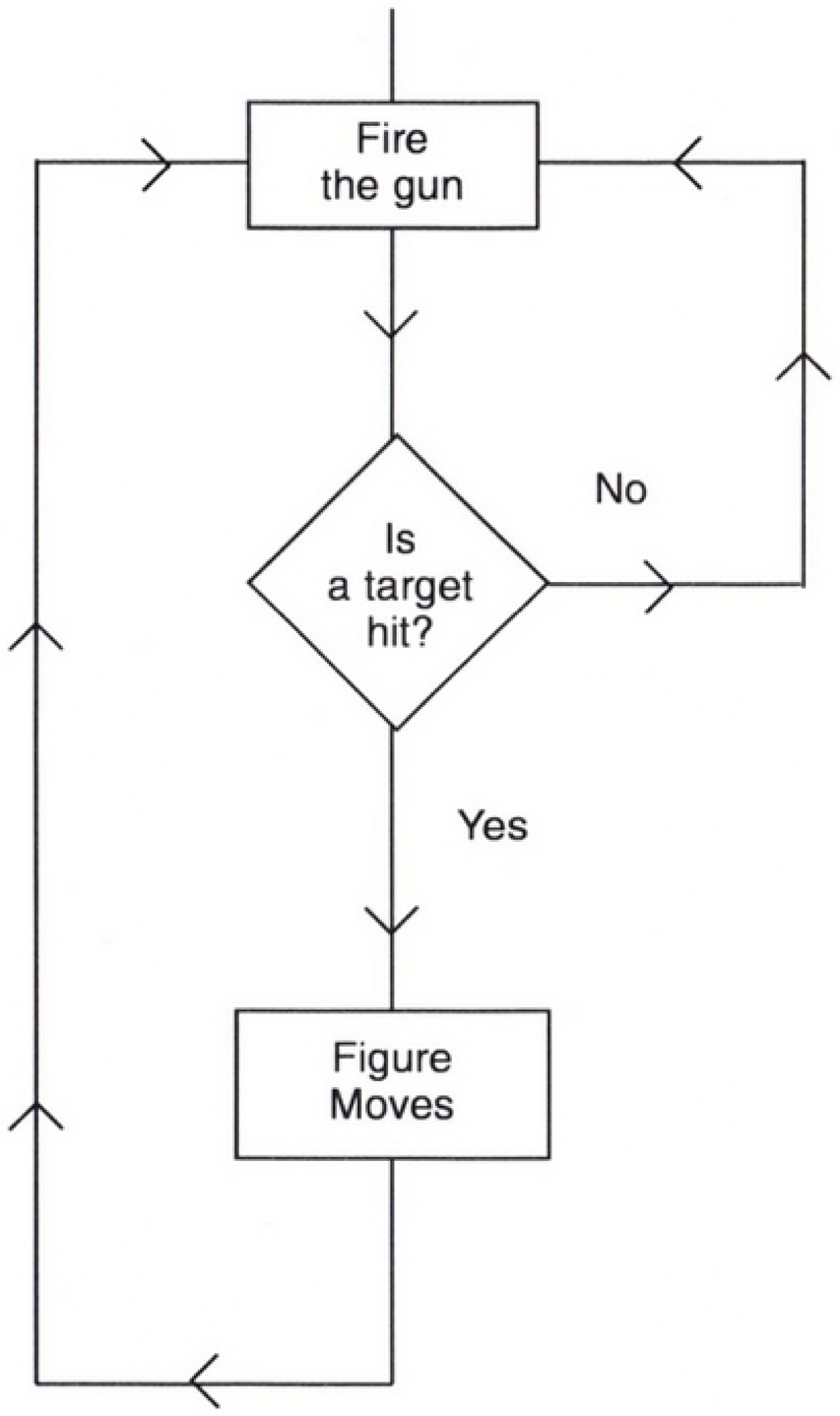
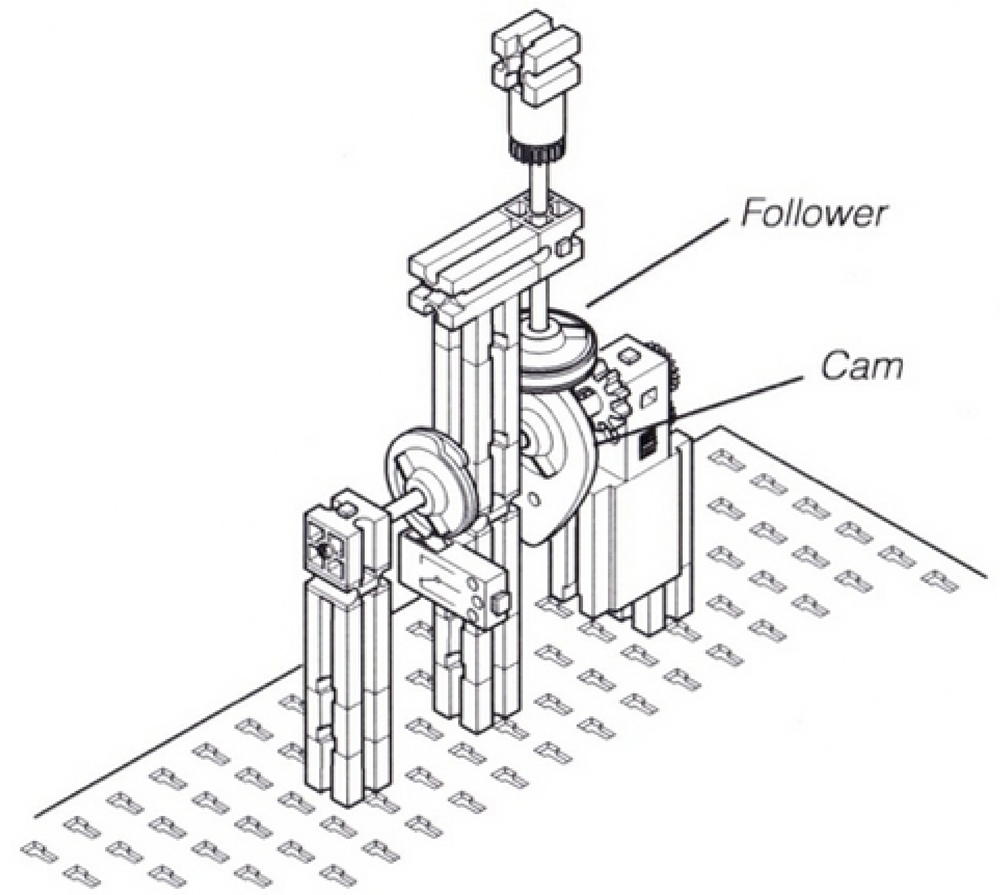


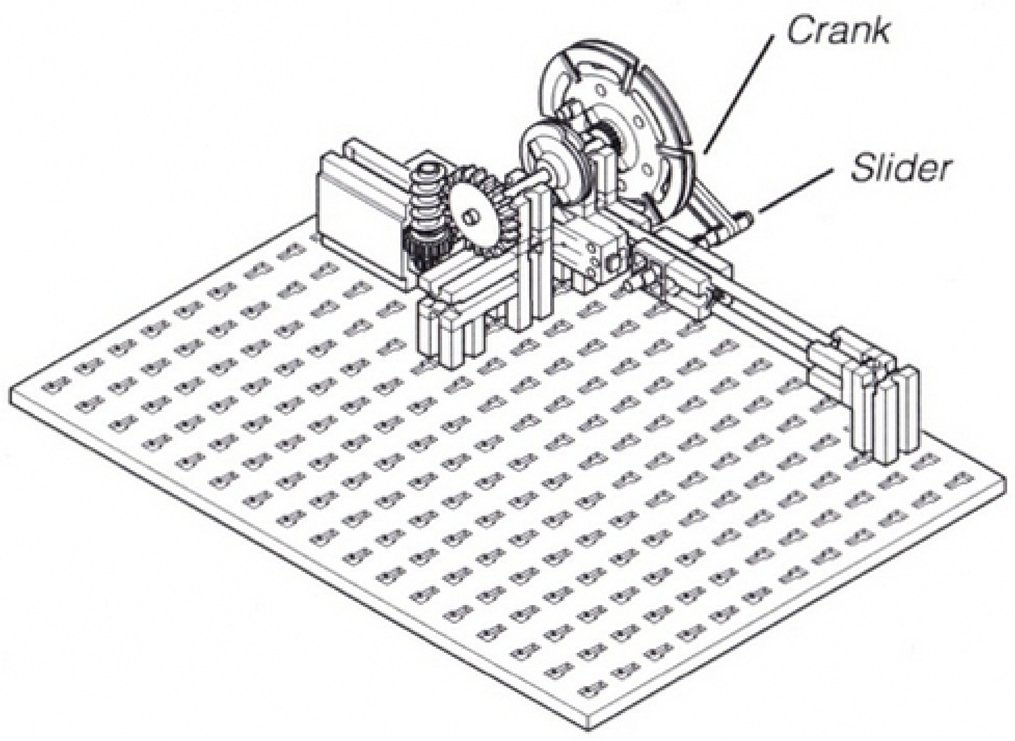
Fig. 1



Cam and Follower

Begin work on the system by looking at the output – the moving figures. They have to pop up or out when their target is hit and then return to their hiding place. The mechanisms shown on the Cam and Follower and Crank and Slider construction sheets will convert the rotary movement of an electric motor into the necessary reciprocating movement.

Build these two models and connect them to the Smart Box as shown on the sheet. Add red and yellow plates from the kit to simulate the figures and their hiding places.



Crank and Slider

Controlling Reciprocating Movement

Both the cam and follower and the crank and slider mechanisms are controlled by the same technique : a switch pressed by an index cam, as shown in fig 2.

Begin by controlling the cam and follower. Use the following instructions.

1) Make sure that the index cam on your model is set as shown in fig 2.

2) Open a MOTOR command and check which way the index cam turns for each setting of the motor (clockwise or anticlockwise?).

3) Build the macro shown in Fig 3.

When the motor is switched on it turns until the cam stops pressing the switch, so set the "switch unpressed?" Decision command to check that the switch is not pressed (0).

The motor keeps turning until the cam presses the switch again, so set the "switch pressed?" Decision command to check that the switch is pressed (1).

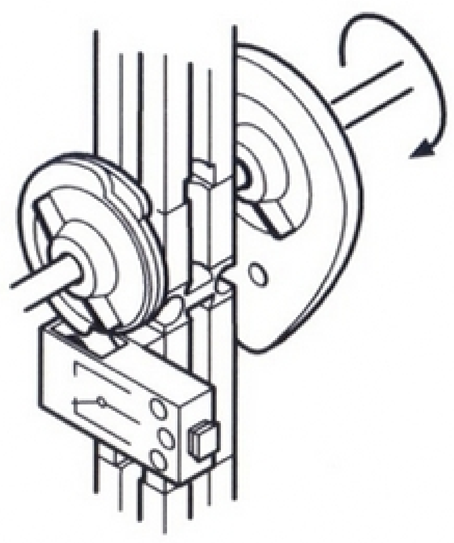
This will make the figure move out and straight back.

NOTE: When you test the macro, use Fast Run (selected from the Options/Speed menu) as it needs fast response to the switch. You may need to experiment with the system to get it working correctly. You could use a short wait (e.g. WAIT 0.5) instead of the "switch unpressed?" Decision command.

Now build and test a similar macro to control the crank and slider.

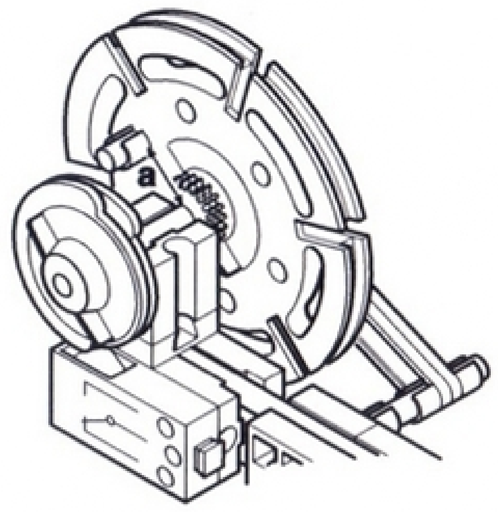
These two macros (see fig 4) provide the output part of the shooting gallery system.

Fig. 2



Cam and Follower

The switch is just pressed when the follower is in its down position.



Crank and Slider

The switch is just pressed when the slider is in the back position.

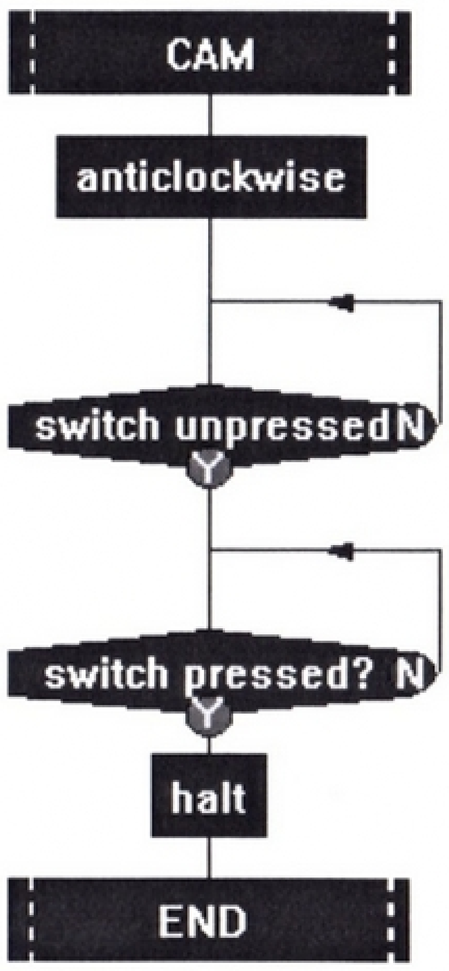


Fig. 3

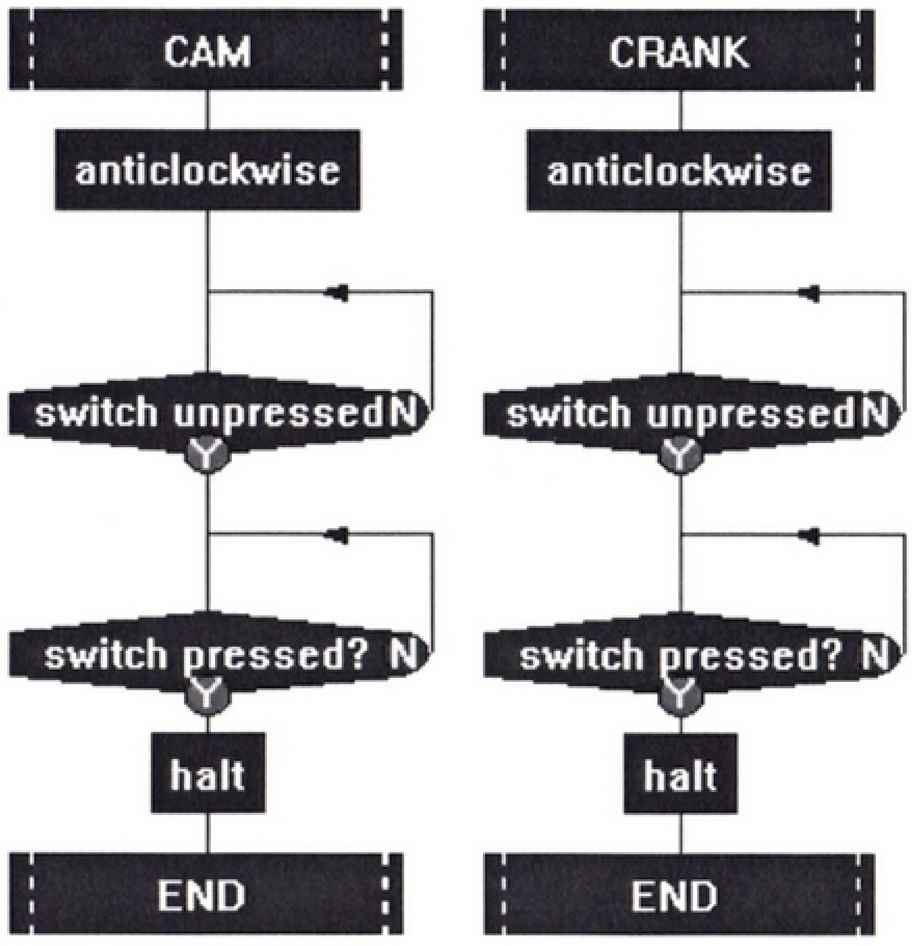


Fig. 4

The Input of the System

a) Targets

Use two light sensors with black masking caps attached as the targets in the shooting gallery. Add one to the baseplate of each model and connect them to Digital Sensors sockets 6 and 7 on the Smart Box as shown in fig 5. You could support them on angle blocks to make them easier or harder to hit.

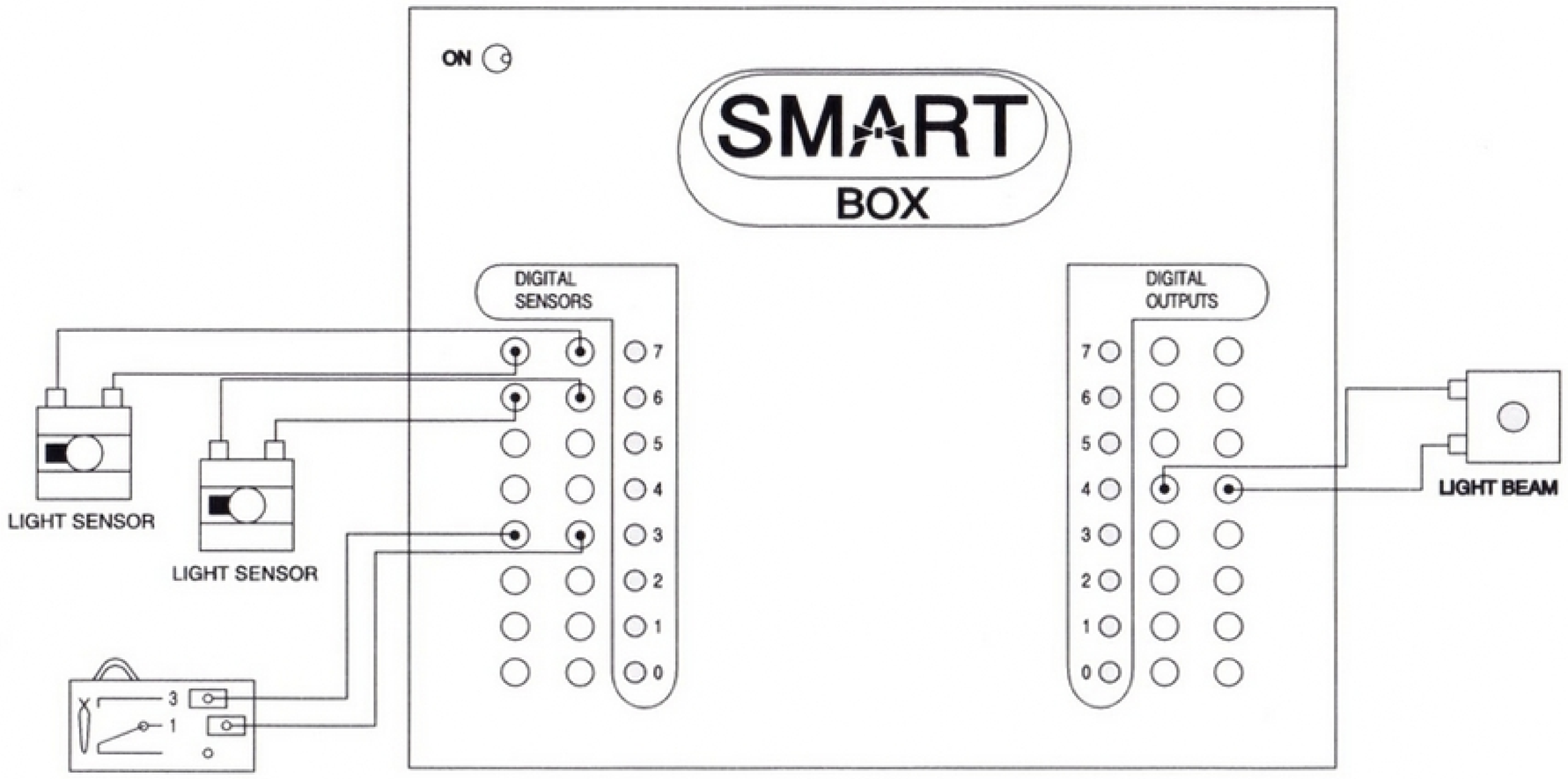


Fig. 5

b) Gun

All you need to make a gun are a switch (to act as a trigger) and a lamp to make a light beam to shine on the sensors. Connect them to the Smart Box as shown in fig 5. You could construct it as shown in fig 6, or you could design your own.

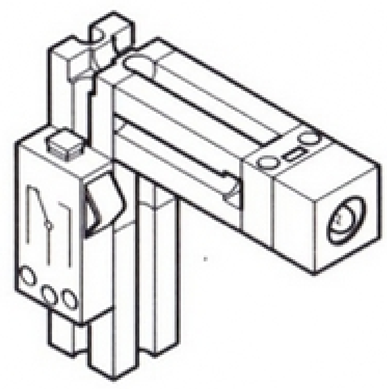


Fig. 6

Build the macro shown in fig 7. It is designed to wait until you press the trigger, and then switch on the lamp for half a second.

Run the macro, point the lamp at a sensor and press the trigger. If the beam shines on the sensor, the sensor's LED indicator on the Smart Box will light. Check that both sensors are responding.

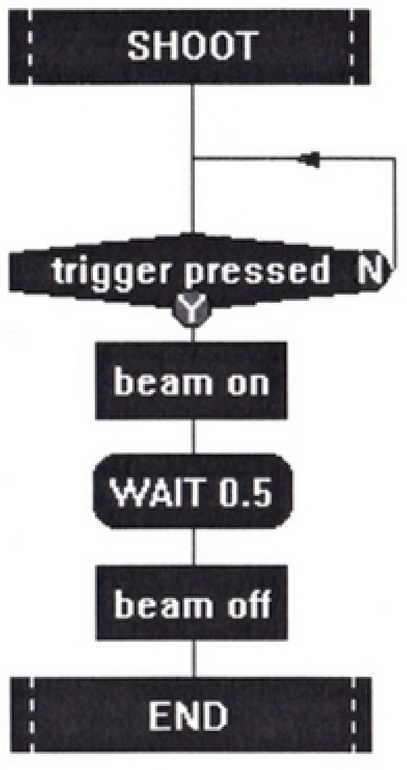
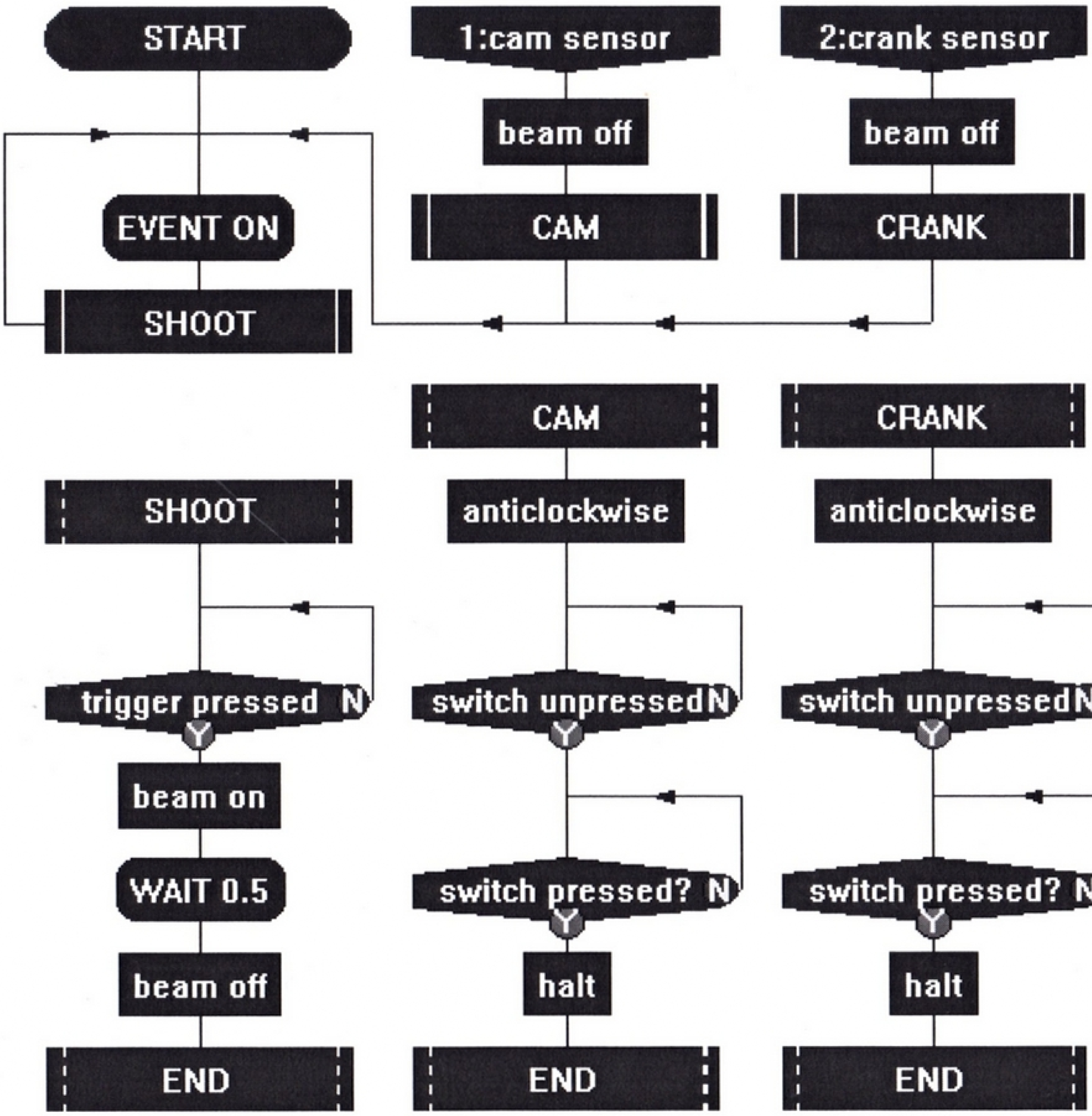


Fig. 7

The Complete System

Fig 8 shows the complete control system for the shooting gallery. Logicator's Event command is used to check if a target sensor is switched on. Each shot lasts only half a second and Events provide the fastest response.



Extending the System

a) Add a coloured lamp to each model. Extend the system so that the lamp lights if its target is hit

b) Develop the system so that each player has only five shots. The gun must stop working after the trigger has been pressed five times. A message on the screen tells the player that their game is over.
The system could also keep a record of each time a target is hit, and display the player's score at the end of a game.

c) Extend the CAM and CRANK macros so that the figures pause for a short time when they are out.

CONTROLLING MOVEMENT 4 : Theme Ride

At Wonderful Worlds theme park, a cliff railway takes passengers to different theme areas located on the side of a hill. They can choose to go to either: Space World, Fantasy World or Nature World.

Fig 1 shows a flow diagram of the control system needed for the ride.

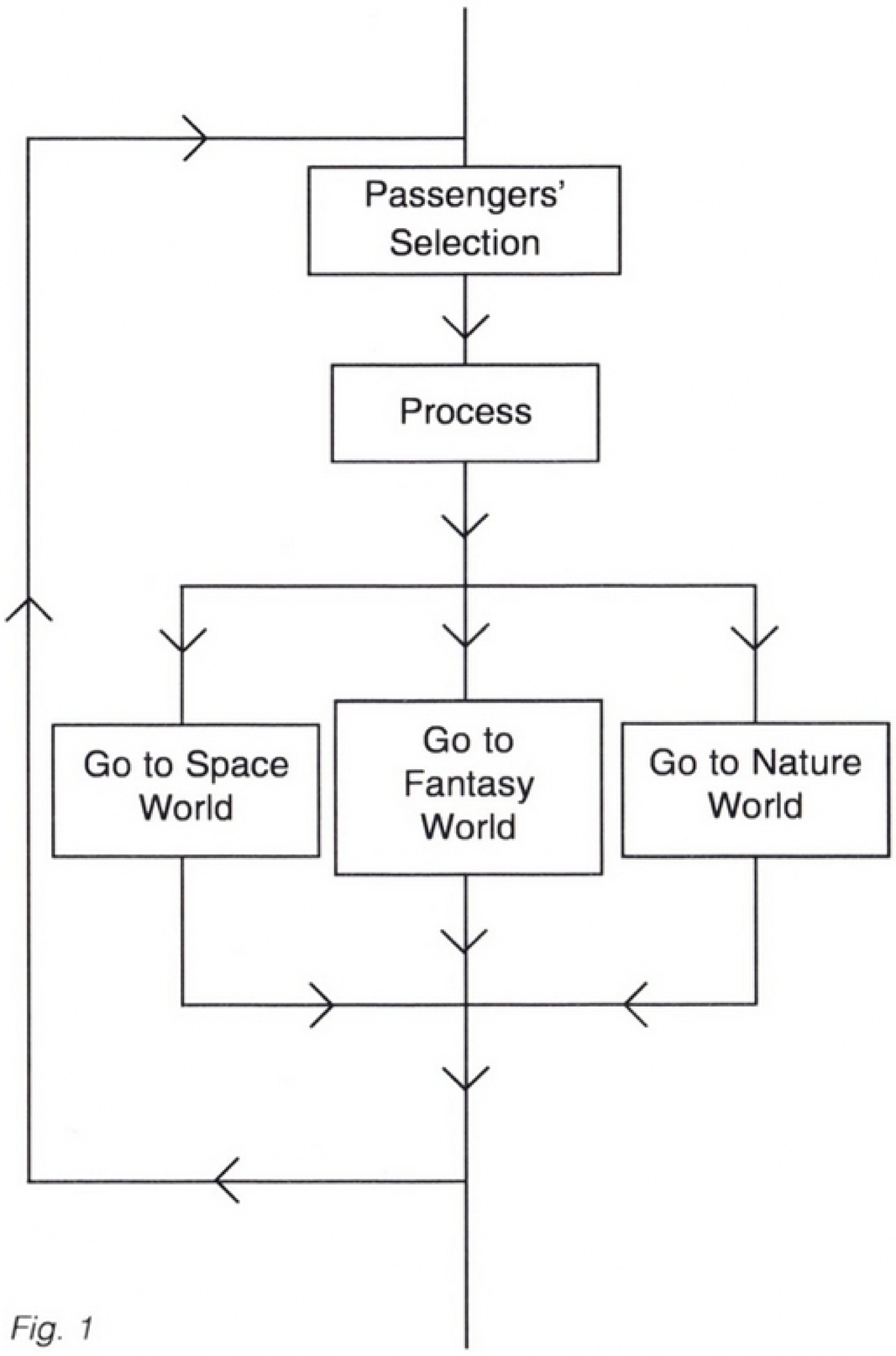
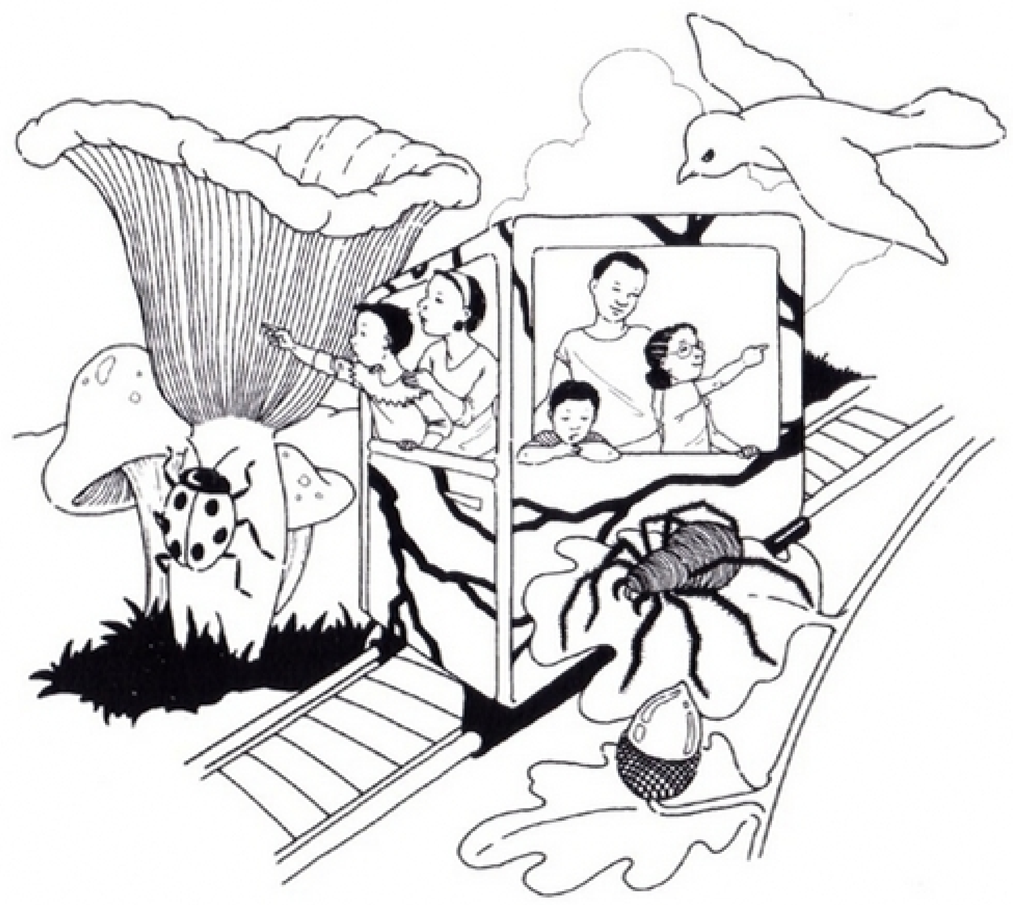


Fig. 1

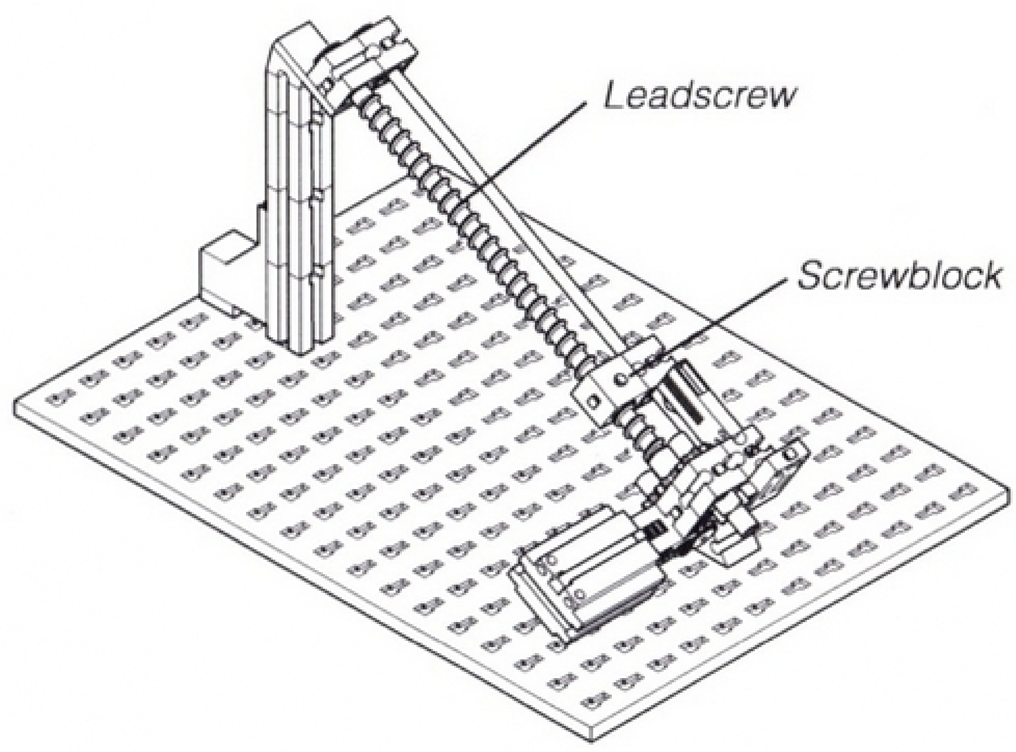
The passengers' choice of destination provides the input to the system. The process section takes the information provided by the input and uses it to select which of the three possible outputs is required. You can develop each section separately and put them together to make a complete system.

Build the model shown on the Leadscrew construction sheet. Connect the motor and switches to the Smart Box as shown on the sheet.



Leadscrew

This mechanism changes the rotary movement of the motor (input) into the linear movement of the screwblock (output).



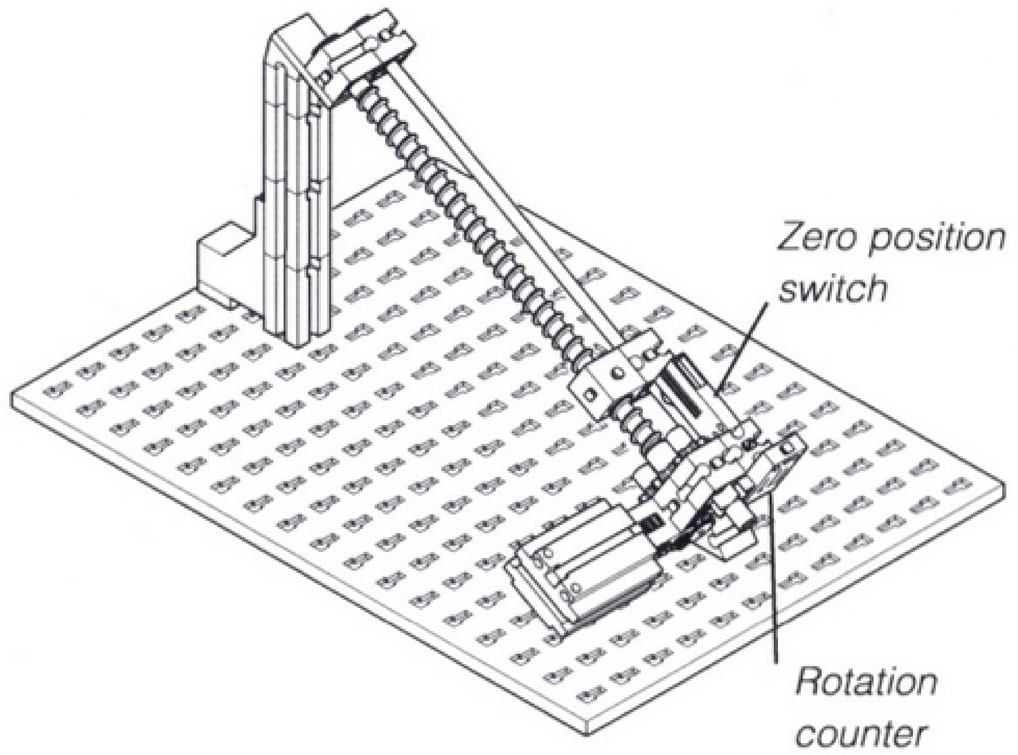
Both a leadscrew and a rack and pinion change rotary to linear movement, but there are at least two differences between them:

i) The output of the leadscrew tends to be much slower.
ii) The pinion can run freely up and down the rack if the motor is disconnected. The leadscrew stays locked in place. This is because the leadscrew is not reversible. It cannot be used to change linear to rotary movement.

Control System for the Leadscrew

The Leadscrew uses the same method of control as the Turntable. A rotation counter provides information about how far the passenger car (the screwblock) has moved up or down the leadscrew.

A zero position switch is used to provide information about the position of the car. When the block presses this switch the system can tell that it is in its start position at the bottom end of the leadscrew.



A COUNT command can be used to count the number of pulses on the rotation counter switch, so that the motor can be stopped when a set number have been counted. The number of pulses needed to take the car to each destination is shown in fig 2.

Build the macros shown in fig 3. Run each one in turn to check that it takes the car to the appropriate destination. Run the ZERO macro between each one so that the car always starts from the zero position.

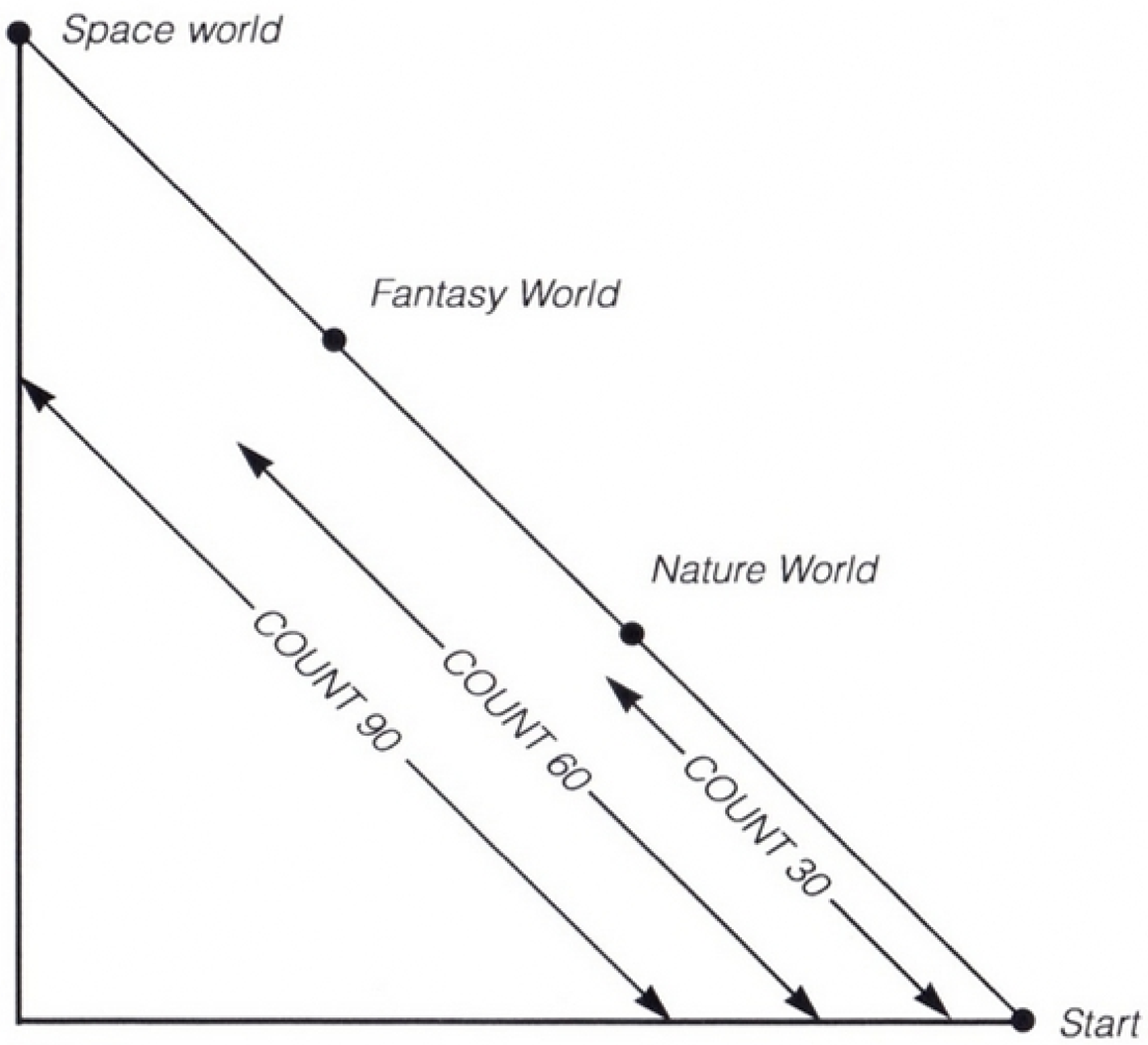


Fig. 2

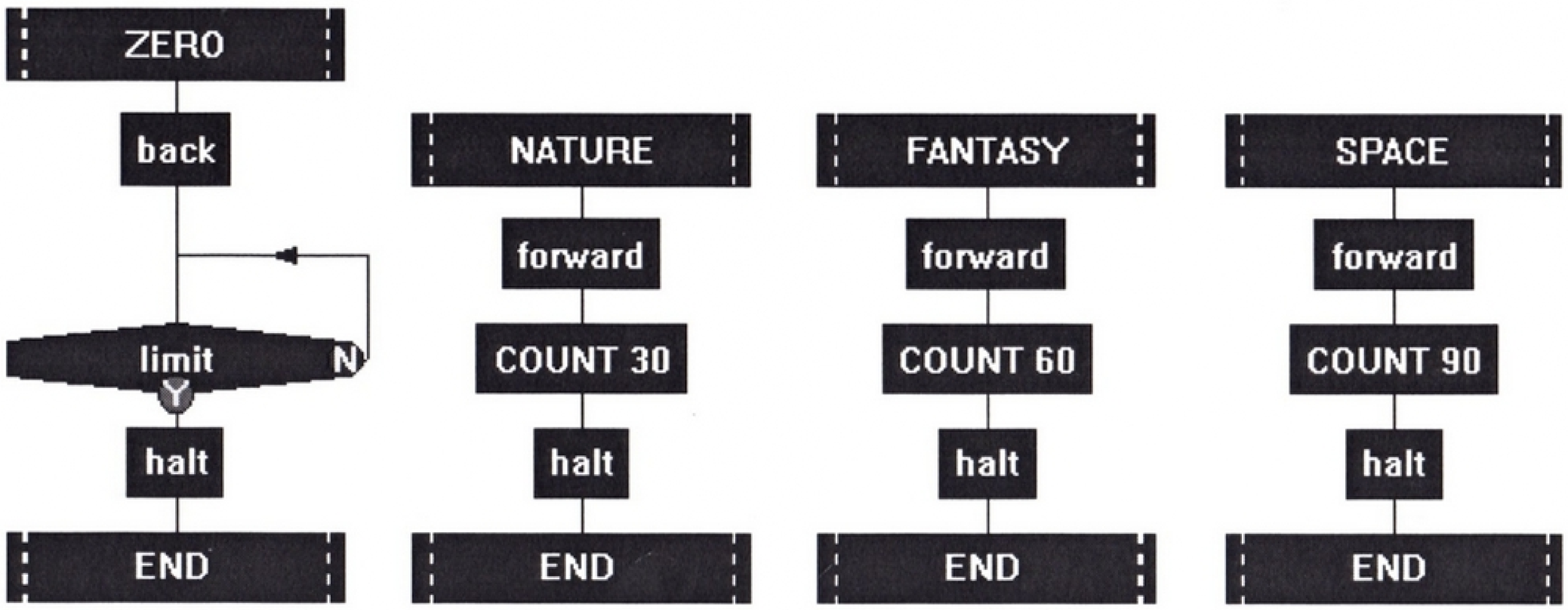


Fig. 3

Input and Process

The input to the Theme Ride system is the passengers' selection of destination. Logicator's INPUT command allows passengers to select their destination by entering a number into variable W (for "World").

Enter: 1 for Nature World
2 for Fantasy World
3 for Space World

When they have made their choice, the system carries out the process section.

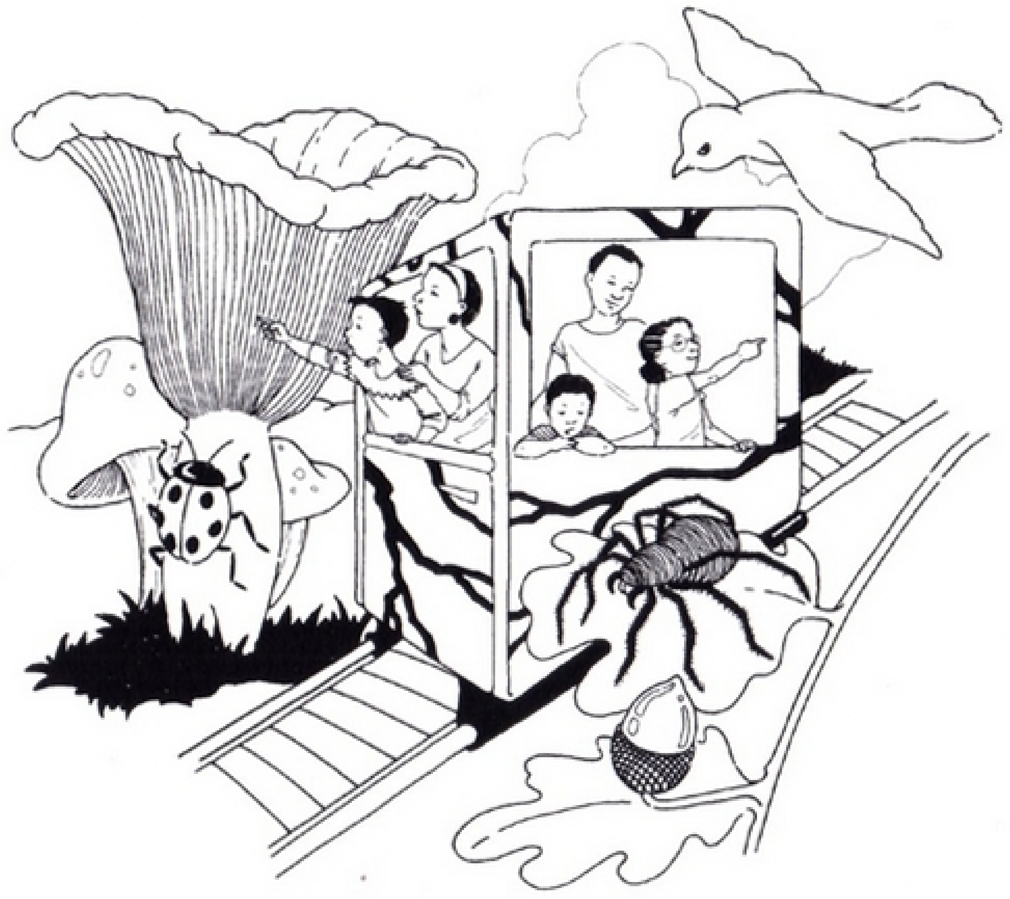
This part of the system takes the information provided by the input and uses it to select which of the three outputs is required. The passengers' selection is held in the variable W, so the PROCESS macro checks the value of W and then carries out the appropriate output macro.

Fig 4 shows the complete Theme Ride system. Add the main routine and PROCESS macro to your existing flowsheet, and then test the system.

Extending the System

The Theme Ride system would be more user-friendly if passengers were informed when they had reached their destination.

Use coloured lights and/or Logicator's MESSAGE, PLAY or SOUND commands to extend the system in this way.



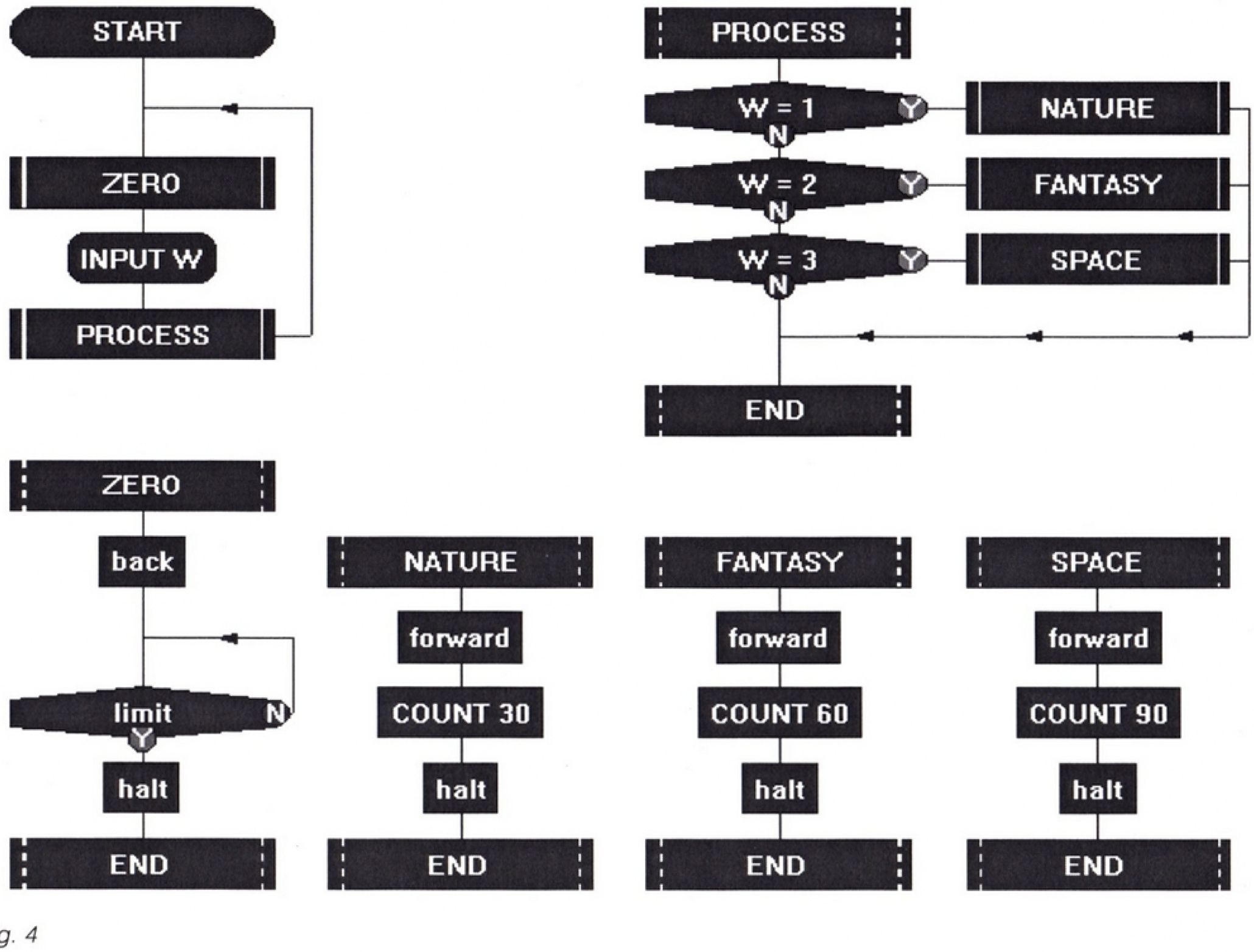


Fig. 4

An Alternative System

This system makes the ride more efficient by moving the car directly from one destination to another without having to return to the zero position before every trip. Read the following explanation, then edit your system to look like fig 5 and test it.

When a destination is selected, the system needs to know: i) which direction to move in, and ii) how many pulses to count.

Three variables are used to calculate this information :

W = the number of the selected destination. Enter 30 for Nature World, 60 for Fantasy World, 90 for Space World.

P = the number of the current position of the car. So, P is set to zero at the end of the ZERO macro. When the car reaches a destination at the end of an UP or DOWN macro, P is set to the number for that destination in the command P = W.

When the next destination is selected, if W is greater than P, the car must move up. If W is less than P, the car must go down.

M = the number of pulses the car has to move to go from its current position to the selected destination.

The commands P=0, P=W, M=W-P, M=P-W are all Expression commands.

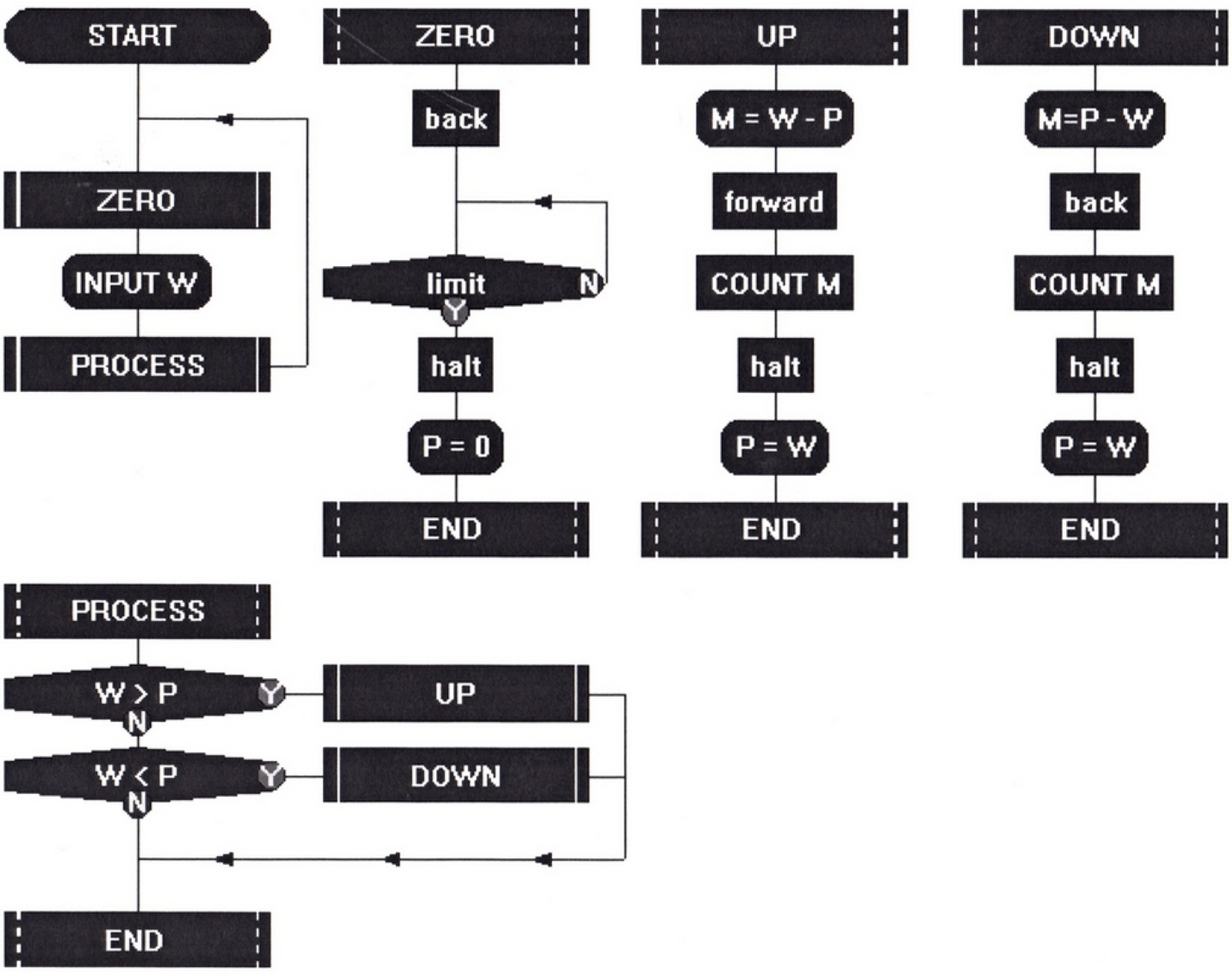


Fig. 5

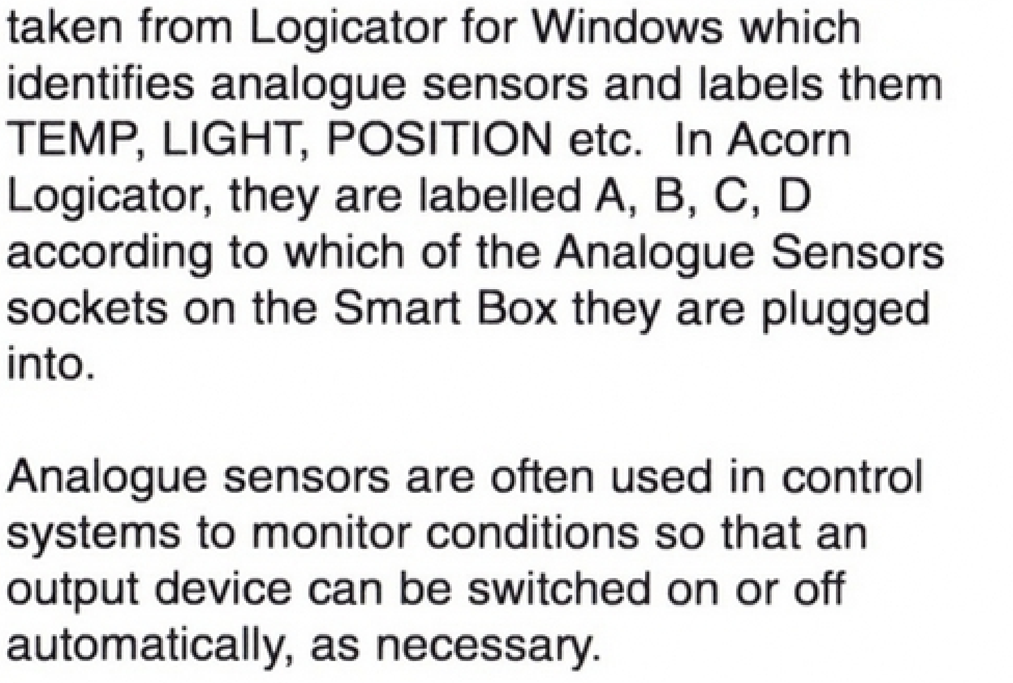
CONTROLLING MOVEMENT 5 : Analogue Sensors

An analogue sensor constantly monitors one part of an environment, such as the humidity in a greenhouse or the light level in a room, or the temperature in an incubator for premature babies. The sensor supplies a reading to the computer through the Smart Box. The reading changes as the heat or light level changes.

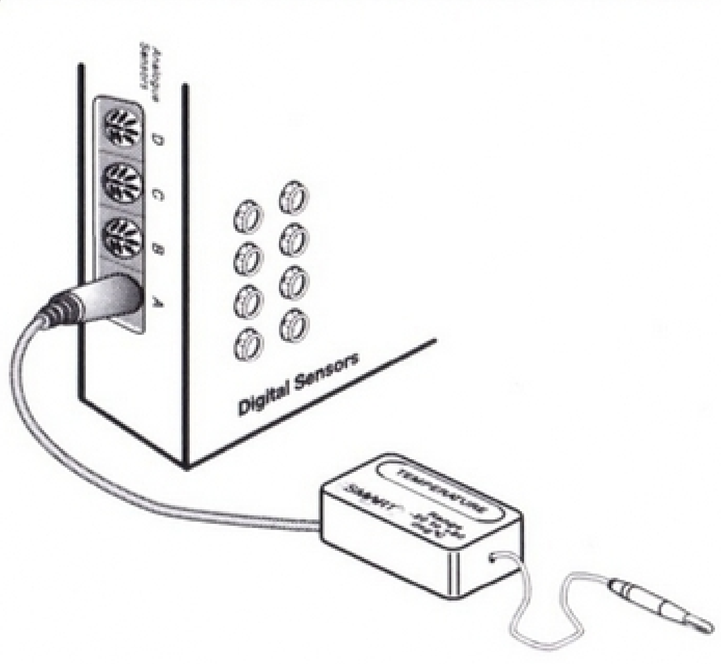
NOTE: The flowsheets shown in this book are taken from Logicator for Windows which identifies analogue sensors and labels them TEMP, LIGHT, POSITION etc. In Acorn Logicator, they are labelled A, B, C, D according to which of the Analogue Sensors sockets on the Smart Box they are plugged into.

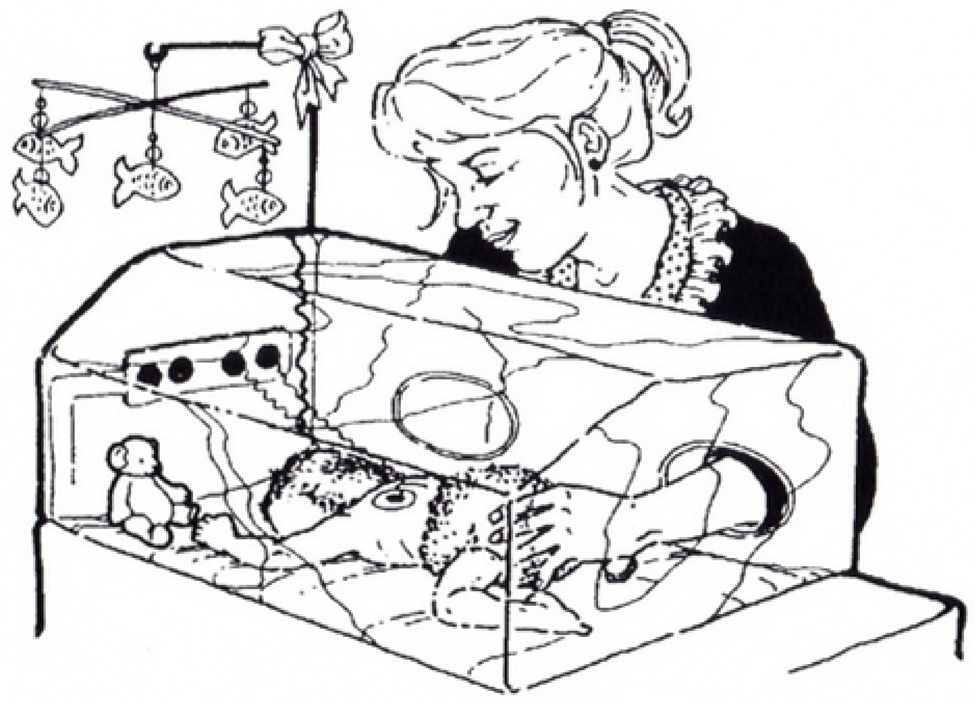
Analogue sensors are often used in control systems to monitor conditions so that an output device can be switched on or off automatically, as necessary.

Build the model fan shown on the Hand Dryer construction sheet. Connect it to the Smart Box.



Plug a Smart temperature sensor into an Analogue Sensors socket on the Smart Box.





Build the flowsheet shown in fig 1. This is designed to switch on the fan when the temperature rises above the threshold, and switch it off when the temperature falls below the threshold.

Set the threshold temperature in the Decision command to a degree or so higher than the reading displayed in the Analog Channels window on the Logicator screen.

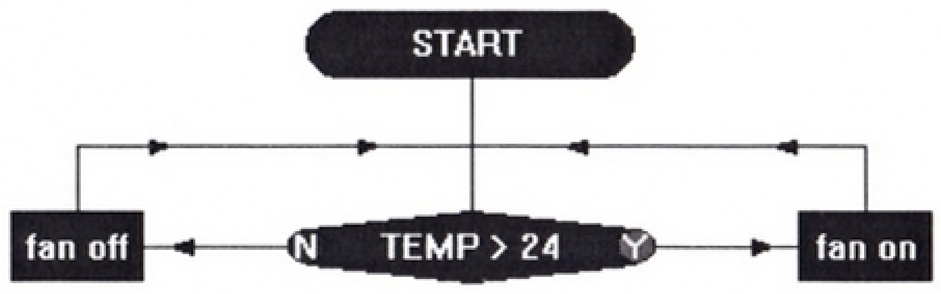


Fig. 1

Run the flowsheet and test the system by warming up the sensor in your hands. Hold it near the fan to cool it down again.

You may find that, when the temperature reading is fluctuating around the threshold, the fan keeps switching on and off. Try editing the flowsheet to make the system switch cleanly.

Varying the Output

Another way to use an analogue sensor in a control system is to make the output vary in response to the varying readings from the sensor.

a) Controlling Speed

Use the fan model and Smart temperature sensor.

Build the flowsheet shown in fig 2. It is designed to speed up the fan as the temperature rises, and slow it down as the temperature falls. If you use the lowest motor speeds, it will be easier to see when the fan changes speed.

Run the flowsheet and test it.

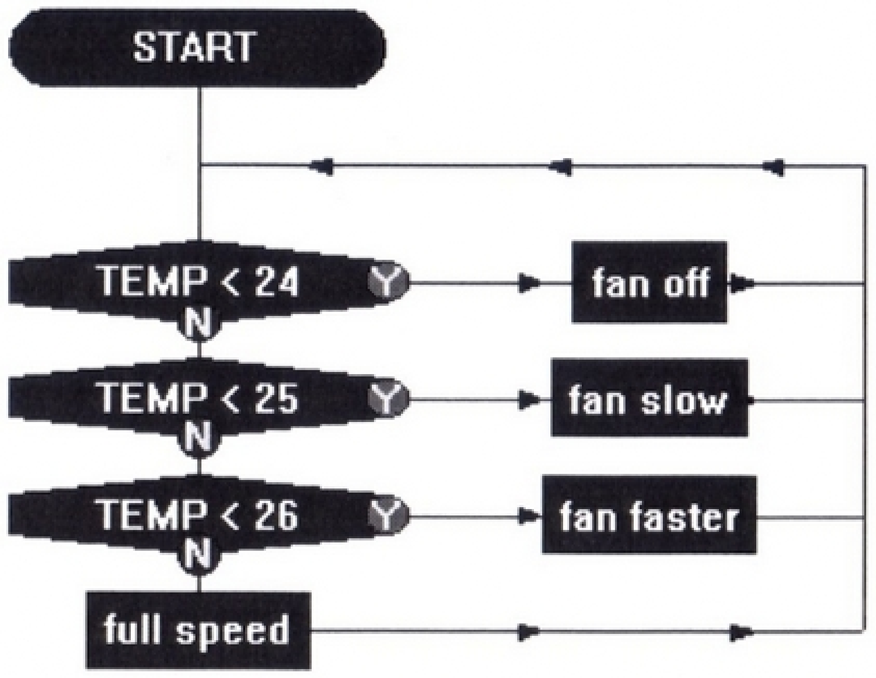


Fig. 2

You could use a similar flowsheet for a system in which a Smart position sensor is used as a speed controller for the single motor buggy shown in fig 3.

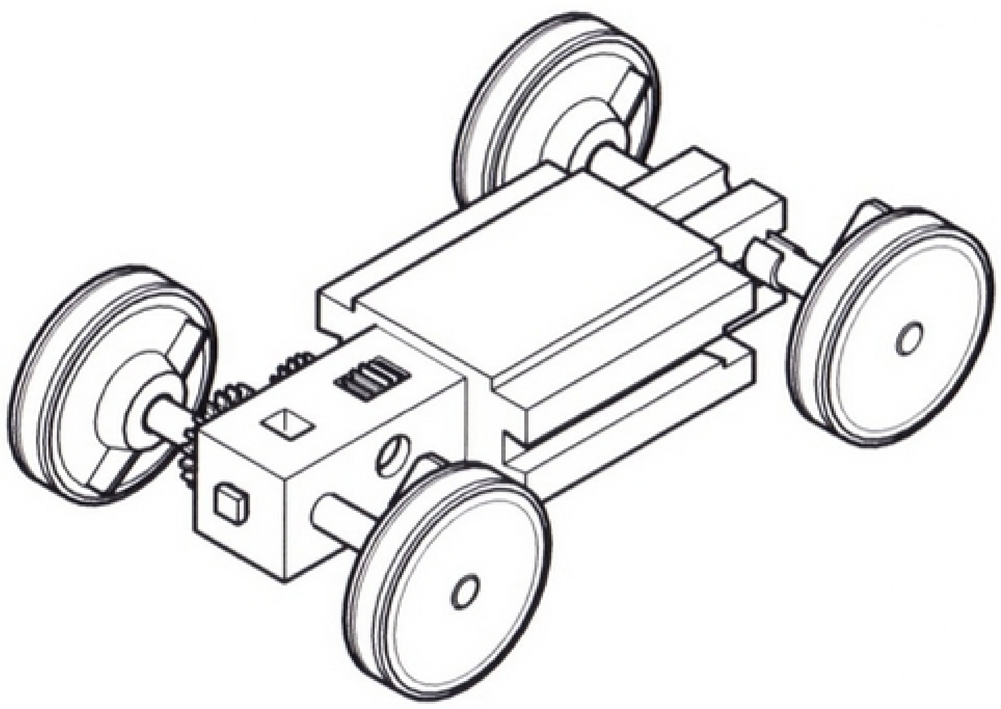


Fig. 3

b) Automatic Lighting

Connect three lamps (Digital Outputs) and a Smart light sensor (Analogue Sensors) to the Smart Box.

Build the flowsheet shown in fig 4. It is designed to switch on the lamps one by one as it gets darker, and switch them off in the same way as it get lighter.

Run the flowsheet and test it.

You could develop this flowsheet to use the eight green LEDs of the Smart Box Digital Outputs as a light meter. The more light shines on the sensor, the more LEDs light up.

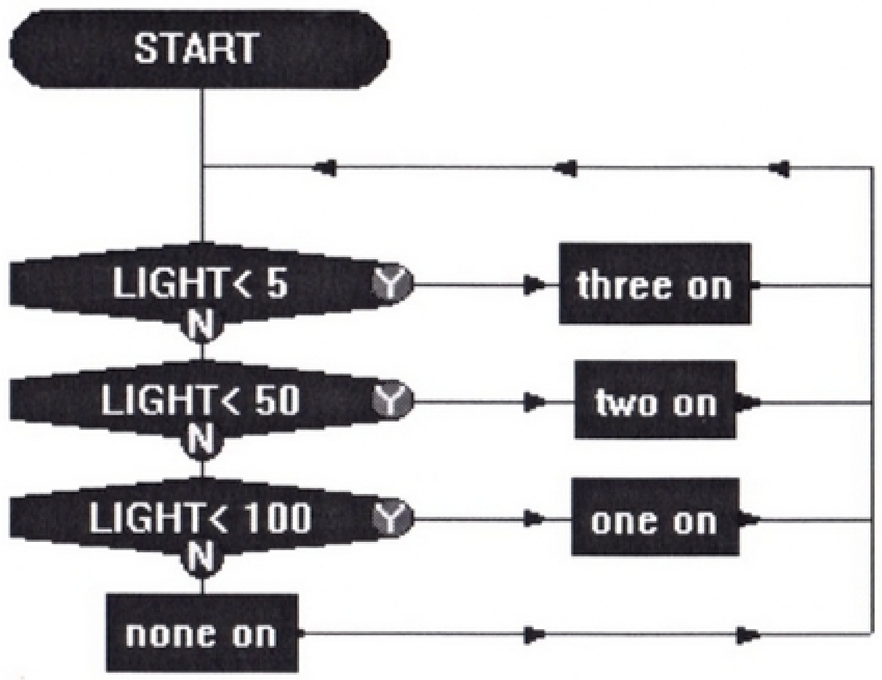


Fig. 4

Monitoring Position

You can use a Smart position sensor to provide accurate feedback on the position of a motor-driven device

a) Turntable Control

Build a turntable onto the position sensor as shown in fig 5. Drive the turntable with a worm gear as shown on the Worm and Wheel construction sheet.

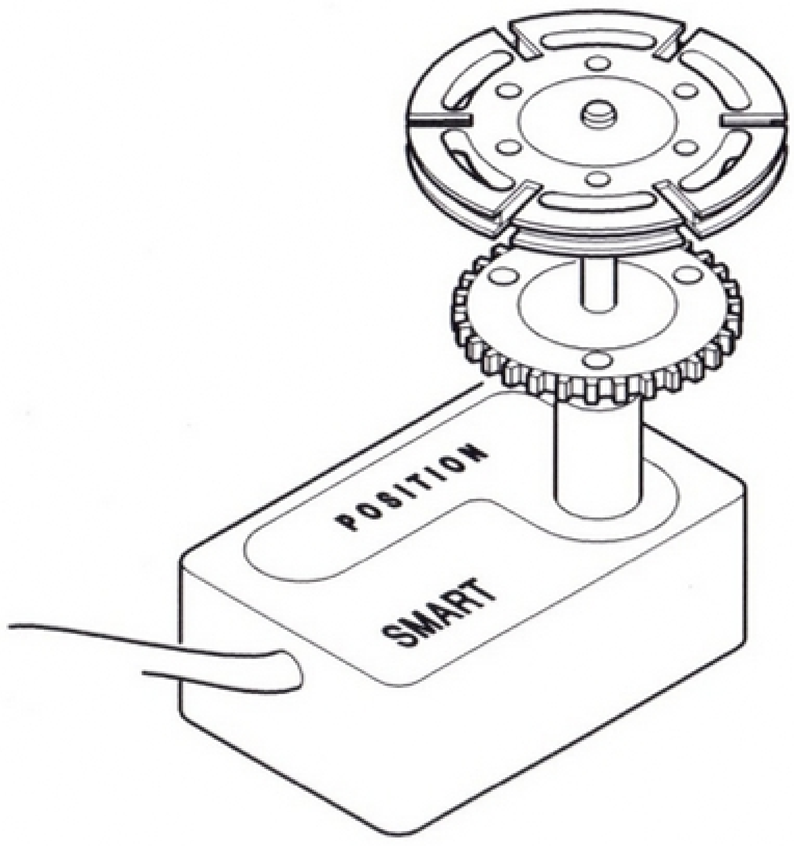


Fig. 5

Fig 6 shows the kind of flowsheet that you can use to drive the turntable accurately from its zero position to any position in its turning circle.

b) Stability Testing

You could develop a system to test the stability of a structure by tilting it and using the position sensor to measure the angle at which it just begins to topple.

Build a platform on which the structure can stand. Use a motor-driven mechanism to tilt the platform so that you can control the speed of tilting. Use a limit switch to stop it at the moment the structure starts to topple.

Fix a weight to the spindle of the position sensor as shown in fig 6. Fix the sensor to the platform. As the platform tilts, the weight will make the spindle turn, so that you can read off the angle at which the structure started to topple.

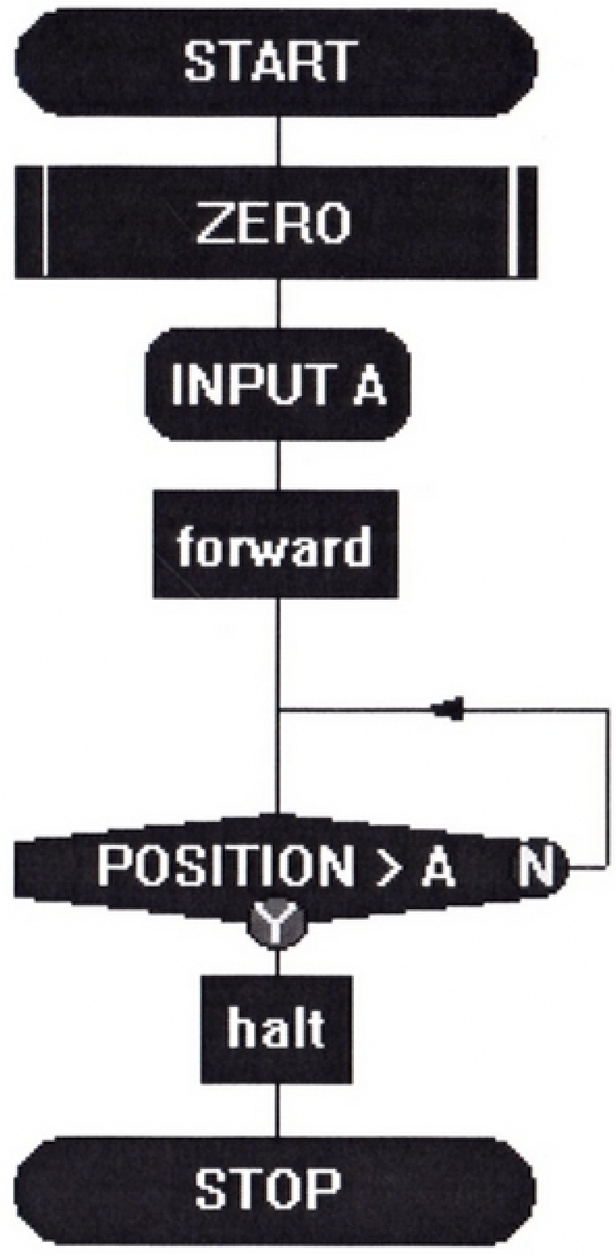


Fig. 6

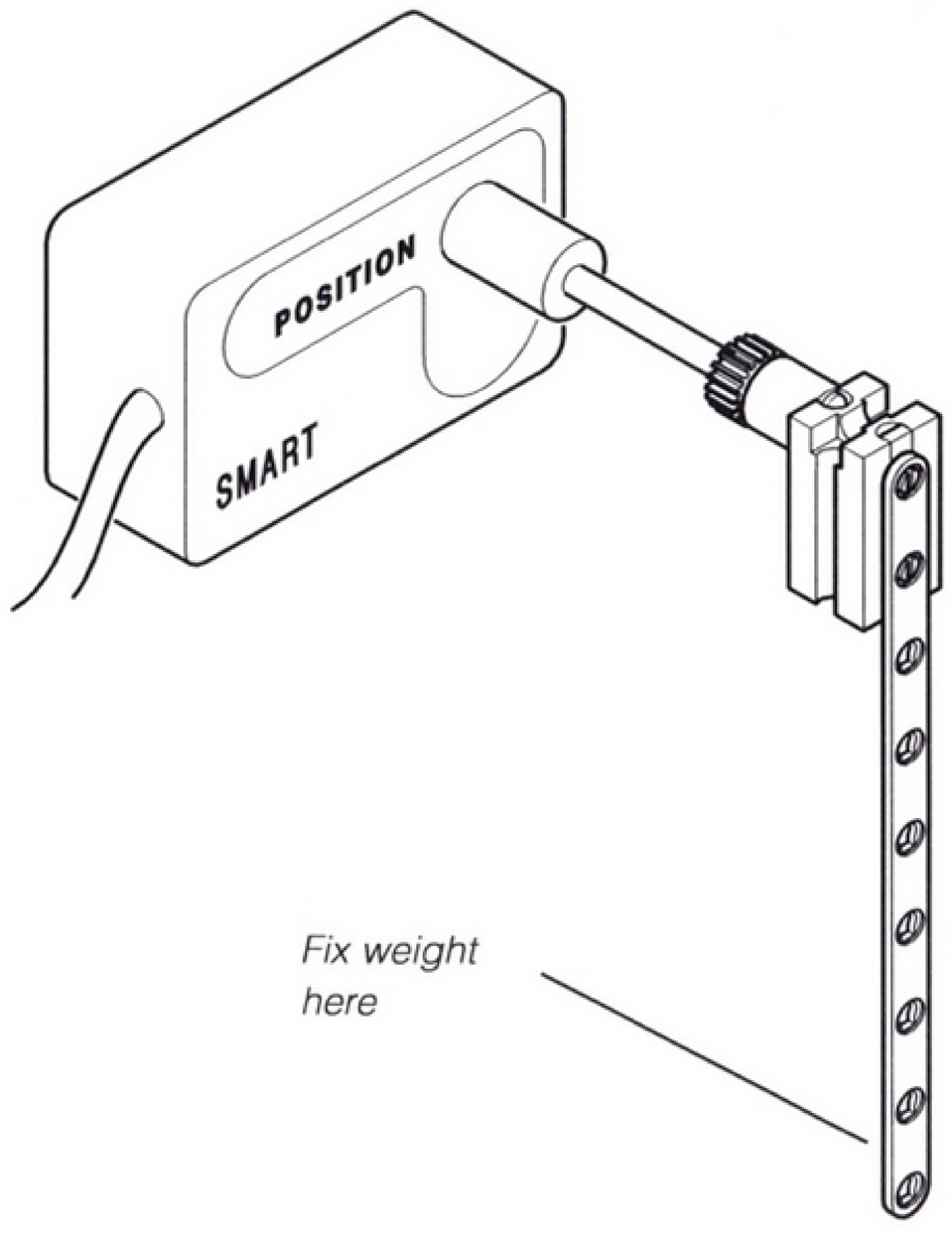


Fig. 7

Smart Move

Help Sheet PC Smart Move

How to Build a Procedure

1. Type: *build <procedure name>*
e.g. *build lamps*

2. Press the Enter key.

3. The screen will change to the Edit screen, showing your procedure name. Type your procedure. Press the Enter key at the end of each line.

How to Edit a Procedure

1. Type: *edit <procedure name>*
e.g. *edit lamps*

2. Press the Enter key.

3. The screen will change to the Edit screen, showing your procedure.

Use word processor methods to edit the text of the procedure.

e.g.

To add new text: place the cursor and type the text.

To delete or change text: place the cursor to the right of the text, press the backspace key, and type the new text.

To make space for a new line in the procedure: move the cursor to the line below where you want the new line. Press the tab key.

How to Run a Procedure

1. Press the Esc key to return to the main screen.

2. Type the name of the procedure that you want to run.
e.g. *lamps*

3. Press the Enter key.

When you want to stop a procedure running, press the Esc key.

How to Enter Direct Commands

This method of entering individual commands one at a time is useful for testing the output devices on your model.

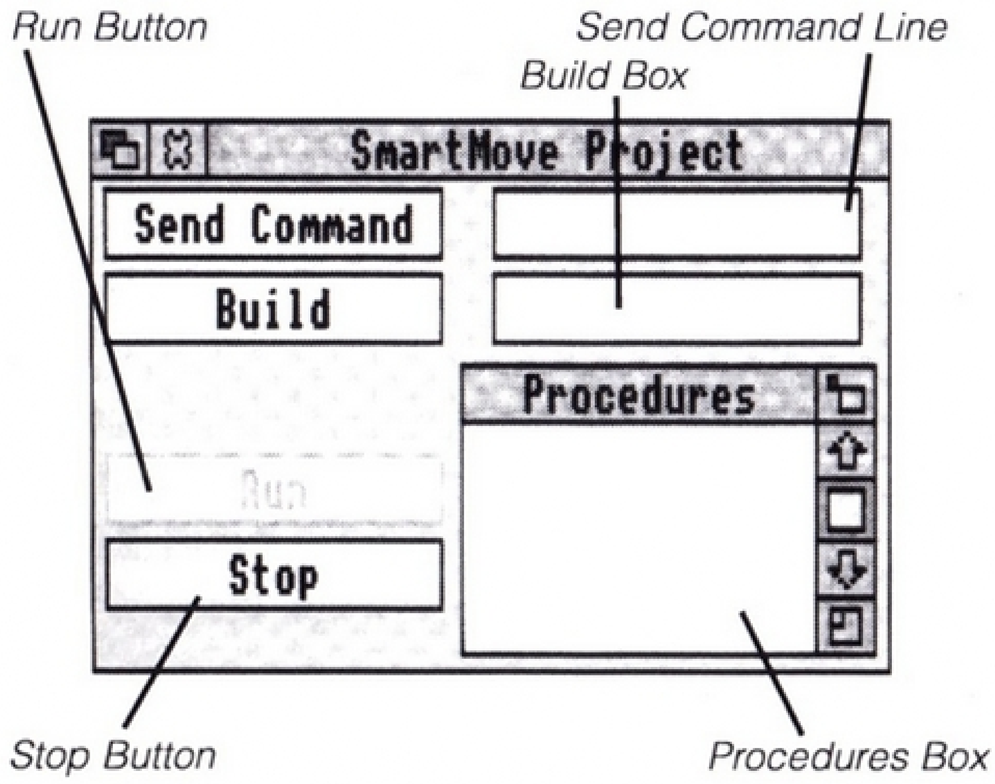
1. Type the command.
eg *switch on 1*

2. Press the Enter key.

The output will stay on until you type: *switch off 1* and press the Enter key.

NOTE: This Help Sheet is also for use with RM186 and BBC versions of Smart Move.

Help Sheet Acorn Smart Move



How to Build a Procedure

1. Click in the Build box.

2. Type the procedure name.

3. Press the Return key.

4. An Edit window will appear. It will have your procedure name. Type your procedure in this window. Press the Return key at the end of each line.

NOTE: When you want to build a new procedure, delete the last procedure name from the Build box and type the new name.

How to Run a Procedure

1. Click the X box on the Edit window.

2. Your procedure name will appear in the Procedures box.

3. Click on the name of the procedure that you want to run.

4. Click the Run button.

When you want to stop a procedure running, click the Stop button.

How to Edit a Procedure

1. Look in the Procedures box for the name of the procedure that you want to edit.

2. Double click on the name.

3. The Edit window for that procedure will appear.

Use word processor methods to edit the text of the procedure.

e.g.

To add new text: click to place the cursor and type the text.

To delete or change text: click to the right of the text, press the backspace key and type the new text.

To make space for a new line in the procedure: click on the line below where you want the new line; press the tab key.

How to Enter Direct Commands

This method of entering individual commands one at a time is useful for testing the output devices on your model.

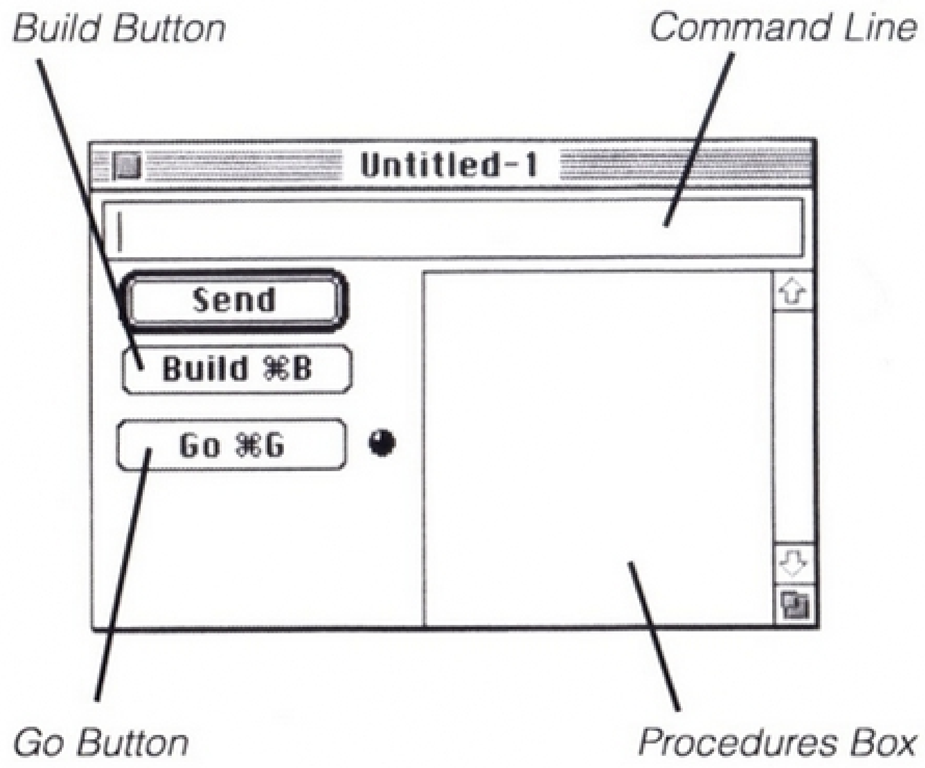
1. Click in the Send Command line

2. Type the command.
e.g. *switch on 1*

3. Click the Send Command button or press Return.

The output will stay on until you delete what's in the Send Command line and type *switch off 1* Press Return.

Help Sheet Macintosh Smart Move



How to Build a Procedure

1. Click the Build button.

2. Type the procedure name in the box that appears.

3. Click OK.

4. An Edit window will appear. It will have your procedure name. Type the procedure into the window. Press the Return key at the end of each line.

How to Run a Procedure

1. Click the close box on the Edit window.

2. Your procedure name will appear in the Procedures box.

3. Click on the name of the procedure that you want to run.

4. Click the GO button.

When you want to stop a procedure running, select Procs_Stop from the Menu bar.

How to Edit a Procedure

1. Look in the Procedures box for the name of the procedure that you want to edit.

2. Double click on the name.

3. The Edit window for that procedure will appear.

Use word processor methods to edit the text of the procedure.

e.g.

To add new text: click to place the cursor and type the text.

To delete or change text: highlight the text, press the delete key and type the new text.

To make space for a new line in the procedure: click on the line below where you want the new line; press the Return key; place the cursor at the start of the new line.

How to Enter Direct Commands

This method of entering individual commands one at a time is useful for testing the output devices on your model.

1. Click in the Command line

2. Type the command.
e.g. *switch on 1*

3. Click the Send button or press Return.

The output will stay on until you delete what's in the Command line and type *switch off 1* and press Return.

Introduction to computer control

Many of the machines that are used in the home and in industry are controlled by a microprocessor that has been programmed to operate the various electrical devices that do the work in the machine.

For example, an automatic washing machine might contain:

an electric motor to rotate the tub,

electrically-operated valves to let water in and out,

a heater to heat the water.

These are called **output** devices.

The washing machine also contains switches and sensors. For example, there might be:

an on/off switch,

a key-pad to select different washing cycles,

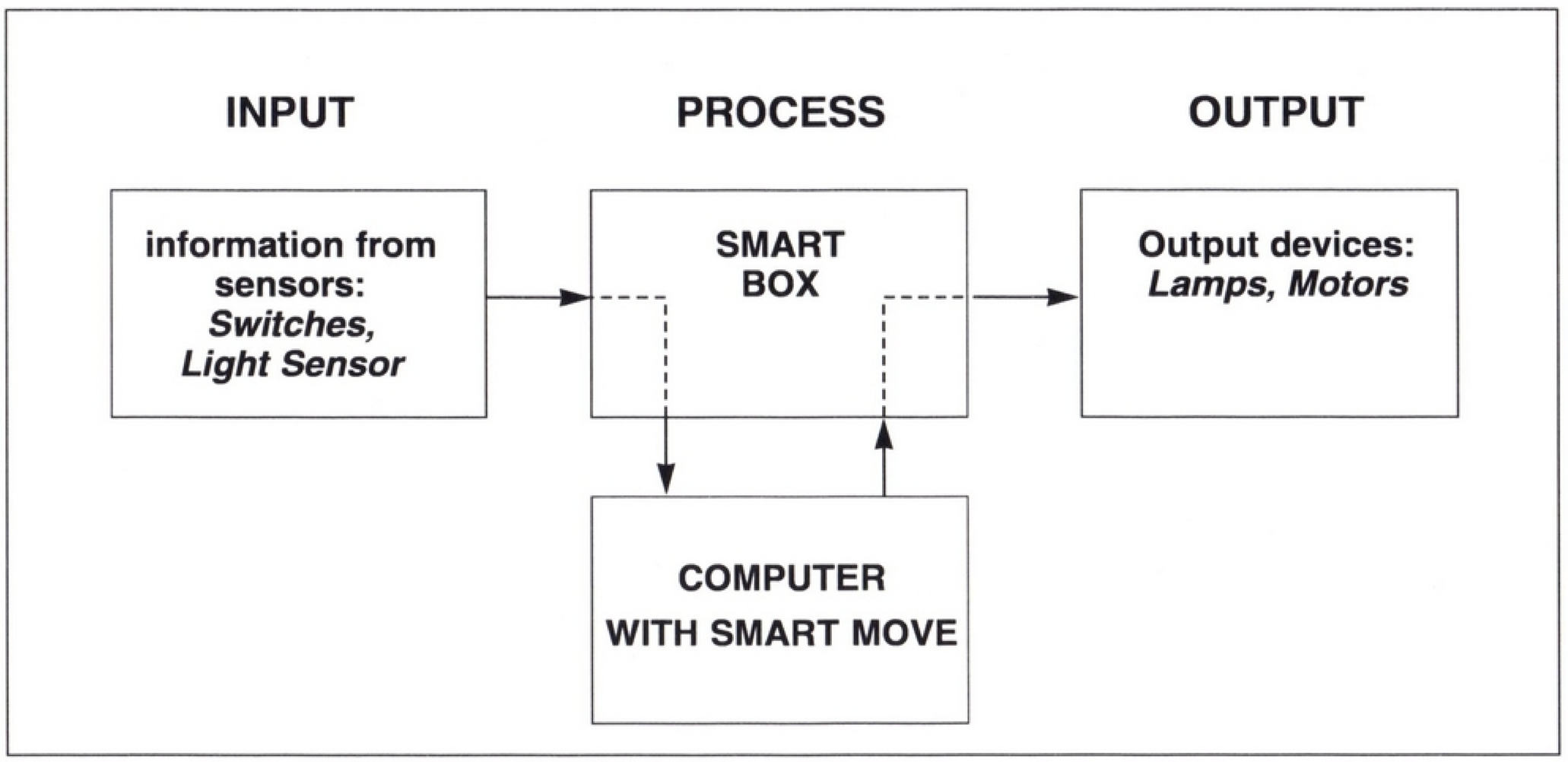
a sensor to check the temperature of the water.

These are called **input** devices.

The job of the microprocessor is to check the state of the input signals from these switches and sensors, and to switch the output devices on and off as they are needed.

This is called the **process**.

All control systems have input, process and output. The diagram below shows how the Smart Box and computer make up the process stage of the control systems that you will build. You will use Smart Move software to design and build the programs that control these systems.



START UP : Control sequence for lamps

Build the model shown on the Hand Dryer construction sheet. There are two lamps on the model (see fig. 1). They simulate the heater.

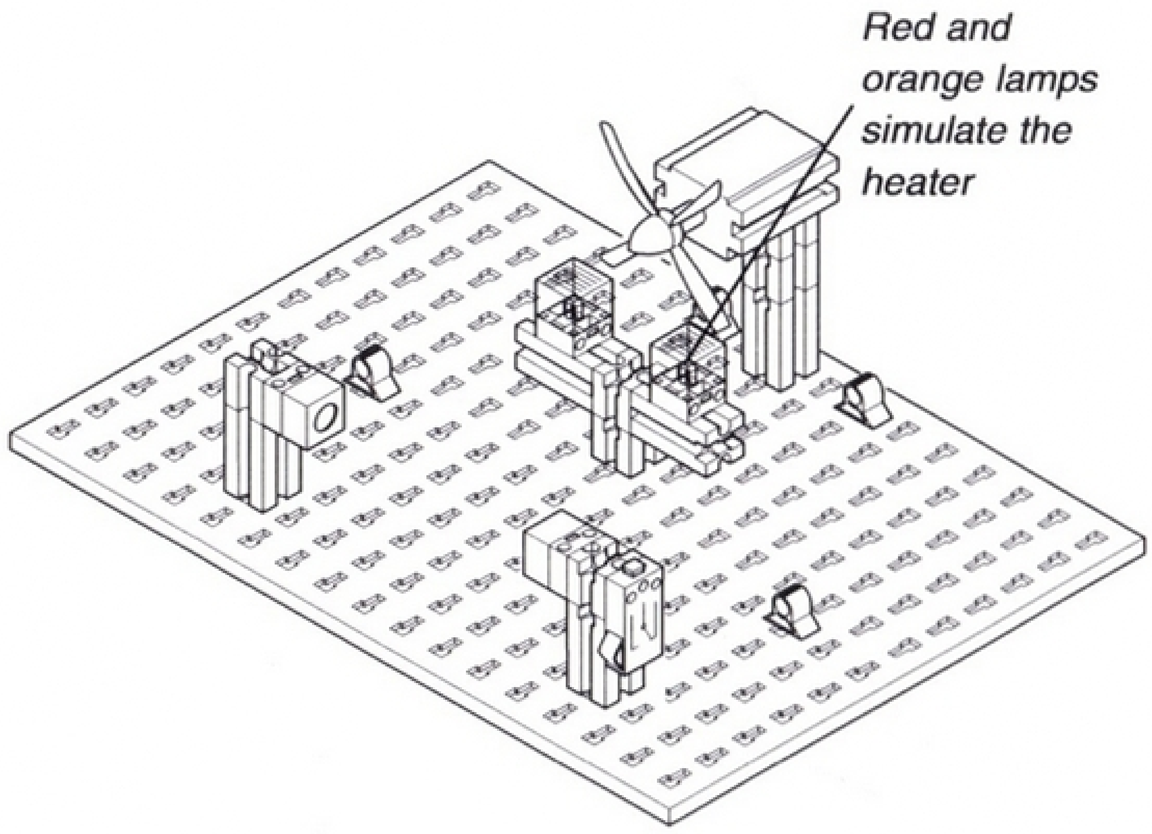
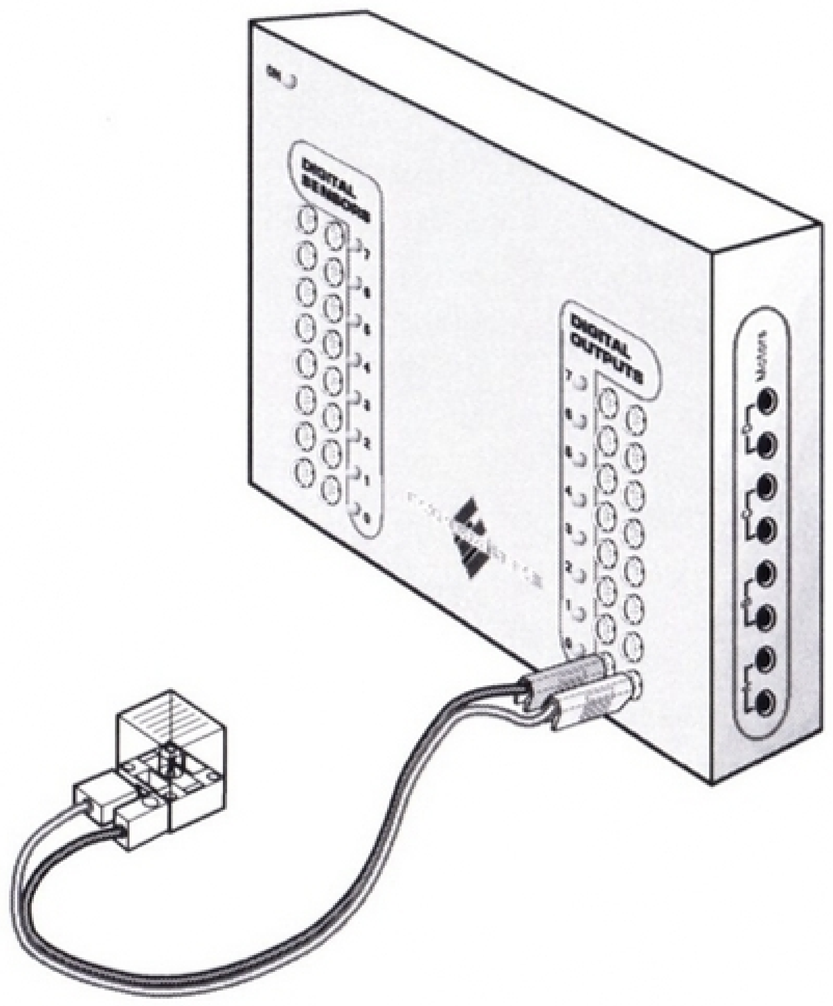


Fig. 1.

The lamps are output devices. They should be connected to Digital Outputs sockets on the Smart Box.

Connect:
Red lamp to Digital Outputs 0
Orange lamp to Digital Outputs 1



Task One

You use the commands SWITCH ON and SWITCH OFF to switch on and off any of the output devices connected to Digital Outputs sockets on the Smart Box.

You use the WAIT command to keep the output devices switched on or off for any length of time you like.

The Help Sheet shows you how to build a procedure. Use it to help you to build the procedure called "lamps" that is shown in fig. 2.

When you have built the procedure, run it to test that it works as it should do. The Help Sheet shows you how to run a procedure.

Procedure: lamps
switch on 0 wait 2 switch off 0

Fig. 2. This procedure will switch on the red lamp for 2 seconds and then switch it off.

Task Two

The Help Sheet shows you how to edit a procedure. Use it to help you to edit your procedure so that it:

switches on the red lamp,

waits 1 second,

switches on the orange lamp, so both lamps are on,

waits 1 second,

switches off both lamps.

Your procedure should look like fig. 3.

Procedure: lamps
switch on 0 wait 1 switch on 1 wait 1 switch off 0,1

Fig. 3. Notice that you can use one command to switch more than one output on or off.

Task Three

Each time you want to start the light sequence, you have to re-run the procedure.

It is more efficient to add a repeat loop as shown in fig. 4 so that the procedure repeats the sequence automatically. Edit your procedure so that it looks like fig. 4.

Run the procedure. The Help Sheet shows you how to stop it running.

Procedure: lamps
repeat switch on 0 wait 1 switch on 1 wait 1 switch off 0,1 wait 1 forever

Fig. 4. Notice that you must include an extra "wait 1" command so that you can see that the lamps have switched off before the red lamp is switched on again.

Task Four

Now edit your procedure to meet the following specification:

Run the procedure to test it.

Specification

The red and orange lamps must switch on together. They stay on for 2 seconds and then they switch off one at a time at intervals of 2 seconds. The sequence must repeat itself until you stop it.

START UP 2 : Controlling a motor

The electric motor on the Hand Dryer model drives a fan that blows air over the heater and then onto the user's hands (see fig. 5).

Connect the motor to Motors socket A on the side of the Smart Box.

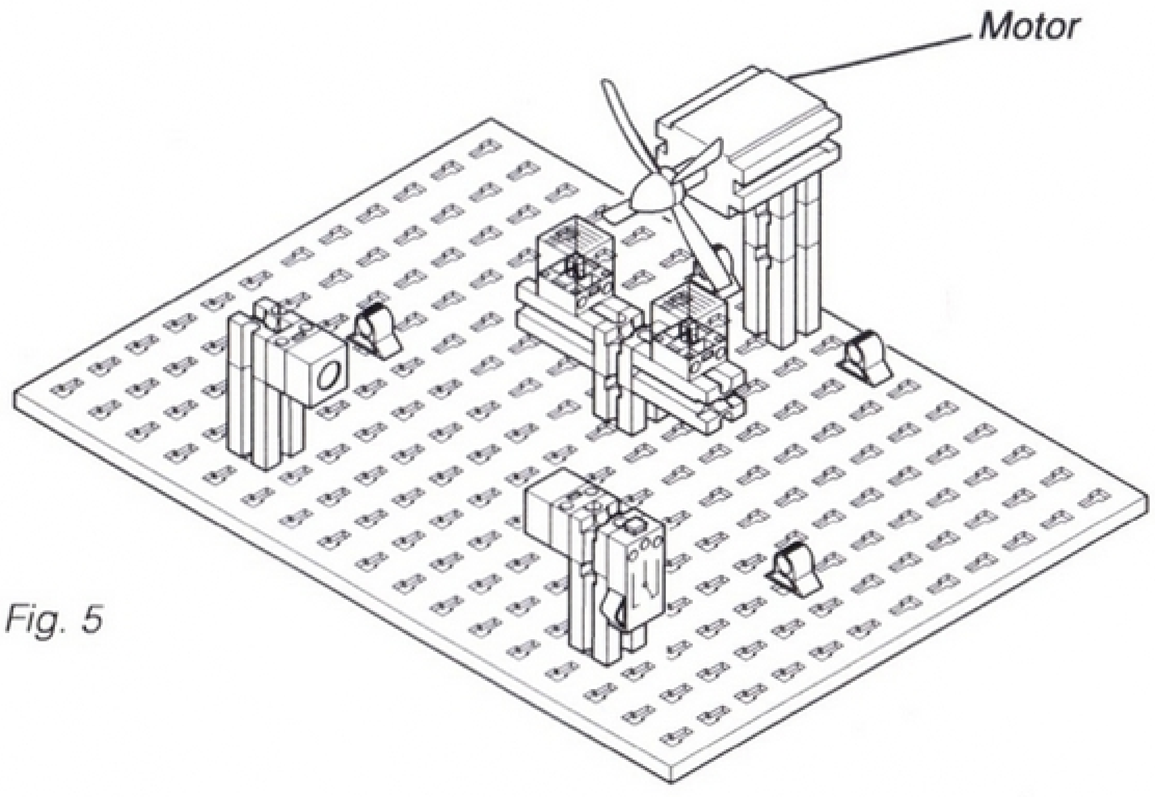
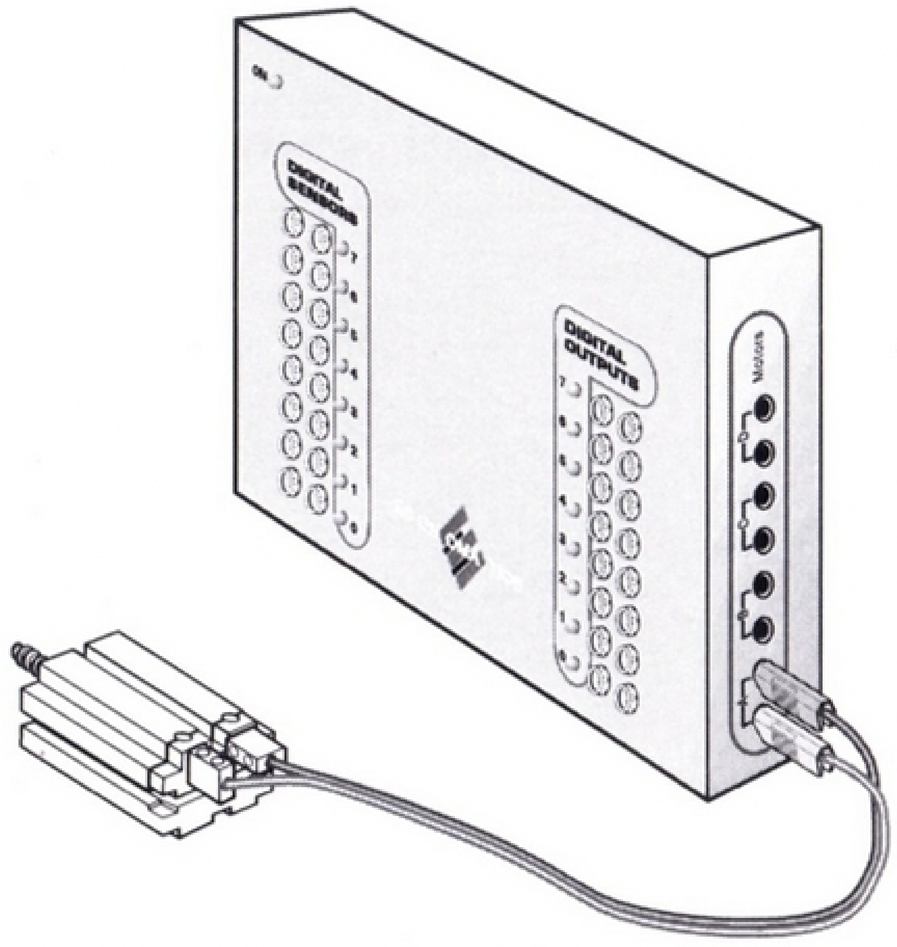


Fig. 5



Task Five

You use the following commands to switch motors on and off:

FORWARD switches the motor on in one direction.
BACKWARD switches the motor on in the opposite direction.
HALT switches the motor off.

Build the procedure shown in fig. 6. When you have built the procedure, run it to test it.

Procedure: motor
forward a wait 2 halt a

Fig. 6. This procedure switches on the motor for 2 seconds, and then switches it off.

Task Six

Edit your procedure to meet the following specification:

Run your procedure to test it.

Specification

The motor must move clockwise for 3 seconds.

It then switches off for 1 second.

It then moves anticlockwise for 3 seconds.

The motor then stops.

START UP 3 : Using input signals from micro-switches

Fig. 7 shows the micro-switch on the model. It can be used to switch the dryer on.

The micro-switch is an input device.Connect the switch to Digital Sensors socket 0 on the Smart Box.

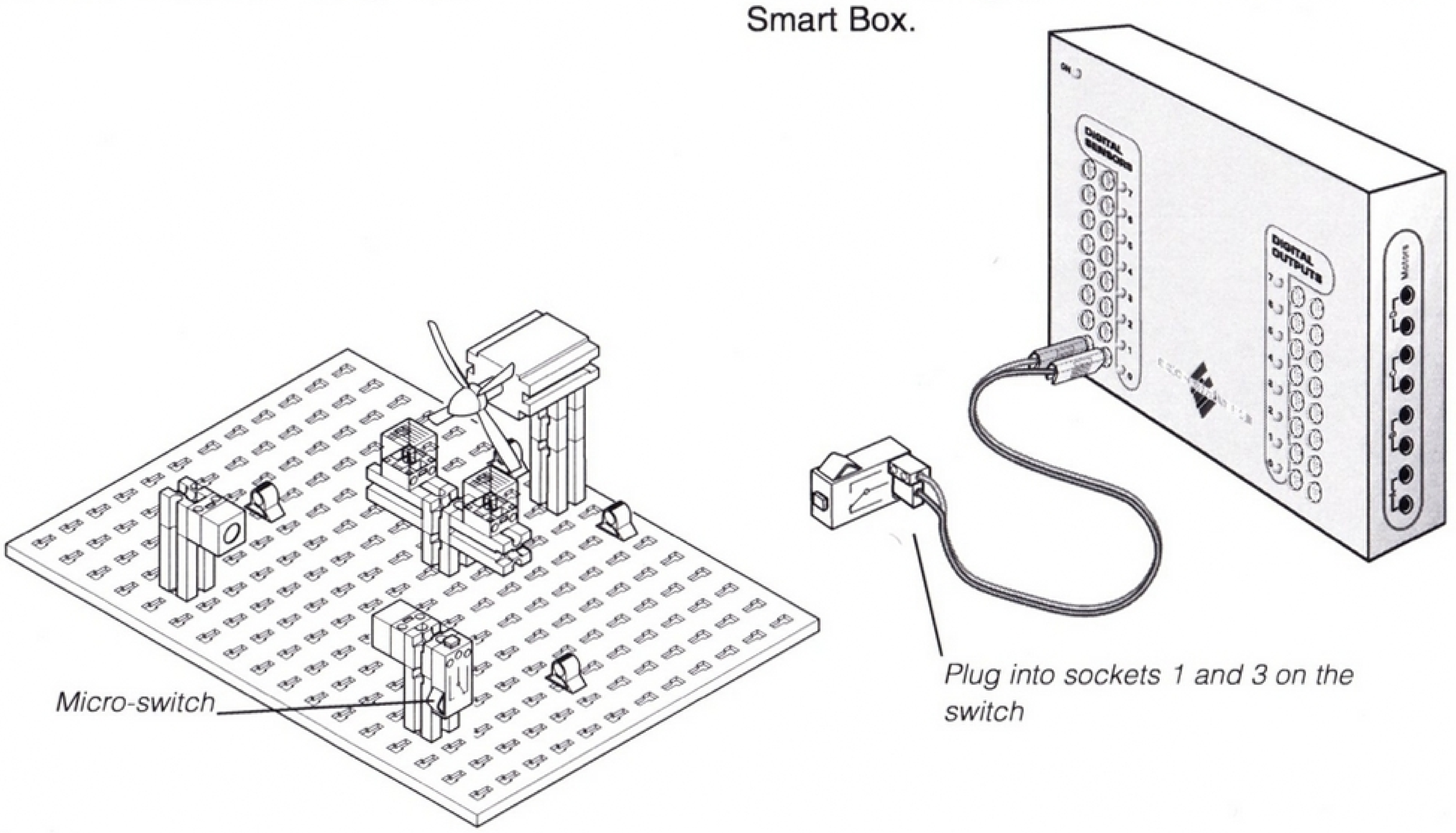


Fig. 7

Task Seven

This task shows how you can use the input signal from a switch to start a sequence.

Build the procedure shown in fig. 8. When you have built the procedure, run it to test it.

Procedure: input
wait until sensor 0 is on switch on 0,1 wait 2 switch off 0,1

Fig. 8. This procedure will switch on both lamps when you press the switch on the model.

Task Eight

Each time you want to use the switch to start the light sequence, you have to re-run the procedure.

It would be more efficient to add a repeat loop, so that the procedure returns automatically to check the switch again when the sequence has stopped.

Edit your procedure so that it does this.

Run the procedure to test it.

START UP 4 : Using input signals for motor control

Task Nine

This task shows how you can use the input signal from a switch to start or stop a motor. Build the procedure shown in fig. 9. When you have built the procedure, run it to test it.

Procedure: input2

```
wait until sensor 0 is on
forward a
wait 3
halt a
```

Fig. 9. This procedure will start the motor when you press the switch.

Before you do Task Ten, follow these instructions to set the motor so that it provides the correct output for a Hand Dryer:

1. Motor Direction

The fan can turn in two different directions. In one direction, it will blow air. In the other direction, it will draw air in. The hand dryer needs to blow air out over the lamps.

Run your procedure and check which way the fan is turning. Use the command that will make the fan blow air out over the lamps.

2. Motor Speed

You use the SPEED command to run the motor at different speeds. The command FORWARD A runs the motor at full speed. The command FORWARD A SPEED 5 runs it at approximately half speed. You can use any number between 1 and 9 to vary the speed (0 switches it off; 10 is full speed).

Edit your procedure so that the fan turns at a medium speed.

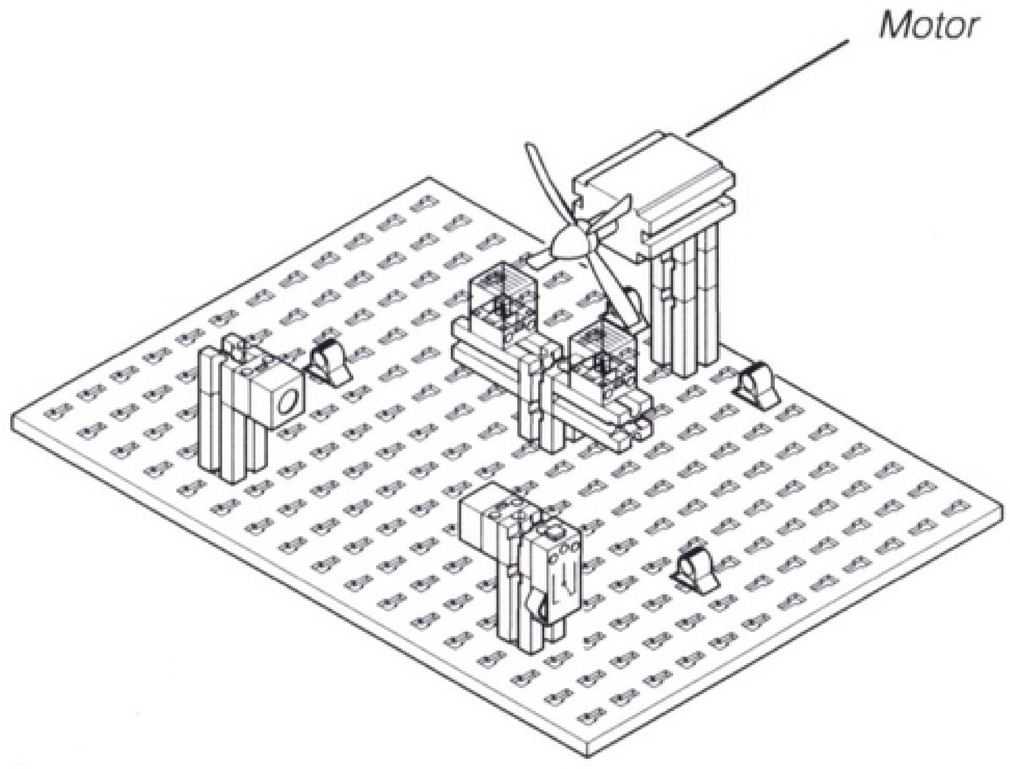
Task Ten

Now edit your procedure to look like fig. 10. This procedure will start the motor immediately and stop it when you press the switch.

Procedure: input2

```
forward a
wait until sensor 0 is on
halt a
```

Fig. 10. This procedure will start the motor immediately and stop it when you press the switch.



START UP 5: Design your own system

Now, use what you have learned about Smart Move to design and test a control system for the Hand Dryer that meets the following specification:

Specification

When you press the switch on the model, the fan and heater (red and orange lamps) must switch on.

They must stay on for 5 seconds to allow the user to dry his or her hands.

The fan and lamps must then switch off.

Run your procedure to check that it meets the specification.

Extensions

Edit your procedure so that it meets one or both of the following additions to the Specification.

Addition 1 to Specification

When the fan and lamps have switched off, you must be able to switch them on again by pressing the switch on the model.

Addition 2 to Specification

Add a green lamp to the model and connect it to Digital Outputs 2. Use the green lamp to indicate that the hand dryer is available for use. It must be switched on when the dryer is not on, and switched off when the dryer is on.

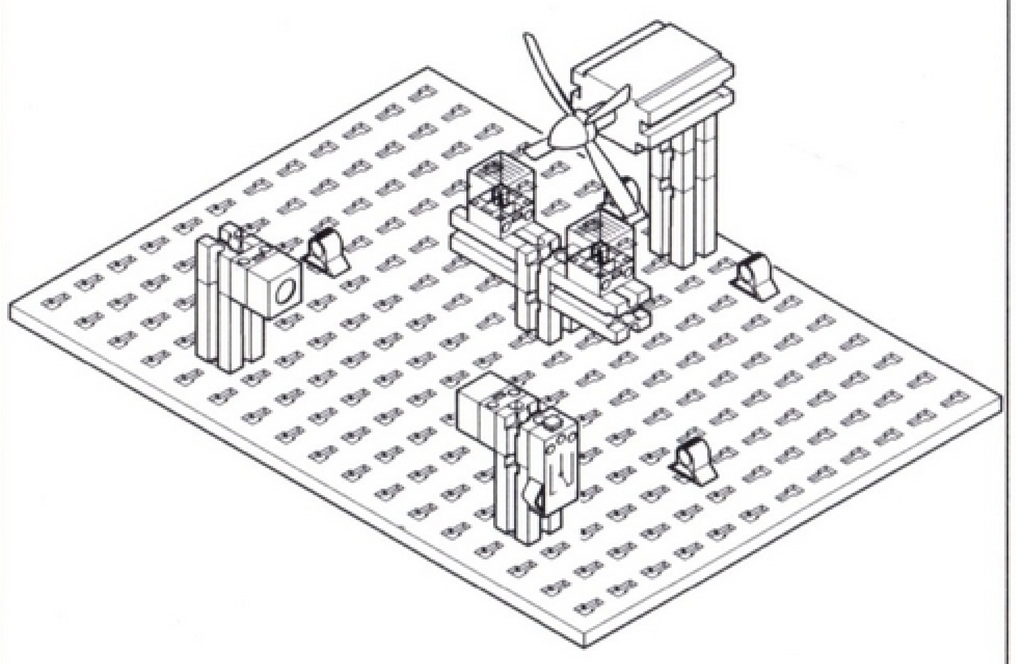
START UP 6 : Building a complex system

The control system that you have built for the dryer is already getting quite complicated. If it stays in the form of one procedure, it will become more and more difficult to follow as you add more features to it.

The most efficient and economical way to develop a more complex system is to build each section as a separate procedure. An advantage of building the system in this way is that it keeps the main procedure short, simple and easy to follow.

This section shows how you can use this method to develop a system to meet the following new specification:

New Specification



Use the green lamp that you added to the model to indicate that the hand dryer is available for use. It must be switched on when the hand dryer is not on, and switched off when the hand dryer is on.

When you press the switch on the model, the fan and heater (red and orange lamps) must switch on.

They must stay on until you press the switch again. They must then switch off.

When the fan and lamps have switched off, you must be able to switch them on again by pressing the switch on the model.

1. Begin by building the procedure shown in fig. 11. When you have built it, run it to test it.

Procedure: dryer
switch on 0,1 forward a wait 5 switch off 0,1 halt a

Fig. 11. This procedure switches on the fan and heater for 5 seconds, and then switches them off.

2. Build the procedure shown in fig. 12.

Procedure: main
repeat wait until sensor 0 is on dryer forever

Fig. 12. When the switch is pressed, the system automatically runs the procedure called "dryer".

Run the procedure called "main". This system meets part of the specification. Now you need to extend it to cover the complete specification.

3. Edit the procedure called "main" to include the green lamp (connected to Digital Outputs 2) as shown in fig. 13. Run the procedure to test it.

Procedure: main
repeat switch on 2 wait until sensor 0 is on switch off 2 dry forever

Fig. 13. The green lamp indicates that the dryer is not switched on.

4. Edit the procedure called "dryer" as shown in fig. 14 so that the dryer switches off automatically when you press the switch.

Procedure: dryer
switch on 0,1 forward a wait until sensor 0 is on switch off 0,1 halt a

Fig. 14.

Run the procedure called "main" to test your system. Check that it meets the New Specification. Do you notice a fault in the system?

5. Improving the System

A possible fault in the system is that the switch is being checked twice in quick succession. The software runs very quickly. If your finger is still on the switch when it is checked for the second time, the dryer will switch off again. Fig. 15 shows an improved version of the "dryer" procedure. It uses the command WAIT UNTIL SENSOR 0 IS OFF to check that you have stopped pressing the switch, before it checks it again.

Edit your "dryer" procedure to look like fig. 15.

Procedure: dryer
wait until sensor 0 is off switch on 0,1 forward a wait until sensor 0 is on switch off 0,1 halt a wait until sensor 0 is off

Fig. 15. Notice that the WAIT UNTIL SENSOR 0 IS OFF command is used twice.

Run the procedure called "main" to test the full system.

A change to the specification

Advertisements often appear near hand dryers because people drying their hands are a captive audience. If the hand dryer had a screen, different kinds of information could be communicated to the users, such as weather reports, traffic information or advertisements.

Edit your system to meet the following specification. Use the information below to help you.

Specification

When you press the switch on the model, the fan and heater must switch on.

When they have switched on, a message must appear on the screen for two seconds. This message must then disappear, and a new message must appear for two seconds and then disappear.

When the second message disappears, the fan and heater must switch off. You must be able to switch them on again by pressing the switch.

Help

To put a message on the screen while the procedure runs, use the command PRINT "type in the message you want to appear" The message must be put inside inverted commas.

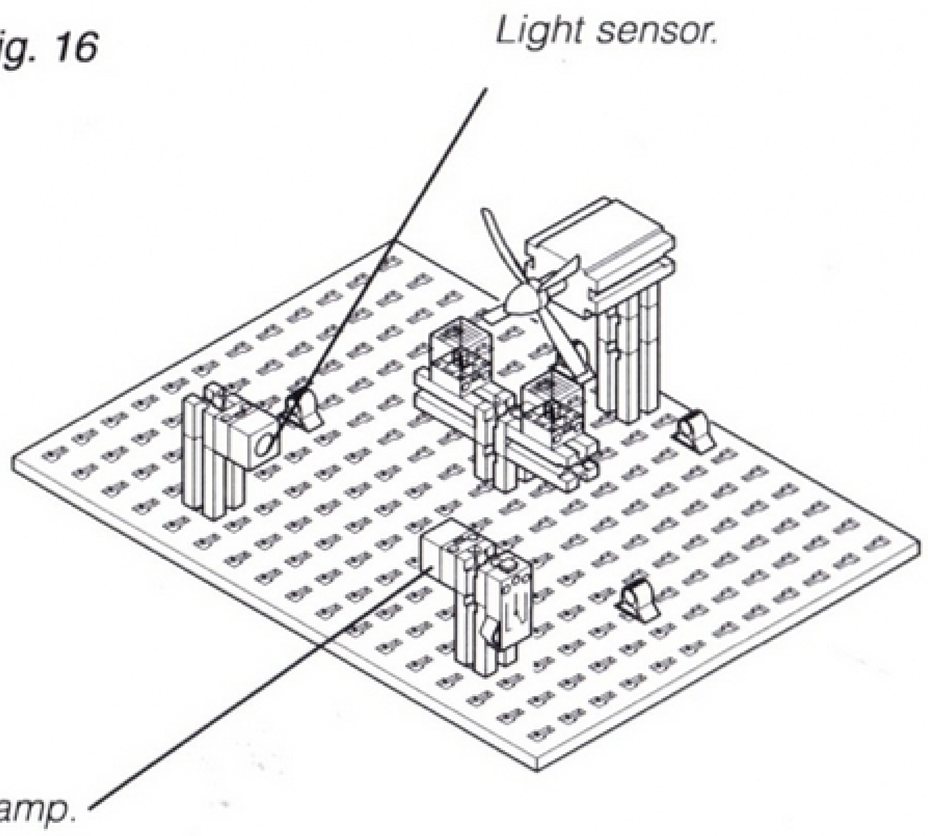
To remove a message from the screen use the command CLS. This stands for: clear screen.

START UP 7 : Responding to a sensor

The Hand Dryer model includes a light beam and digital light sensor (see fig. 16) which can be used to feed back information to the system by signalling that someone has put his or her hands under the dryer. This makes the system into an automatic hand dryer.

When light from the lamp shines on the digital light sensor, the sensor is on. When somebody blocks the light, the sensor switches off.

Fig. 16



The Help Sheet shows how to enter direct commands. Use them to test the light beam and sensor as described in the following instructions.

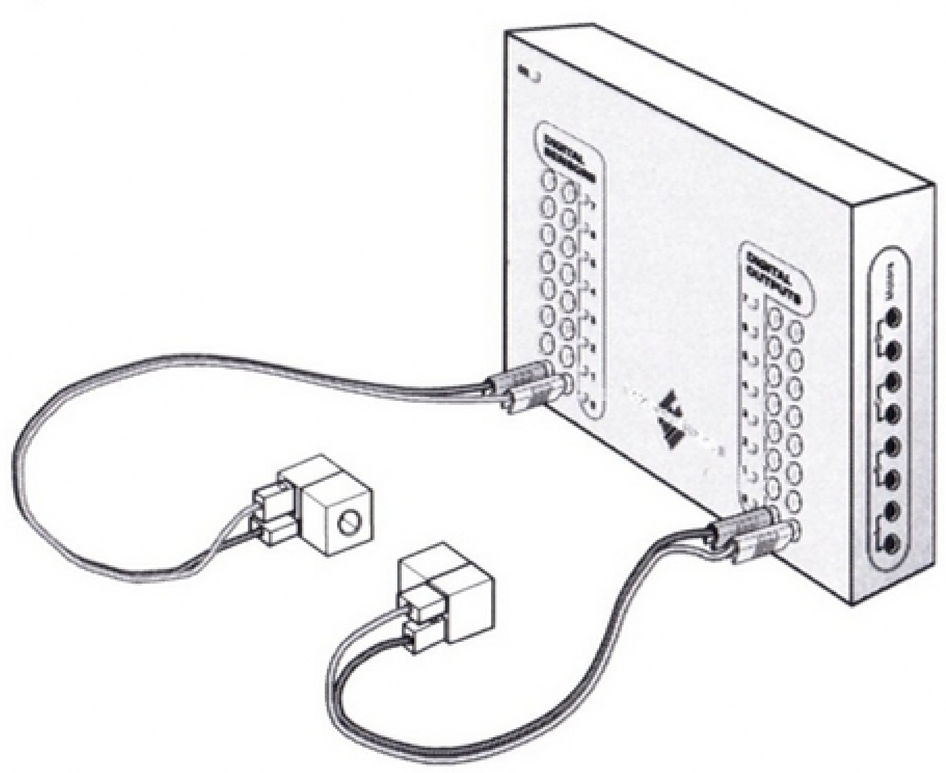
1. Enter a direct command: *switch on 3*.

Look at the Smart Box. Two indicator lamps should be on - the green indicator light by the socket connected to the lamp, and the yellow indicator light by the socket connected to the light sensor.

2. Put your hand between the lamp and the sensor. The yellow indicator light should switch off. If the sensor is not responding, try swapping over the two plugs that connect it to the Smart Box.

3. Enter a direct command: *switch off 3*.

Connect:
The lamp to Digital Outputs 3
The light sensor to Digital Sensors 1



Build the two procedures shown in fig. 17. This is a basic control system for the model. It switches on the dryer for 5 seconds when you break the light beam.

Procedure: main

```
switch on 3
wait 0.5
repeat
if sensor 1 is off then dryer
forever
```

Procedure: dryer

```
switch on 0,1
forward a
wait 5
switch off 0,1
halt a
```

Fig. 17. Basic system using the light sensor. The command WAIT 0.5 is used to allow time for the lamp to shine on the sensor before the procedure checks for it to be off.

Now edit your system so that it meets the following full specification:

Specification

Use the green lamp that you have added to the model to indicate that the hand dryer is available for use. It must be switched on when the dryer is not on, and switched off when the dryer is on.

When you put your hand between the light and the light beam on the model, the fan and heater (red and orange lamps) must switch on.

They must stay on until you remove your hand. They must then switch off.

When the fan and lamps have switched off, you must be able to switch them on again by breaking the light beam.

Extension

The system that you built in Section 6 could waste energy if users don't bother to switch off the dryer after they have used it.

Build a system to meet the following specification. Use the information in the help panel to help you.

Specification

When you press the switch on the model, the fan and heater must switch on.

They must switch off either when the switch is pressed again, or after ten seconds.

Help

1. OR logic

One WAIT UNTIL..... command can test two conditions, so that the procedure will continue when either one or the other happens.

e.g. wait until sensor 6 is on or sensor 7 is on

The procedure will continue if either of the sensors is on.

2. TIME

The procedure shown in fig. 18 shows how to use the TIME command. It will switch on the red lamp for 5 seconds, and then switch it off. Build the procedure and test it.

Procedure: timer

```
reset clock
start clock
switch on 0
wait until time > :5:00
switch off 0
```

Fig. 18

Note that the time value must begin with a colon. Other examples of time values:

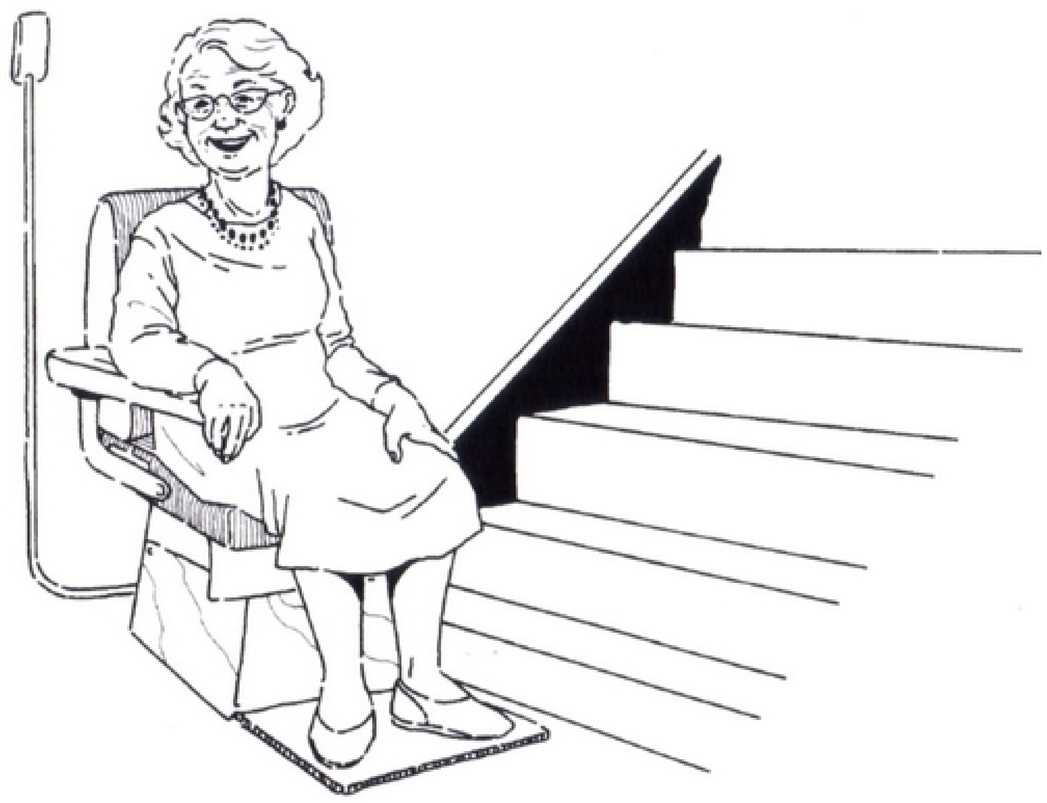
:2:00:00 = 2 minutes, 0 seconds,0 centiseconds
:30:50 = 30 seconds, 50 centiseconds
:25 = 25 centiseconds (a quarter of a second)

CONTROLLING MOVEMENT 1 : Stairlift

A motorised stairlift makes life much easier for people who have difficulty walking up and down stairs.

The output of the stairlift system is the movement of the chair. The input is provided by the push buttons fixed on the arm of the chair. The passenger uses these buttons to select whether the chair moves up or down the staircase.

Fig 1 shows how the control system for the stairlift could be sketched out as a flow diagram.



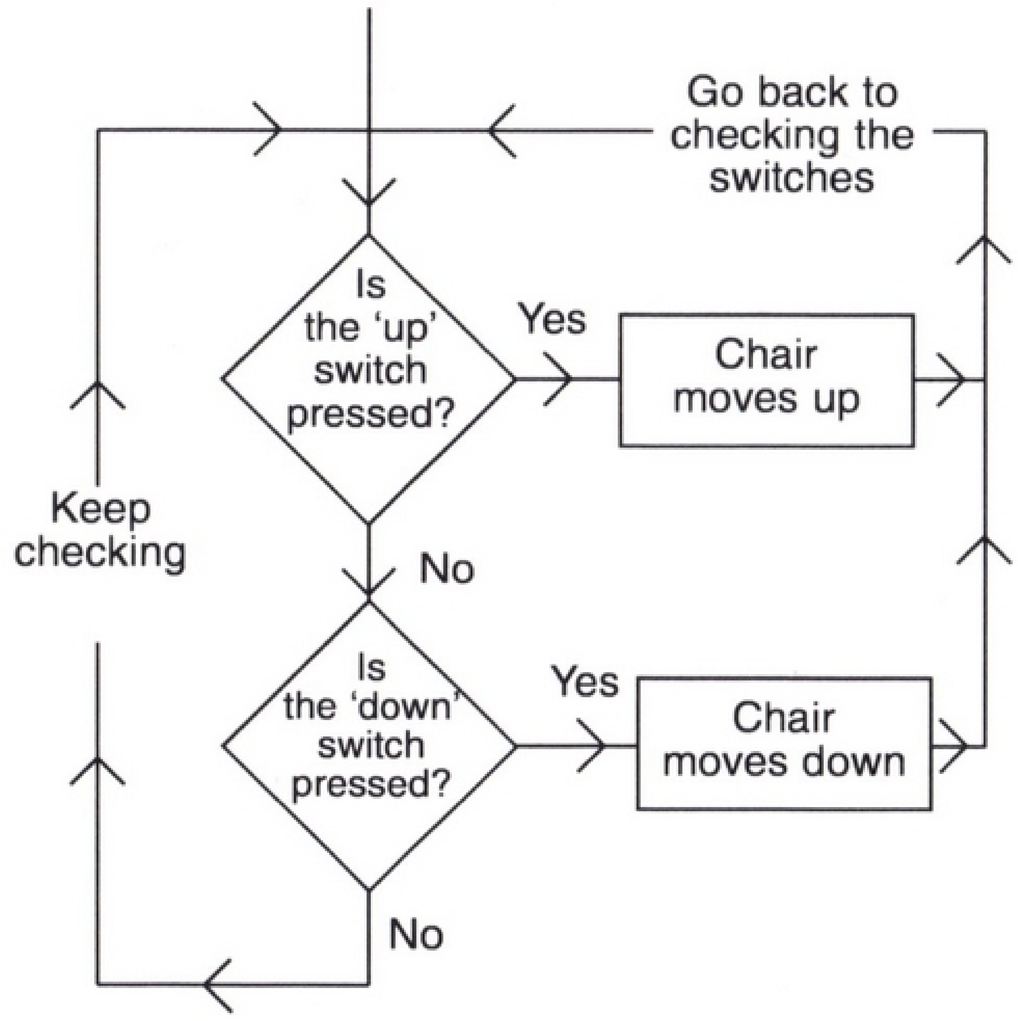
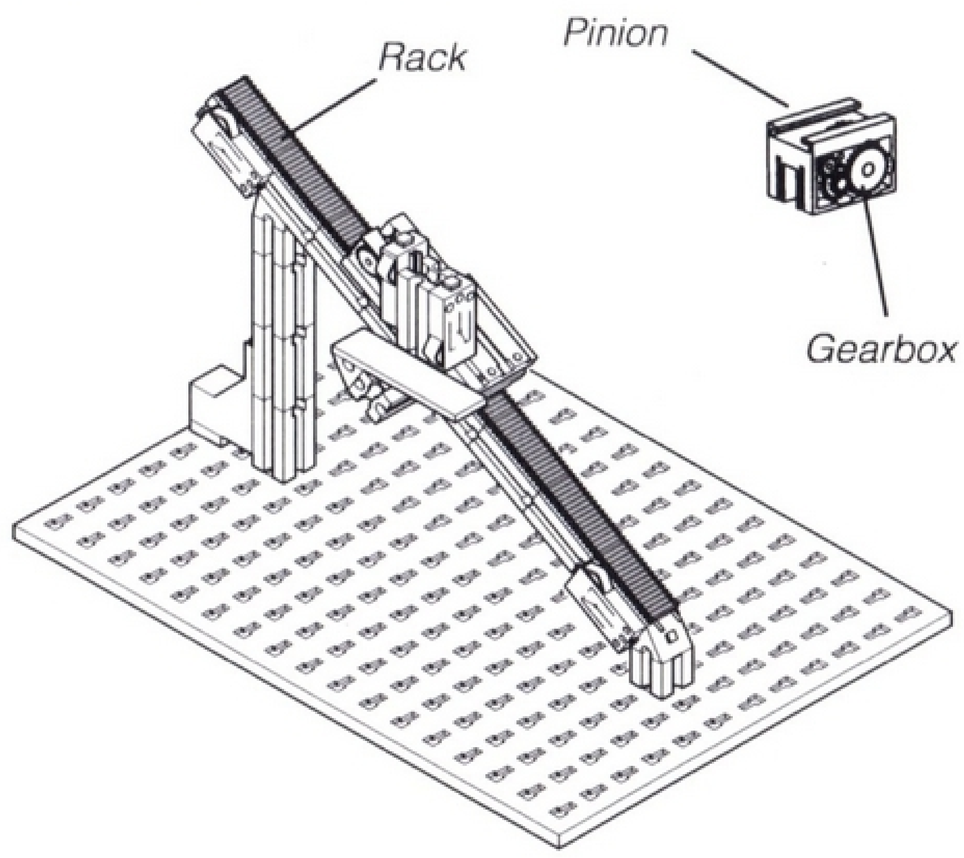


Fig 1

Build the model of the stairlift shown on the Rack and Pinion 1 construction sheet. Connect the motor and switches to the Smart Box as shown on the sheet.

Rack and Pinion 1

The mechanism of the stairlift model is made up of two mechanical systems : the gearbox and the rack and pinion.



Rack and Pinion 1

The input of the gearbox system is the fast rotary movement of the worm drive on the motor. The output is the much slower rotary movement of the gear that meshes with the rack. This gear is called a pinion.

The input of the rack and pinion system is the rotary movement of the pinion. The output is the linear movement of the motor and gearbox up and down the rack.

Limit Switches

Switches 1 and 2 on the model are the limit switches (see fig 2). They feedback information to the system about the position of the chair. For example, when the gearbox presses switch 2, the chair is at the top of the staircase.

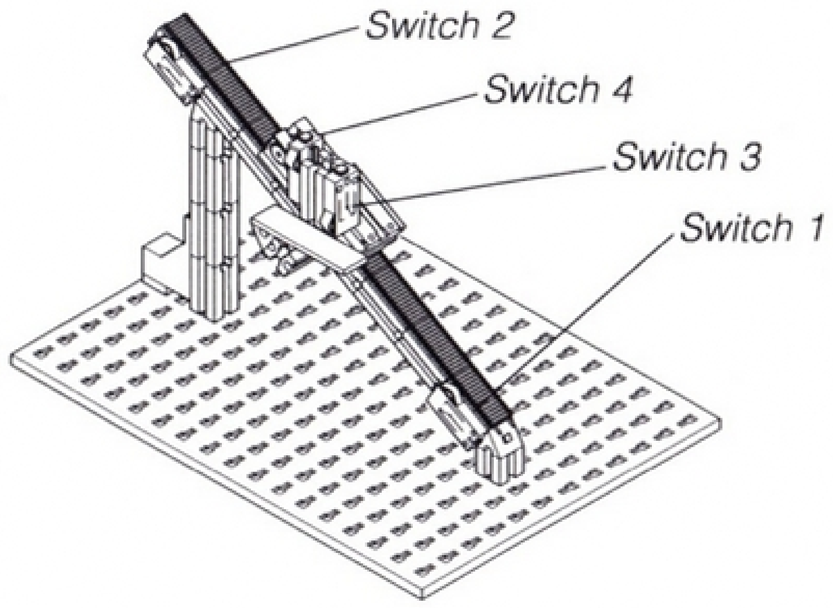


Fig 2.

So the way to control the movement of the stairlift is to switch on the motor and then wait until the limit switch is pressed. When it is pressed, then switch off the motor.

Build the procedure shown in fig 3 and test it to check that it moves the chair to the top of the staircase and then stops.

```
Procedure: up

forward a
wait until sensor 2 is on
halt a
```

Fig. 3

The Complete System

The two switches fixed to the chair (switches 3 and 4) can be used to select which direction the lift moves in. The complete system needs to check these two switches and carry out the appropriate up or down procedure if one of them is pressed. Fig 4 shows how this can be done.

```
Procedure: main

repeat
if sensor 3 is on then down
if sensor 4 is on then up
forever

Procedure: down

backward a
wait until sensor 1 is on
halt a

Procedure: up

forward a
wait until sensor 2 is on
halt a
```

Fig 4.

Fig 4. Notice that a repeat loop is used so that:

i) the switches are checked all the time when the chair isn't moving
ii) the sytem goes back to checking the switches as soon as the chair has reached its destination.

Extending the System

1. Add a red light and a green light to the model. Develop the control system so that the red light shines while the chair is moving. When the chair stops, the red light should switch off and the green light should switch on to show that it is safe to get off.

2. Redesign the system so that at any time on the upward journey, the passenger can press the "down" switch and go back to the bottom. At any time in the downward journey, they can press the "up" switch and go back to the top.

CONTROLLING MOVEMENT 2 : Packing System

In the packing department of a sweet factory, boxes are sealed and printed with a `best by' date on an indexing table. The table moves round step by step to bring each open box under the sealing machine in turn.

Fig 1 shows the control system required. The most efficient way to build the system is to take each section separately.

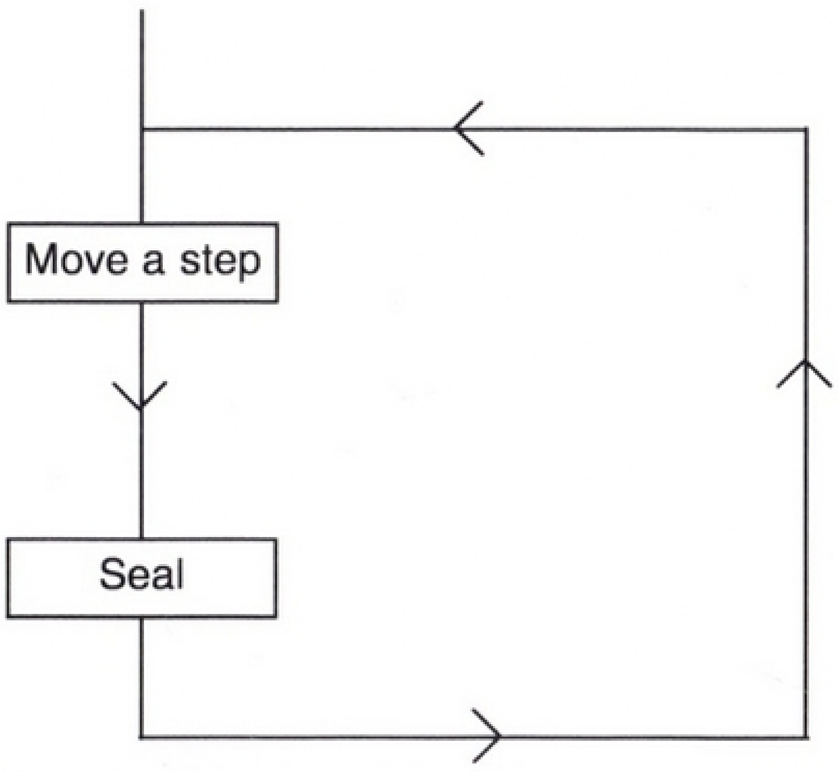
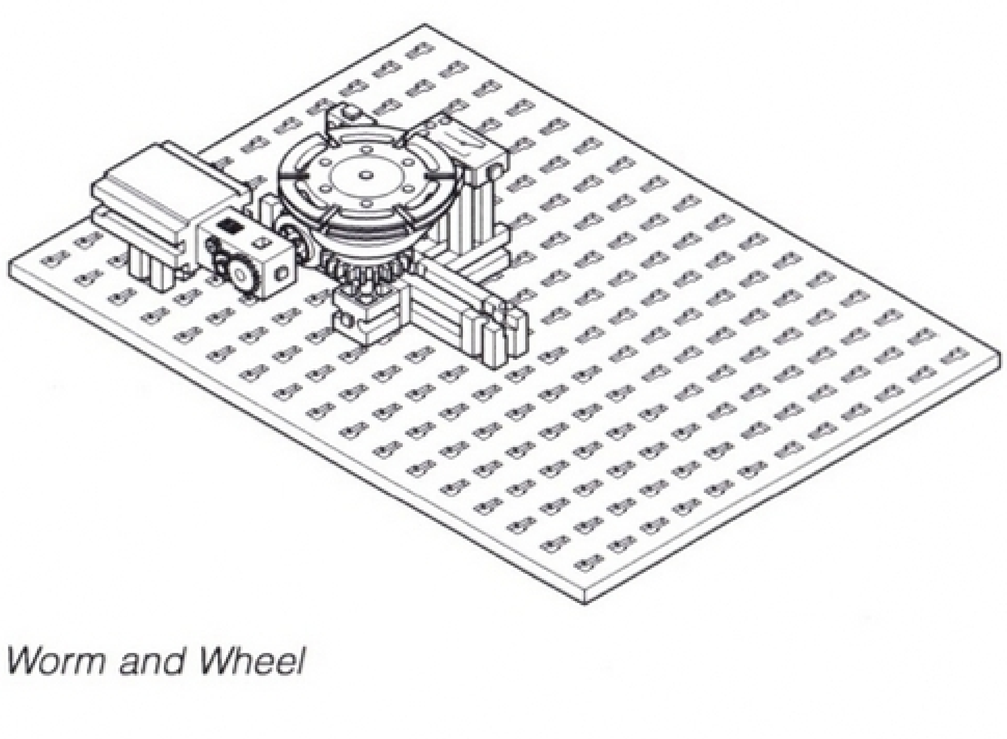


Fig. 1

1. Move a Step

Build the model of the turntable shown on the Worm and Wheel construction sheet. Connect it to the Smart Box as shown on the sheet.

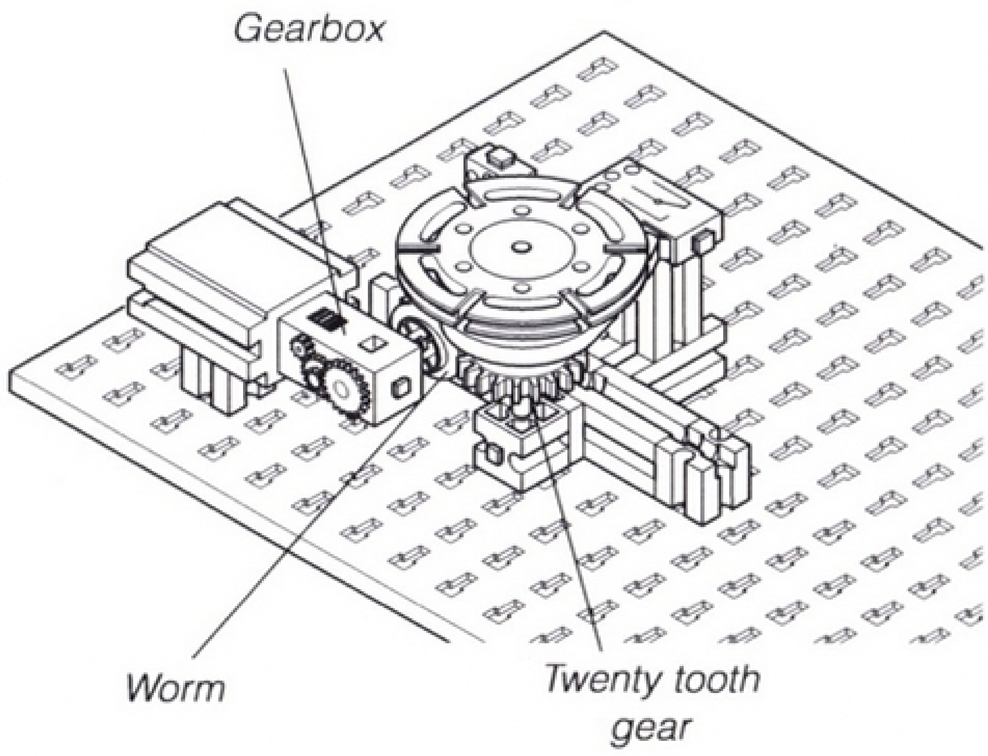


Worm and Wheel

Worm and Wheel

The mechanism of the turntable combines:

i) A gearbox which reduces the speed of movement and increases the torque provided by the motor,



ii) A worm and wheel. The input of this system is the rotary movement from the gearbox. The output is the rotary movement of the turntable. The worm and wheel (the process) changes the direction of the movement through 90°. The worm is really a one-tooth gear so when it is used with a twenty-tooth gear as on this model, speed is reduced by a ratio of 1:20. There is a corresponding increase in torque.

Control System for the Turntable

The turntable has six slots in it. Each box is held in one of these slots. Therefore each step in the system must be the distance between two slots.

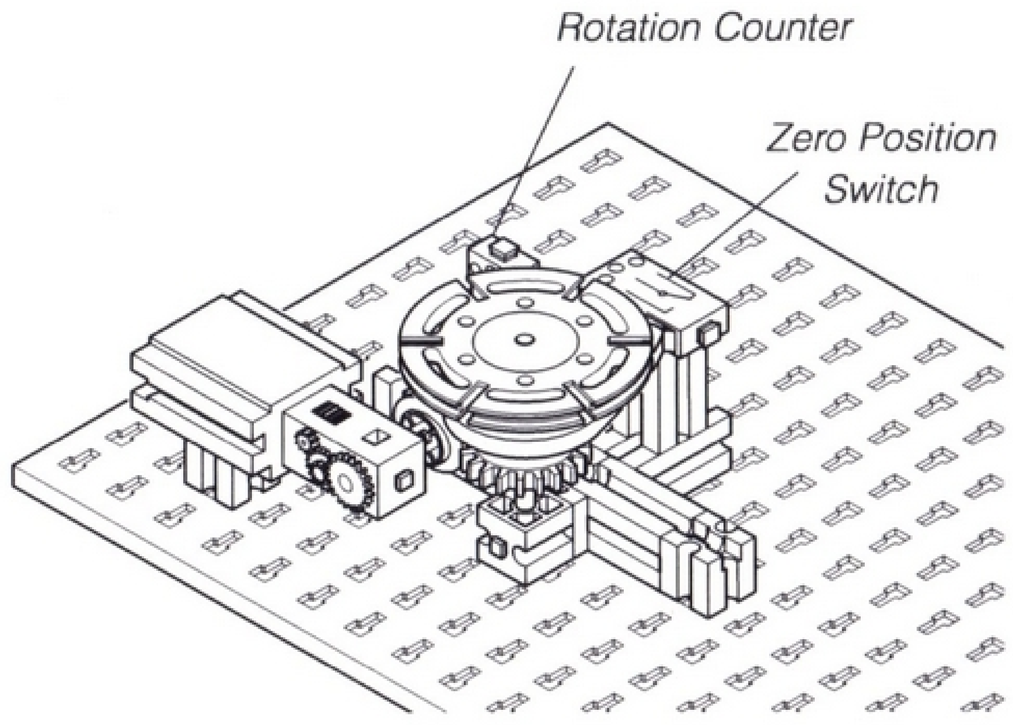
The system needs information about the position and movement of the turntable. This information is provided by a rotation counter and a zero position switch.

The rotation counter is made up of a switch and a pulsing wheel. The pulsing wheel is connected to the motor, so that as the motor rotates, the wheel pulses the switch. This sensor provides information about how far the turntable has moved.

Smart Move's COUNT command is used to count the number of pulses on the switch, so that the motor can be stopped when a set number have been counted.

The zero position switch is used to provide information about the position of the turntable. When the cam presses this switch the system can tell that the turntable is in its start position.

Build the procedures shown in fig 2. The "zero" procedure sends the turntable to its start position. Use the information in the Help panel to build the "step" procedure.



Procedure: zero

```
backward a
wait until sensor 1 is on
halt a
```

Procedure: step

```
zero
repeat
forward a
wait until count 0=17
halt a
wait 1
forever
```

Fig 2

Help

The command "wait until count 0 = 17" counts 17 pulses on the rotation counter switch which is connected to Digital Sensors 0.

You may need to use a different number for the count to fit the way your model runs.

Each COUNT command counts pulses from a particular input. The commands for inputs 0, 1, 2, 3 are linked with Motor Outputs A, B, C, D respectively, so that any command given to that motor will automatically reset the appropriate count to 0. This can be very useful because it means you don't have to keep resetting the count in the procedure.

2. Seal

Build the model shown on the Rack and Pinion 2 construction sheet. Fix it on to the baseplate by the turntable and connect it to the Smart Box as shown on the sheet.

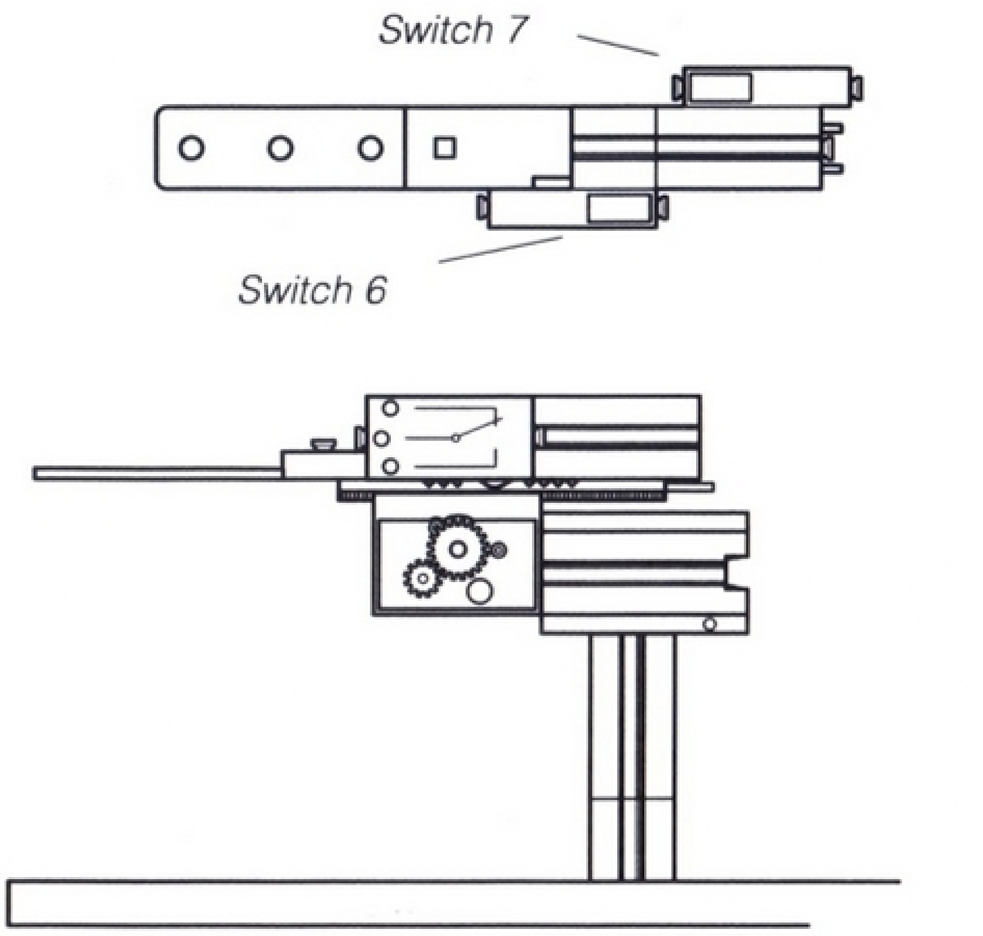
Arrange the model so the slider will move out and back over a slot in the turntable to simulate the sealing machine sealing a box held in the slot.

Although it uses the same mechanism as the stairlift (Rack and Pinion 1), the Rack and Pinion 2 model is different because the motor and gearbox are fixed and the rack moves. Limit switches are also used differently.

The motor moves the rack forward until switch 6 is no longer pressed. It moves it backward until switch 7 is no longer pressed. You can increase or decrease the distance the slider moves by moving the switches. If you need to move the rack by hand, ease the gearbox up to disengage it from the worm drive on the motor.

Build the "seal" procedure shown in fig 3. Run the procedure to test that this part of the system works correctly.

Then edit your "step" procedure and build the "main" procedure to make the complete packing system shown in fig 3.



Rack and Pinion 2

Extending the System

1. Add red and green indicator lamps to the model. Use the red light to show that the machine is switched on so operators must stay clear of moving parts. Use the green light to show that the machine is switched off.

2. Develop the system so it keeps a count of how many boxes have been sealed, and displays this information on the screen.

Procedure : main

```
zero
repeat
step
seal
forever
```

Procedure : zero

```
backward a
wait until sensor 1 is on
halt a
```

Procedure : seal

```
forward b
wait until sensor 6 is off
halt b
wait 1
backward b
wait until sensor 7 is off
halt b
```

Procedure : step

```
forward a
wait until count 0 = 17
halt a
```

Fig 3. Complete packing system.

CONTROLLING MOVEMENT 3 : Shooting Gallery

The novelty shooting gallery shown here has several targets. Each one contains a light sensor. The gun fires a light beam when the trigger is pressed. If the gun is aimed straight at a target, it switches the sensor on and a figure or object pops out from behind a part of the scenery.

Fig 1 shows a flow diagram sketch of the control system needed for the shooting gallery.

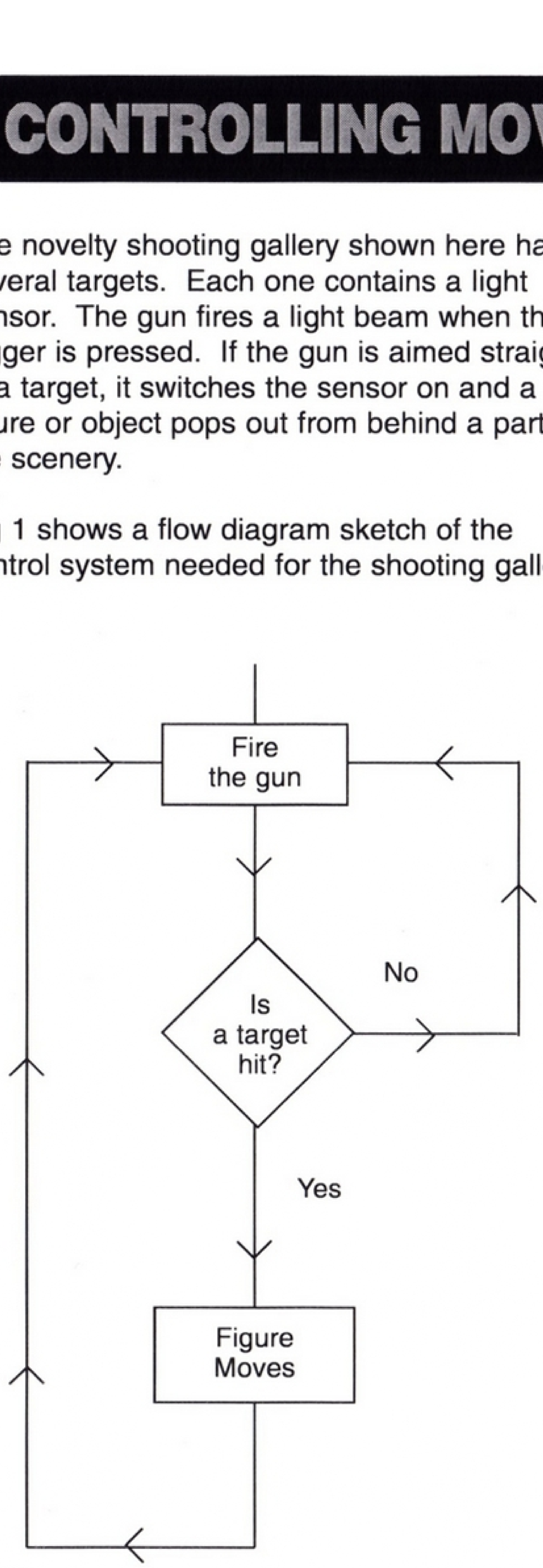
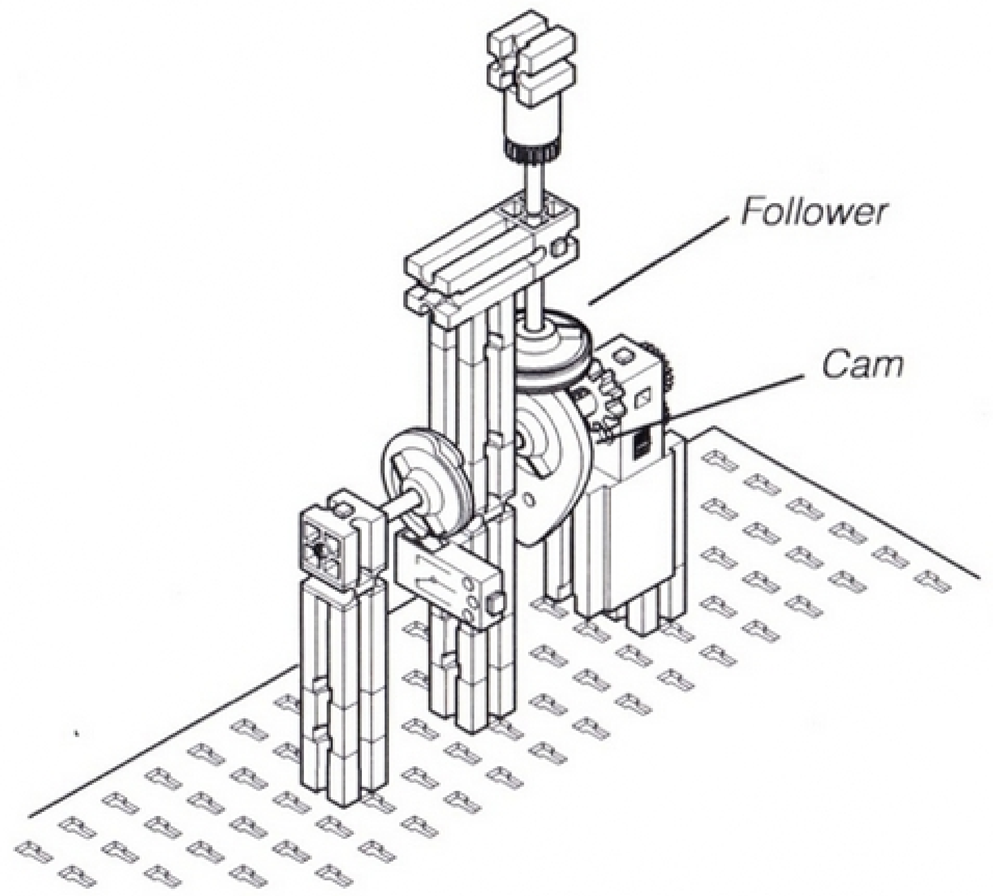
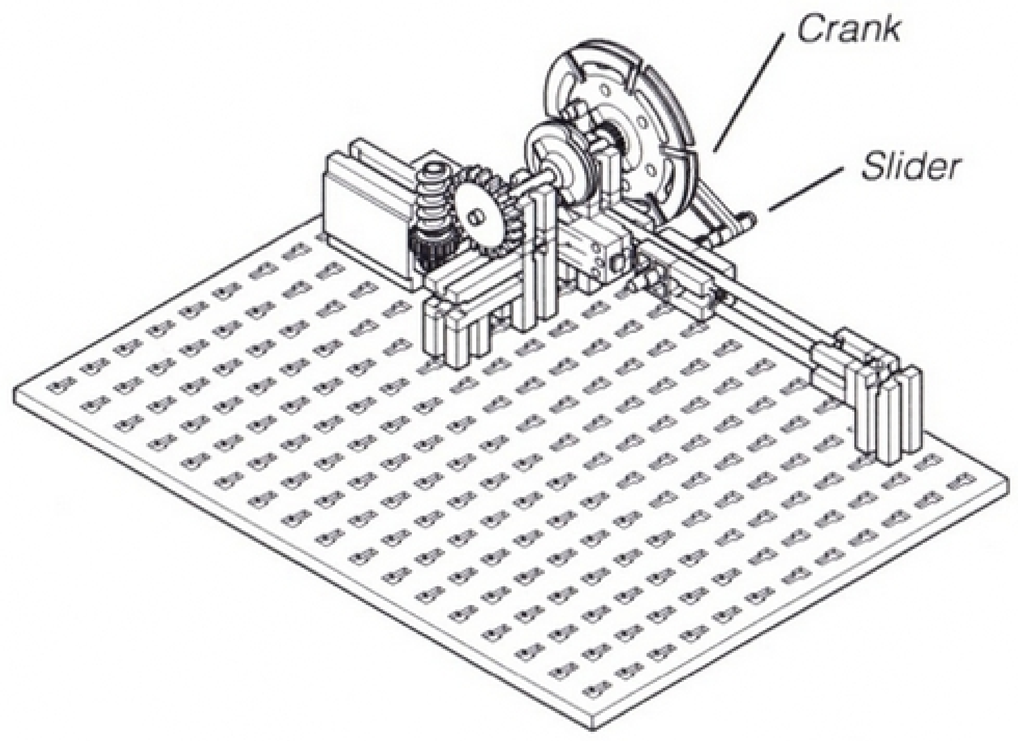


Fig. 1



Cam and Follower

Begin work on the system by looking at the output – the moving figures. They have to pop up or out when their target is hit and then return to their hiding place. The mechanisms shown on the Cam and Follower and Crank and Slider construction sheets will convert the rotary movement of an electric motor into the necessary reciprocating movement.



Crank and Slider

Build these two models and connect them to the Smart Box as shown on the sheet. Add red and yellow plates from the kit to simulate the figures and their hiding places.

Controlling Reciprocating Movement

Both the cam and follower and the crank and slider mechanisms are controlled by the same technique : a switch pressed by an index cam, as shown in fig 2.

Begin by controlling the cam and follower. Use the following instructions.

1) Make sure that the index cam on your model is set as shown in fig 2.

2) Use the Direct Command method to check which way the index cam turns for each setting of the motor. The forward command should turn it clockwise.

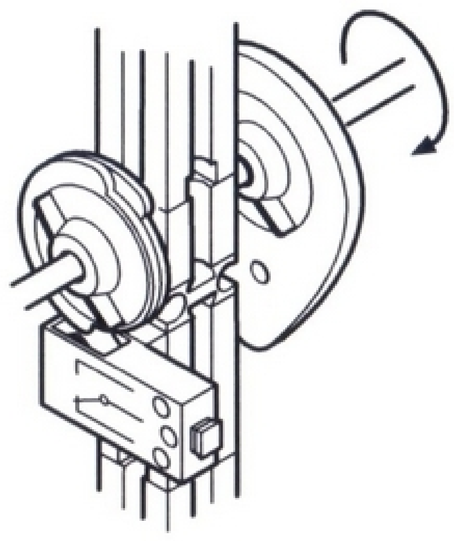
3) Write the procedure shown in fig 3.

When the motor is switched on it turns until the cam stops pressing the switch, and keeps turning until the cam presses the switch again.

This will make the figure move out and straight back.

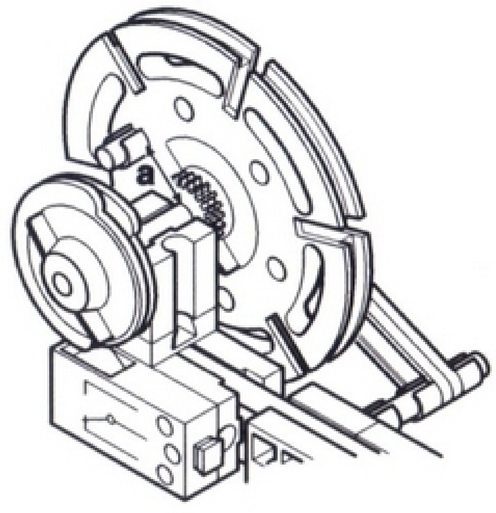
Now build and test a similar procedure to control the crank and slider (see fig 4) . These two procedures provide the output part of the shooting gallery system.

Fig. 2



Cam and Follower

The switch is just pressed when the follower is in its down position.



Crank and Slider

The switch is just pressed when the slider is in the back position.

Procedure : cam
backward b wait until sensor 1 is off wait until sensor 1 is on halt b

Fig. 3

Procedure : crank
backward a wait until sensor 0 is off wait until sensor 0 is on halt a

Fig. 4

The Input of the System

Use two light sensors with black masking caps attached as the targets in the shooting gallery. Add one to the baseplate of each model and connect them to Digital Sensors sockets 6 and 7 on the Smart Box as shown in fig 5. You could support them on angle blocks to make them easier or harder to hit.

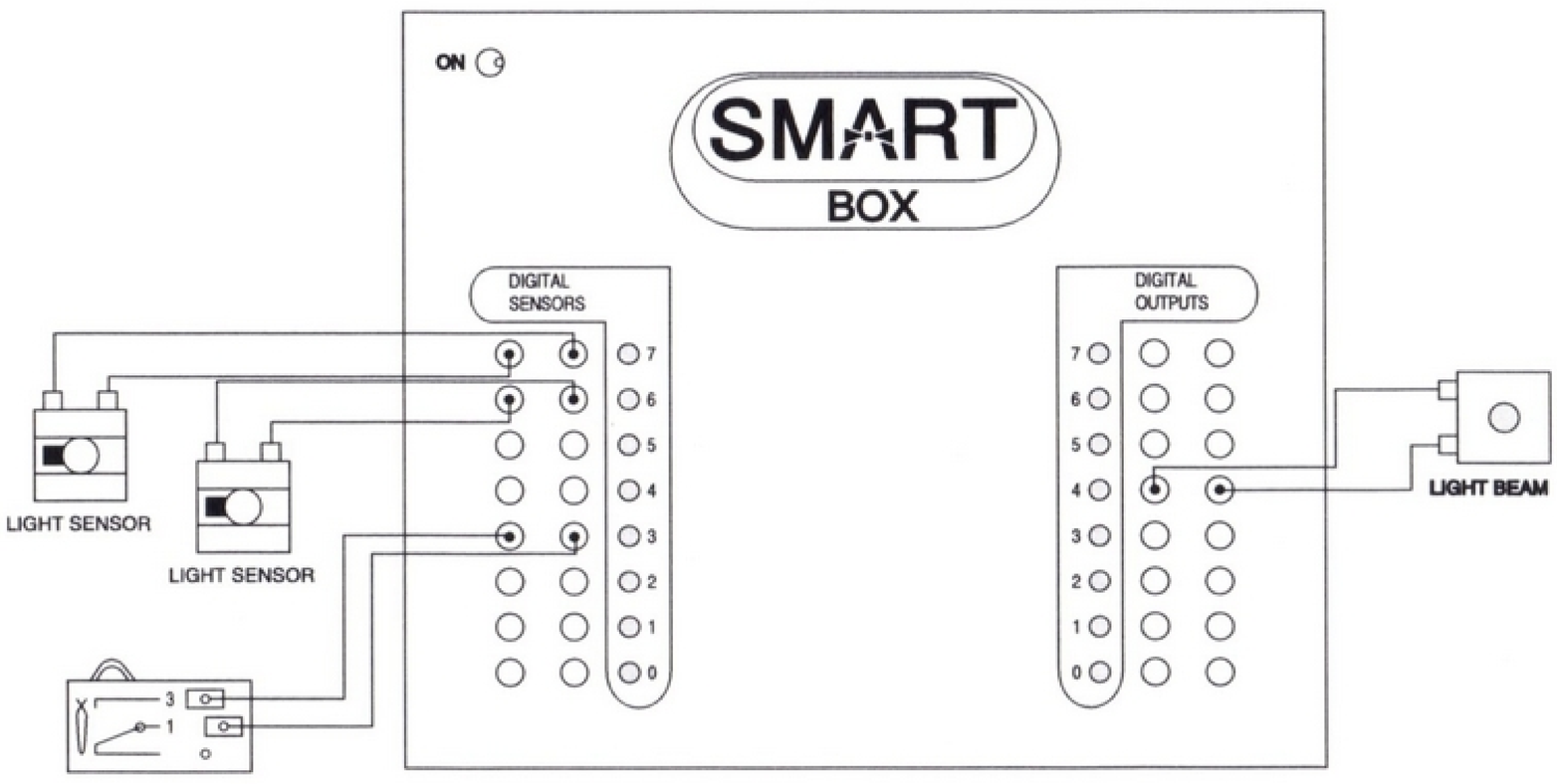


Fig. 5

All you need to make a gun are a switch (to act as a trigger) and a lamp to make a light beam to shine on the sensors. Connect them to the Smart Box as shown in fig 5. You could construct it as shown in fig 6, or you could design your own.

Build the procedure shown in fig 7. It is designed to wait until you press the trigger, and then switch on the lamp for half a second.

Procedure : shoot

```
wait until sensor 3 is on
switch on 3
wait 0.5
switch off 3
```

Fig. 7

Run the procedure, point the lamp at a sensor and press the trigger. If the beam shines on the sensor, the sensor's LED indicator on the Smart Box will light.

Check that both sensors are responding.

The Complete System

The final part of the system is the procedure that carries out the process part of the system i.e. it checks the input and chooses the appropriate output. Fig. 8 shows the procedures. It also shows how the "shoot" procedure should be edited to include it.

Extending the System

a) Add a coloured lamp to each model. Extend the system so that the lamp lights if its target is hit

b) Develop the system so that each player has only five shots. The gun must stop working after the trigger has been pressed five times. A message on the screen tells the player that their game is over.

The system could also keep a record of each time a target is hit, and display the player's score at the end of a game.

c) Extend the "cam" and "crank" procedures so that the figures pause for a short time when they are out.

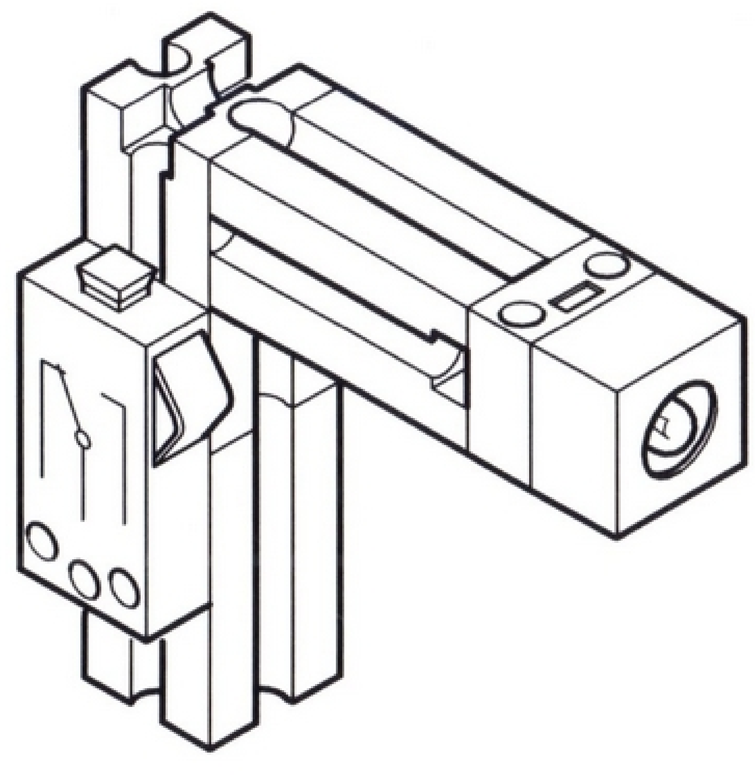


Fig. 6

Procedure : shoot

```
repeat
wait until sensor 3 is on
switch on 3
wait 0.5
check
switch off 3
forever
```

Procedure : check

```
if sensor 7 is on then cam
if sensor 6 is on then crank
```

Procedure : cam

```
backward b
wait until sensor 1 is off
wait until sensor 1 is on
halt b
```

Procedure : crank

```
backward a
wait until sensor 0 is off
wait until sensor 0 is on
halt a
```

Fig. 8

CONTROLLING MOVEMENT 4 : Theme Ride

At Wonderful Worlds theme park, a cliff railway takes passengers to different theme areas located on the side of a hill. They can choose to go to either: Space World, Fantasy World or Nature World.

Fig 1 shows a flow diagram of the control system needed for the ride.

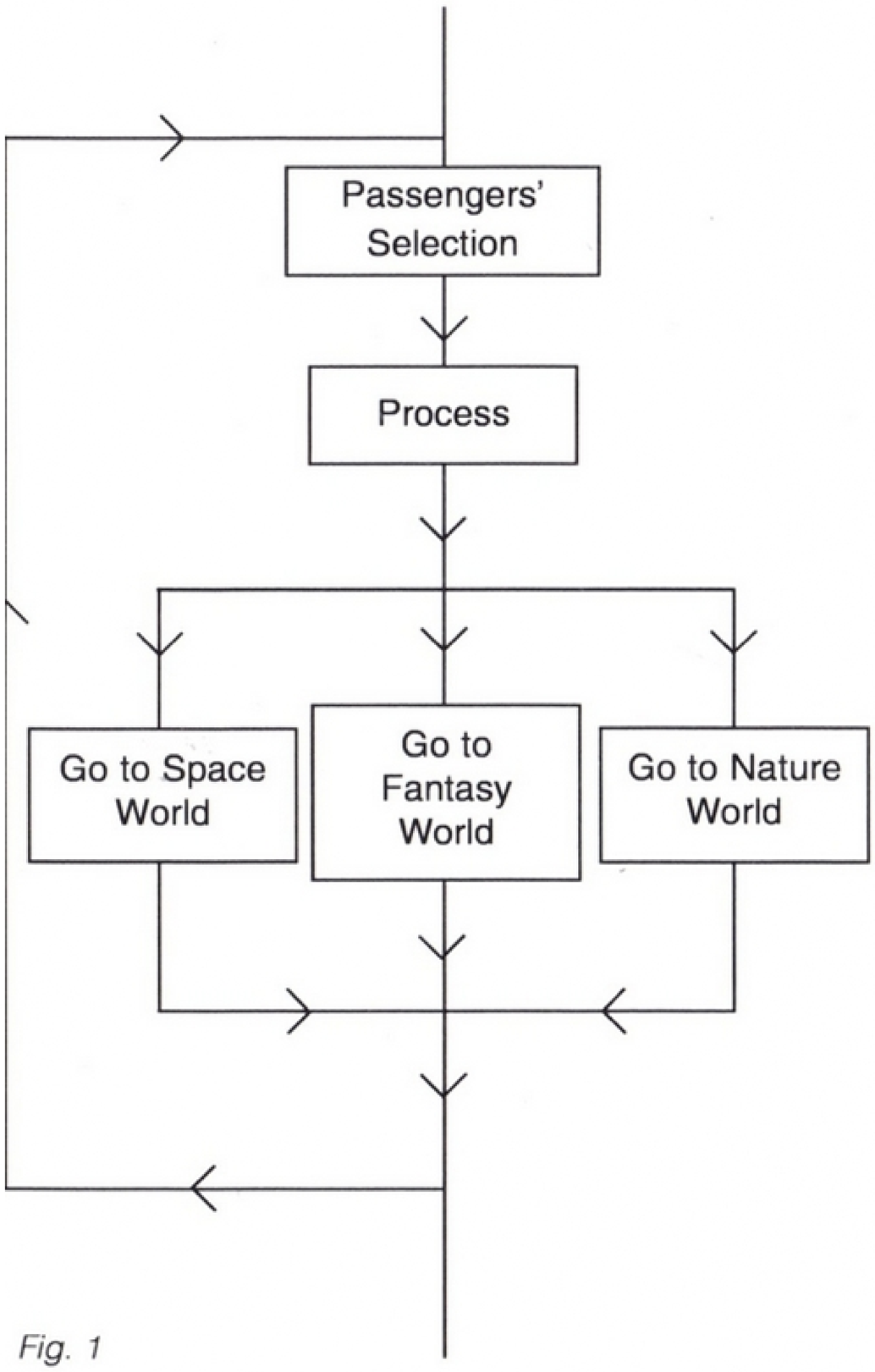
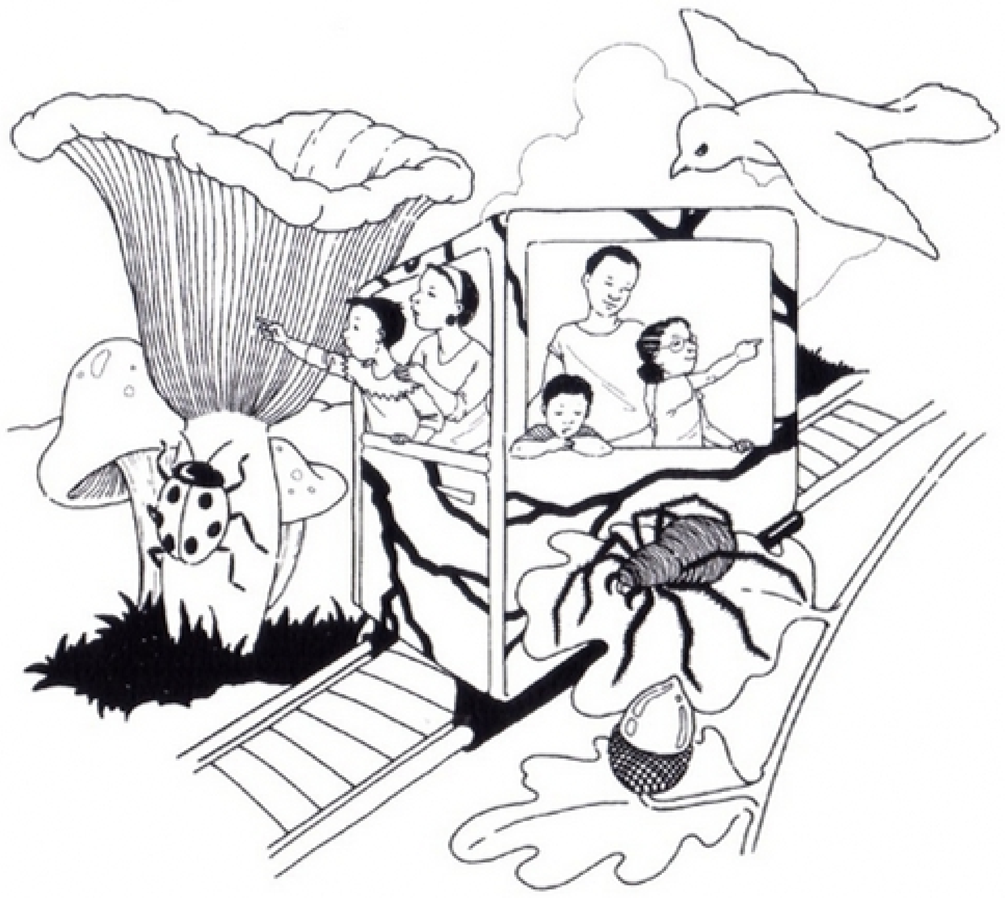


Fig. 1

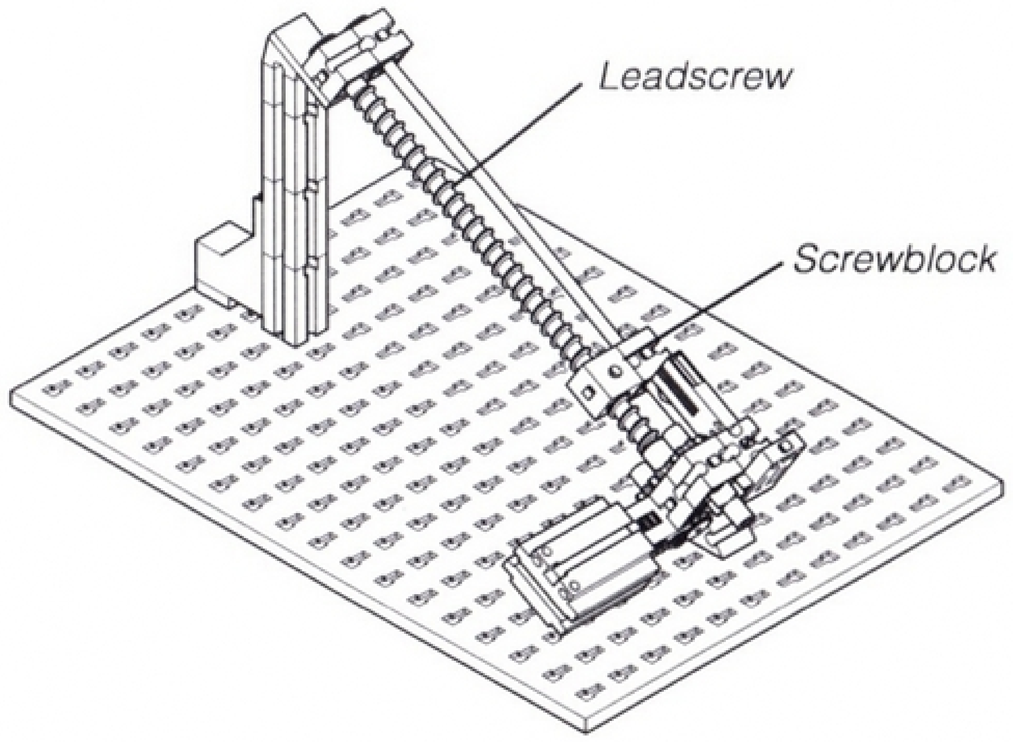
The passengers' choice of destination provides the input to the system. The process section takes the information provided by the input and uses it to select which of the three possible outputs is required. You can develop each section separately and put them together to make a complete system.

Build the model shown on the Leadscrew construction sheet. Connect the motor and switches to the Smart Box as shown on the sheet.



Leadscrew

This mechanism changes the rotary movement of the motor (input) into the linear movement of the screwblock (output).



Both a leadscrew and a rack and pinion change rotary to linear movement, but there are at least two differences between them:

i) The output of the leadscrew tends to be much slower.
ii) The pinion can run freely up and down the rack if the motor is disconnected. The leadscrew stays locked in place. This is because the leadscrew is not reversible. It cannot be used to change linear to rotary movement.

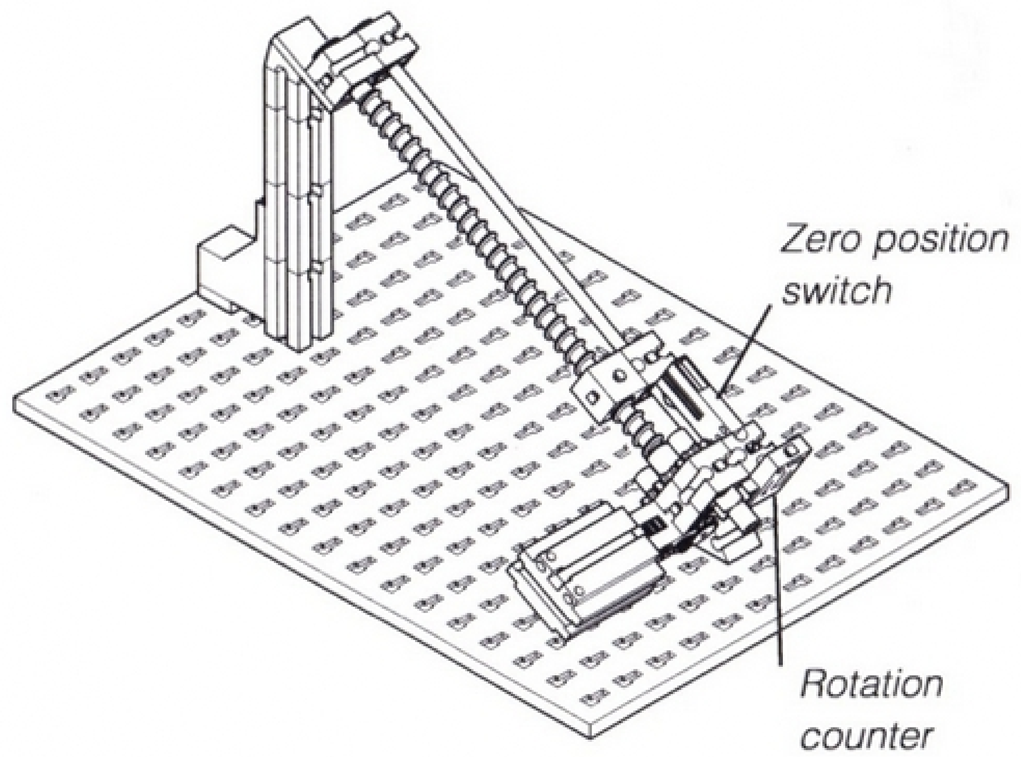
Control System for the Leadscrew

The Leadscrew uses the same method of control as the Turntable. A rotation counter provides information about how far the passenger car (the screwblock) has moved up or down the leadscrew.

A zero position switch is used to provide information about the position of the car. When the block presses this switch the system can tell that it is in its start position at the bottom end of the leadscrew.

A COUNT command can be used to count the number of pulses on the rotation counter switch, so that the motor can be stopped when a set number have been counted. The number of pulses needed to take the car to each destination is shown in fig 2.

Build the procedures shown in fig 3. Run each one in turn to check that it takes the car to the appropriate destination. Run the "zero" procedure between each one so that the car always starts from the zero position.



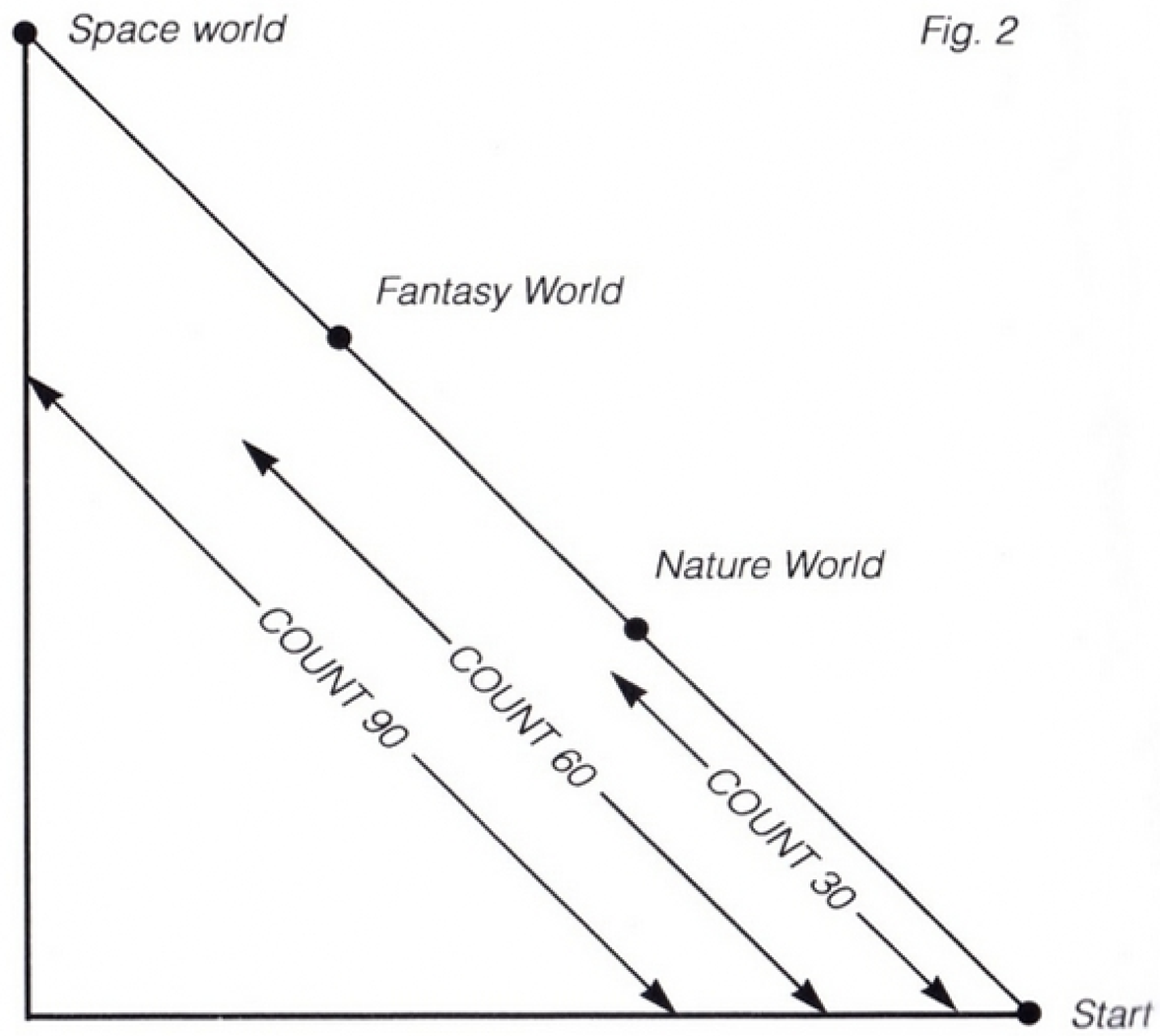


Fig. 2

Procedure : zero

```
back a
wait until sensor 1 is on
halt a
```

Procedure : nature

```
forward a
wait until count 0 = 30
halt a
```

Procedure : fantasy

```
forward a
wait until count 0 = 60
halt a
```

Procedure : space

```
forward a
wait until count 0 = 90
halt a
```

Fig. 3

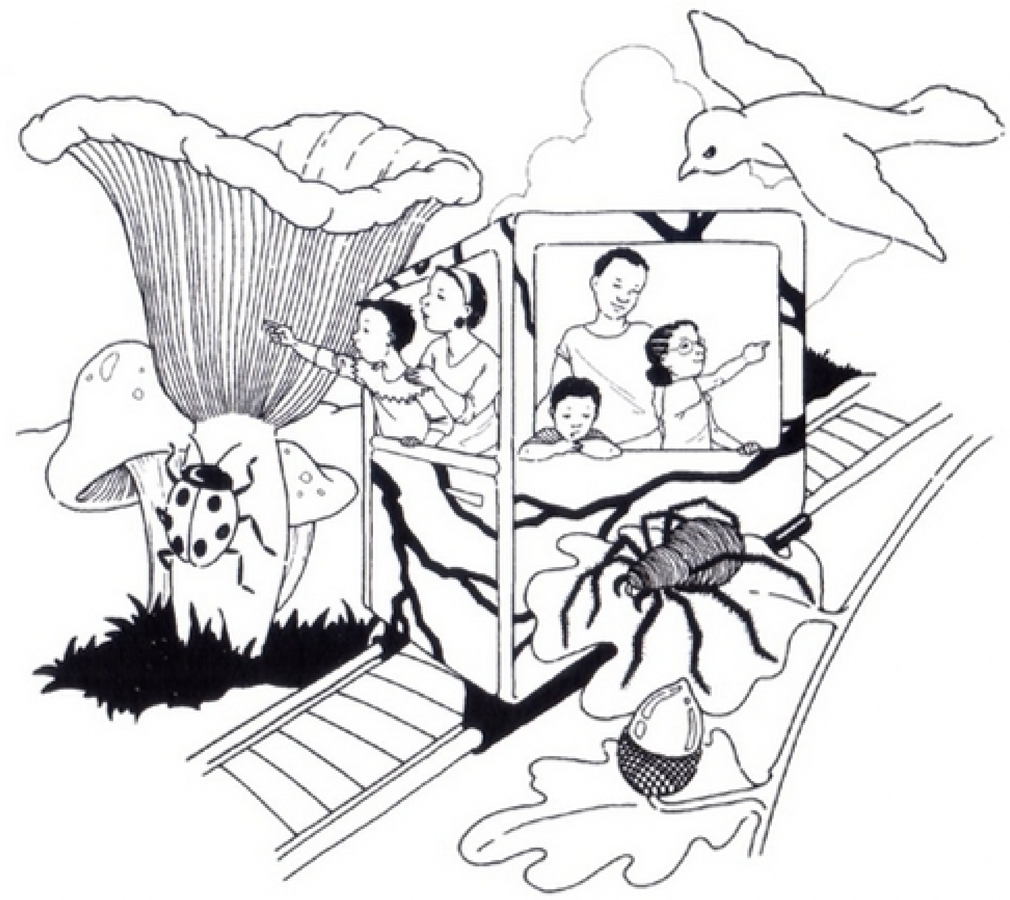
Input and Process

The input to the Theme Ride system is the passengers' selection of destination. This section of the system needs two parts:

i) a way for passengers to make their selection.

ii)Information to tell them how to do it.

Fig 4 shows how this can be written.



Procedure : main

```
repeat
print "enter 1 for nature world" ' "2 for fantasy world" ' "3 for space world"
ask n, "please enter your destination number",w
cls
process
forever
```

Fig 4

Smart Move's ASK command is used to allow passengers to make their selection. When the procedure runs, they are asked to enter an identification number for their destination. The information printed on the screen tells them which number to enter. When they have made their choice, the system carries out the process section.

The process section of the system takes the information provided by the input and uses it to select which of the three outputs is required. The passengers' selection is held in the variable w (for "World"). So, the "process" procedure checks the value of w and then carries out the appropriate output procedure. Build the "process" procedure shown in fig 5.

Procedure: process

```
if w = 1 then nature
if w = 2 then fantasy
if w = 3 then space
```

Fig 5

The Complete System

You have now created all the parts that make up the complete Theme Ride system. It contains the following procedures : main, process, nature, fantasy, space, zero.

Run "main" to operate the system. Check that it sends the car to the selected destination.

Extending the System

The Theme Ride System would be more user-friendly if passengers were informed when they had reached their destination. Use coloured lights and/or Smart Move's PRINT command to extend the system in this way.

An Alternative System

This system makes the ride more efficient by moving the car directly from one destination to another without having to return to the zero position before every trip. Read the following explanation, then edit your system to look like fig 6 and test it.

When a destination is selected, the system needs to know : i) which direction to move in, and ii) how many pulses to count.

Three variables are used to calculate this information :

w = the number of the selected destination. Enter 30 for Nature World, 60 for Fantasy World, 90 for Space World.

p = the number of the current position of the car. So, p is set to zero at the end of the "zero" procedure. When the car reaches a destination at the end of an "up" or "down" procedure, p is set to the number for that destination in the command "let p = w".

When the next destination is selected, if w is greater than p, the car must move up. If w is less than p, the car must go down.

m = the number of pulses the car has to move to go from its current position to the selected destination.

Procedure : main

```
zero
repeat
ask n, "please enter your destination number",w
if w > p then up
if w < p then down
forever
```

Procedure : zero

```
backward a
wait until sensor 1 is on
halt a
let p = 0
```

Procedure : up

```
let m = w-p
forward a
wait until count 0 = m
halt a
let p = w
```

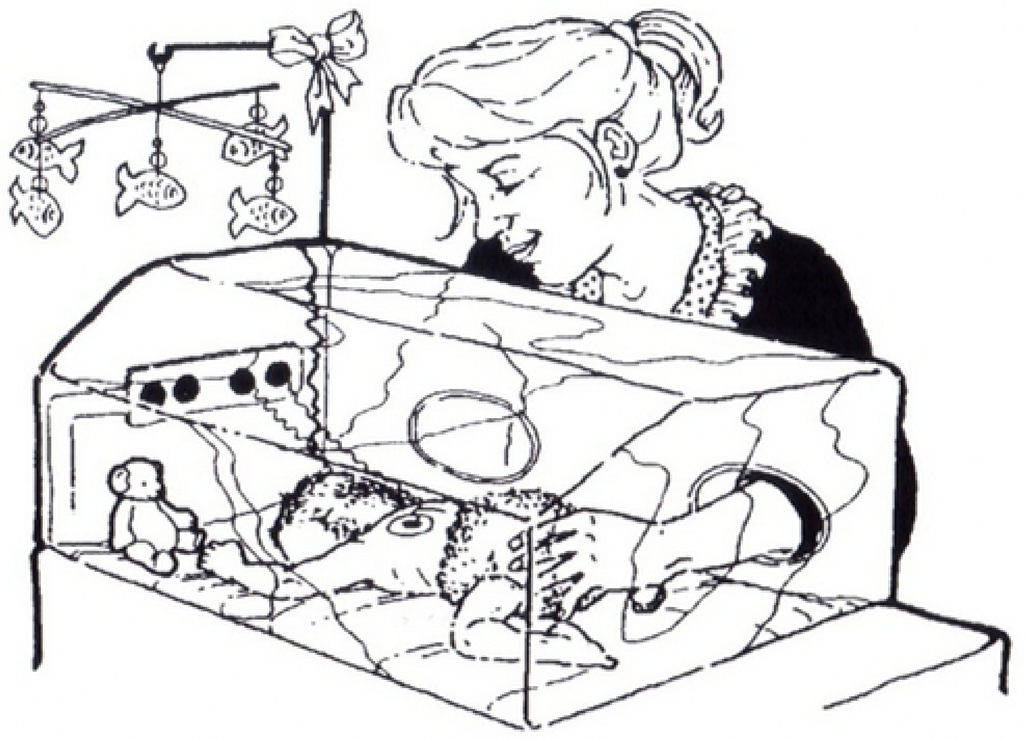
Procedure : down

```
let m = p-w
backward a
wait until count 0 = m
halt a
let p = w
```

Fig 6

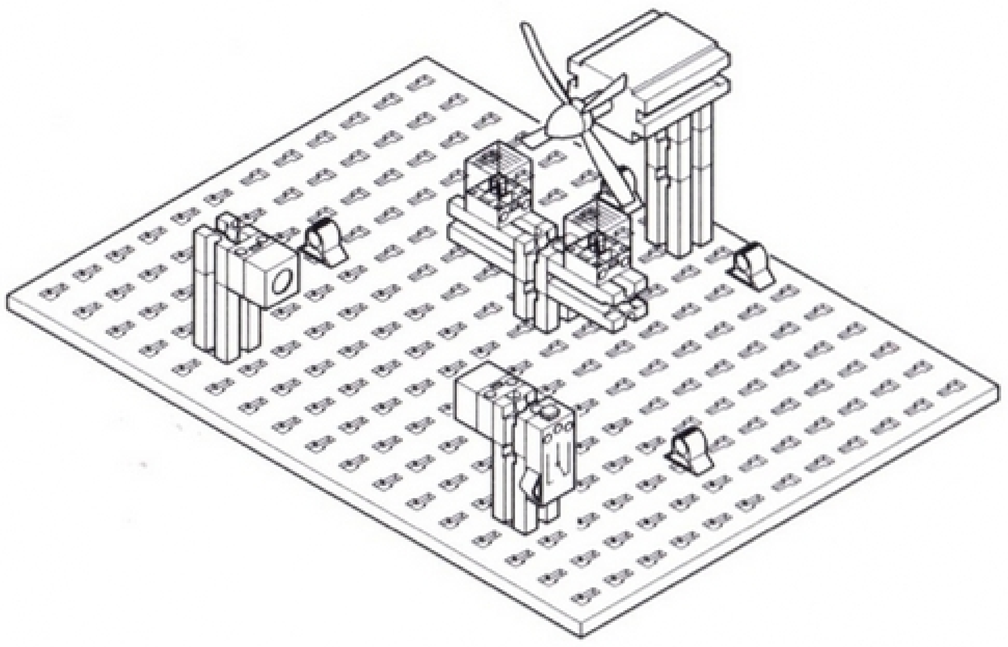
CONTROLLING MOVEMENT 5 : Analogue Sensors

An analogue sensor constantly monitors one part of an environment, such as the humidity in a greenhouse or the light level in a room, or the temperature in an incubator for premature babies. The sensor supplies a reading to the computer through the Smart Box. The reading changes as the heat or light level changes.

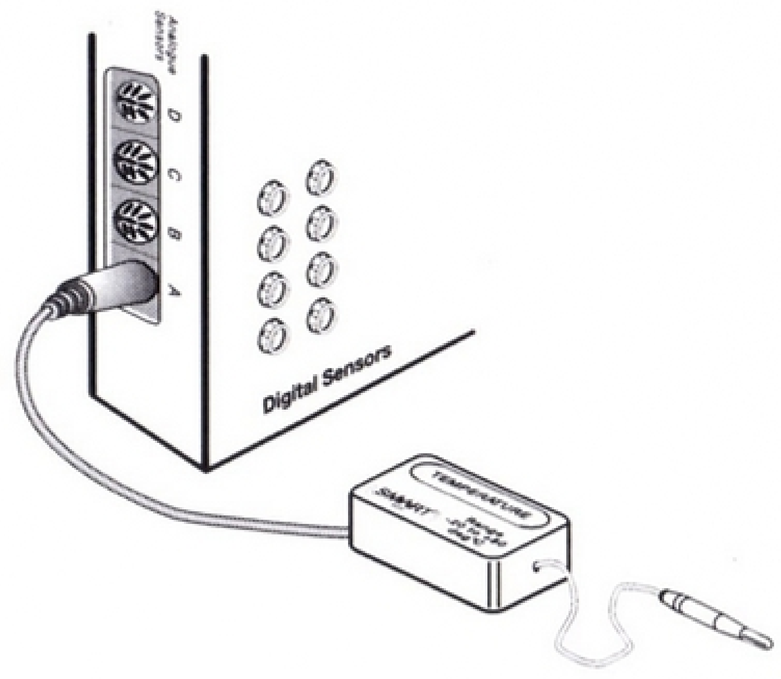


Analogue sensors are often used in control systems to monitor conditions so that an output device can be switched on or off automatically, as necessary.

Build the model fan shown on the Hand Dryer construction sheet. Connect it to the Smart Box.



Plug a Smart temperature sensor into an Analogue Sensors socket on the Smart Box.



Build the procedure shown in fig 1. This is designed to switch on the fan when the temperature rises above the threshold, and switch it off when the temperature falls below the threshold.

Procedure : fan
repeat if temp > 25 then forward a else halt a forever

Fig. 1

Set the threshold temperature to a degree or so higher than the reading displayed in the Monitor on the Smart Move screen.

Run the procedure and test the system by warming up the sensor in your hands. Hold it near the fan to cool it down again.

You may find that, when the temperature reading is fluctuating around the threshold, the fan keeps switching on and off. Try editing the procedure as shown in fig 2, to make the system switch cleanly.

Procedure : fan
repeat if temp < 24 then halt a if temp > 25 then forward a forever

Fig. 2 The fan doesn't switch on until the temperature reaches 26°, and doesn't switch off until it falls to 23°.

Varying the Output

Another way to use an analogue sensor in a control system is to make the output vary in response to the varying readings from the sensor.

Use the fan model and Smart temperature sensor for this activity. Build the procedure shown in fig 3. It is designed to speed up the fan as the temperature rises, and slow it down as the temperature falls. If you use the lowest motor speeds, it will be easier to see when the fan changes speed. Run the procedure and test it.

You could use similar procedure for:

1. A system in which a Smart position sensor is used as a speed controller for the single motor buggy shown on the Car Park Barrier construction sheet.

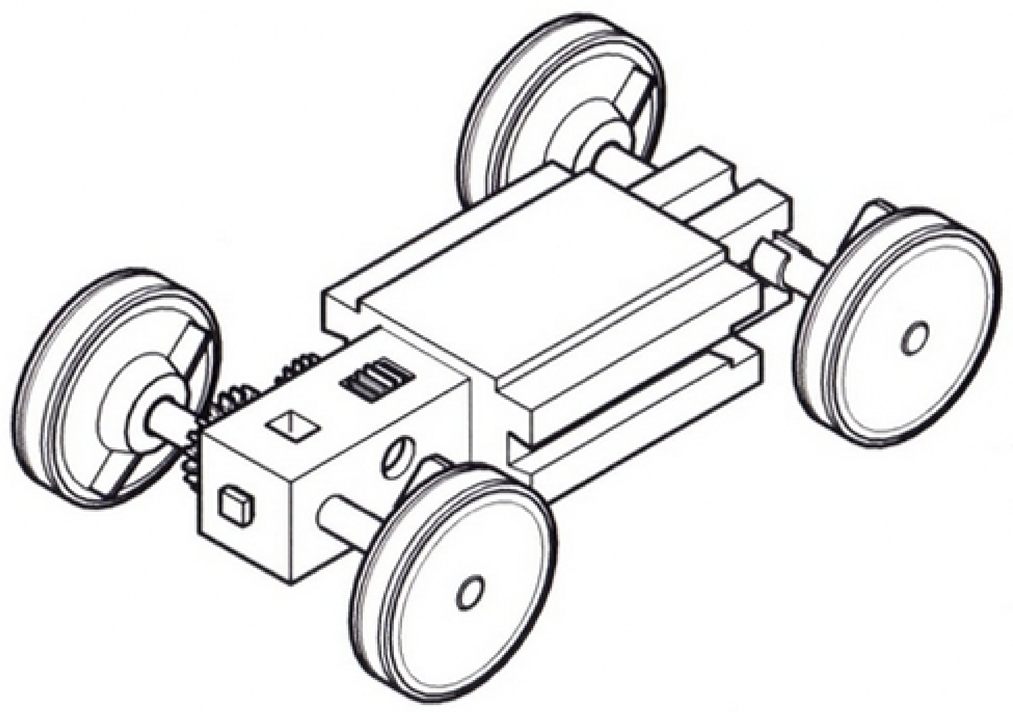
2. An automatic lighting system with three lamps and a Smart light sensor which switches on the lamps one by one as it gets darker, and switches them off in the same way as it get lighter.

You could develop this procedure to use the eight green LEDs of the Smart Box Digital Outputs as a light meter. The more light shines on the sensor, the more LEDs light up.

Procedure : fanspeed

```
repeat
let t = temp
if t < 24 then halt a
if t >= 24 and t < 25 then forward a speed 2
if t >= 25 and t < 26 then forward a speed 5
if t >= 26 then forward a
forever
```

Fig. 3



Single Motor Buggy

Monitoring Position

You can use a Smart position sensor to provide accurate feedback on the position of a motor-driven device. Build a turntable onto the position sensor as shown in fig 4. Drive the turntable with a worm gear as shown on the Worm and Wheel construction sheet.

Fig 5 shows the kind of procedure that you can use to drive the turntable accurately from its zero position to any position in its turning circle.

Procedure : turn

```
zero
ask n, "enter position", p
forward a
wait until position >= p
halt a
```

Fig. 5

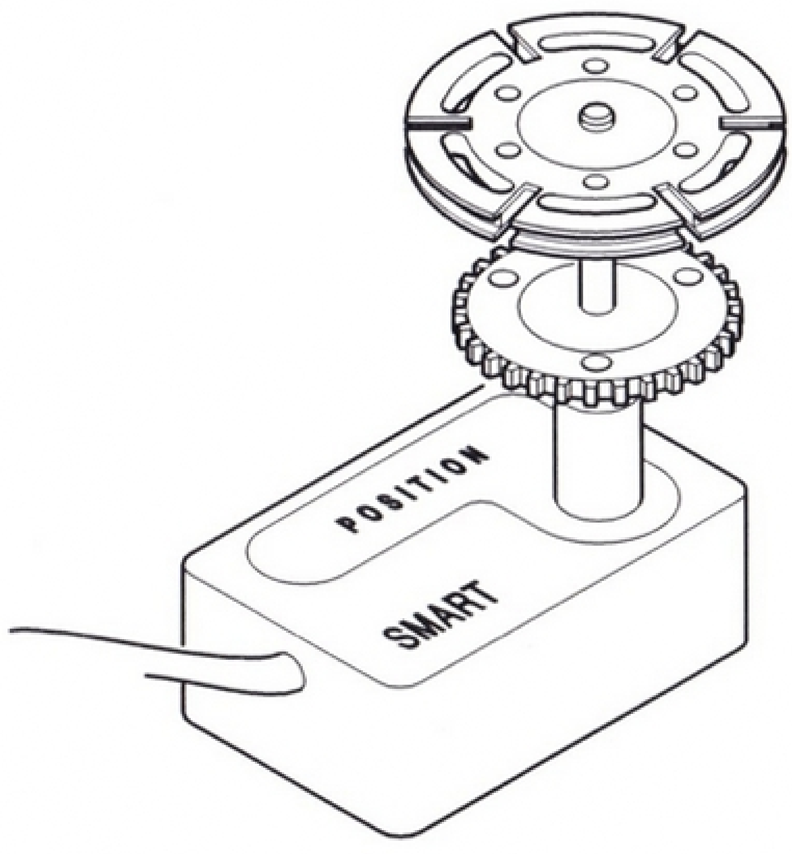
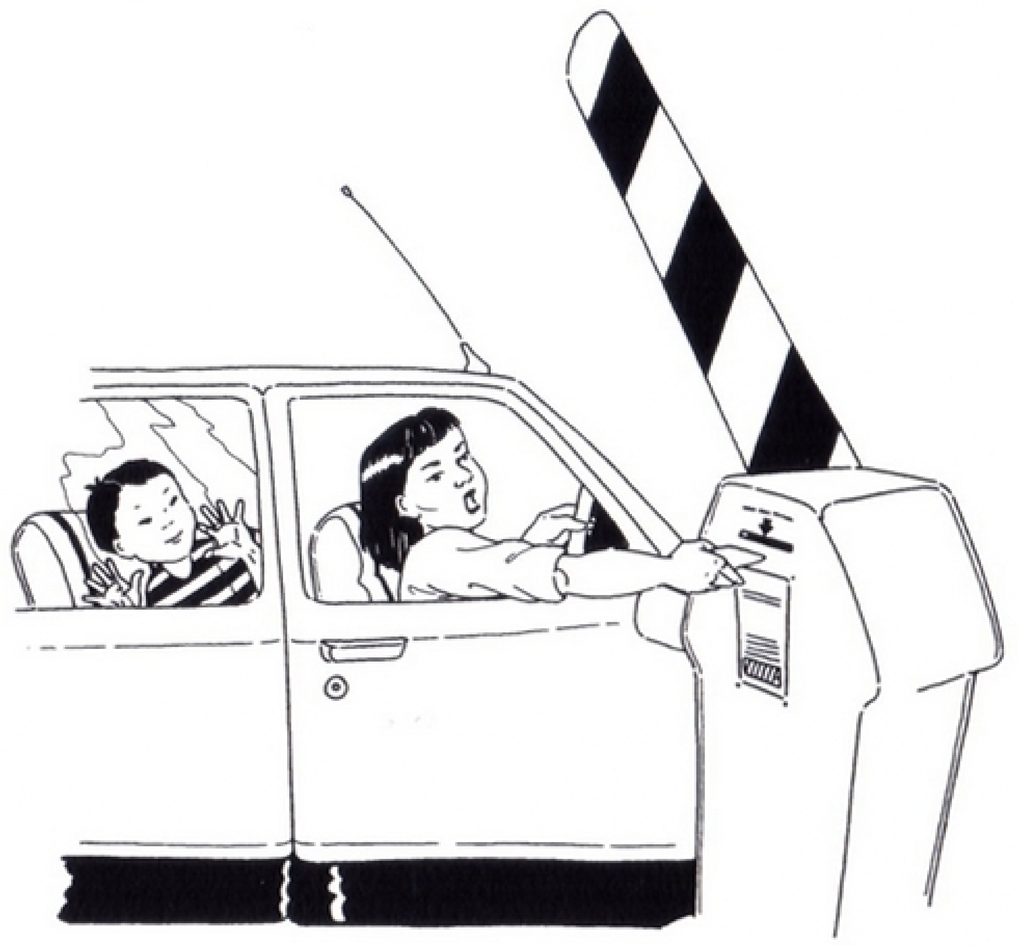


Fig. 4

Project File

PROJECT FILE : Car Park Barrier

A company who own a chain of multi-storey car park need a demonstration model to show the control system for the automatic barrier at the entrance to each of their car parks.



Specification

The model should include : a simple motor driven barrier with red and green lights, and a model car.

It must demonstrate the following system:

When a car approaches the barrier, a red light comes on with a message saying STOP, TAKE TICKET. As soon as the driver has taken a ticket the red light goes out. The green light comes on and the barrier is raised.

When the car has passed safely though, the barrier comes down and the green light is switched off. An important safety feature of the system is that the barrier will not come down while a car is still underneath it.

Build the model and design the control system to carry out the demonstration.

Planning

a) You could design and build your own model, or use the Car Park Barrier construction sheet to help you. If you plan to design your own model, remember that you must be able to control the speed of the barrier and the distance it moves.

b) Your system must include sensing:

i) that a car has approached the barrier,
ii) that the driver has taken a ticket,
iii) that the car has passed under the barrier.

NOTE : The construction sheet shows how to build a simple motor driven car which is controlled as part of the whole demonstration system. You could use a hand-pushed car instead and use the second motor in the kit to build a ticket dispenser (the roller mechanism shown on the Reader construction sheet is one way of dispensing tickets).

Extending The System

You could develop the system so that it keeps a count of cars coming in and going out. When all the available spaces are taken, a message is displayed saying CAR PARK FULL. When spaces are available, the message says SPACES.

CAR PARK FULL

SPACES

PROJECT FILE : Controllable Vehicle

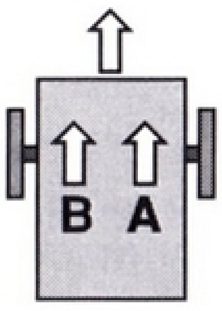
Using a computer to control a vehicle has always played an important part in the study of robotics - and it can be fun. In this project, you can investigate some of the basic ways of controlling a vehicle.

Build the model shown in part one of the Buggy construction sheet and use it to test the routines you write in the following investigations.

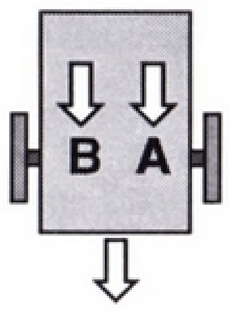
1. Steering

The buggy has two motors – one to drive each wheel. This means that you can control the direction in which it moves, as shown in fig 1.

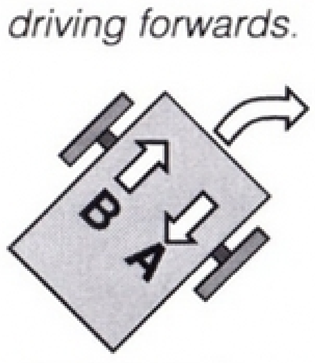
Fig 1.



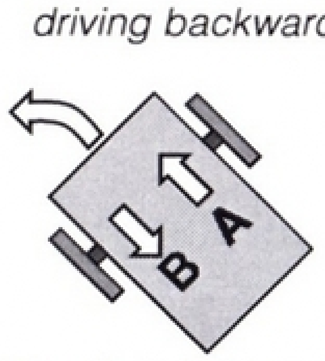
Forward. Both motors driving forwards.



Backward. Both motors driving backwards.



Turn Right. Right motor driving backwards, left motor driving forwards.



Turn Left. Right motor driving forwards, left motor driving backwards.

Write routines to make the buggy carry out some simple manoeuvres e.g. moving in a square, reversing around a corner.

2. Avoiding Obstacles

Add the bumper to the model as shown in part two of the Buggy construction sheet. Write a routine to make the buggy move around obstacles in its path. The following information will help you :

When the buggy comes up against an obstacle, the bumper presses the switch. It should stop briefly, and then go back far enough to allow it enough space to turn. It should then turn right and move forward far enough to allow it to turn and move past the obstacle. Finally it should turn left and then carry on forward.

3. Following a Path

Add the sensor bar to the buggy as shown in part three of the Buggy construction sheet.

Fig 2.

Make a path for the buggy by cutting strips of matt black paper about 80mm wide and sticking them to a large sheet of white paper (see fig 2). Write a routine to make the buggy follow the path. The following information will help you :

The lamp shines downwards and the sensors respond to light from the lamp reflected off the surface on which the buggy is travelling. If it is shining on a matt black surface, there will be little or no reflection and the sensor will be off. If it is shining on a white surface, the reflected light will switch the sensor on. If the right sensor switches on, the buggy must move left until the sensor is off again. If the left sensor switches on, the buggy will need to turn right.

PROJECT FILE : Security

Security is very important in all kinds of buildings such as offices, factories, banks, homes. Many of these places have doors that are operated by security locks.

The input to some security locks is made by pressing the right combination of buttons. In others, a coded card is used. In this project, you can investigate simulations of both methods. You could develop the project to design and build a motor operated lock mechanism as the output of the system.

1. Push button combination

Connect four switches, a red light and a green light to the Smart Box.

The user must press two of the switches to open the lock. Choose which of the two switches they will be. Then design a system that switches on the green light when these two are pressed, and switches on the red light if any other combination of switches is pressed.

2. Card operated lock

Build the model shown on the Reader construction sheet.

The model is designed to read how many holes are punched in a single row on a card (Fig 1 shows an example of this kind of card).

It uses a combination of lamp and light sensor to do this. If the space between the lamp and the sensor is blocked by solid card, the sensor will be off. If there is a hole in the card that allows light to shine through, the sensor will be on. The front sensor will sense if a card is pushed into the machine.

The motor-driven rollers will pull the card through at the right speed for the front sensor to read the card.

The back sensor will sense when the card is fully into the machine so there are no more holes to be read, and the card can be ejected.

Design a card of the right size, shape and material to work efficiently in the model.

Design a control system to use the model to find out how many holes are punched in the card.

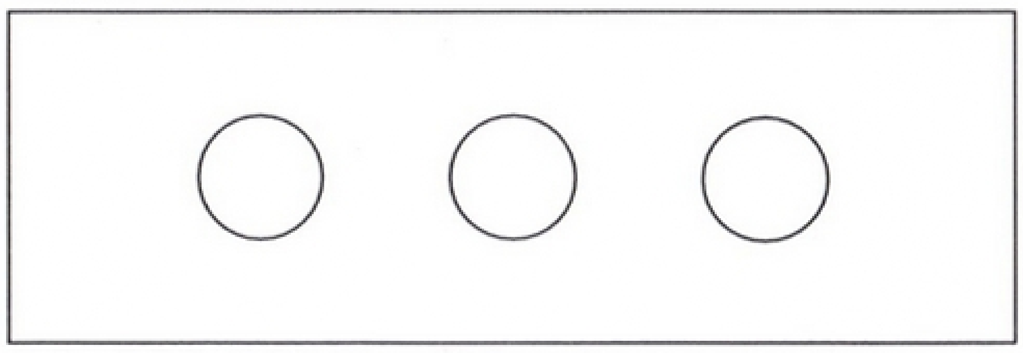


Fig 1 Card with three holes

PROJECT FILE : Vending Machine

Machines that dispense crisps, drinks, chocolate bars etc. are popular at places like swimming pools and leisure centres.

A designer has begun to plan one of these machines and has come up with a model of the part that holds chocolate bars. This is shown in fig 1. Instructions for building the model are on the Dispenser construction sheet.

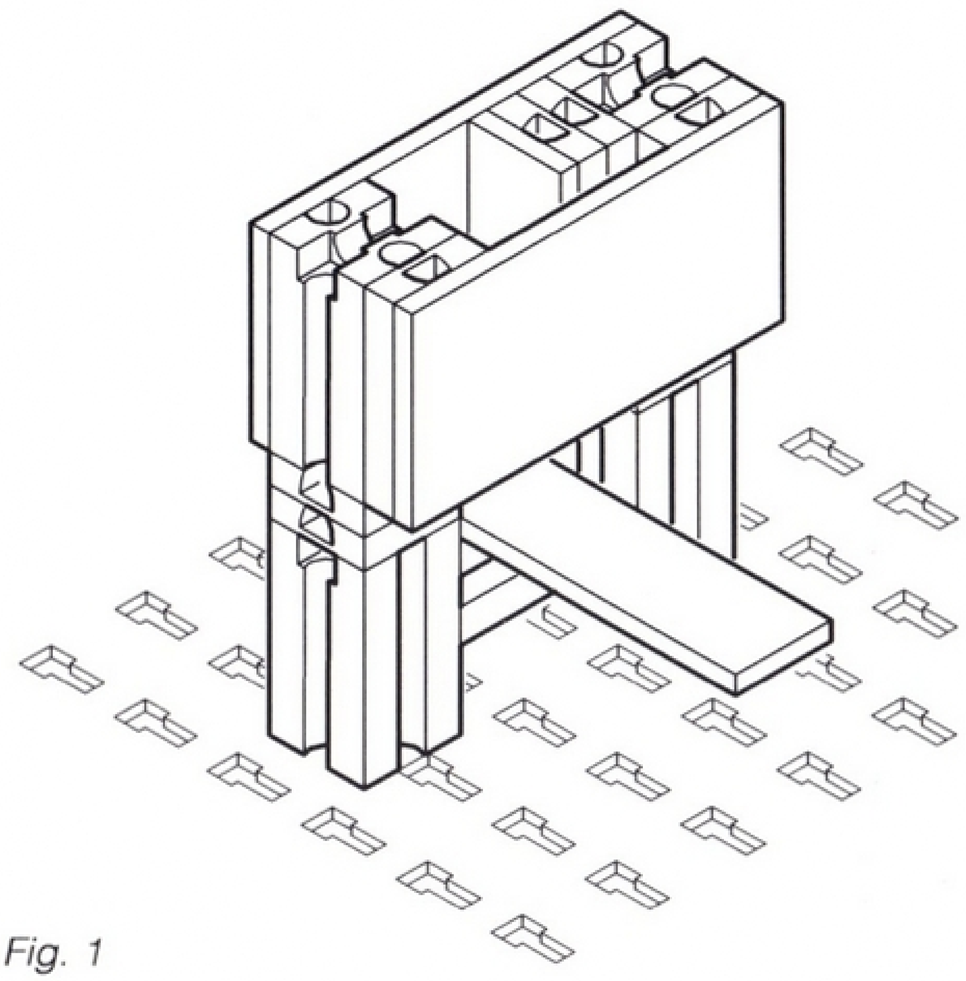


Fig. 1

Build the model and add a motor driven mechanism that will push out one bar at a time.

Design the control system to operate the machine to the following specification.

Specification

The system must sense when a customer puts a coin into the machine, and then dispense one bar. It must also include some way of indicating when the holder is empty.

PROJECT FILE : High Rack Warehouse

The modern way to store large quantities of goods is to have giant floor to ceiling racking systems up to 20m high, in rows up to 100m long. The racks are loaded and unloaded by operators sitting in computer-controlled enclosed platforms that move up, down and accross between the thousands of locations in the system.

A company which makes these systems would like a small model that can be used to demonstrate how their product works.

The model must meet the following specification.

Specification

The model should have nine locations arranged as shown in fig. 1.

The control sysytem must allow the operator to input two numbers - one number for the x axis, and one for the y axis. The operator's platform will then move automatically to the required location, as quickly as possible.

Build the demonstration model, and design the control system to operate it.

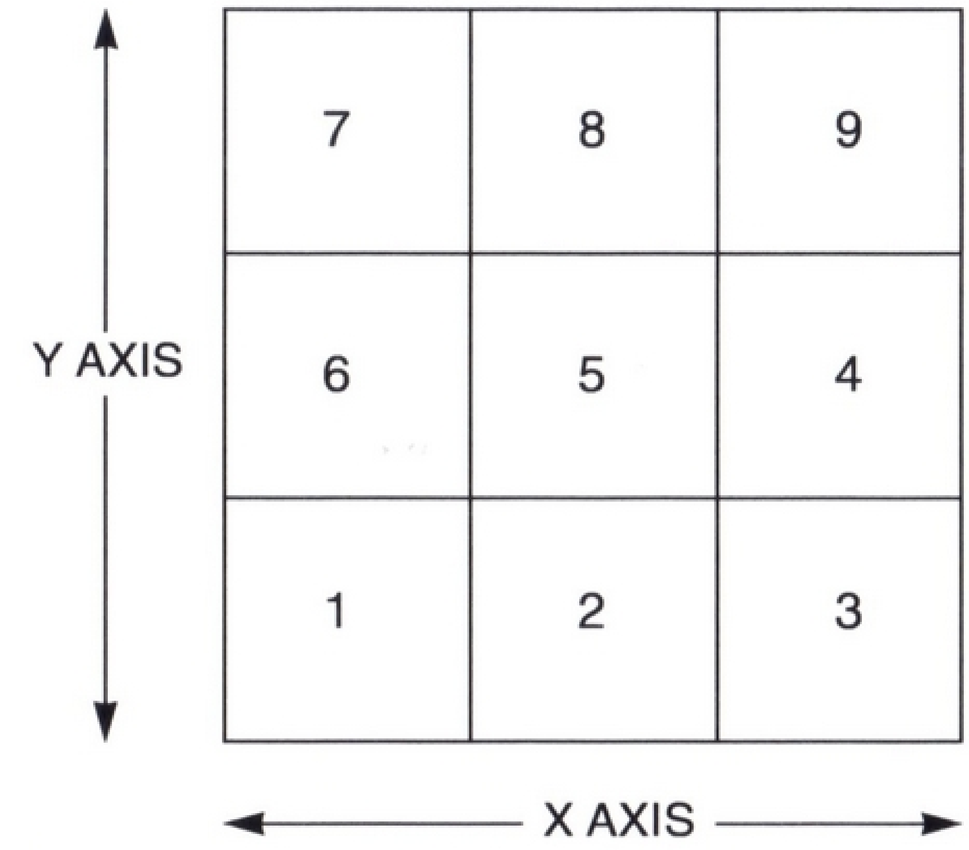


Fig 1.

Planning

The model needs two sources of linear movement - one for each axis. The Leadscrew and Rack and Pinion 1 construction sheets show the basis. You will need to design a way of combining them into a stable structure. Other changes you will need to make will include:

i) Use a lamp and light sensor, instead of a switch, to sense the zero position of the leadscrew.
ii) Three limit switches are needed on the rack and pinion - one at each end as shown, and an additional one in the middle position.
iii) Design a different way for the rack and pinion limit switches to be pressed. With the existing method, the platform will stop at different middle positions depending on whether it is going up or down.

PROJECT FILE : Carousel Storage System

Automated storage systems are very common. Many of them use a carousel system, in which components are stored on different segments of a computer-controlled turntable.

When a particular component is needed, the operator inputs its number and the turntable rotates automatically to bring the appropriate segment under the device that picks up the component and places it in a collection tray. See fig. 1.

A company which makes these systems would like a small model that can be used to demonstrate how their product works.

The model must meet the following specification.

Specification

Six different components are stored in six locations on the turntable. The control systems must allow the storekeeper to input a part number. It must then move the turntable so the location of that component is under the pick-up device.

Much valuable time would be lost if the turntable had to go back to its zero position between each selection. The system must move the turntable directly from one location to another as each new component is selected.

Build the demonstration model, and design the control system to operate it.

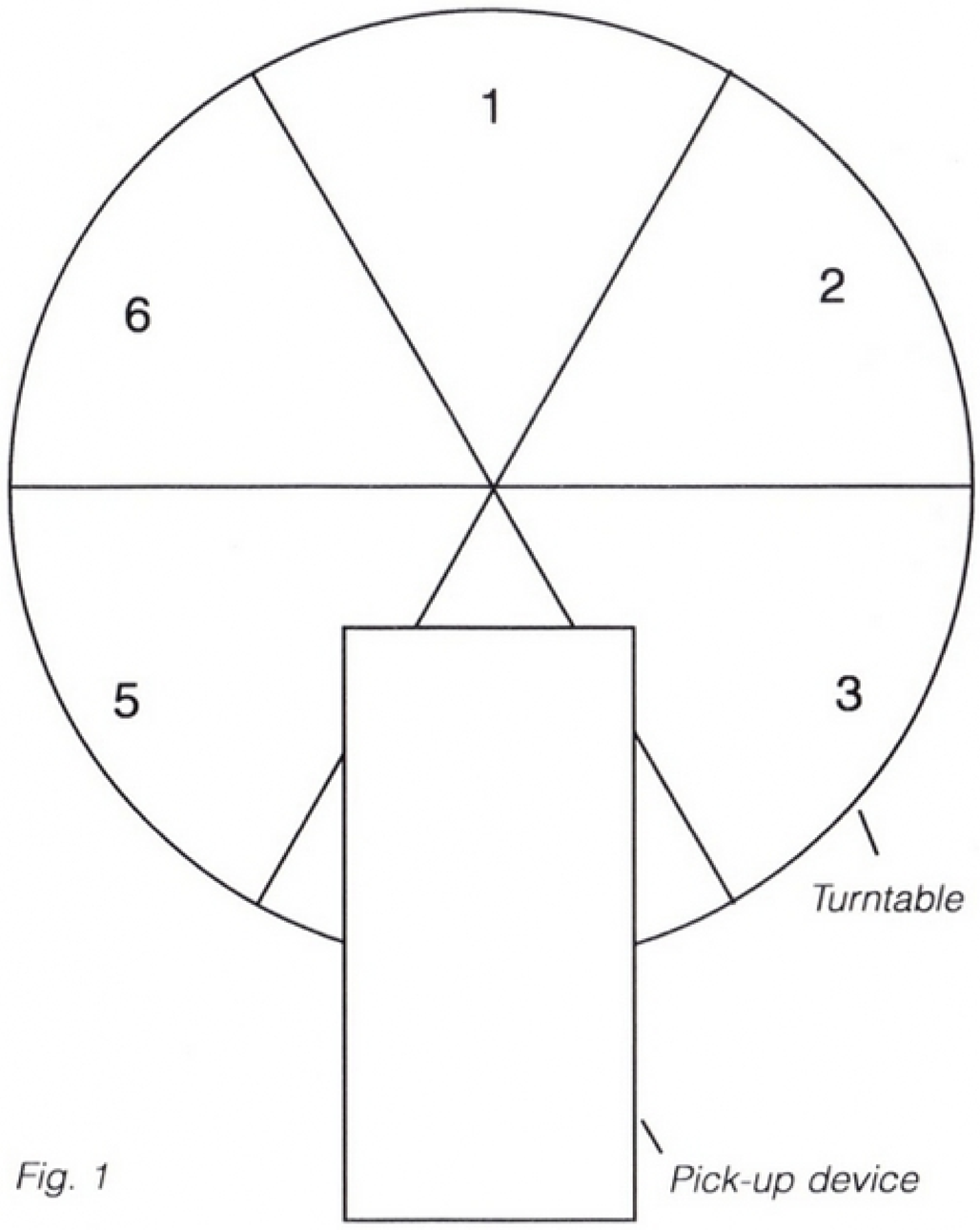


Fig. 1

Planning

The model shown on the Worm and Wheel construction sheet can be used to simulate the turntable. The model on the Rack and Pinion 2 sheet could be added to simulate the pick-up device.

It is a good idea to put a number label on at least one of the segments of the turntable to help you keep track of where it is.

A count of 17 or thereabouts will move the turntable round by one segment.

Construction Sheets

THE FISCHERTECHNIK SYSTEM

1. Blocks

Join blocks by SLIDING pegs in slots (Fig. 1)
The T-connector turns a slot into a peg (Fig. 2)
The link joins blocks side by side (Fig. 3)
Keep axles in place with either a washer or a spring clip (Fig. 4)

Fig. 1

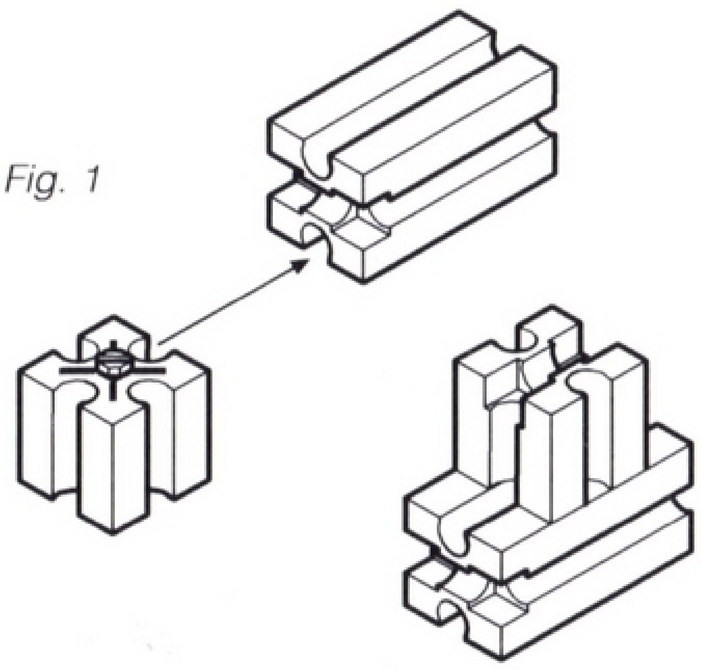


Fig. 2

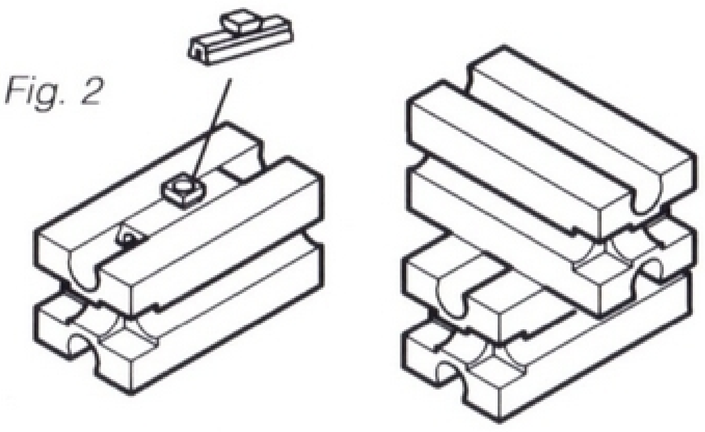


Fig. 3

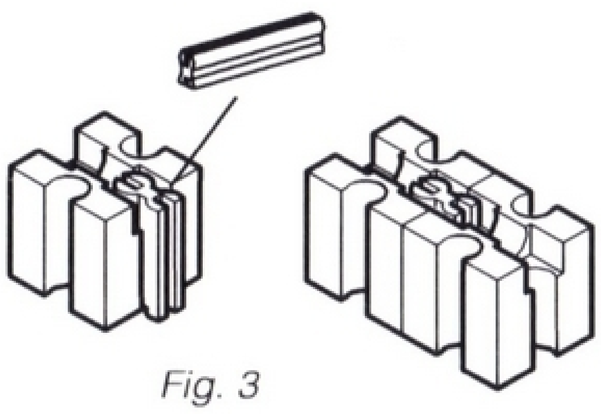
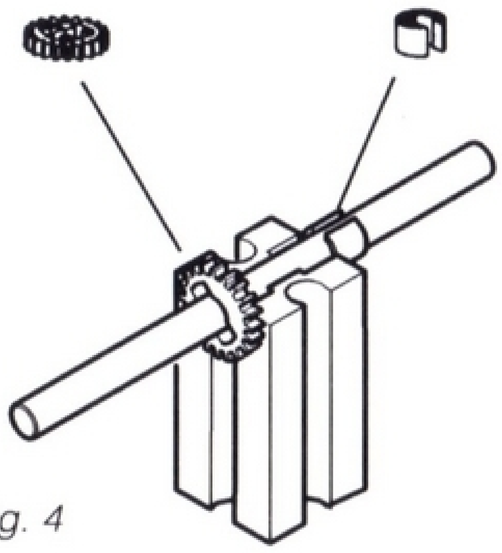


Fig. 4



2. Wheels

Most wheels are locked onto axles by the hub nut and collet system (Fig. 5)

Fig. 5

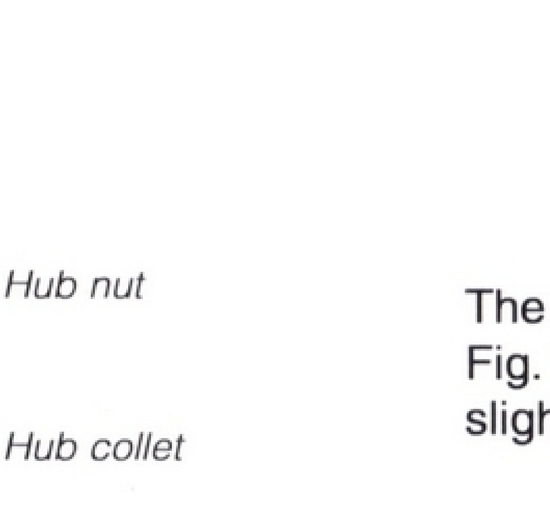
Hub nut

Hub collet

The components shown in Fig. 6 use the collet system slightly differently.

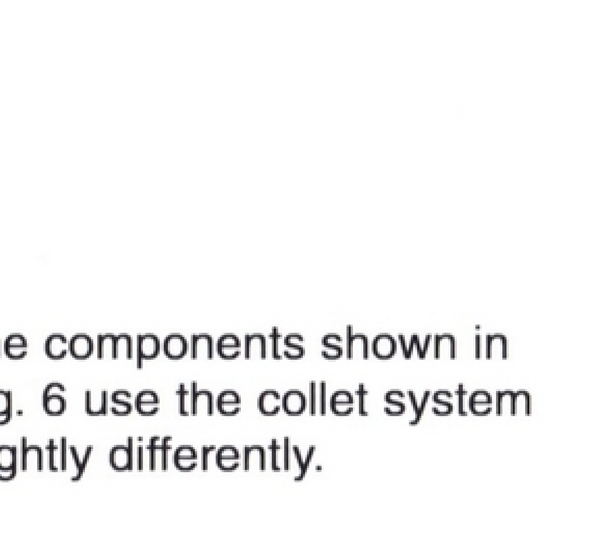
i) Slide the hub collet onto the axle.

ii) Place the outer part of the wheel onto the collet.



iii) Screw on the hub nut.

Fig. 6

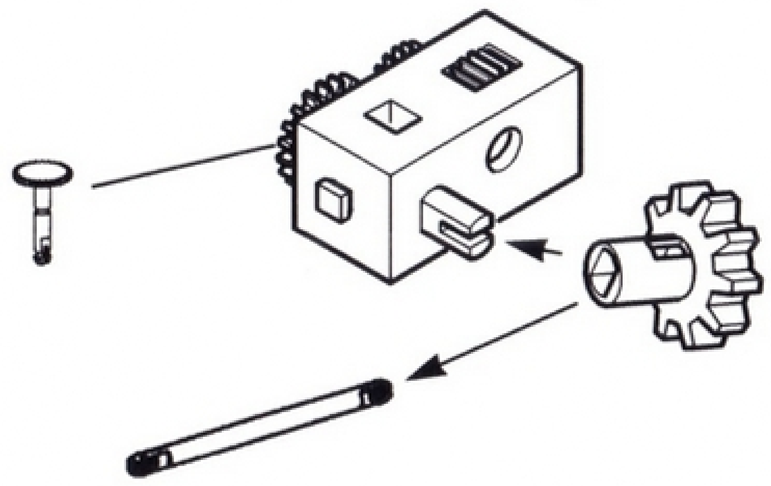


3.Push-Fit Components

The components shown in Fig. 7 push-fit onto the black clip axles (see Fig. 8).

Fig. 7

Fig. 8



4. Switch

In the models shown in this book, the switch is always used as a push to make (normally open) switch, so connection must be made to sockets 1 and 3 (see Fig. 9).

The switch can also be used as a push to break (normally closed) switch by connecting to sockets 1 and 2 (see Fig. 10).

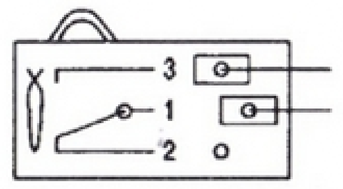


Fig. 9

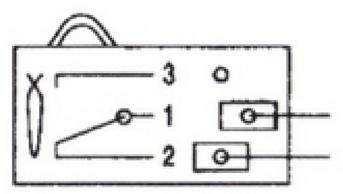


Fig. 10

5.Lamp

Connect plugs into the lampholder as shown in Fig. 11.

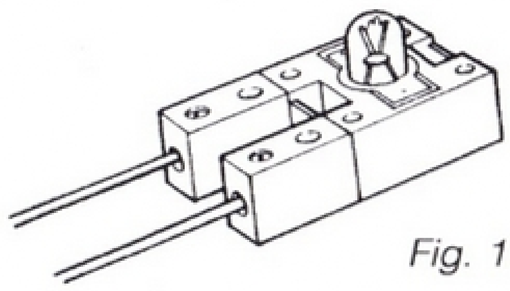


Fig. 11

6.Light Sensor

The light sensor is polarity - dependent. It must be connected to the Smart Box as shown in Fig. 12.

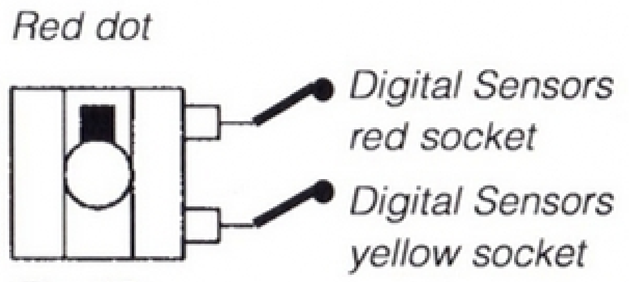


Fig. 12

7.Connecting Leads

There are 11 components in the kit that can be connected to the Smart Box (2 motors, 3 lamps, 4 switches, 2 light sensors). Wire, Fischertechnik plugs and a screwdriver are supplied to make up 12 x 1m double strand connecting leads. Each lead will also need 2 x 4mm plugs (not supplied) for connection to the Smart Box. Use the red clip shown in Fig. 13 to keep wires clear of moving parts.

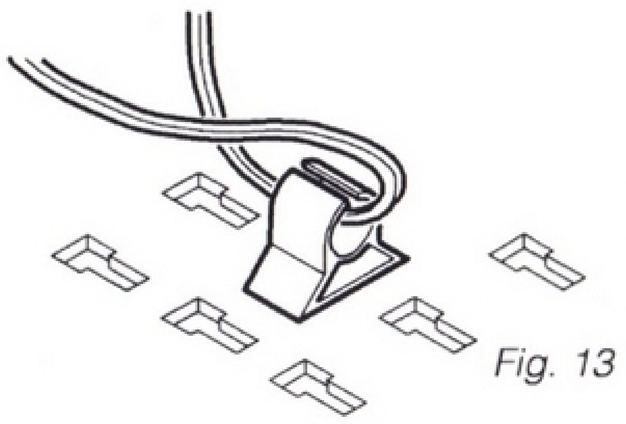


Fig. 13

HAND DRYER

3 x

3 x

1 x

1 x

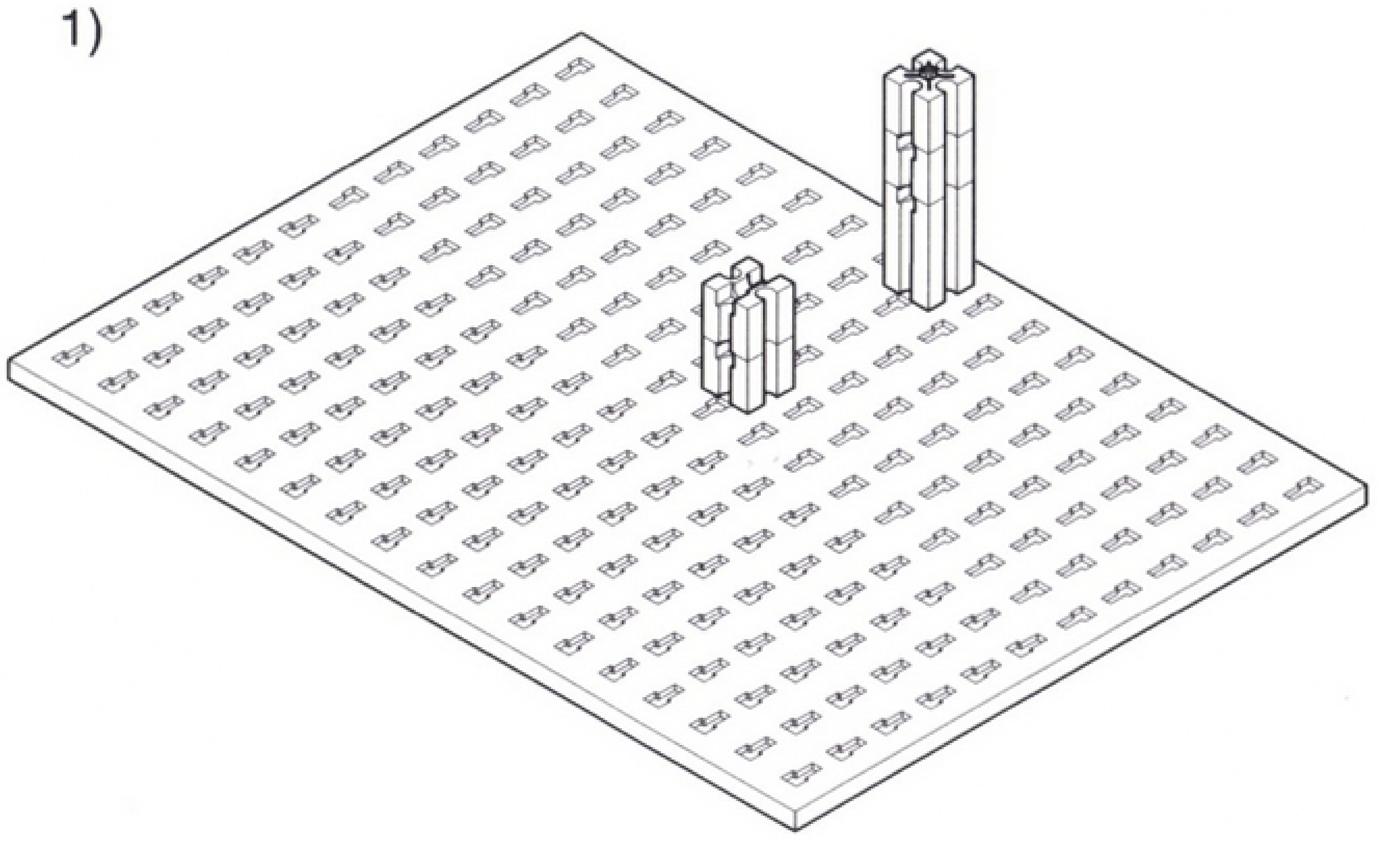
1 x

2 x

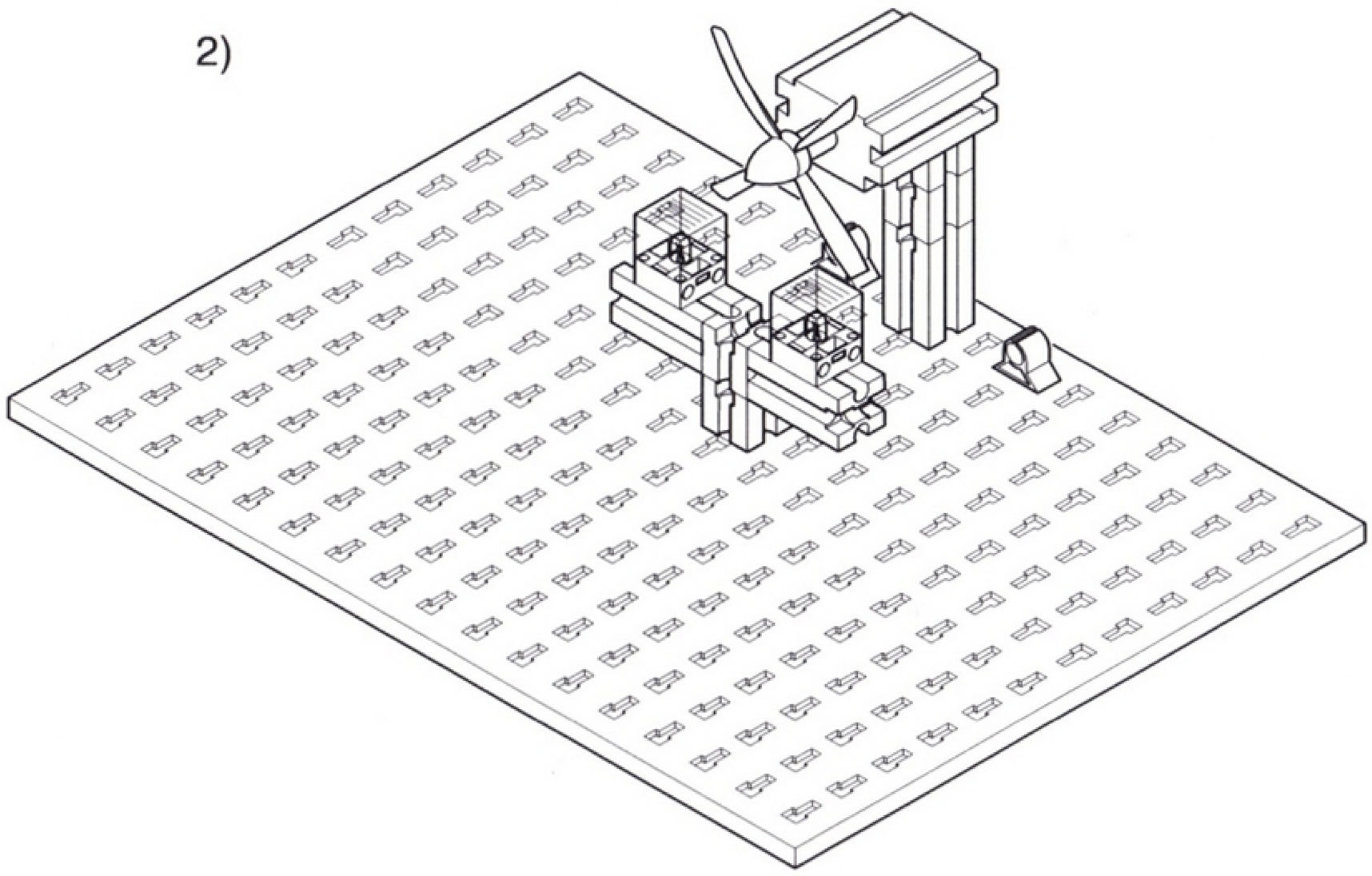
2 x

2 x (orange, red)

1)

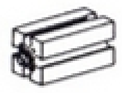


2)



cm 10 30 60 80 90 100 110

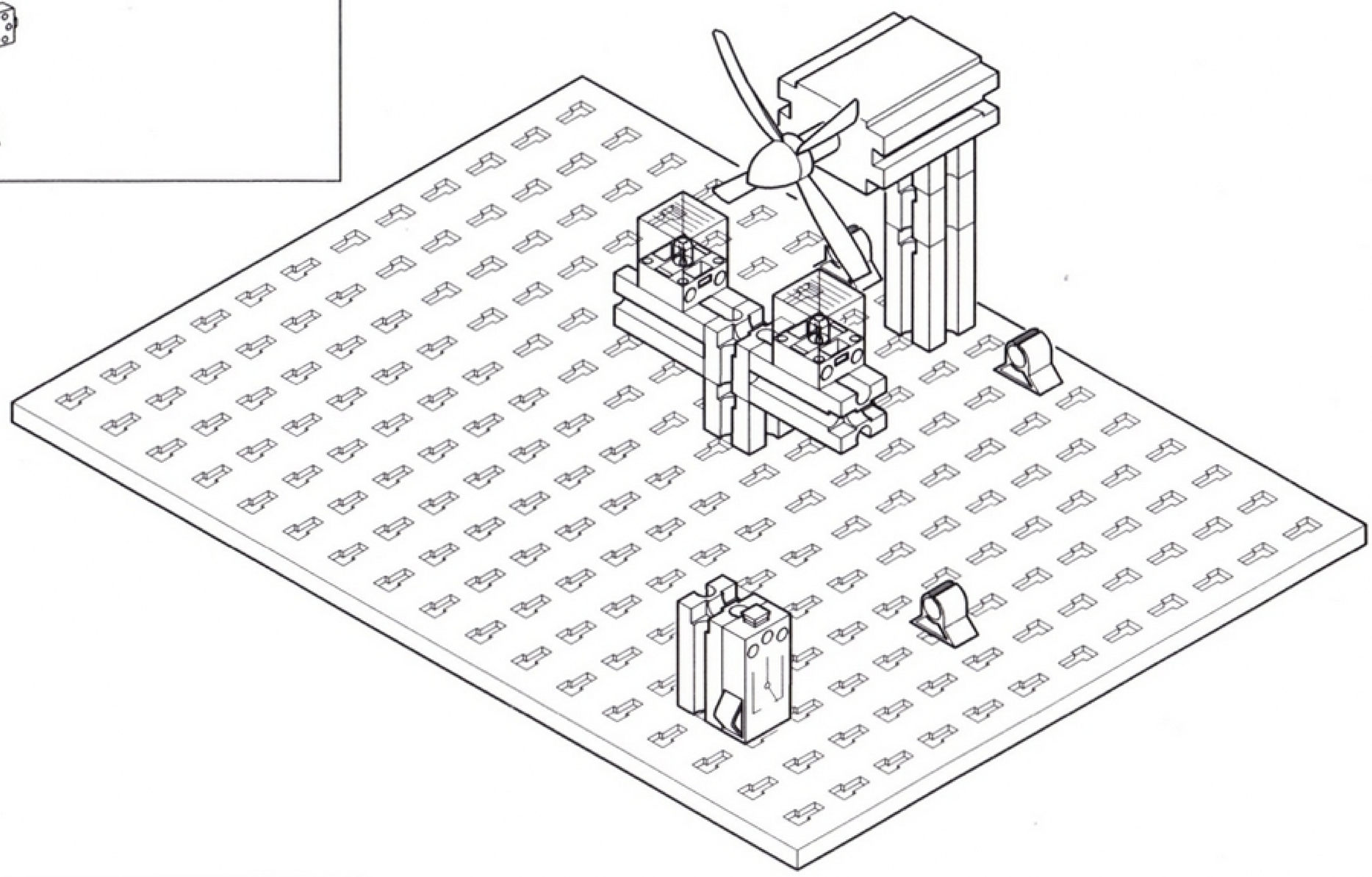
HAND DRYER

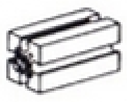
1 x 

1 x 

1 x 

3)



1 x 

1 x 

2 x 

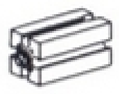
2 x 

1 x 

1 x 

4)

RACK AND PINION 1

13 x 

2 x 

1 x 

1 x 

2 x 

2 x 

4 x 

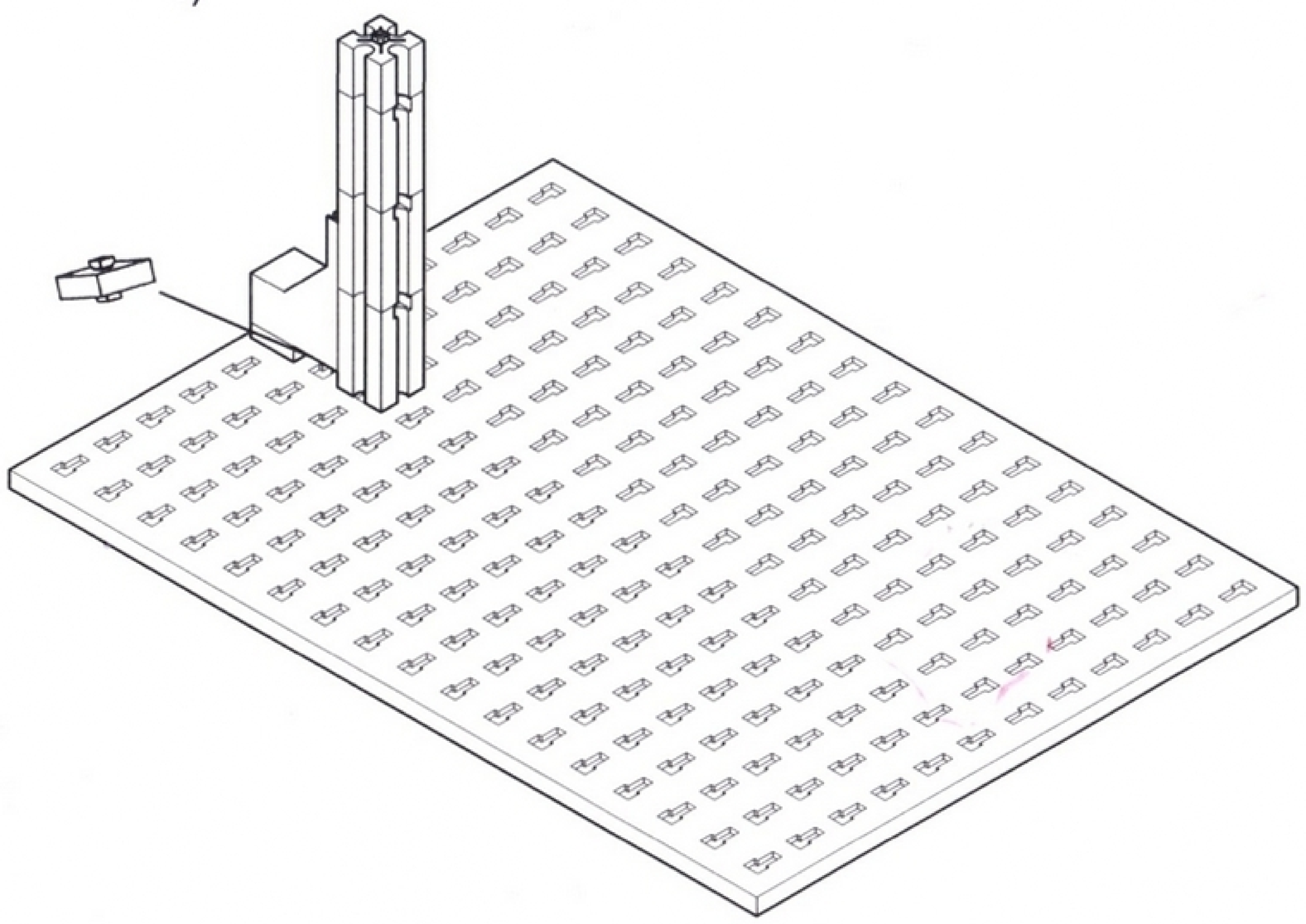
1 x 

1 x 

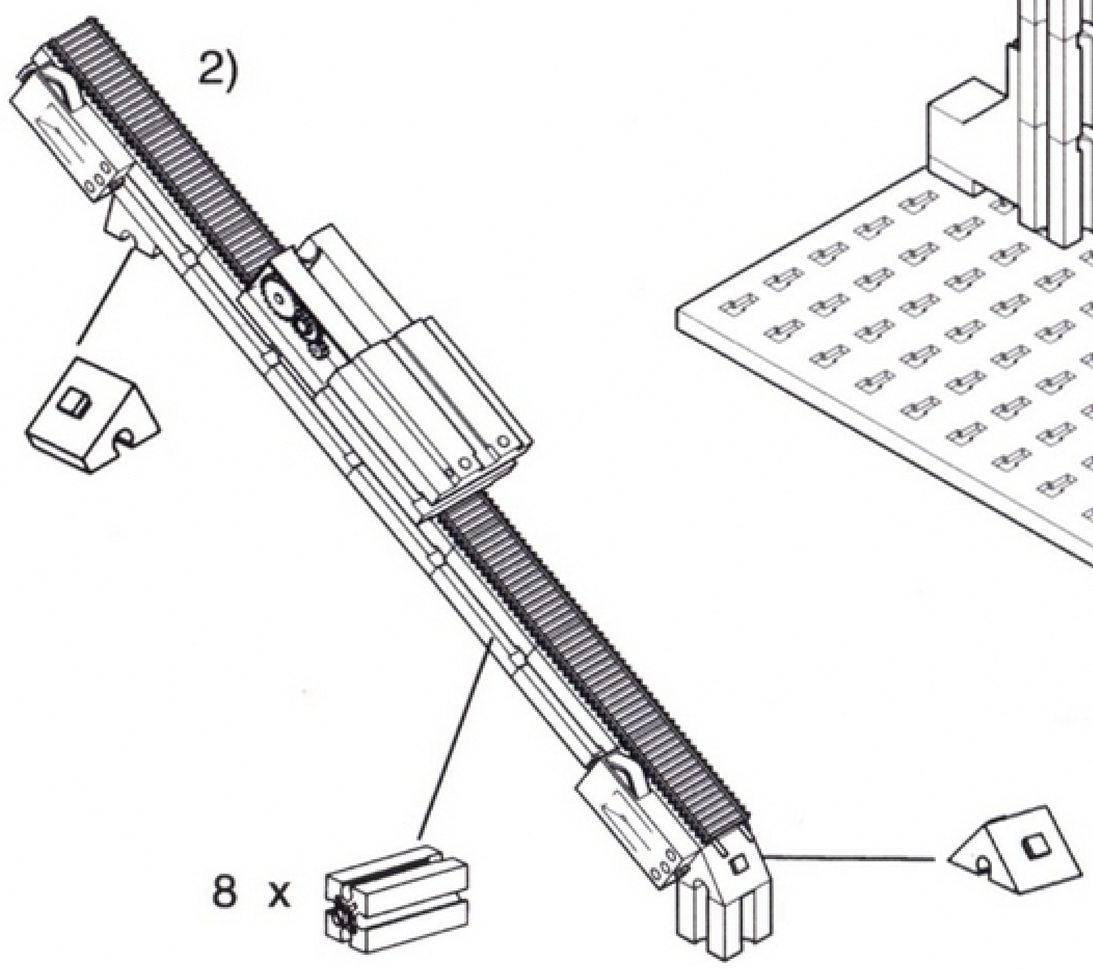
1 x 

4 x 

1)

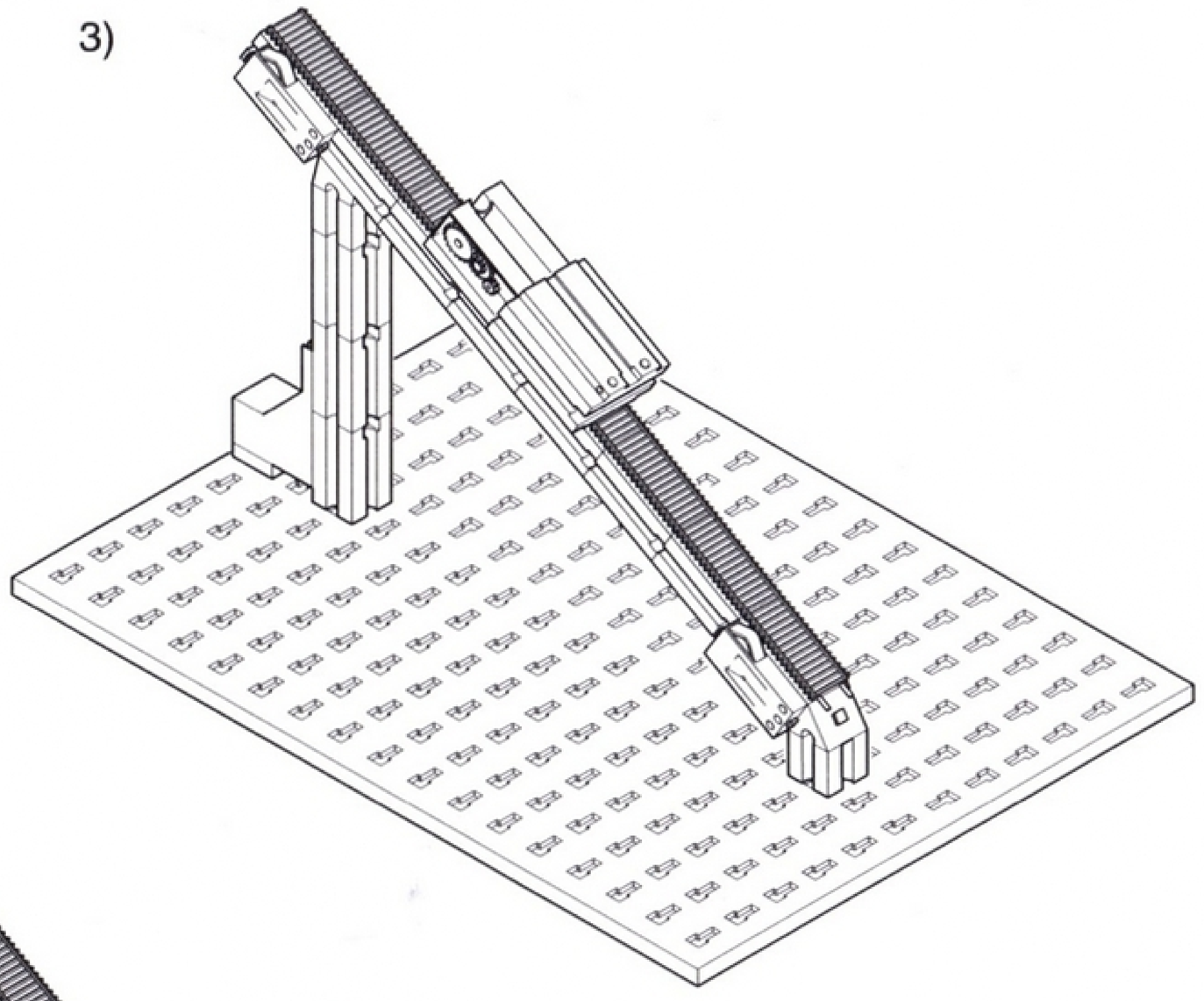


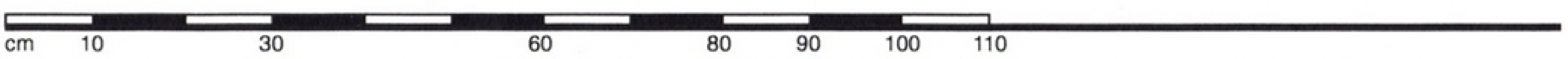
2)



8 x

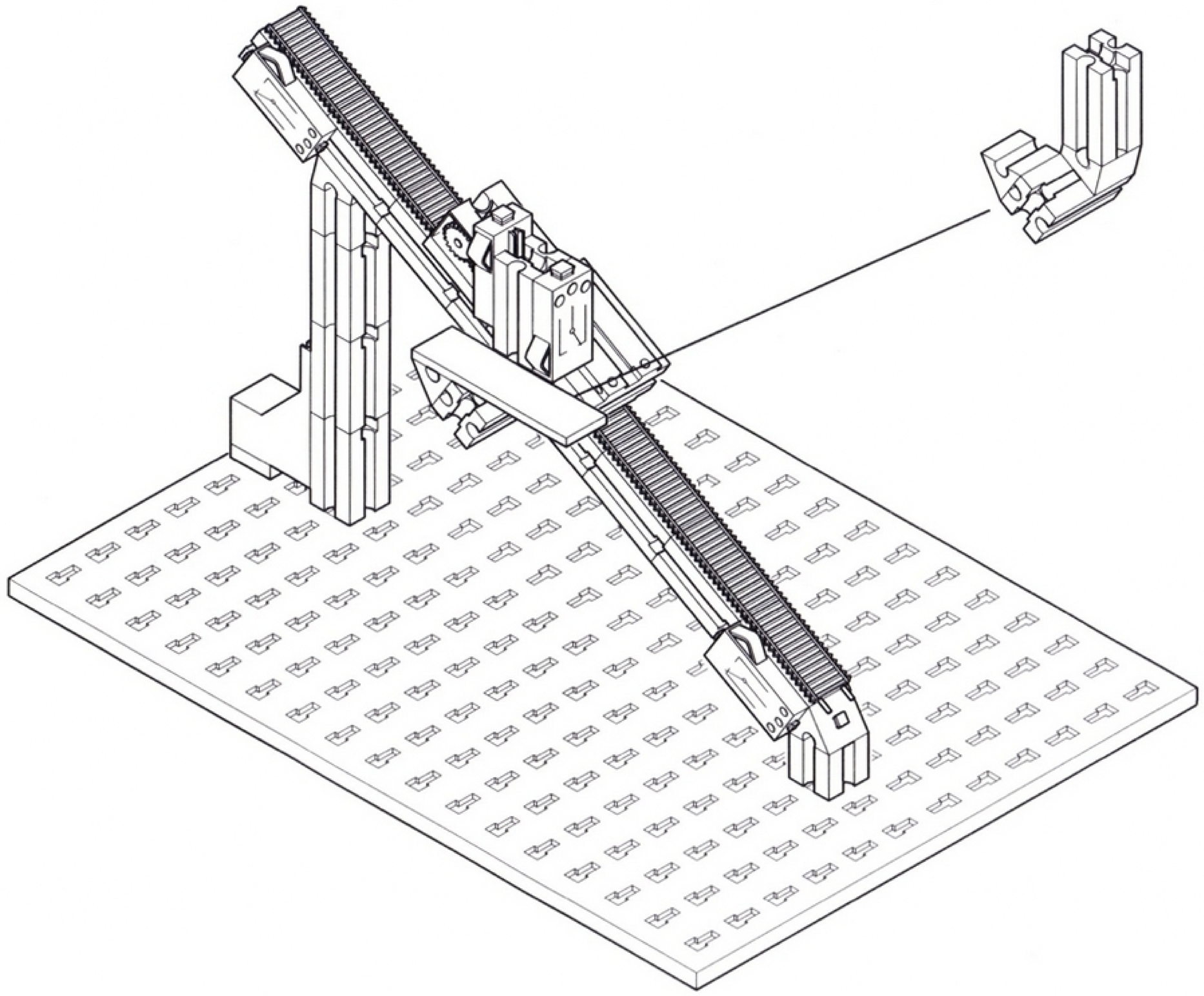
3)





RACK AND PINION 1

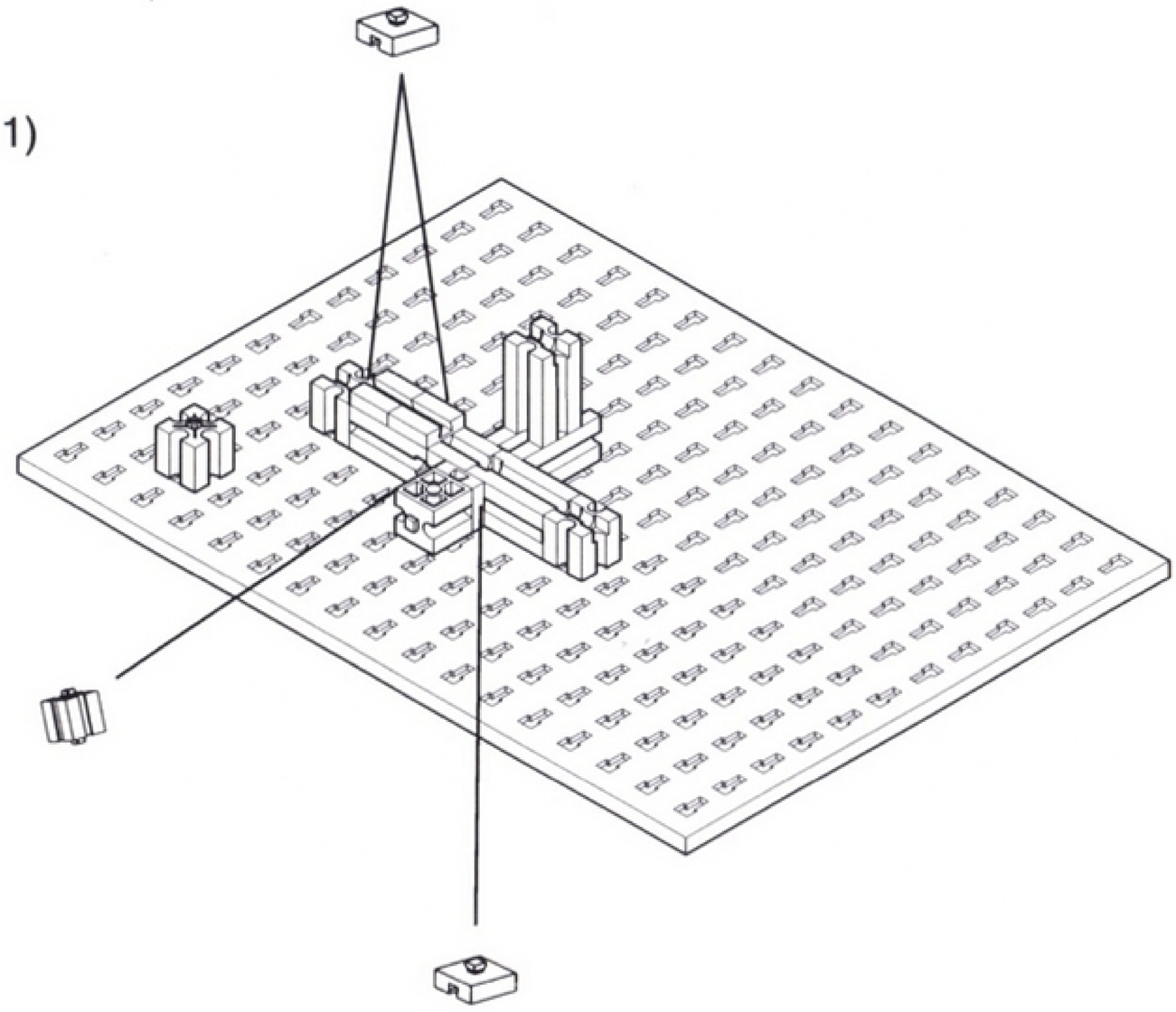
4)



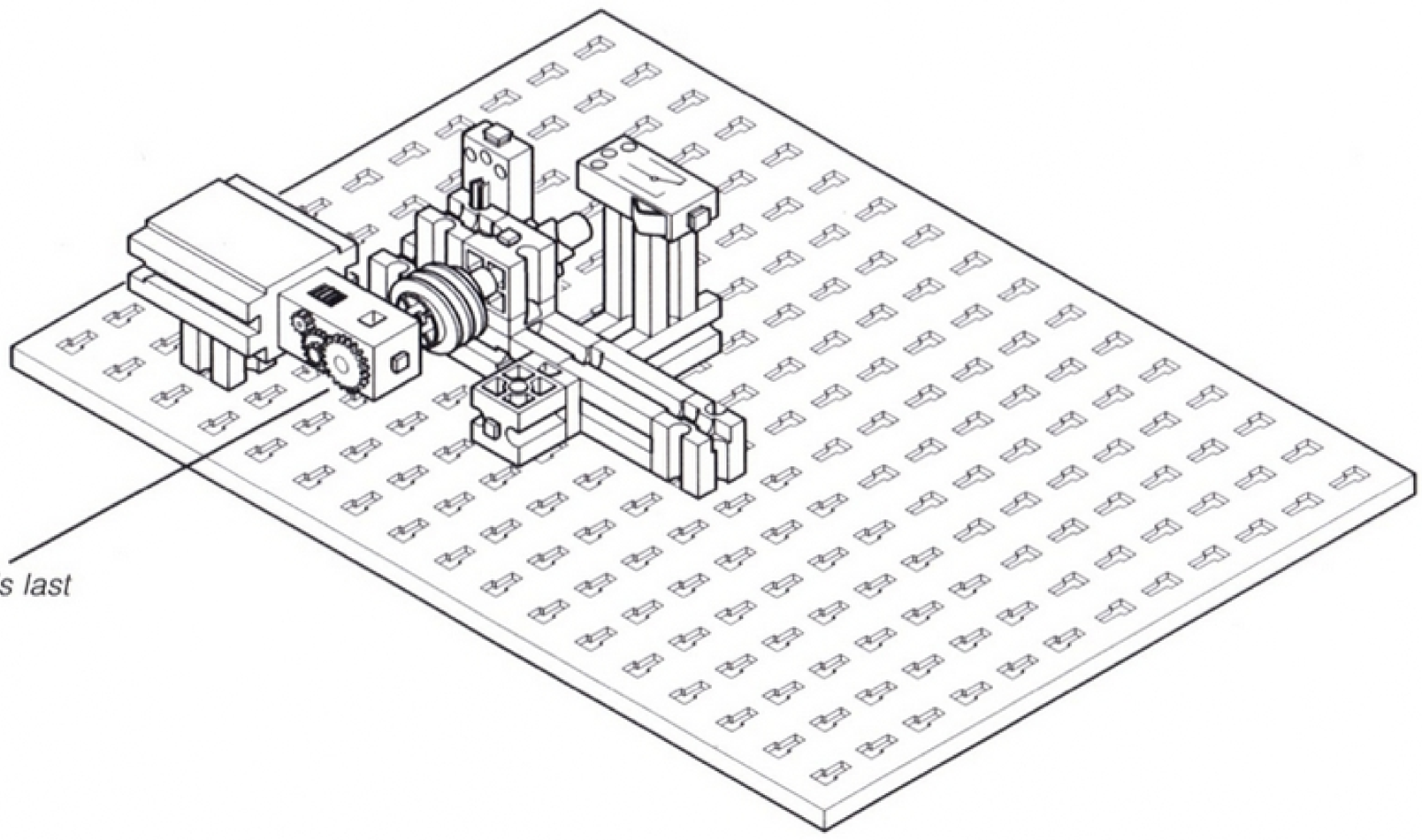
WORM AND WHEEL

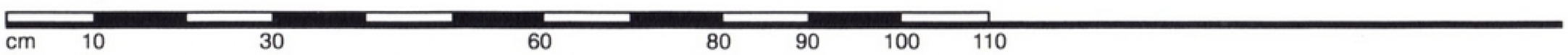
4 x		1 x	
3 x		1 x	
2 x		1 x	
3 x		1 x	
2 x		3 x	
1 x		3 x	
1 x		1 x	 (60mm)
1 x		1 x	
1 x	 (30mm)	2 x	

1)



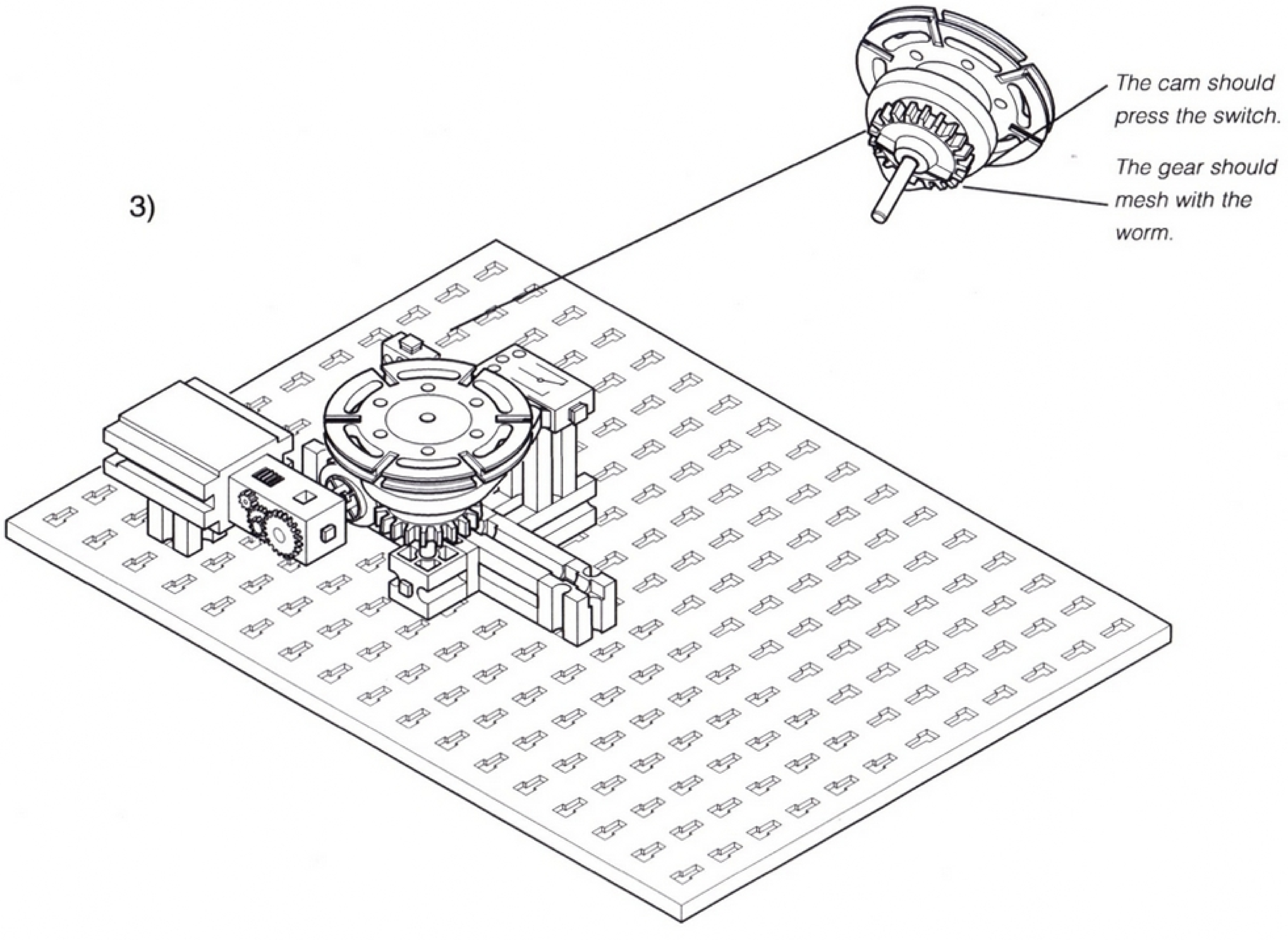
2)

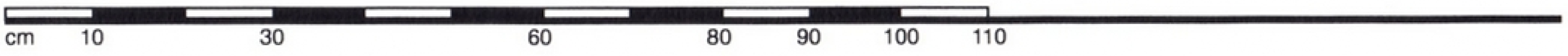




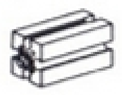
WORM AND WHEEL

3)





RACK AND PINION 2

2 x 

1 x 

1 x 

1 x 

1 x 

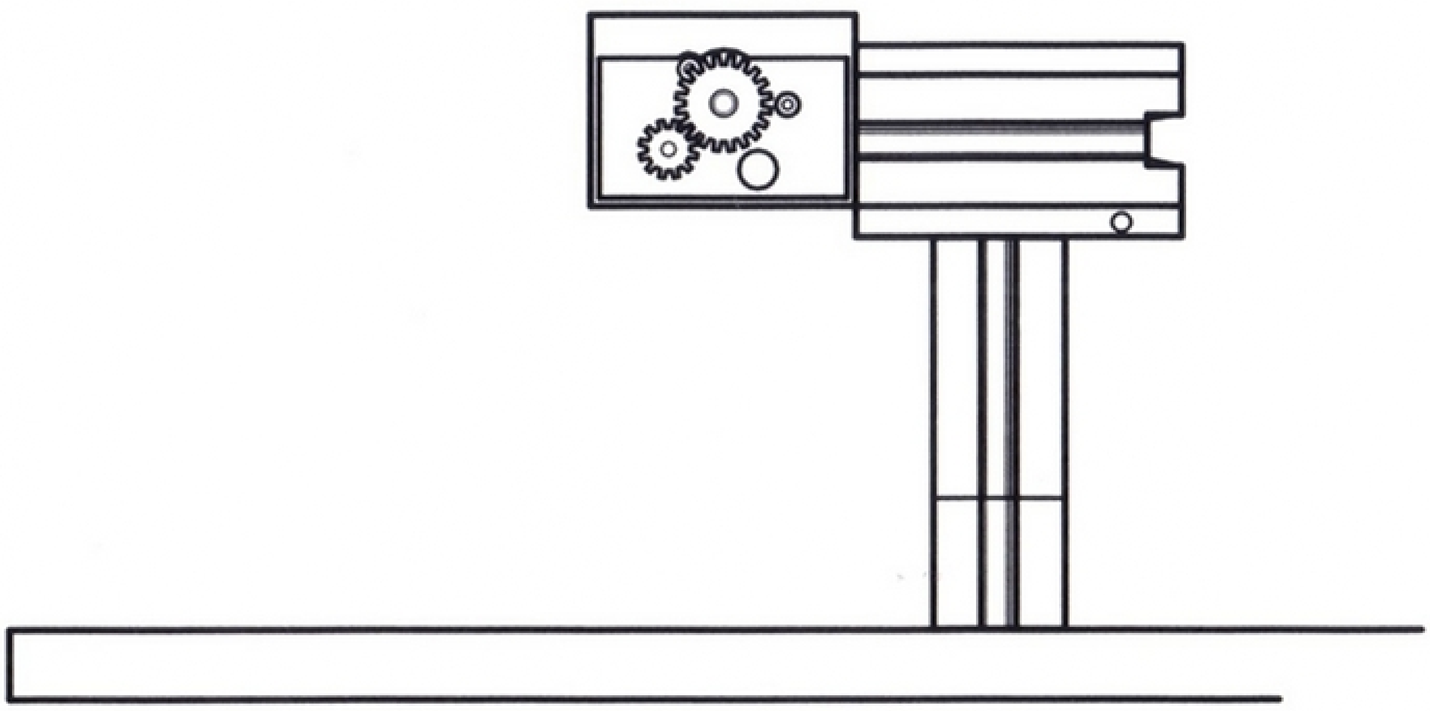
1 x 

1 x 

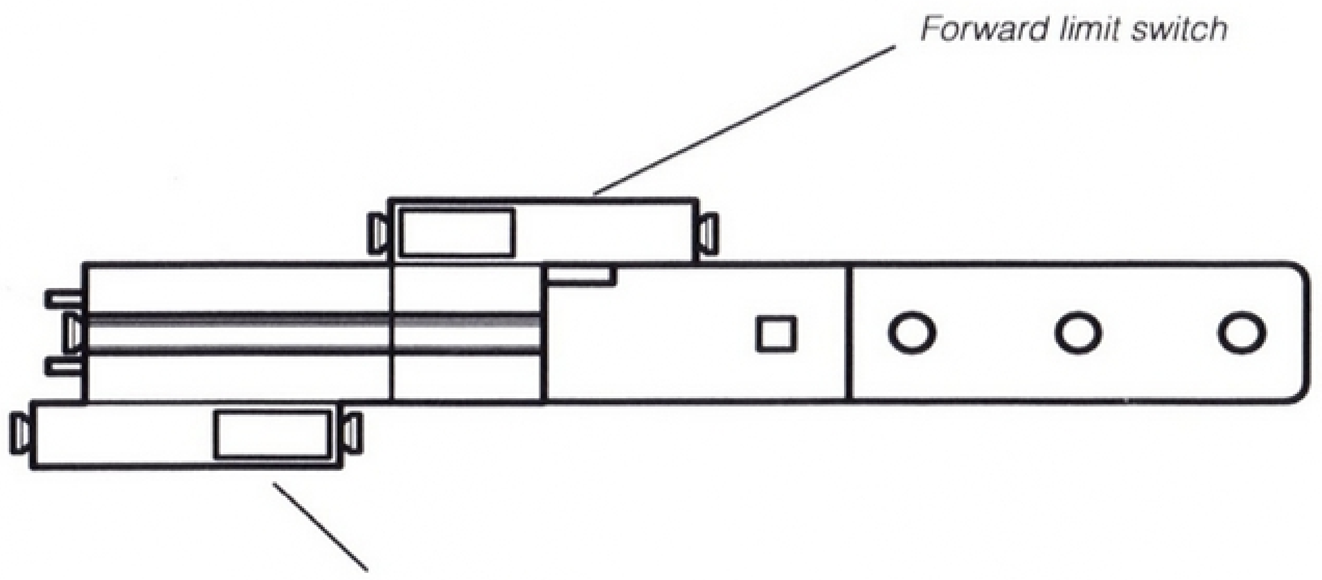
1 x 

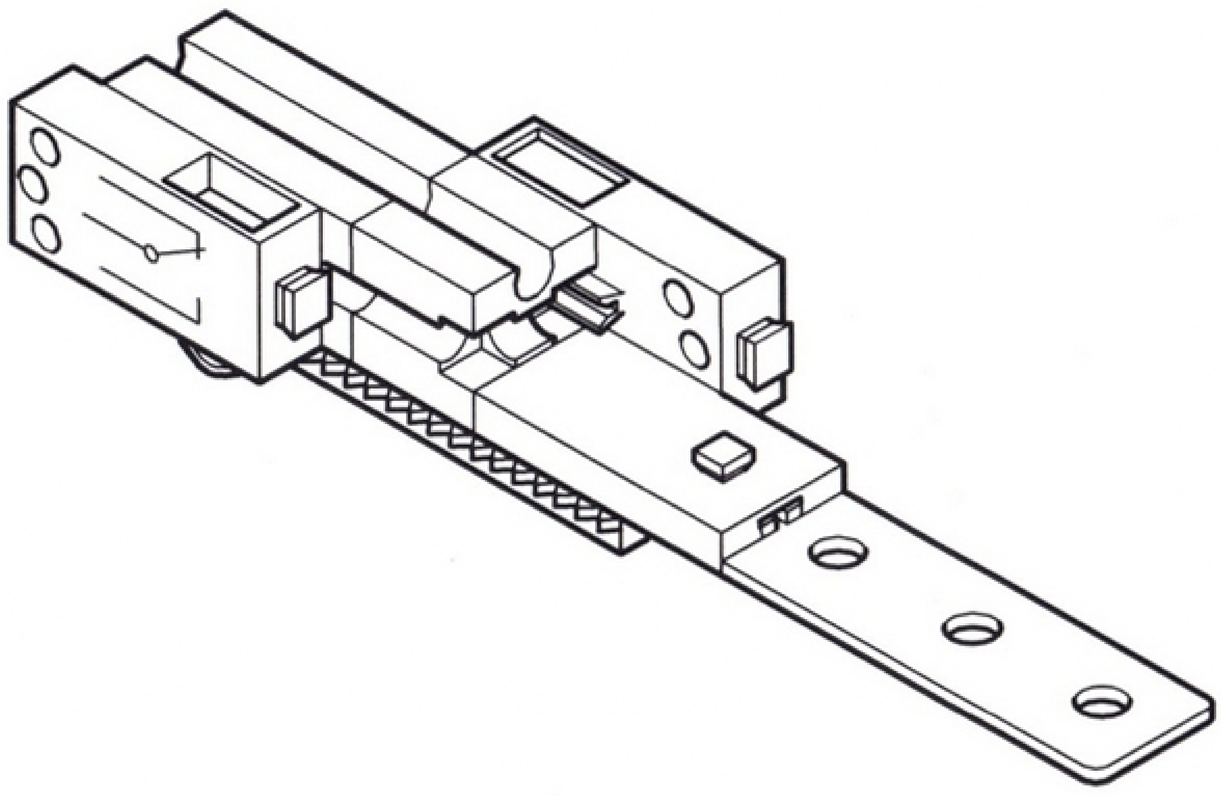
2 x 

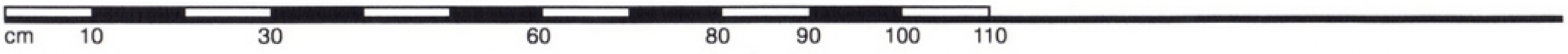
1)



2)

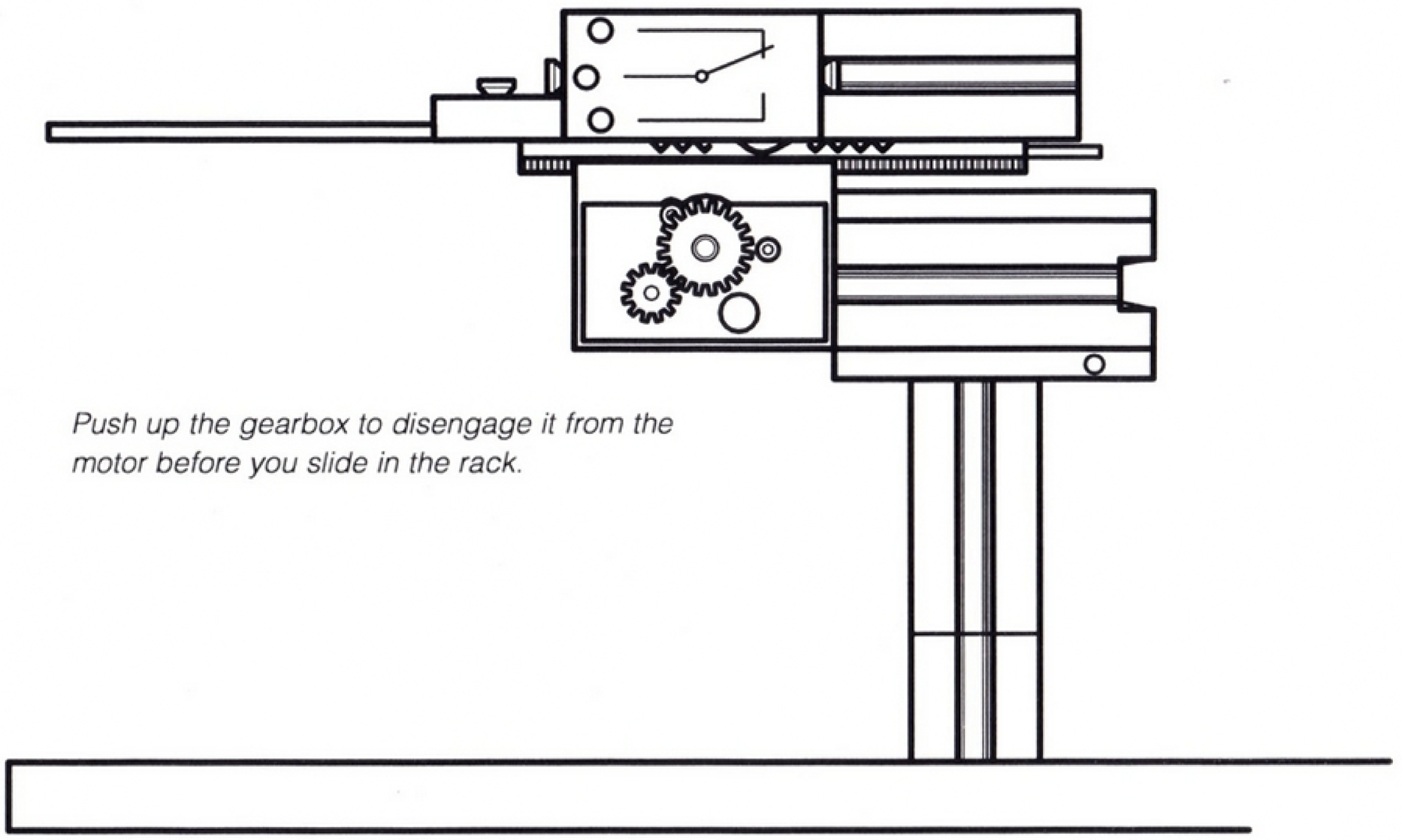




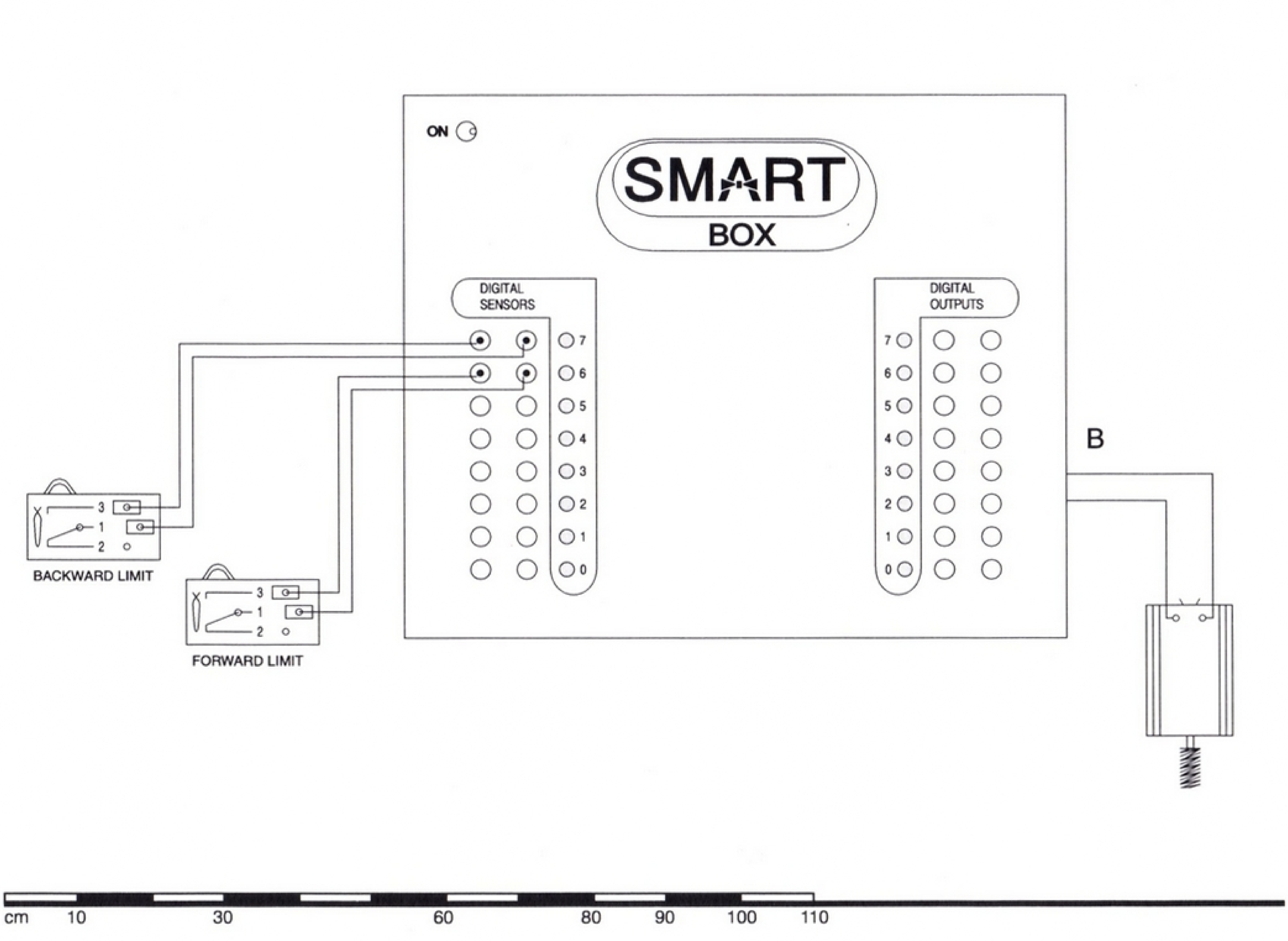


RACK AND PINION 2

3)



Push up the gearbox to disengage it from the motor before you slide in the rack.

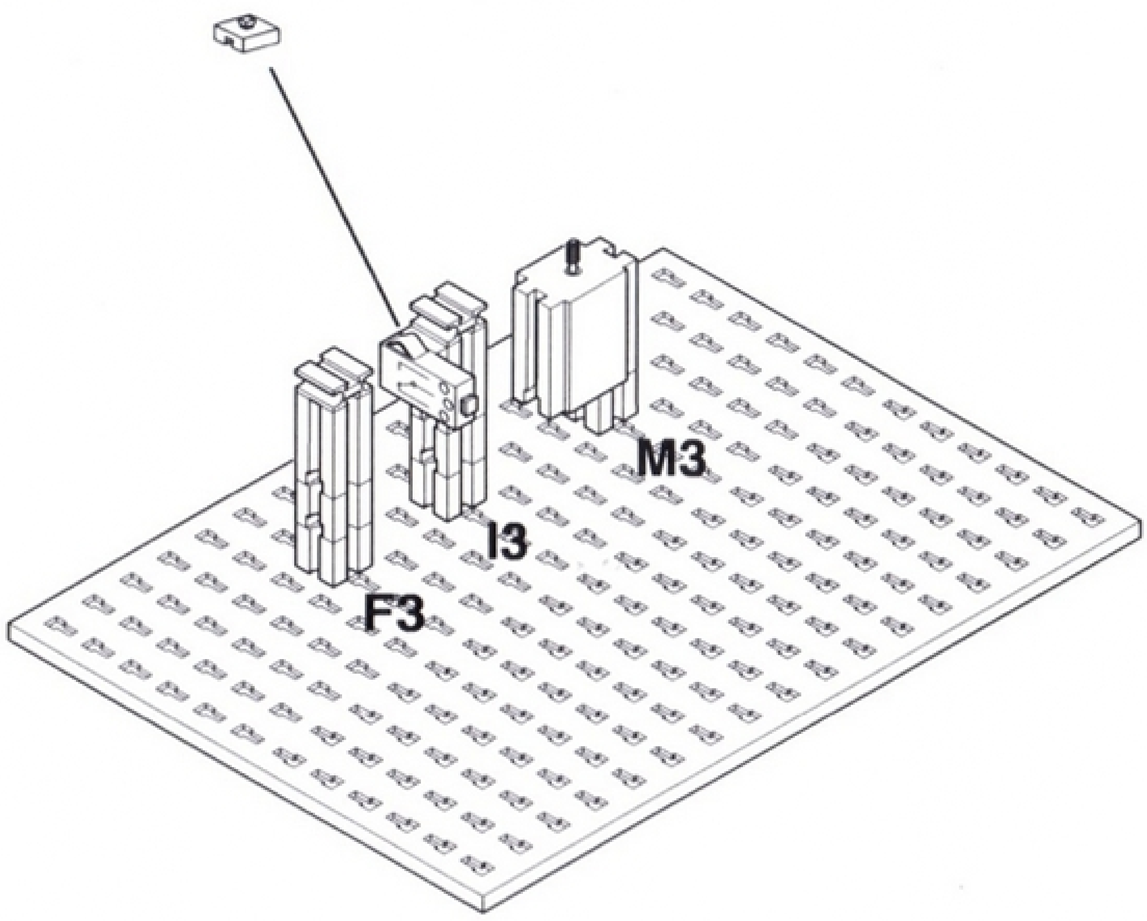


CAM AND FOLLOWER

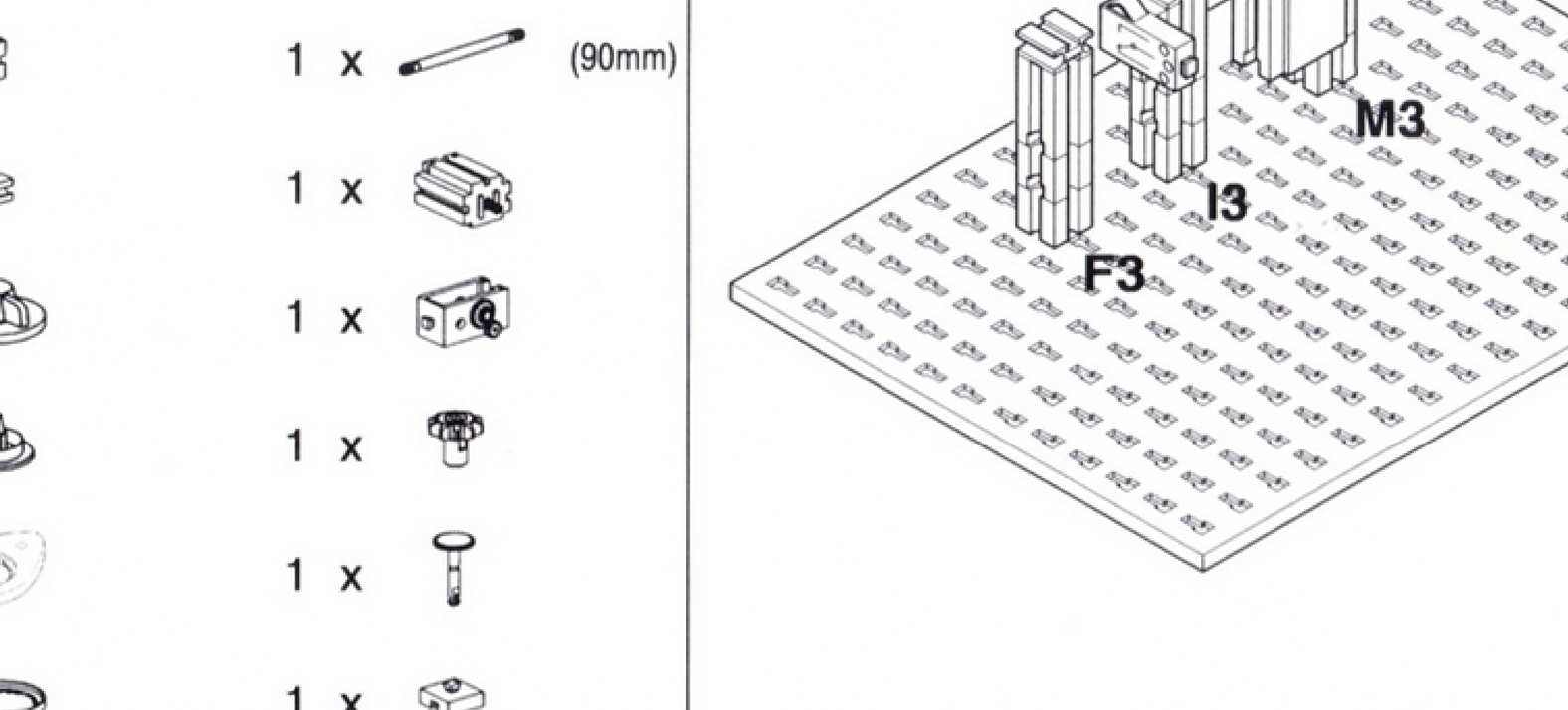
4 x
4 x
3 x
3 x
2 x
3 x
3 x
1 x
2 x

1 x
1 x
1 x (80mm)
1 x (90mm)
1 x
1 x
1 x
1 x
1 x
1 x

1)

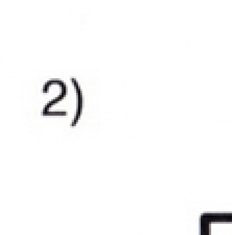


2)



This gear is used here to connect the 90mm clip axle to the gearbox.

Index Cam

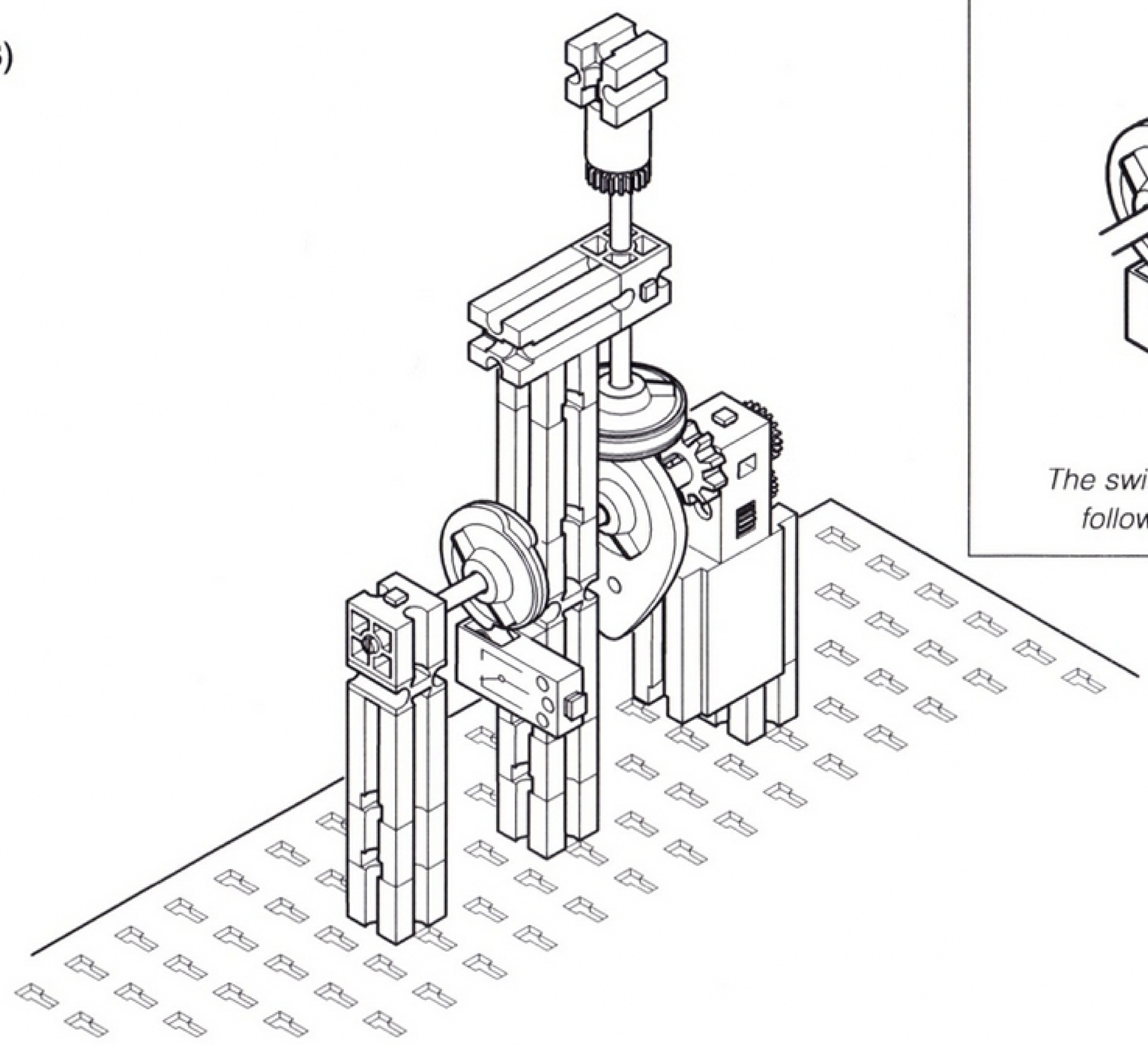


1) Place one cam section this way up on the hub collet.

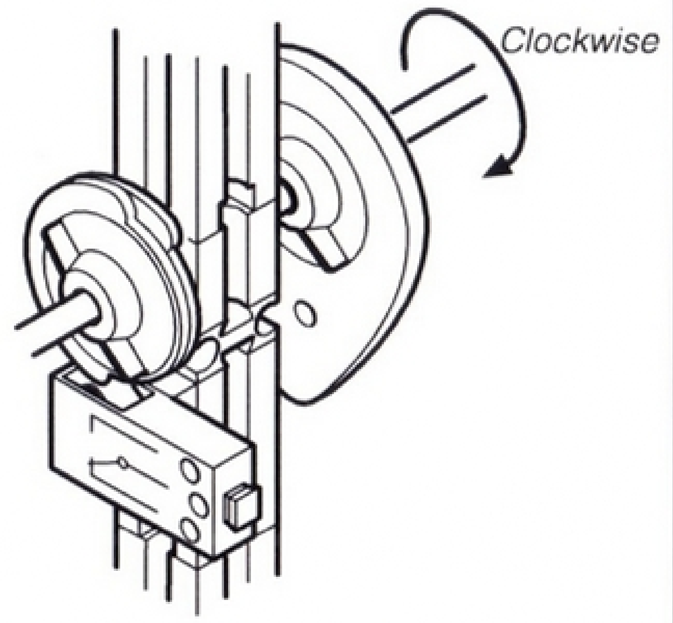
2) Overlap the second cam section and place it this way up. Screw on the hub nut.

CAM AND FOLLOWER

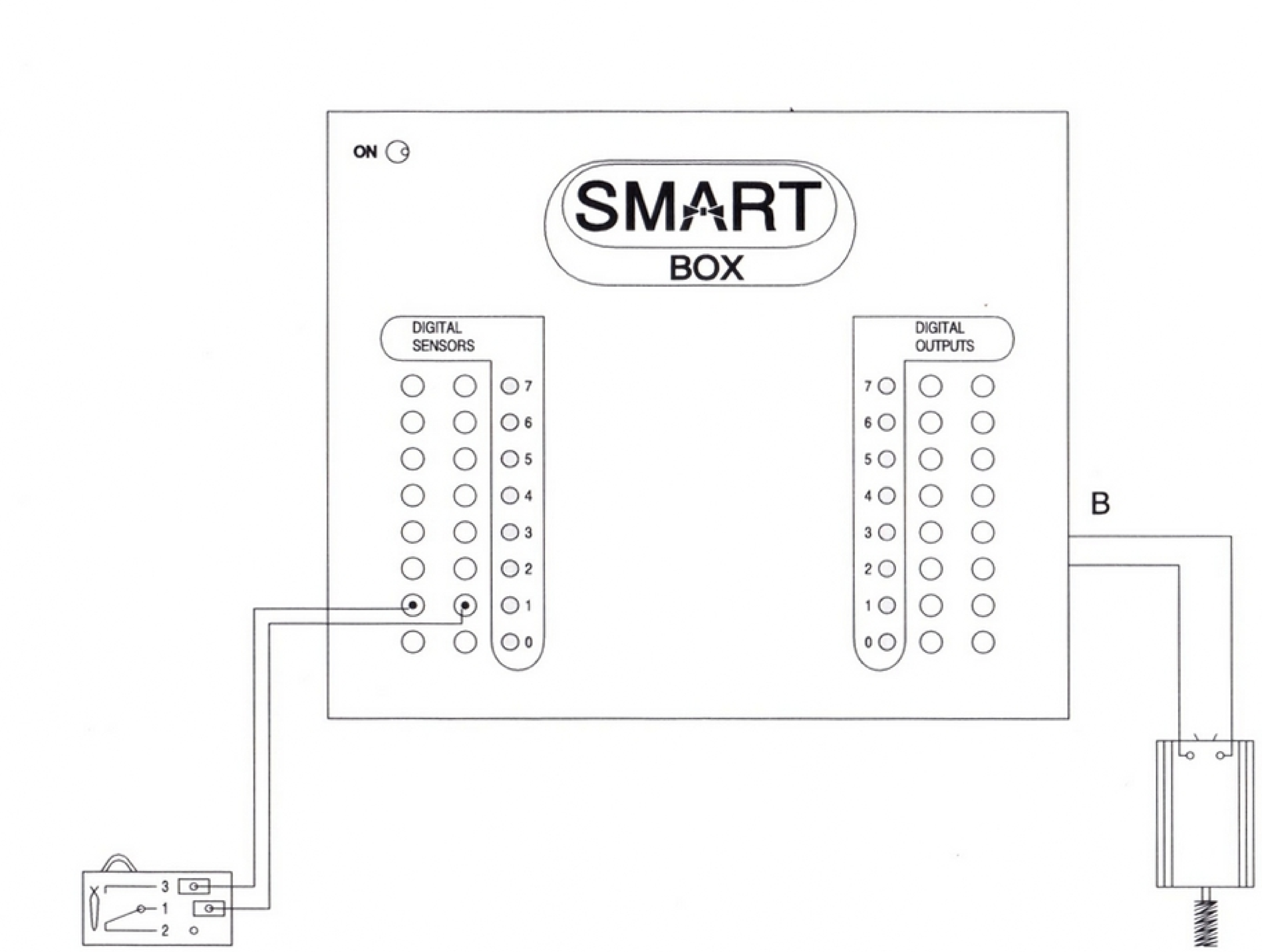
3)



Setting The Index Cam



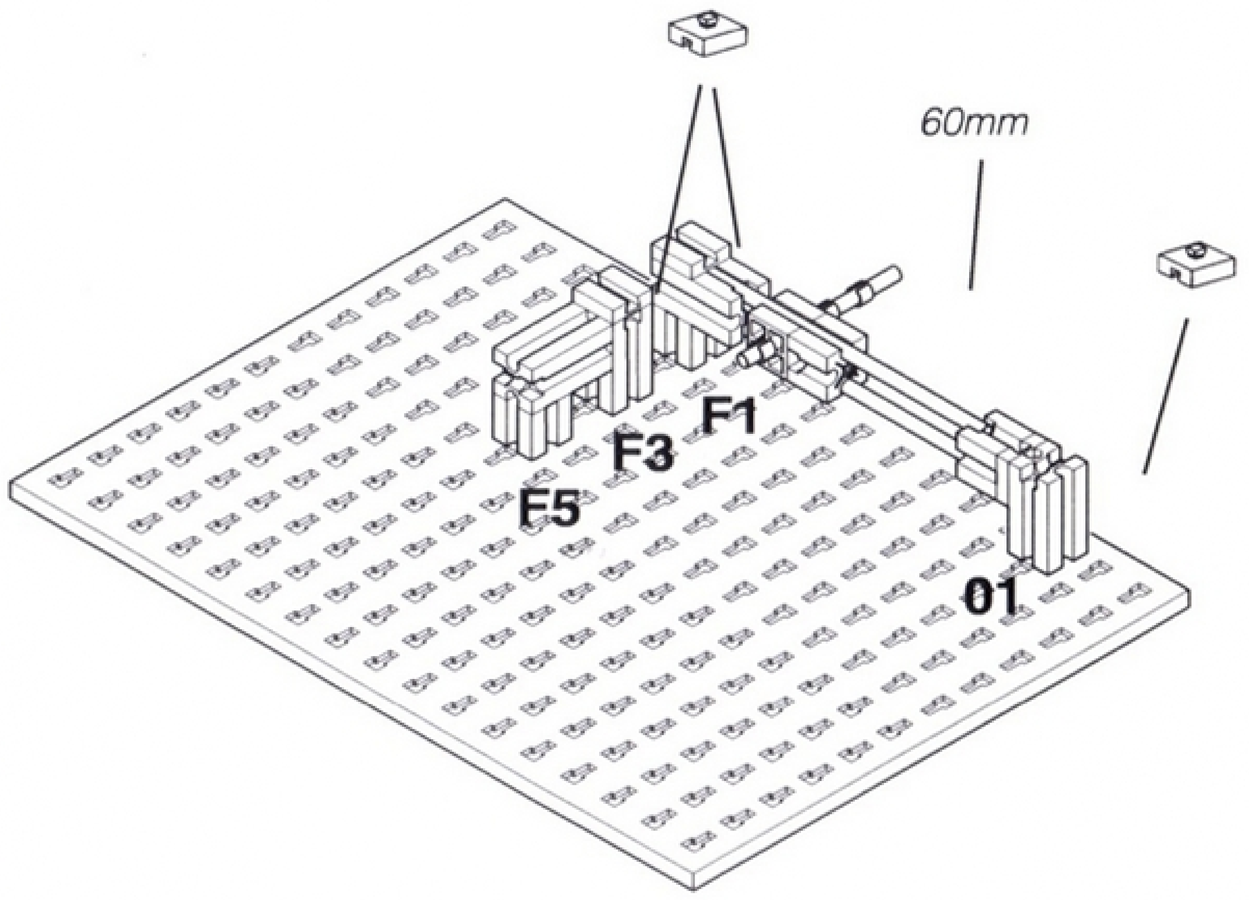
The switch is just pressed when the follower is in its down position.



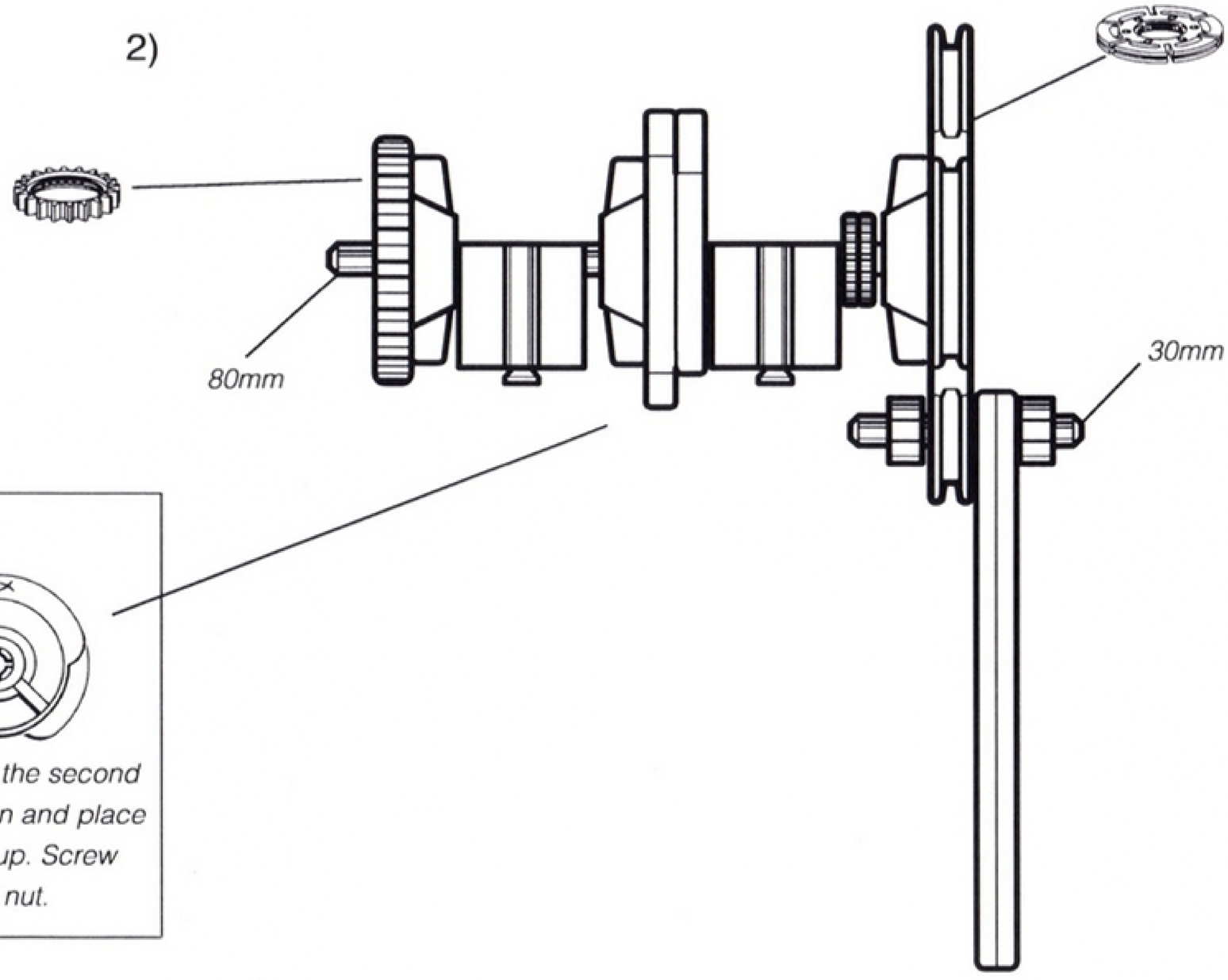
CRANK AND SLIDER

4 x		1 x	
4 x		1 x	
2 x		1 x	
1 x		2 x	(60mm)
4 x		2 x	(110mm)
3 x		1 x	(80mm)
3 x		1 x	(60mm)
1 x		1 x	(30mm)
1 x		2 x	
2 x		6 x	

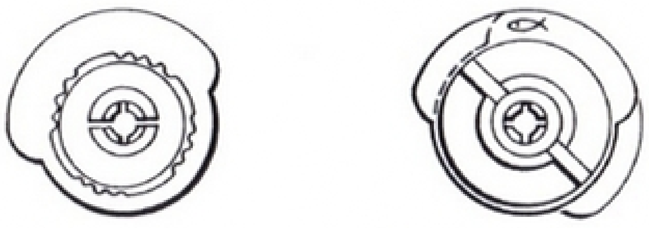
1)



2)

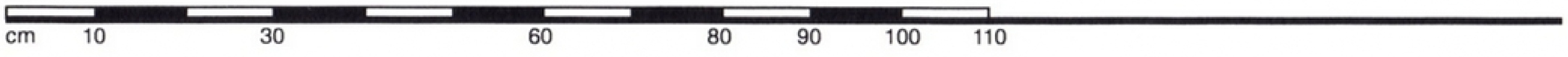


Index Cam



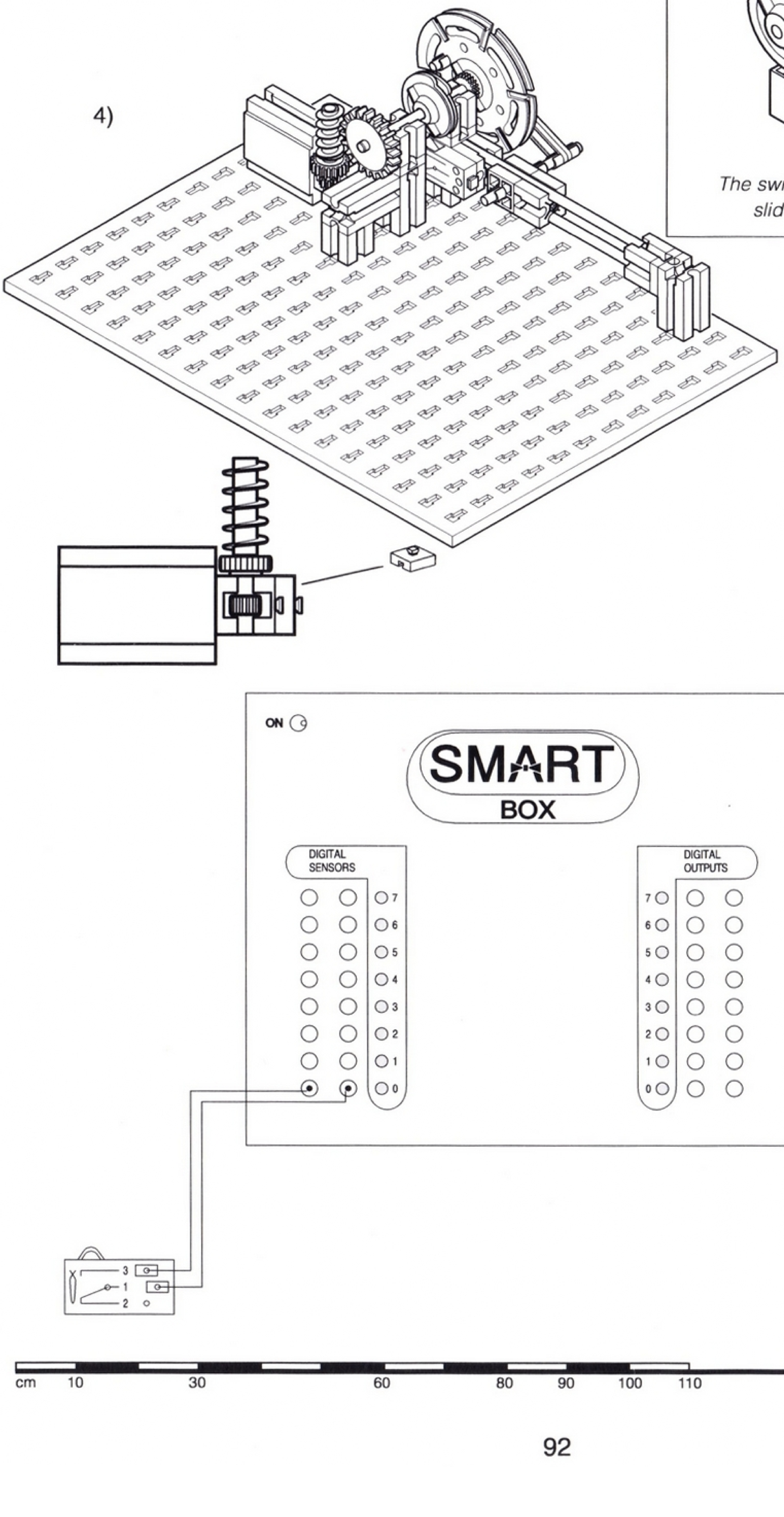
1) Place one cam section this way up on the hub collet.

2) Overlap the second cam section and place it this way up. Screw on the hub nut.

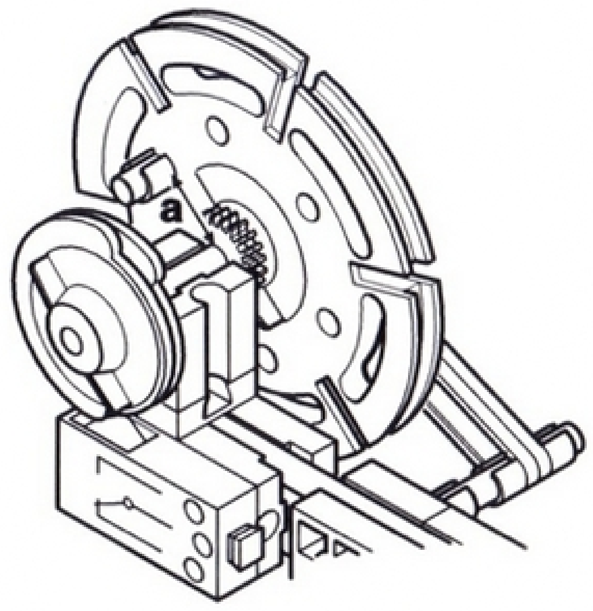


CRANK AND SLIDER

4)

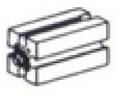
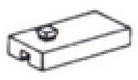


Setting The Index Cam

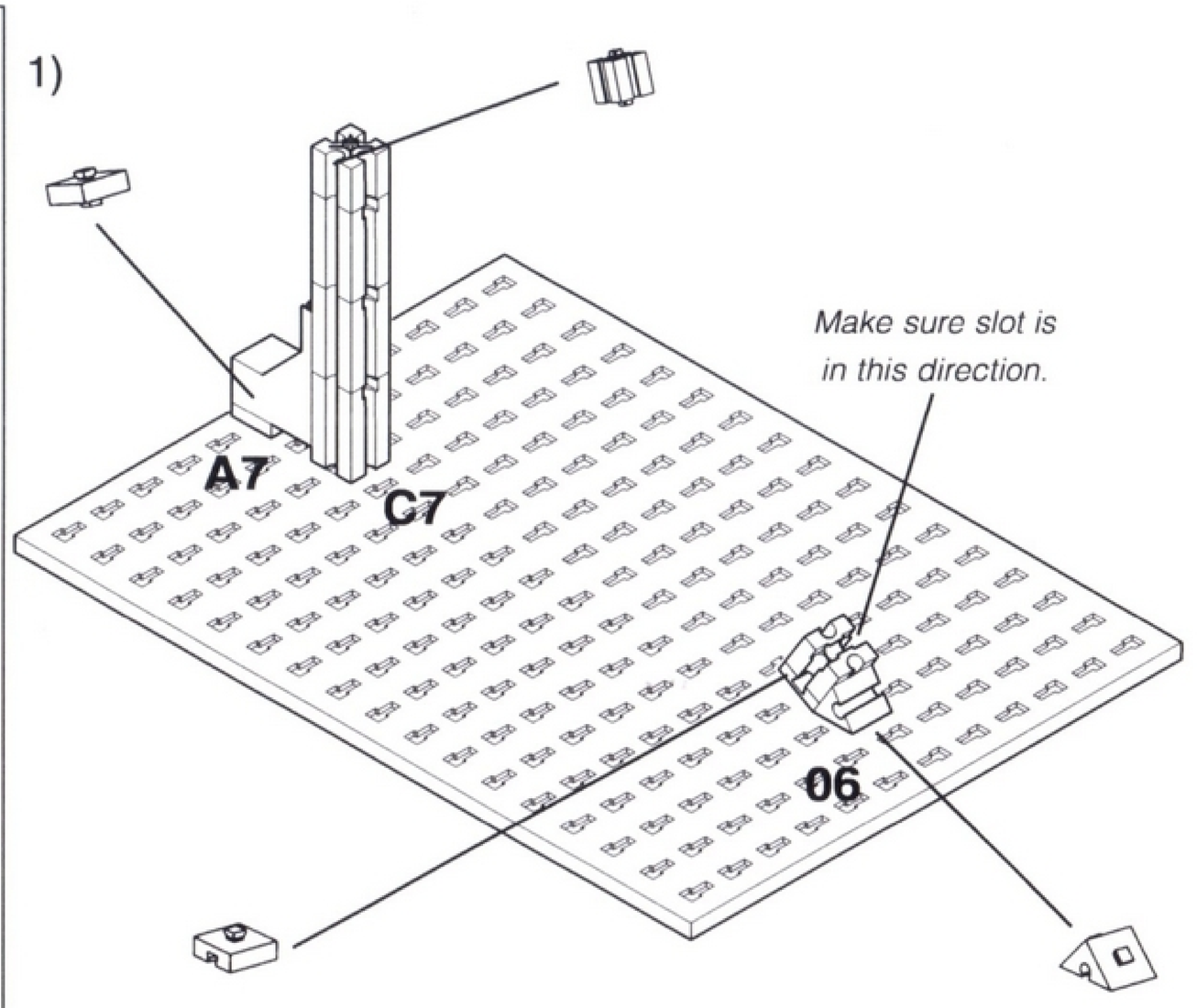


The switch is just pressed when the slider is in its back position.

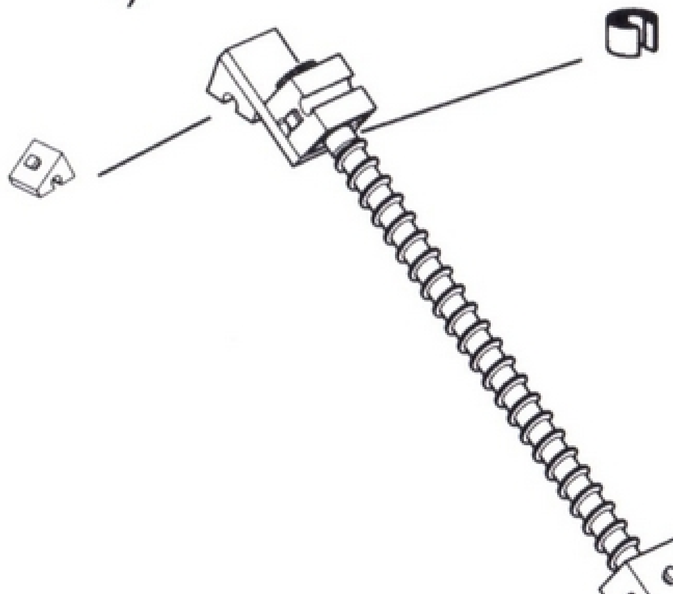
LEADSCREW

3 x		4 x	
2 x		1 x	
1 x		2 x	 (200mm)
1 x		4 x	
1 x		1 x	
1 x		1 x	
1 x		1 x	
6 x		1 x	 (30mm)
1 x		2 x	
1 x		1 x	
1 x		1 x	
1 x		2 x	

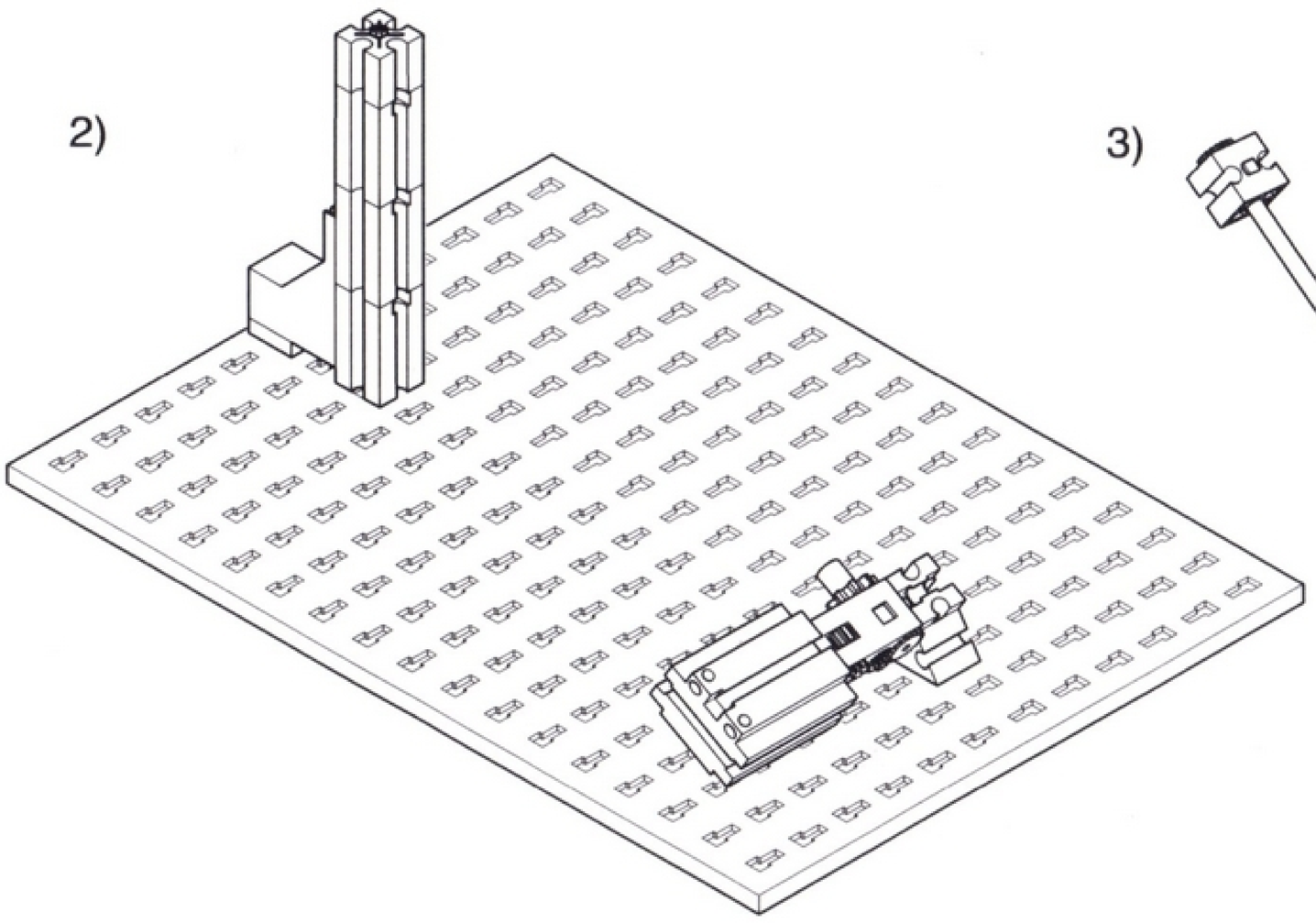
1)



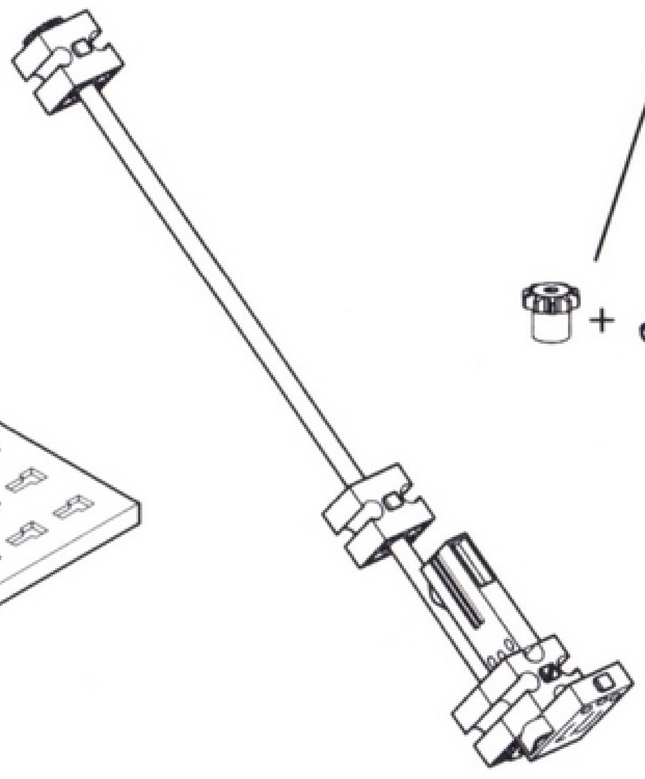
4)



2)

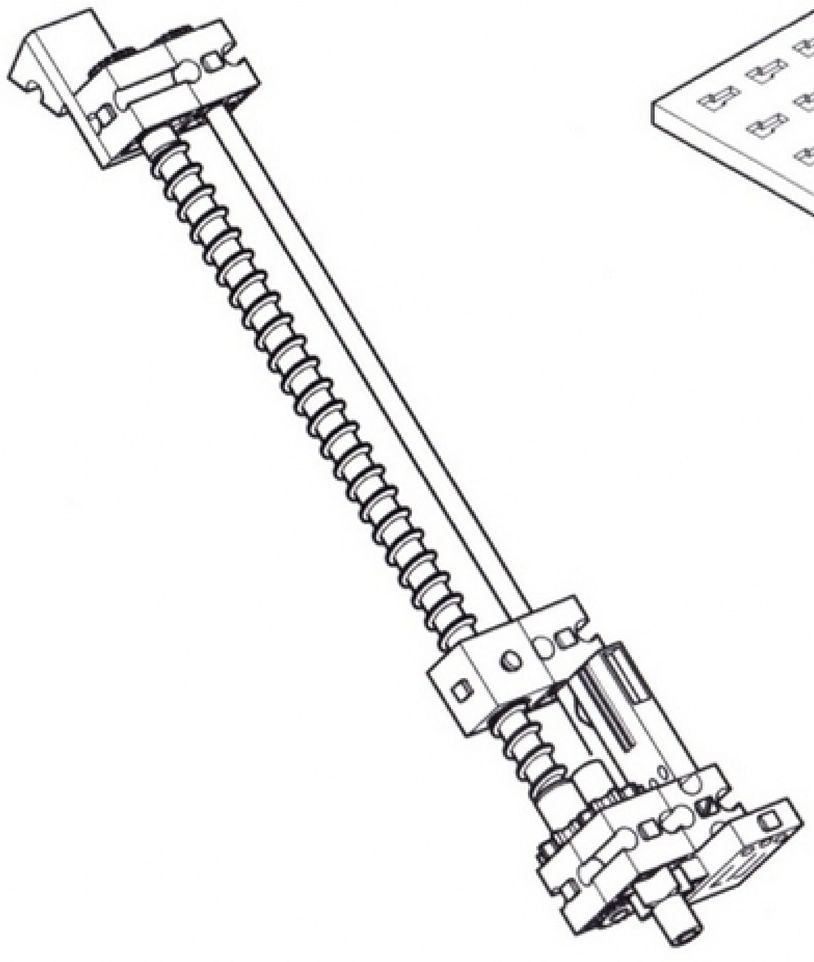


3)

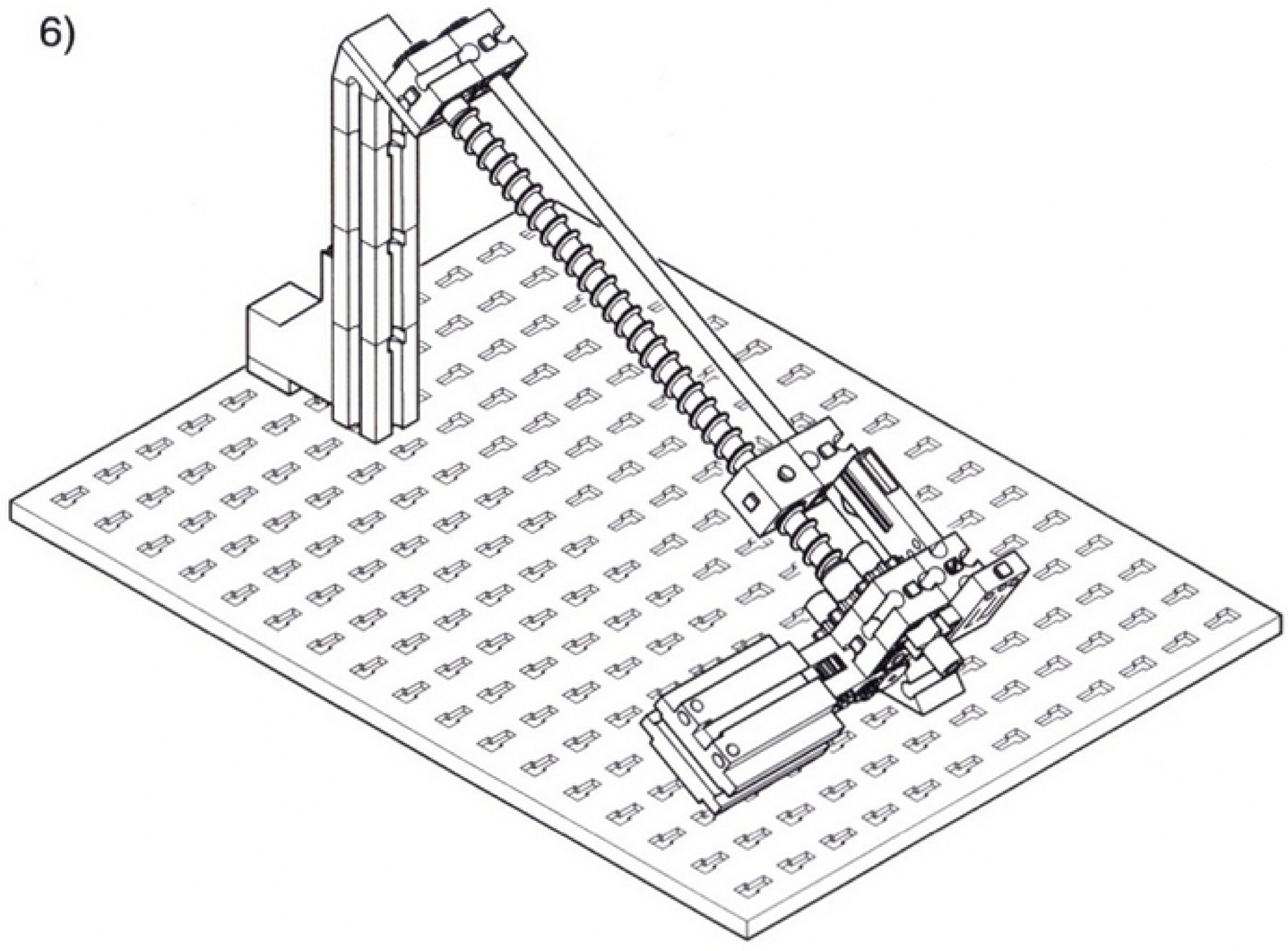


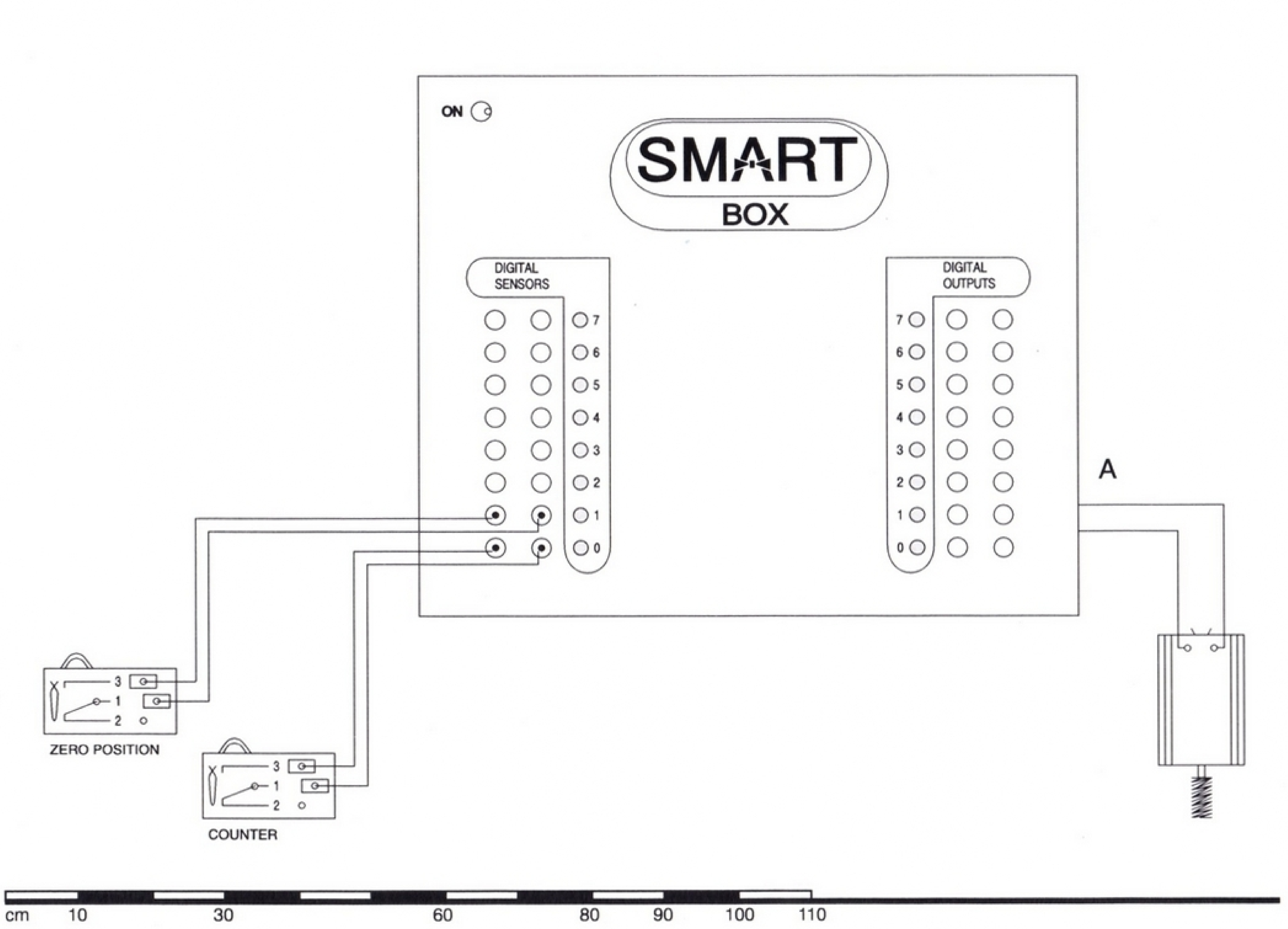
LEADSCREW

5)

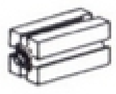


6)

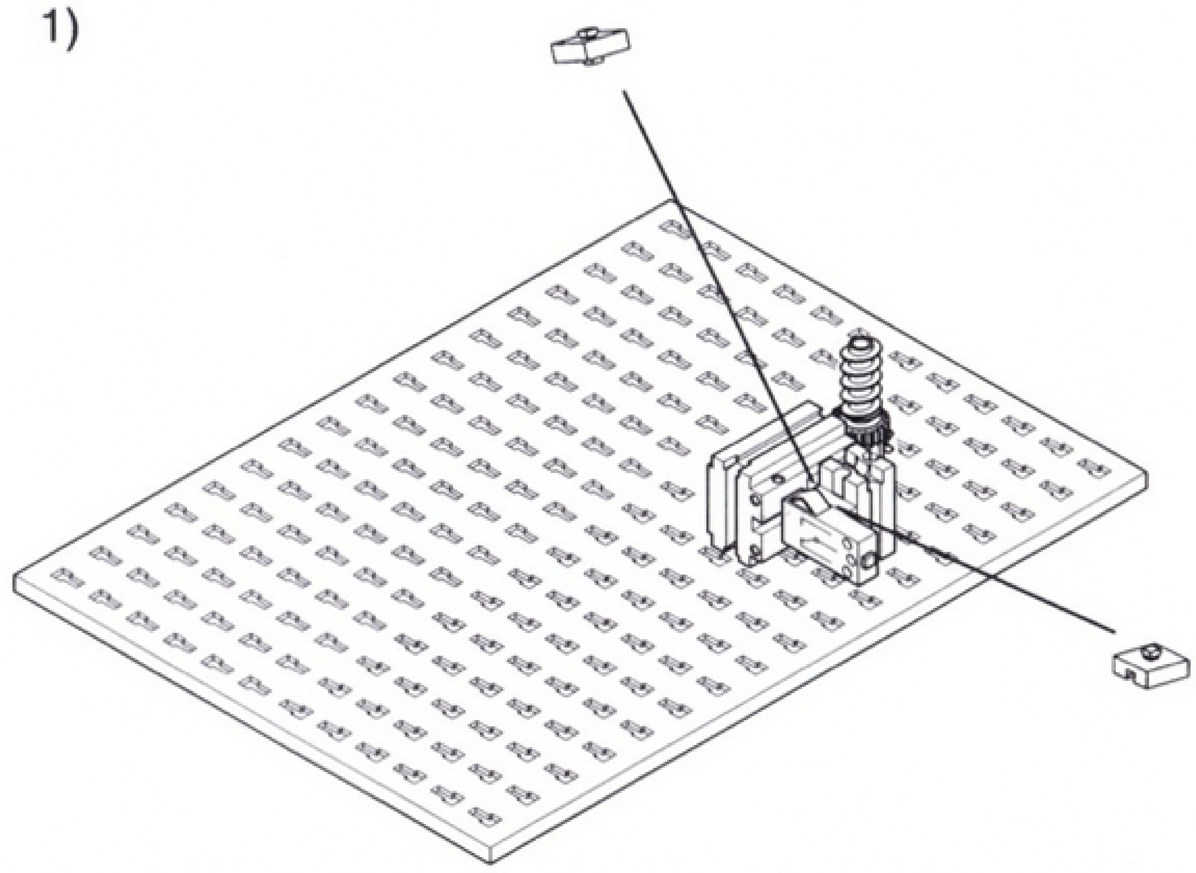




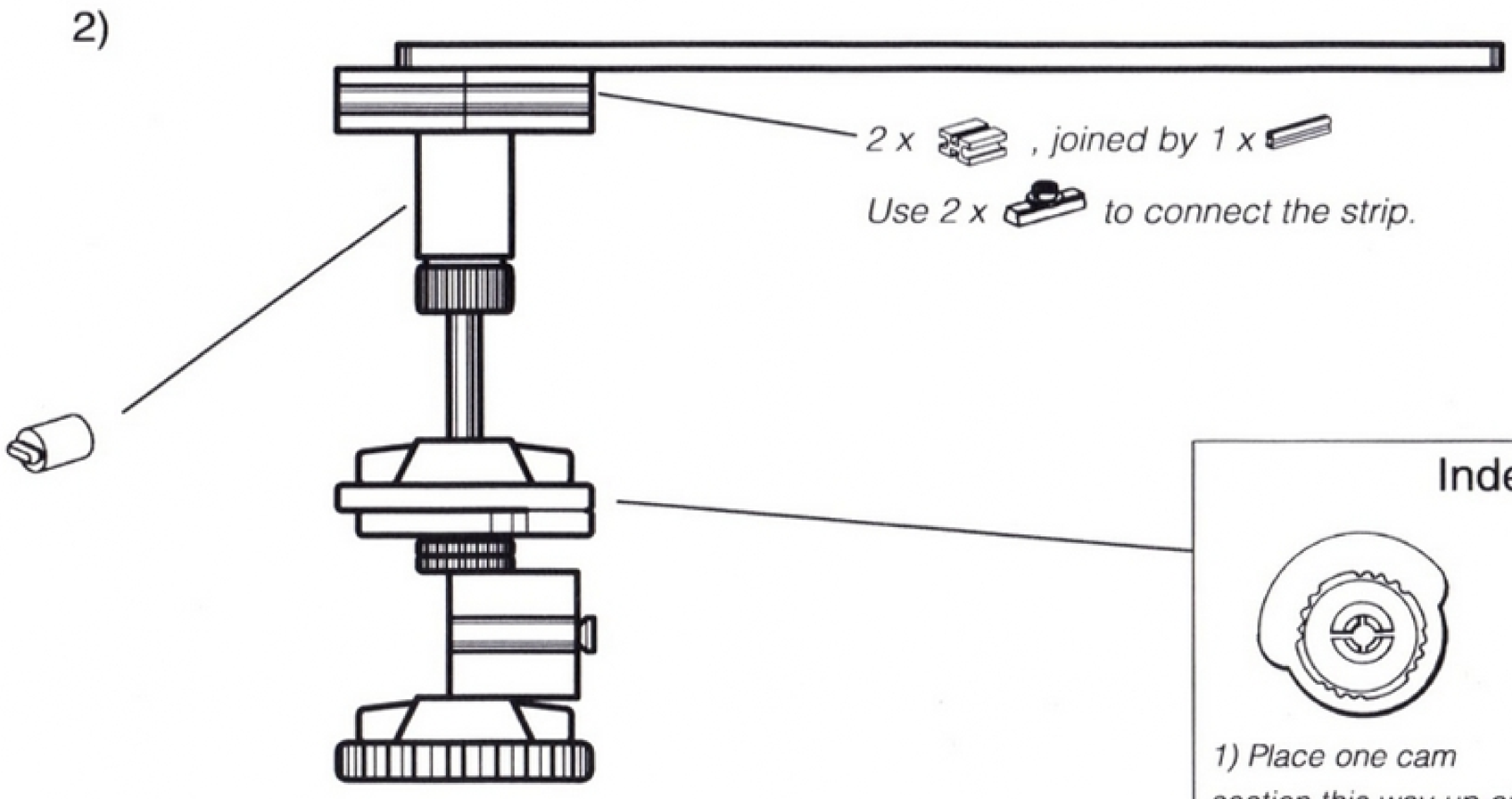
CAR PARK BARRIER

1 x		2 x		
1 x		1 x		(30mm)
1 x		1 x		(80mm)
1 x		1 x		(120mm)
2 x		1 x		
2 x		1 x		
2 x		1 x		
2 x		1 x		
1 x		1 x		
2 x				

1)

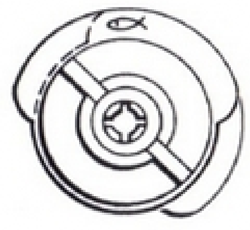


2)



Index Cam

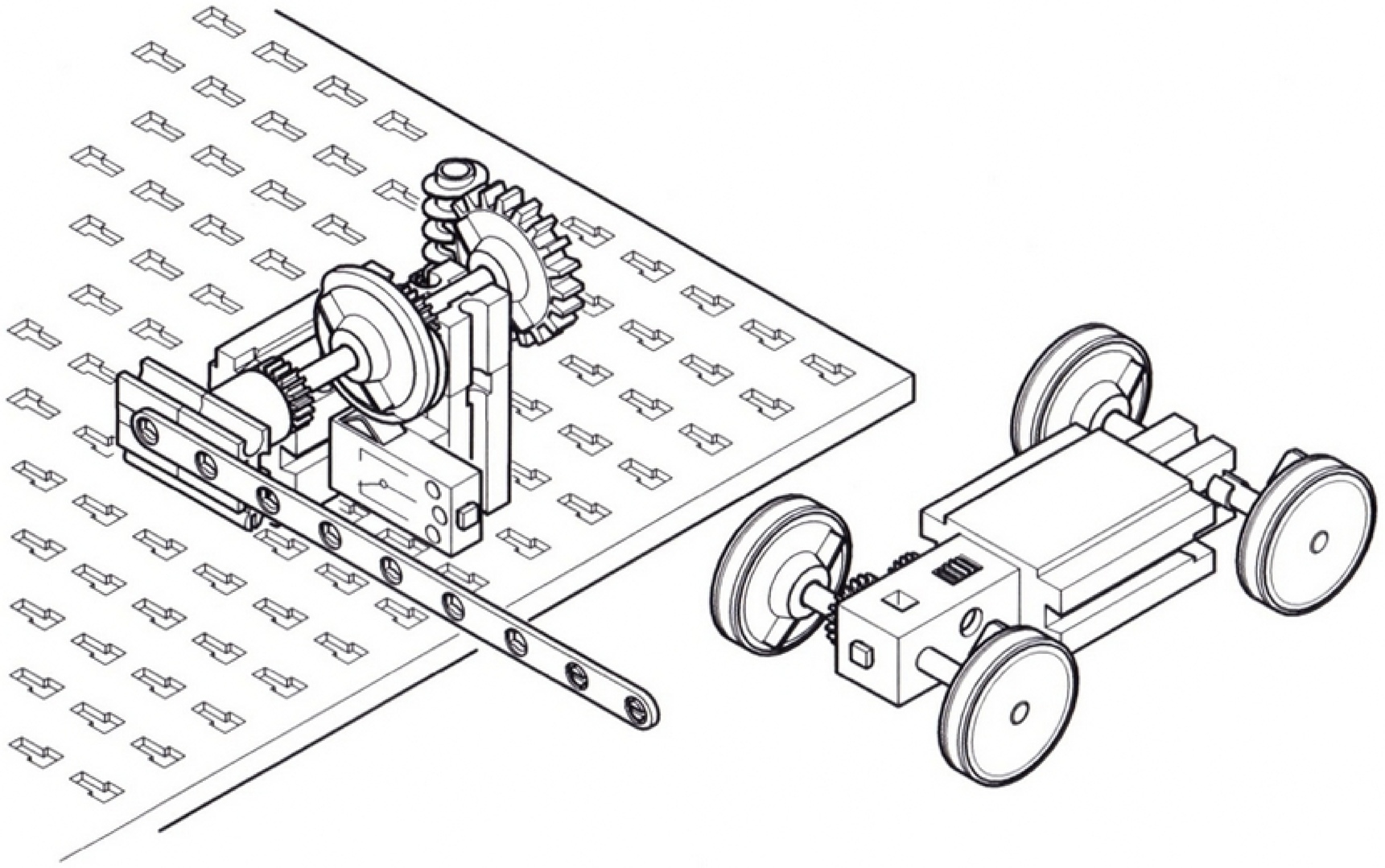
1) Place one cam section this way up on the hub collet.



2) Overlap the second cam section and place it this way up. Screw on the hub nut.

CAR PARK BARRIER

3)

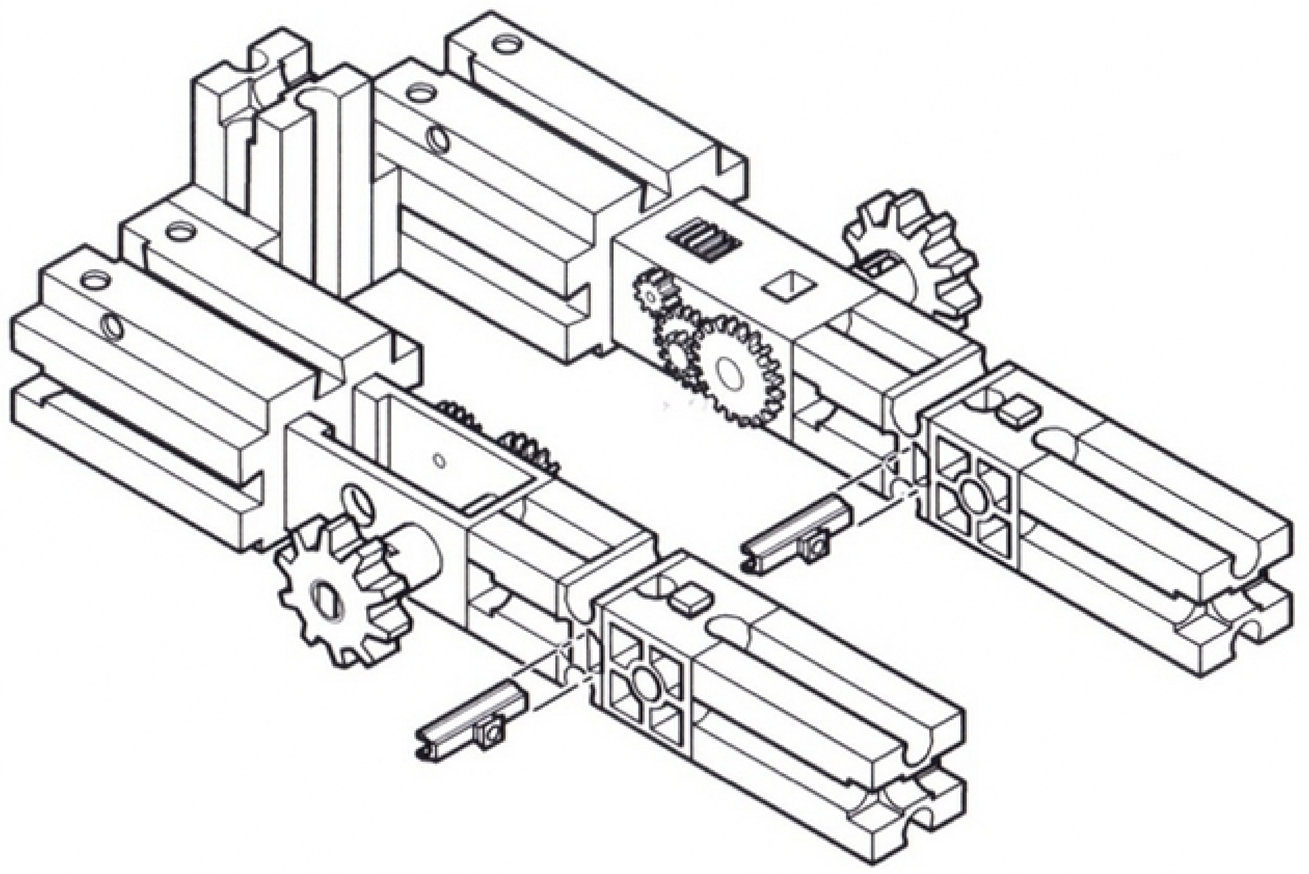


Single Motor Buggy

BUGGY

4 x		2 x	
2 x		2 x	
2 x		2 x	
1 x		2 x	
3 x		2 x	
2 x		1 x	
2 x		2 x	
2 x		4 x	
2 x	(60mm)	3 x	
2 x			

Part One
1)

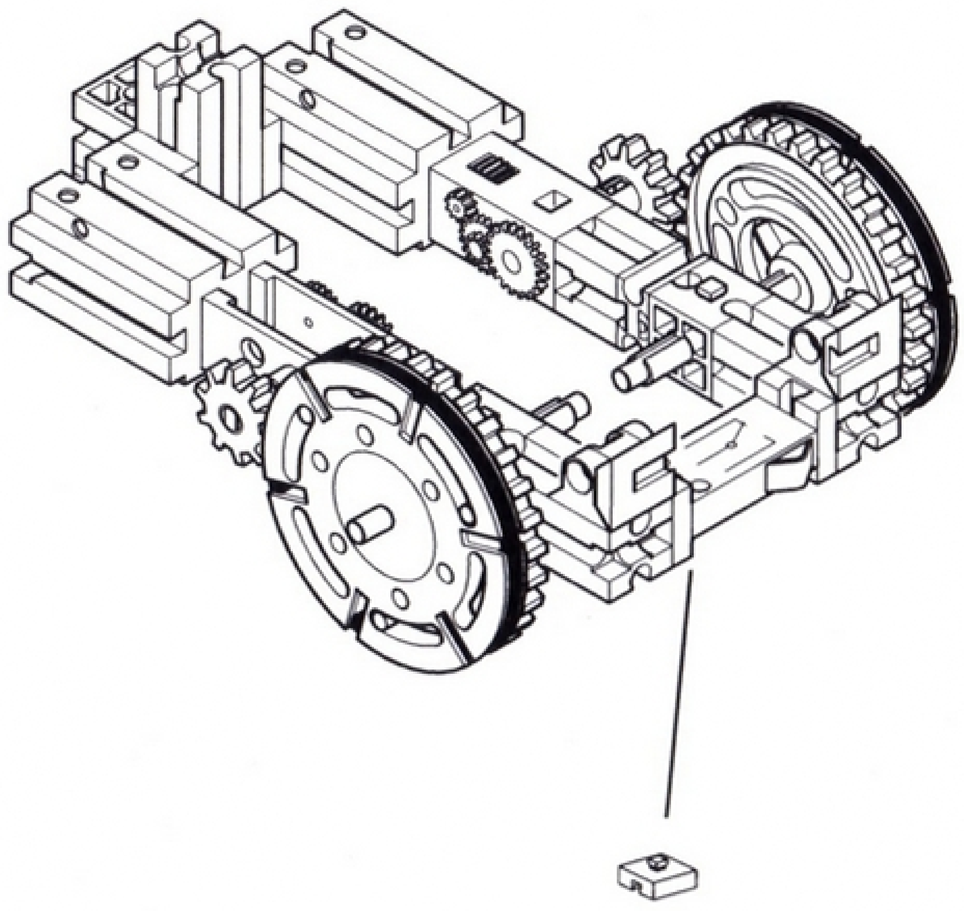


2)

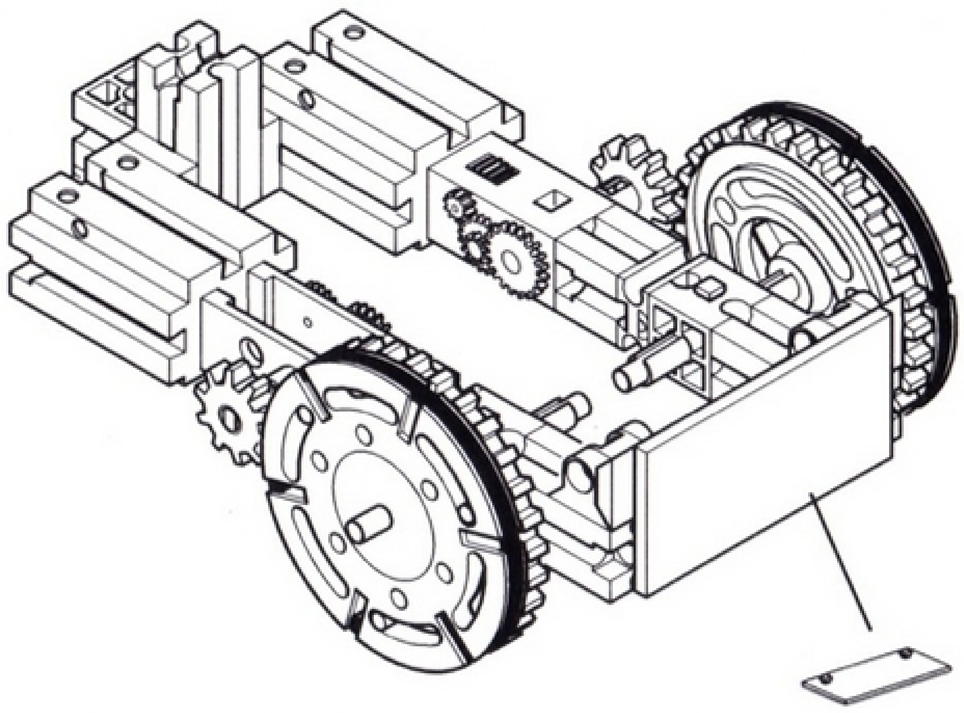
cm 10 30 60 80 90 100 110

BUGGY

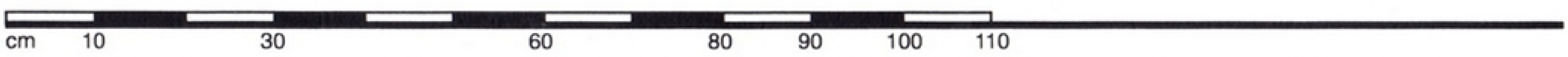
Part Two
1)



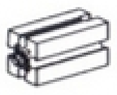
2)



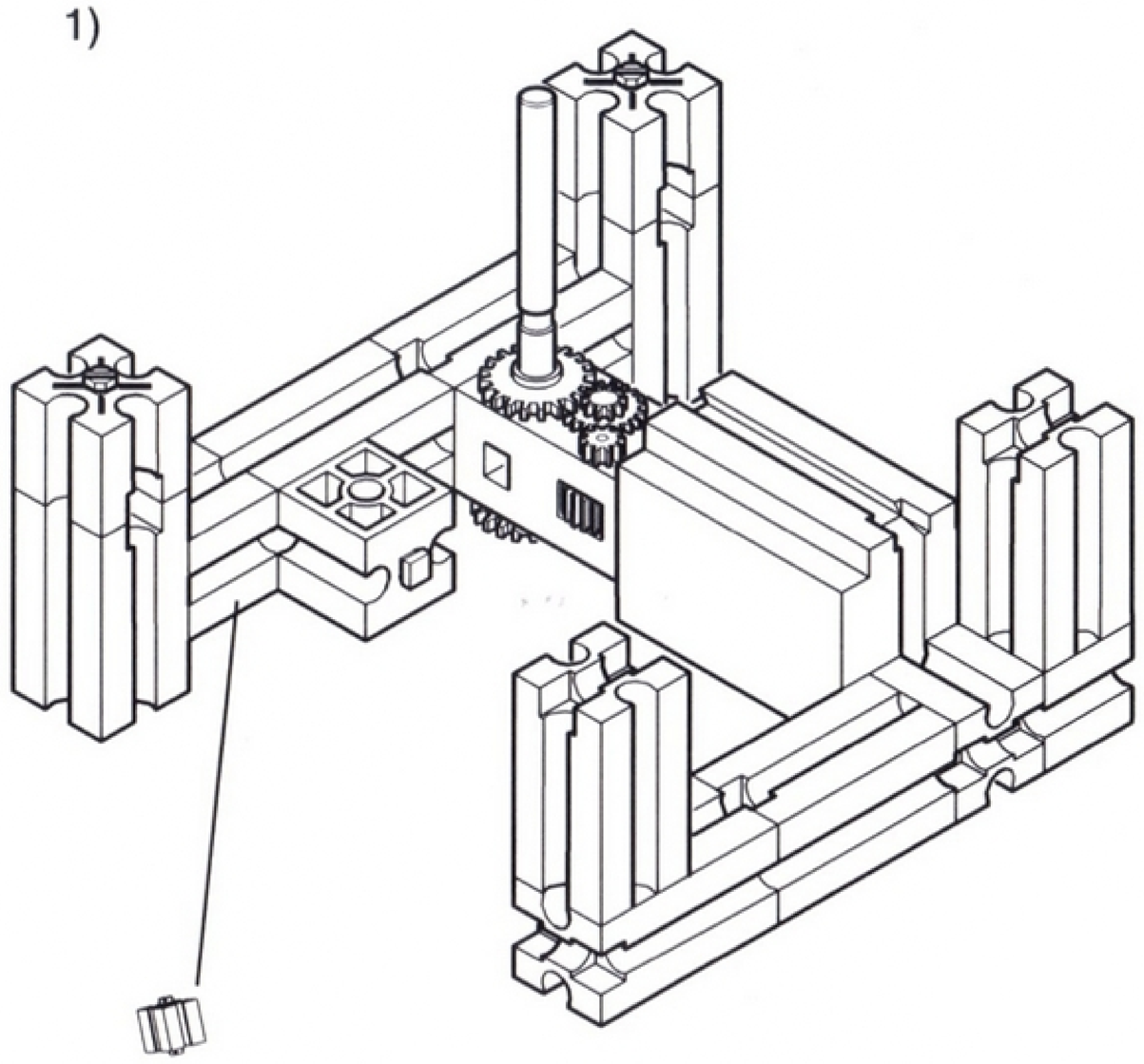
Part Three



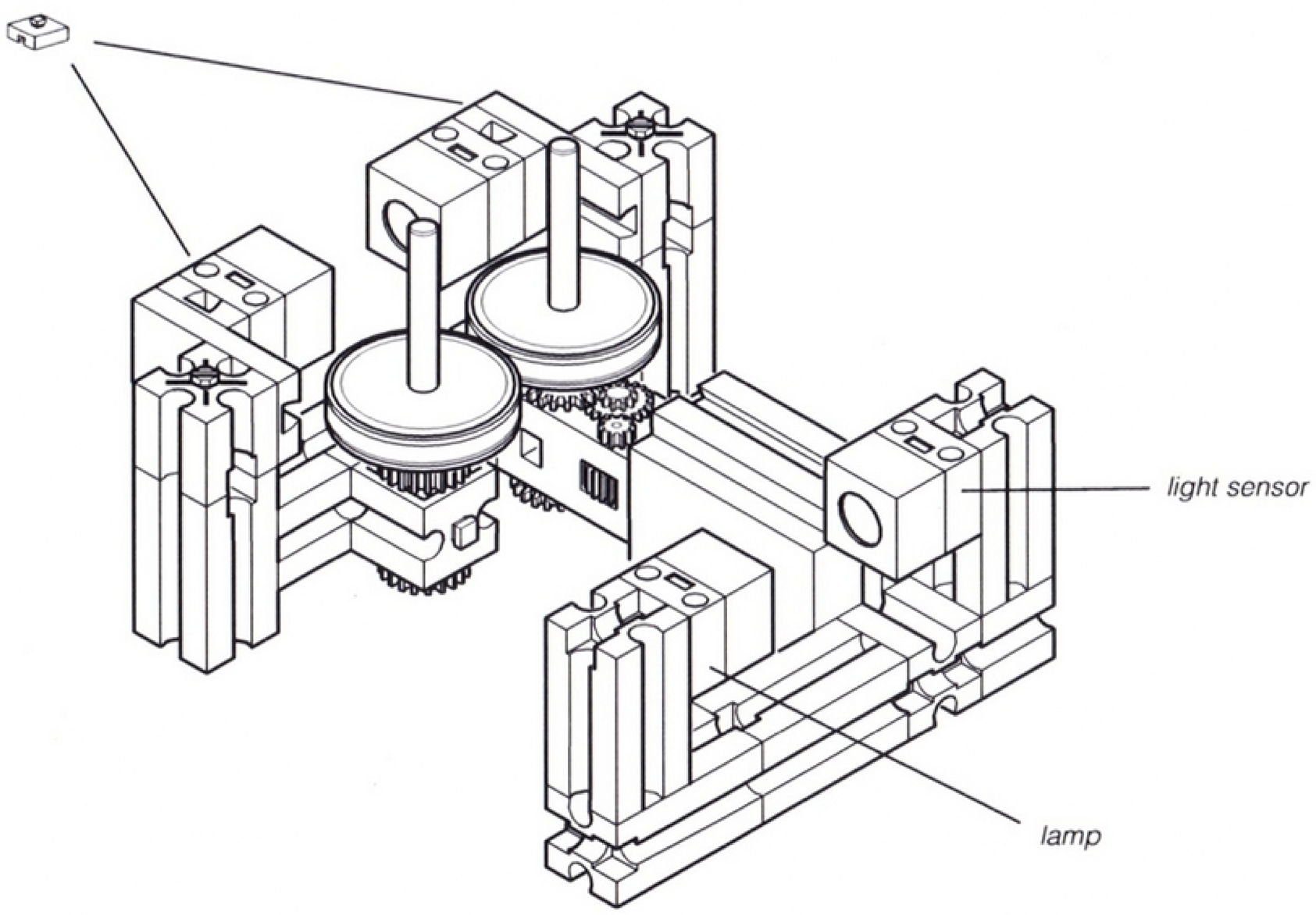
READER

8 x		3 x	
4 x		1 x	 (60mm)
1 x		1 x	
1 x		2 x	
1 x		2 x	
1 x		4 x	
2 x		2 x	
2 x		2 x	
2 x			

1)



2)



DISPENSER

4 x

1 x

5 x

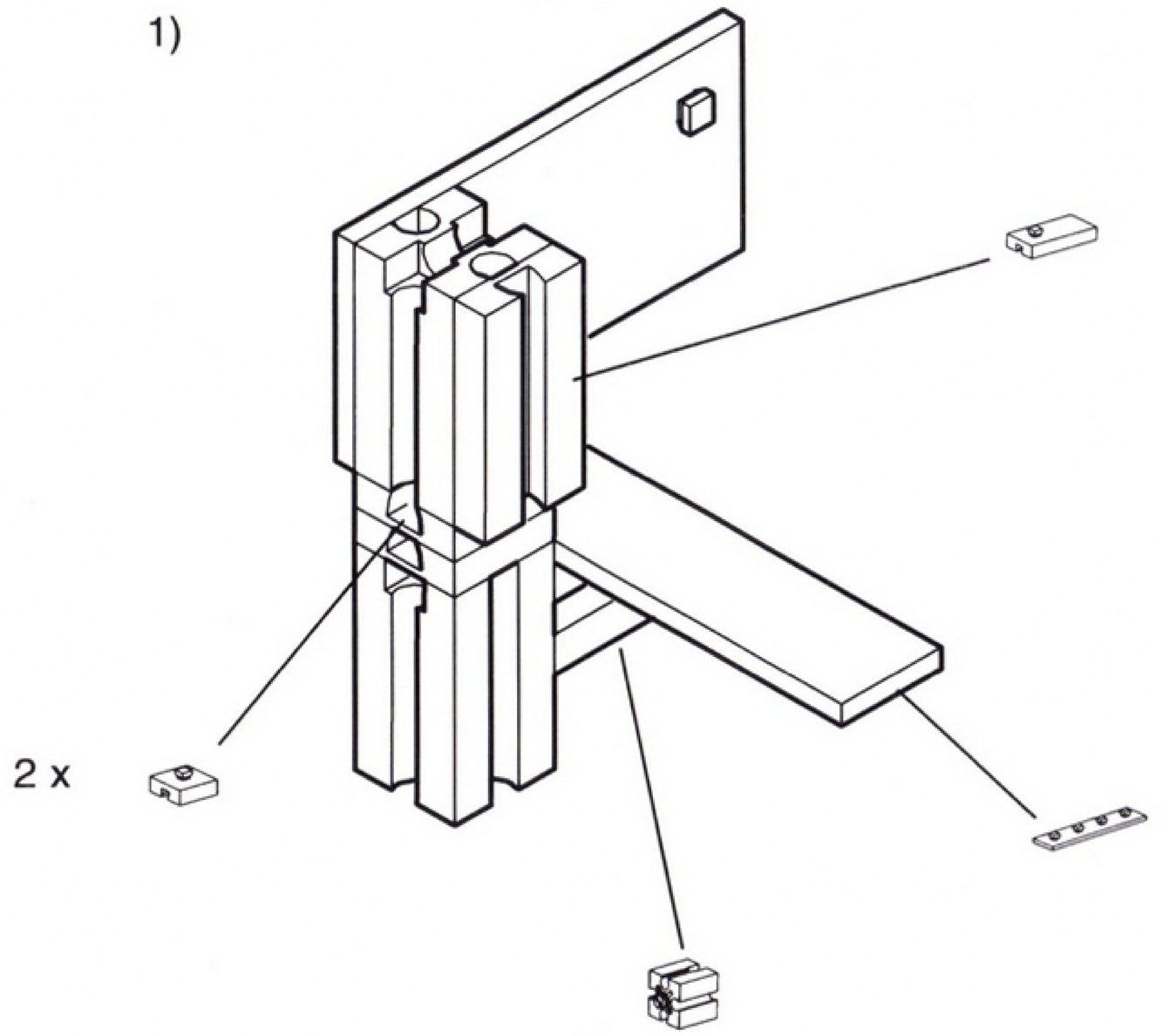
3 x

2 x

1 x

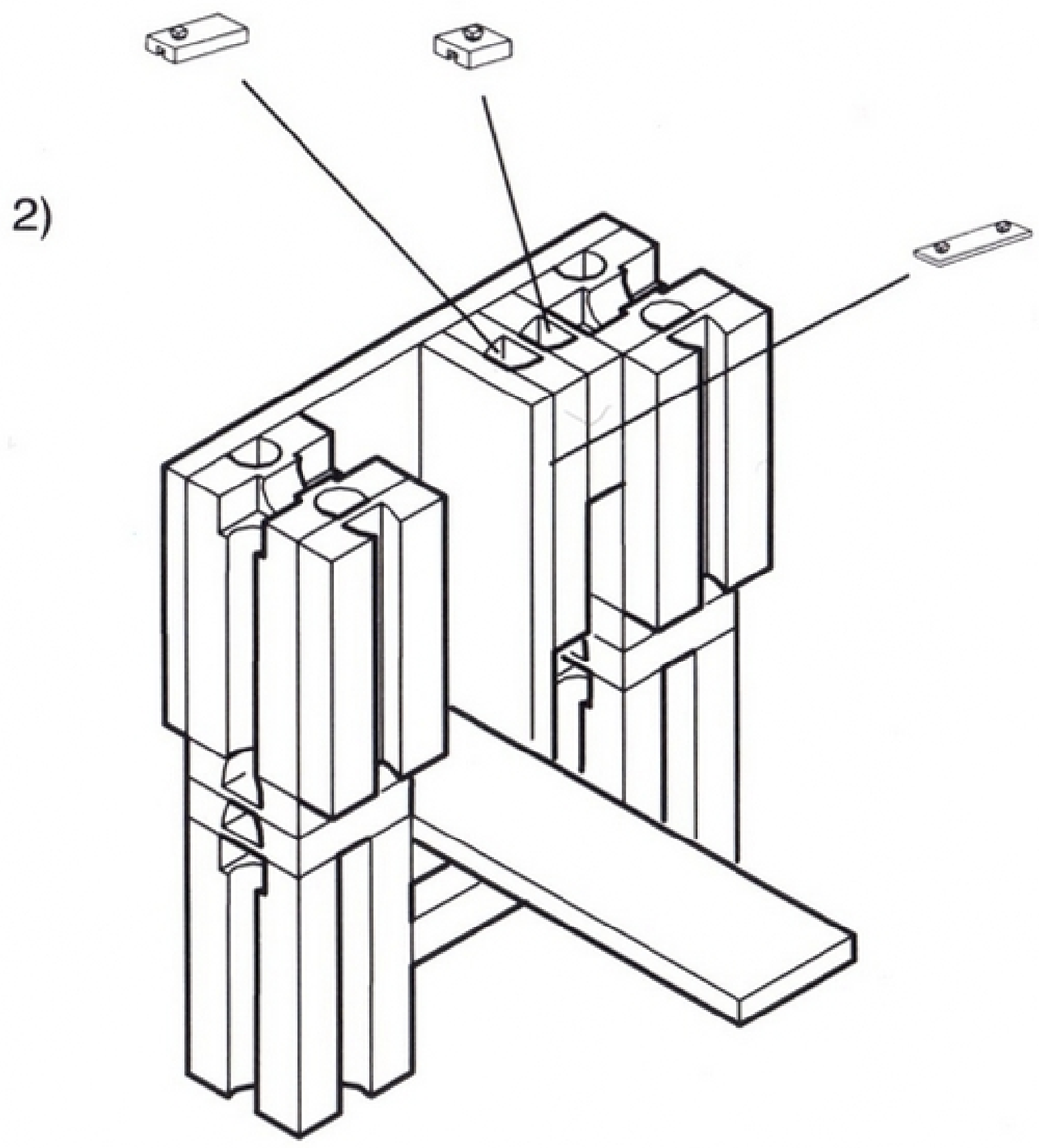
1 x

1)

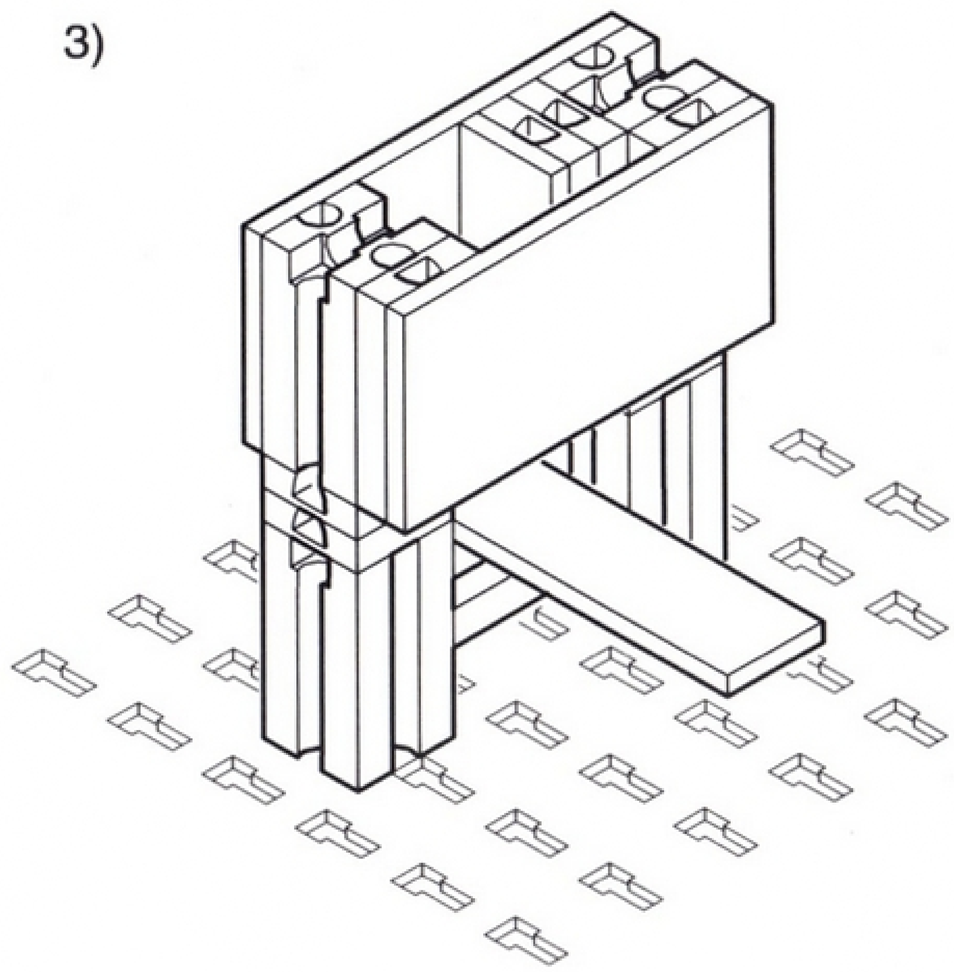


2 x

2)



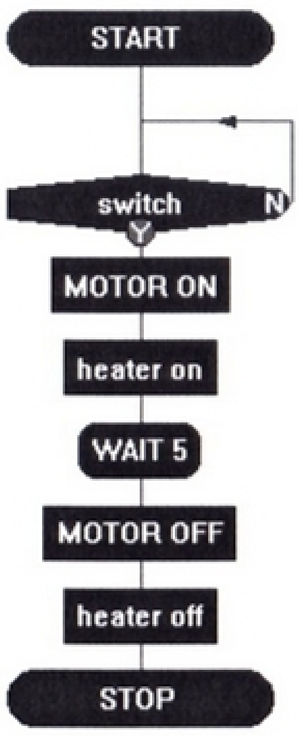
3)



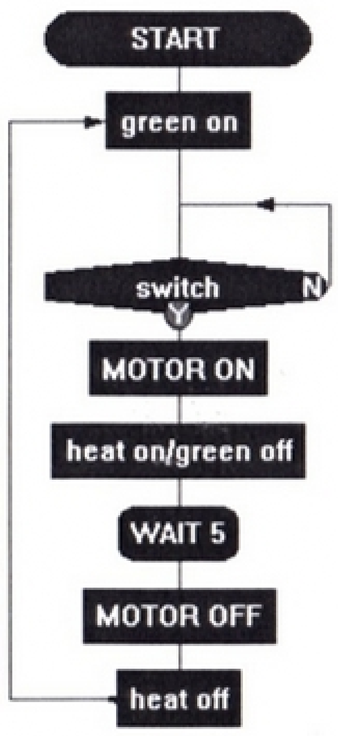
Suggested Solutions

SUGGESTED SOLUTIONS - LOGICATOR

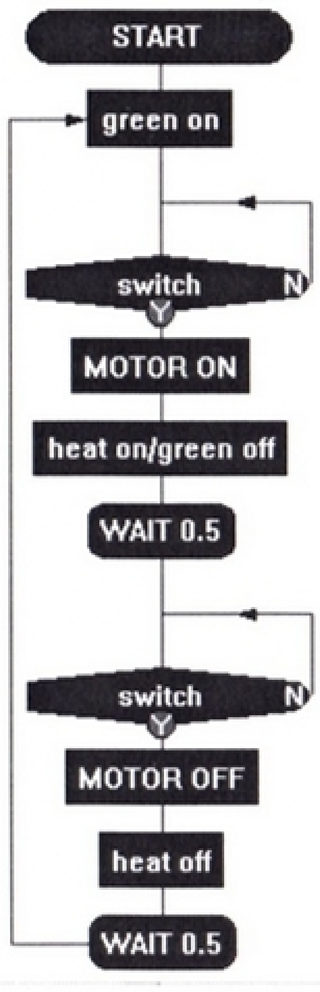
Start Up 5: Design your own system



Addition 1



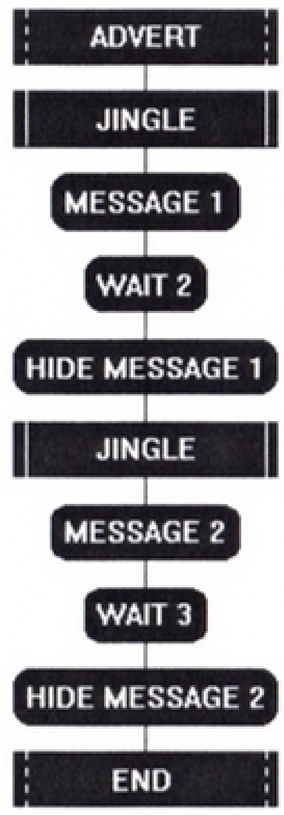
Addition 2



Start Up 6: Building a complex system

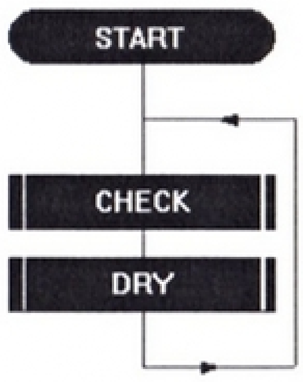
Addition

The new version of the ADVERT macro is shown below; the rest of the flowsheet is unchanged from that shown in fig 14.



Start Up 7: Responding to a sensor

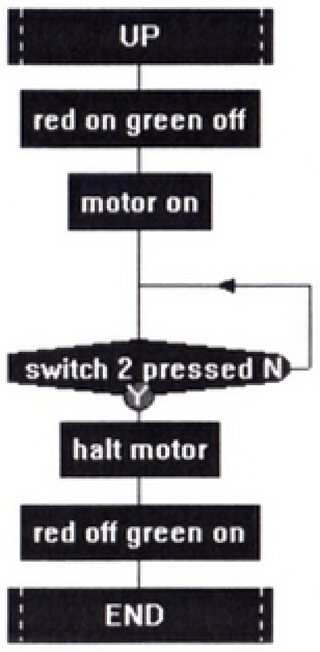
The new version of the main routine is shown below; the rest of the flowsheet is unchanged from that shown in fig 14.



Controlling Movement 1: Stairlift

Extending the System

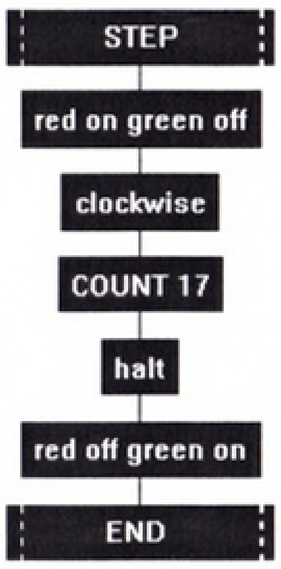
The new version of the UP macro is shown. The DOWN macro should be edited in the same way. The rest of the flowsheet is unchanged from that shown in fig 4.



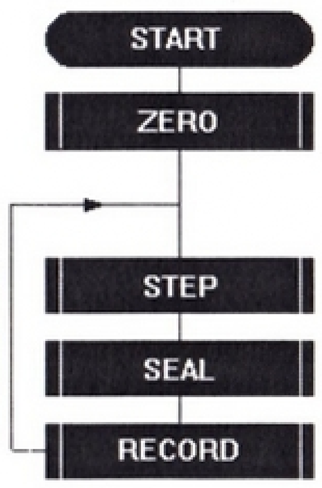
Controlling Movement 2: Packing System

Extending the System

1. The new version of the STEP macro including the indicator lights is shown below. The SEAL macro should be edited in the same way.



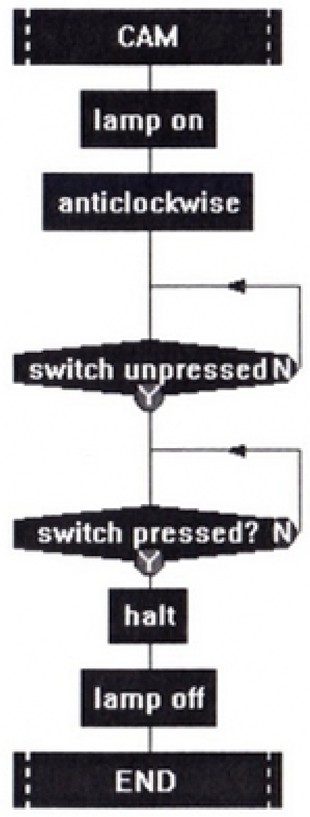
2. The new version of the main routine including the RECORD macro is shown below.



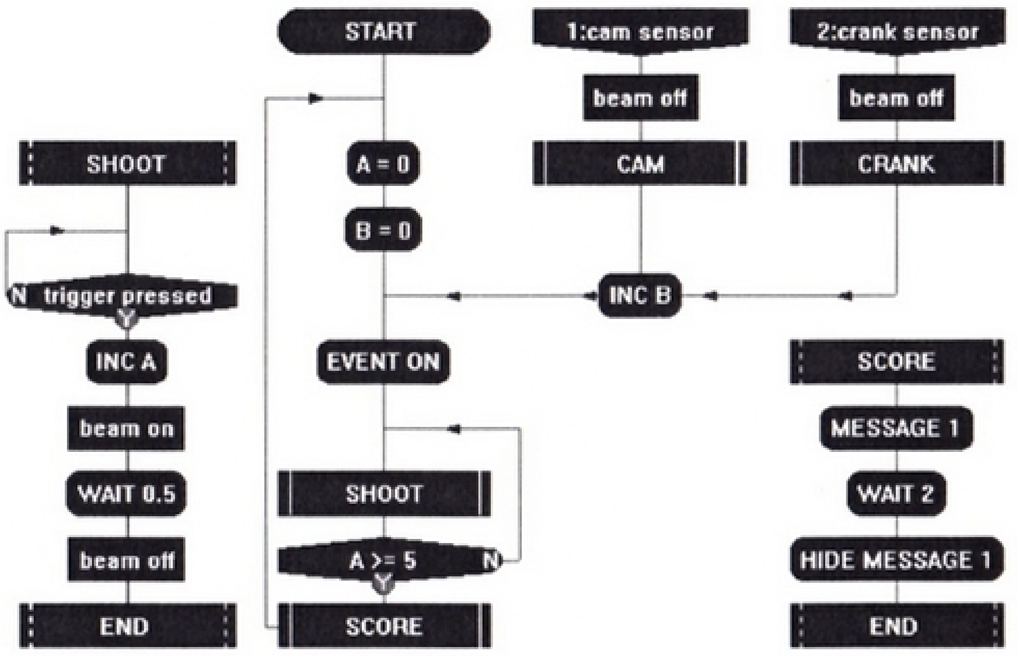
Controlling Movement 3: Shooting Gallery

Extending the System

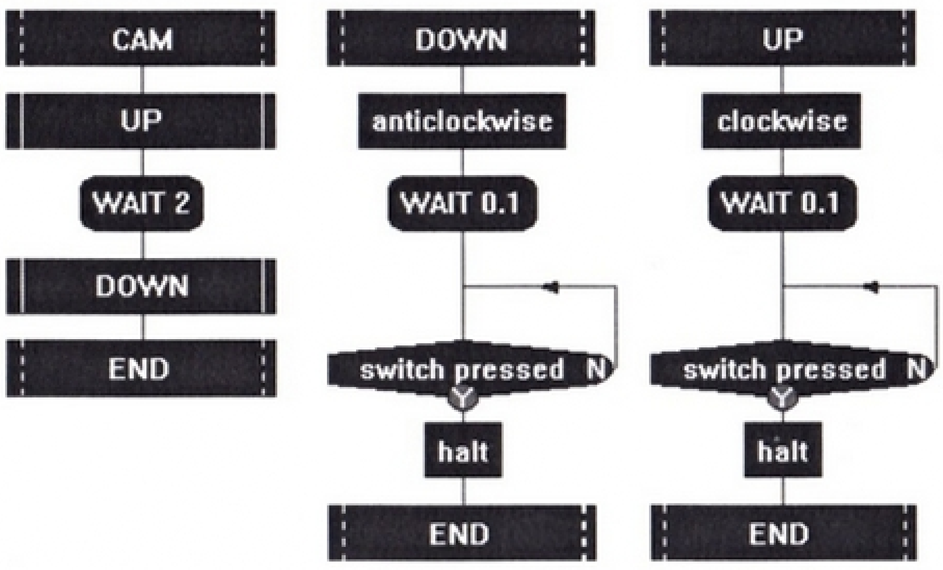
1. The new version of the CAM macro including the indicator light is shown. The CRANK macro should be edited in the same way.



2. The flowsheet shown below allows each player five shots and displays the score. Variable B is used to count the hits. The text in the MESSAGE command should read something like “Game over. Your score is [B]”. The CAM and CRANK macros are not shown, but they should also be on the flowsheet.



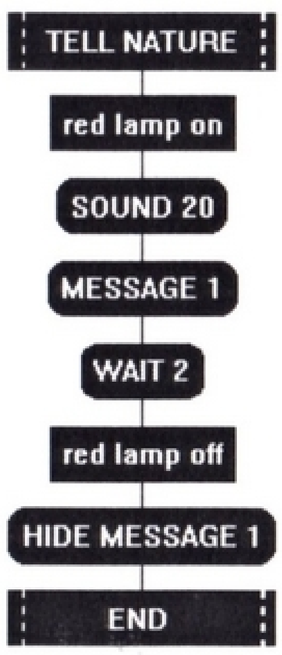
3. The new version of the CAM macro which makes the figure pause for 2 seconds when it is out, is shown below. The CRANK macro should be edited in the same way.



Controlling Movement 4: Theme Ride

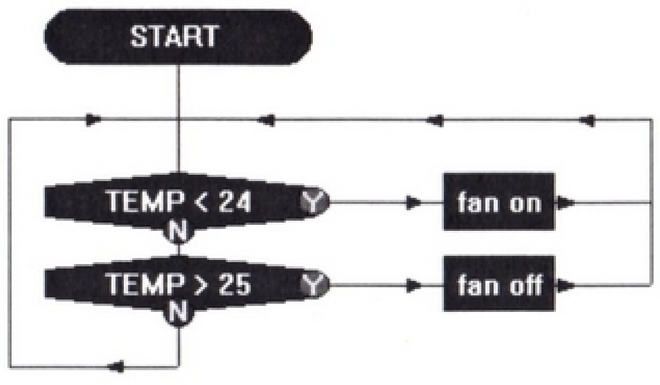
Extending the System

The TELL NATURE macro shown below can be used to inform passengers that they have arrived at Nature World. A Do Macro command labelled TELL NATURE must be included just before the end of the NATURE macro. Equivalent macros, using different coloured lamps, sounds and message text can be used to inform passengers of arrival at other destinations. See Logicator Help for information on how the PLAY command could be used, for example, to play music or a recorded message on arrival.



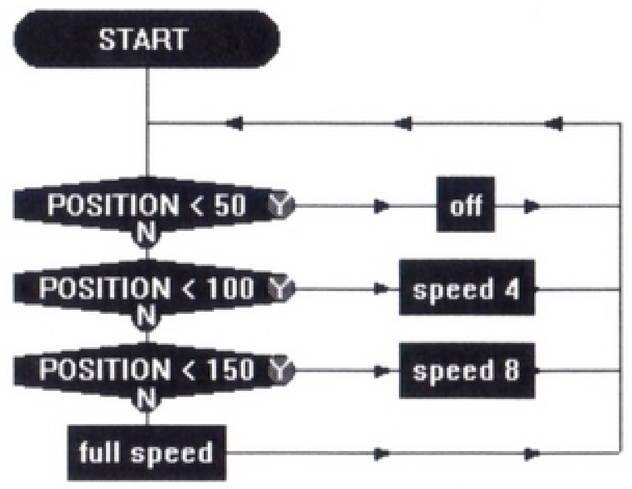
Controlling Movement 5 : Analogue Sensors

A flowsheet to switch the fan more cleanly in response to the temperature sensor is shown below. In this system, the fan does not switch on until the temperature reaches 26 degrees; and it does not switch off until it falls to 23 degrees.

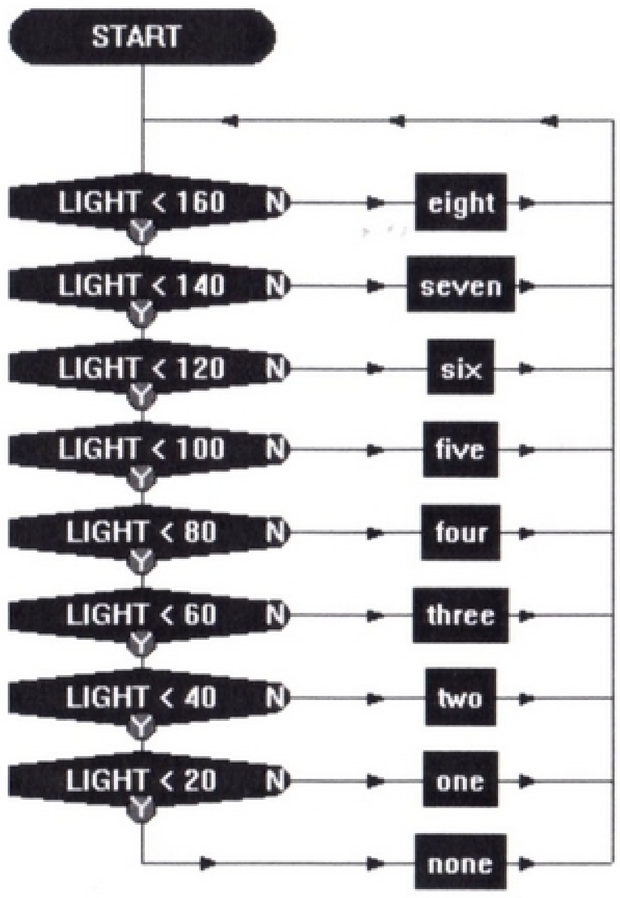


Varying the Output

1. A flowsheet for using a Smart position sensor to control the speed of a single motor buggy is shown.

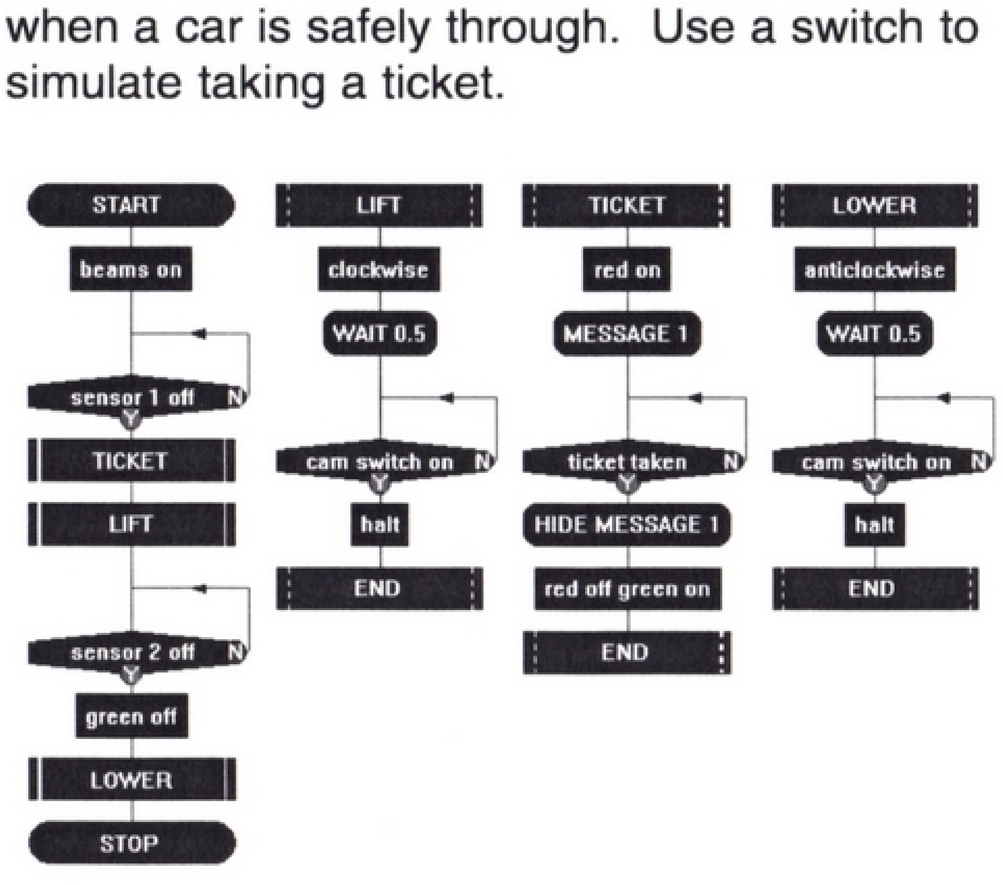


2. The flowsheet shown below uses the eight green LEDs on Smart Box as a light meter.



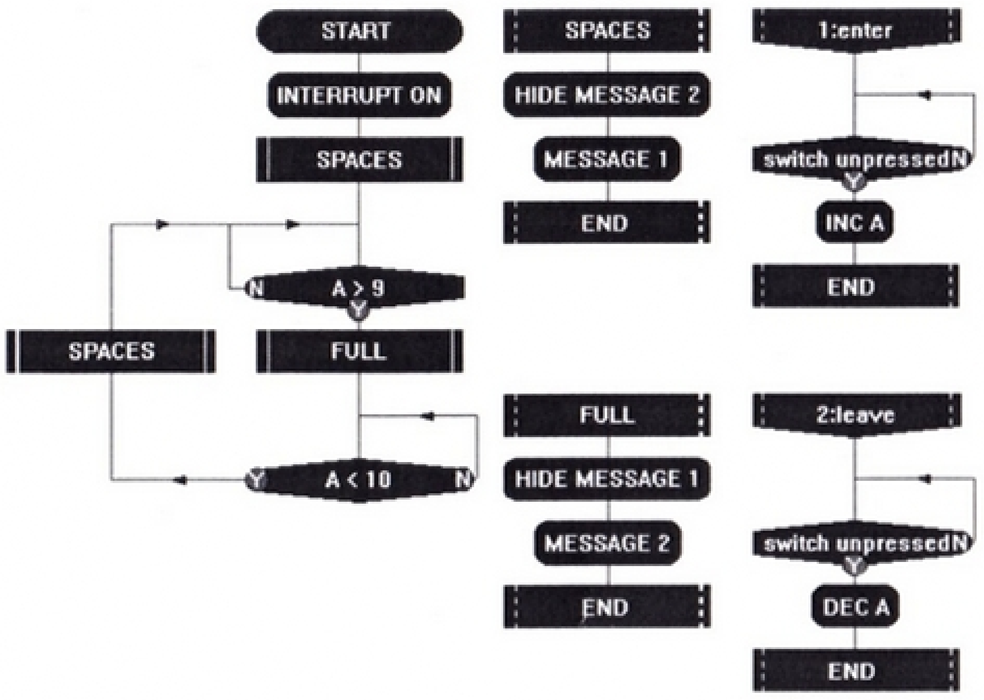
Project File - Car Park Barrier

The flowsheet shown below meets the main specification for one demonstration, using a hand-pushed car. MOTOR commands can be added to drive a motorised car model up to the barrier, halt, and then drive under the barrier. The system requires one lamp and light sensor in front of the barrier to sense the approach of a car, and another behind the barrier to sense when a car is safely through. Use a switch to simulate taking a ticket.



Extending the System

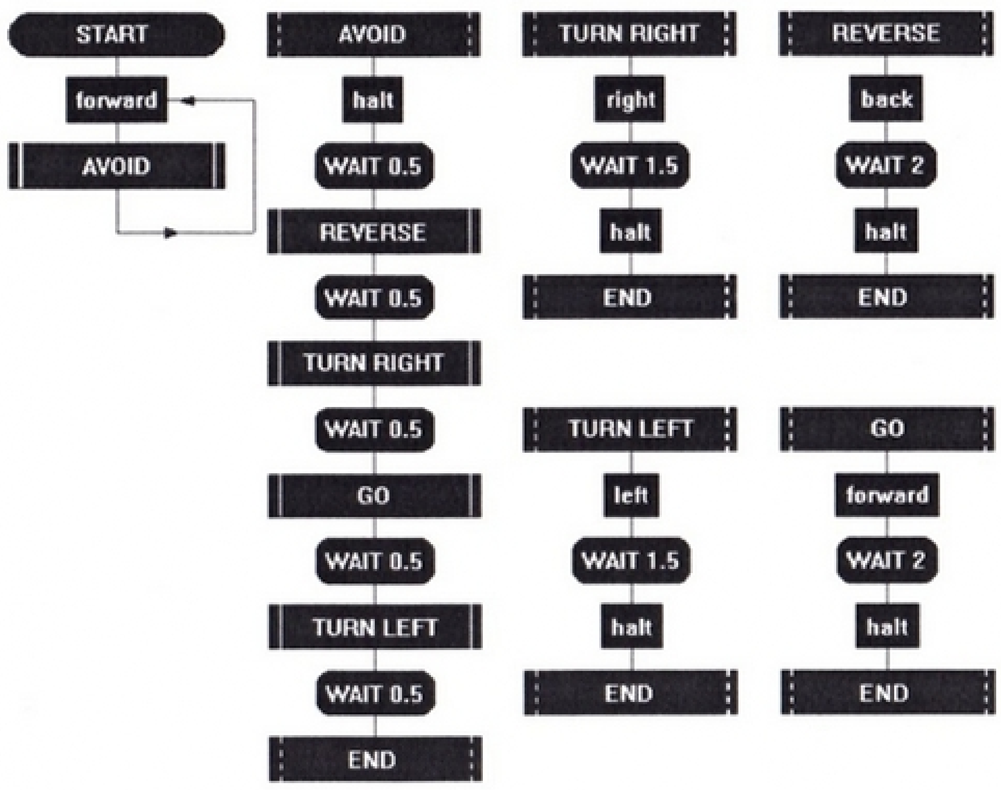
A flowsheet for counting cars and displaying appropriate messages is shown below. Use two switches to simulate pressure pads at the entrance and exit to the car park. Interrupt commands are used to check these switches. Decision commands are used to check that the switches are unpressed again before the count is made. This "debouncing" effect ensures an accurate count.



Project File - Controllable Vehicle

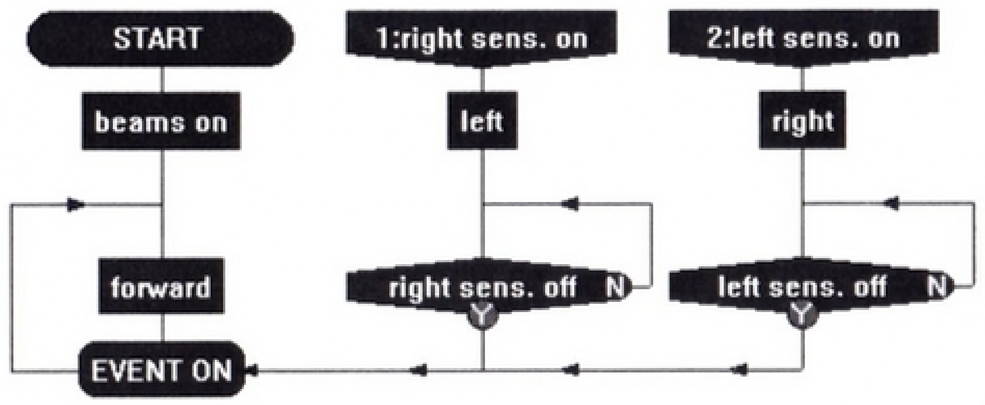
2. Avoiding Obstacles

A flowsheet for moving round an obstacle in the vehicle's path is shown below. The WAIT times will vary according to the speed of the vehicle and the size of the obstacle.



3. Following a Path

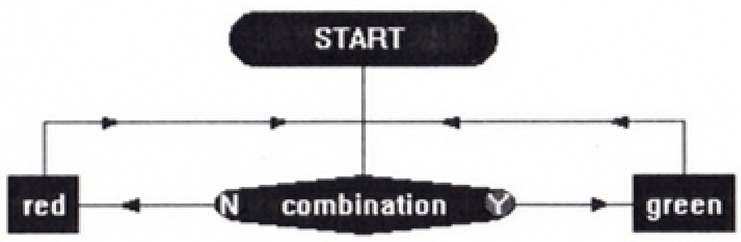
The flowsheet shown uses two light sensors to keep the vehicle on the black path.



Project File - Security

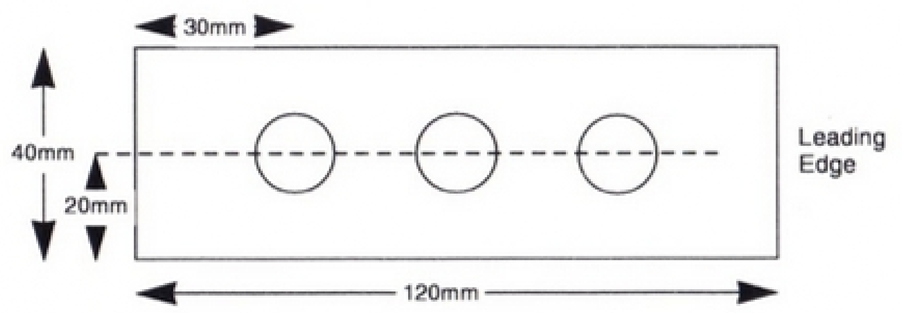
1. Push button combination

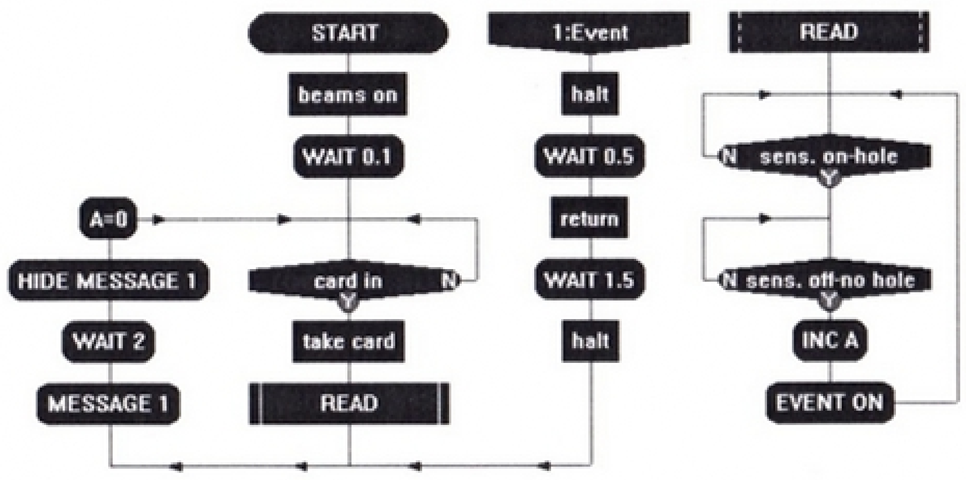
Set the Decision command to check for the two selected switches.



2. Card operated lock

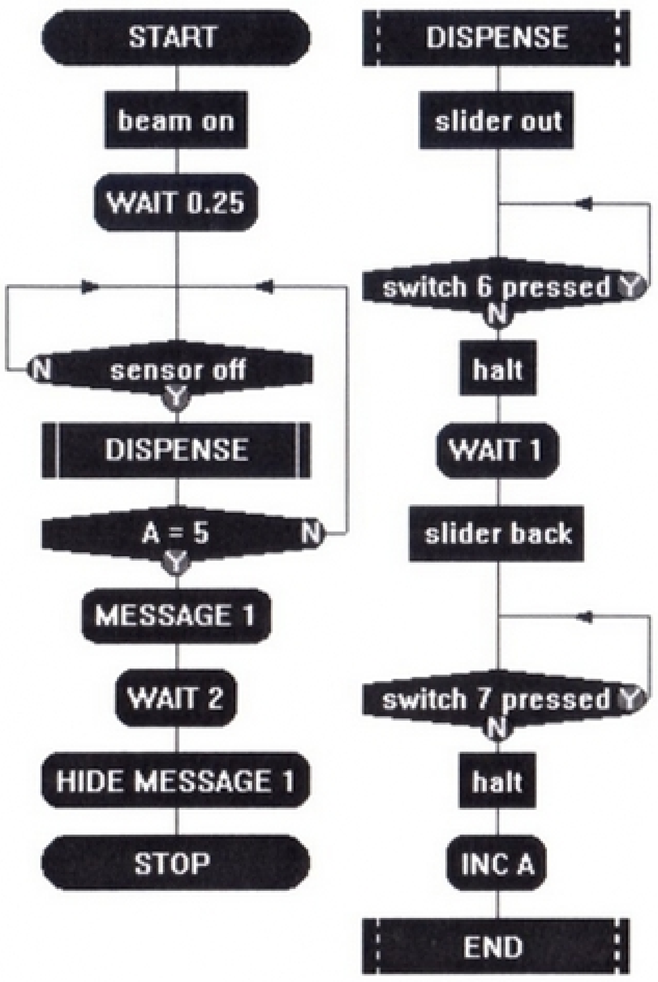
Black matt card is the best material to use, stiff but not too thick. Approximate dimensions are shown below. The flowsheet shows a possible control system for taking in, reading and ejecting the card.





Project File - Vending Machine

The flowsheet shown below uses the mechanism shown on the Rack and Pinion 2 construction sheet to dispense a chocolate bar. The crank and slider could also be used, in which case the DISPENSE macro would be based on the CRANK macro used in the "Controlling Movement 3" section.

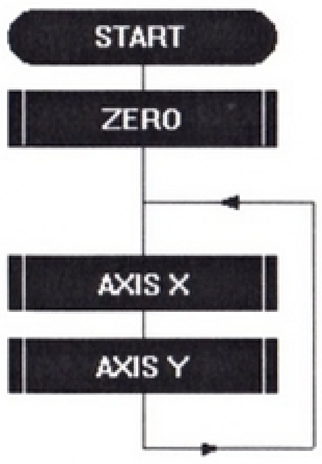


A lamp and light sensor are used to sense if a coin has been put into the machine. You could also experiment with other methods of sensing a coin, e.g. using two switches set at a measured distance apart so that only the right size of coin will press both at the same time to provide the necessary input signal.

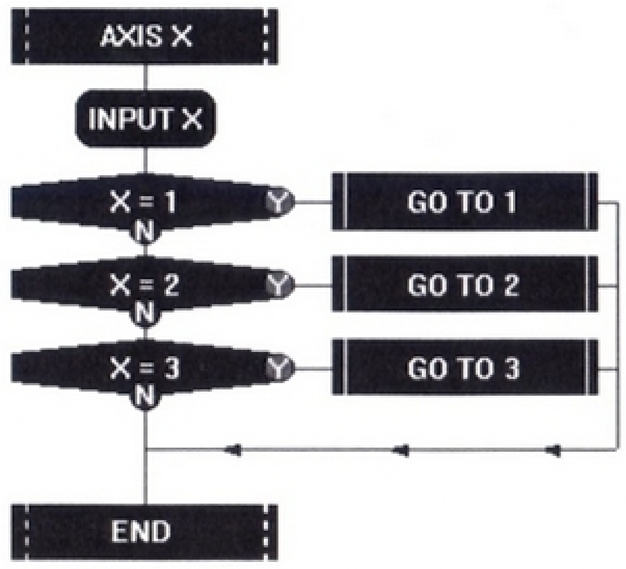
After five bars have been dispensed, a message indicates that the machine needs to be filled.

Project File - High Rack Warehouse

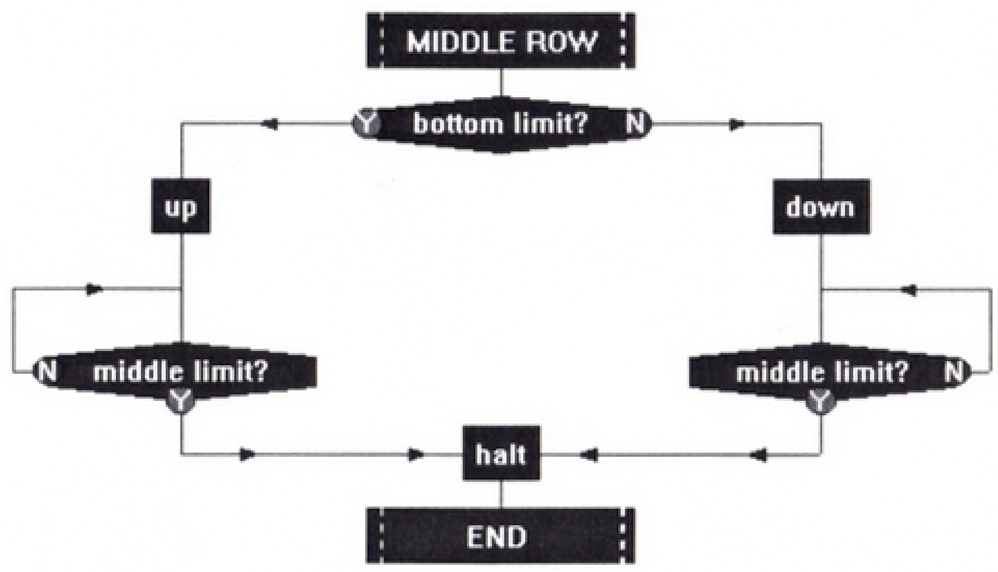
This suggested solution uses the leadscrew for the X axis, and the rack and pinion for the Y axis. The main route is shown below.



The AXIS X macro as shown below is very similar to the PROCESS macro used in the "Controlling Movement 4" section. The GO TO 1 macro will be similar to the NATURE macro and so on.



The AXIS Y macro will use the same algorithm, with the macros within it being based on those used in the "Controlling Movement 1" section. Unlike the Stairlift system, this system requires a MIDDLE ROW macro as shown below. This tests the bottom limit switch first to find out which direction to move in to go to the middle row.



Use a red angle block or angle bracket attached to a black block to the side of the motor to press the limit switches on the rack and pinion.

Project File - Carousel Storage System

The flowsheet shown as "Alternative System" in the "Controlling Movement 4" section can easily be adapted to meet this specification, substituting equivalent CLOCKWISE and ANTICLOCKWISE macros for the UP and DOWN macros.

SUGGESTED SOLUTIONS - SMART MOVE

Start Up 5: Design your own system

Procedure: main

```
wait until sensor 0 is on
switch on 0,1
forward a
wait 5
switch off 0,1
halt a
```

Addition 1

Procedure: main

```
repeat
wait until sensor 0 is on
switch on 0,1
forward a
wait 5
switch off 0,1
halt a
forever
```

Addition 2

Procedure: main

```
repeat
switch on 2
wait until sensor 0 is on
switch off 2
switch on 0,1
forward a
wait 5
switch off 0,1
halt a
forever
```

Start Up 6: Building a complex system

Change to the specification

Build a new procedure:

Prodedure: info

```
print " message one"
wait 2
cls
print "message two"
wait 2
```

Edit the "dryer" procedure as shown below:

Procedure: dryer

```
switch on 0,1
forward a
info
switch off 0,1
halt a
```

The "main" procedure is unchanged.

Start Up 7: Responding to a sensor

Procedure: main

```
switch on 2,3
wait 0.5
repeat
if sensor 1 is off then dryer
forever
```

Procedure: dryer

```
switch off 2
switch on 0,1
forward a
wait until sensor 1 is on
switch off 0,1
halt a
switch on 2
```

Extension

Procedure: main

```
repeat
wait until sensor 0 is on
wait until sensor 0 is off
dryer
forever
```

Procedure: dryer

```
reset clock
start clock
switch on 0,1
forward a
wait until sensor 0 is on or time > :10:00
switch off 0,1
halt a
wait until sensor 0 is off
```

Controlling Movement 1: Stairlift

Extending the System

1. The new version of the "up" procedure is shown below. The "down" procedure should be edited in the same way.

Procedure: up

```
switch off 6
switch on 7
forward a
wait until sensor 2 is on
halt a
switch off 7
switch on 6
```

2. The system for changing direction in mid journey is shown below. The "up" and "down" procedures are unchanged.

Procedure: main

```
repeat
checksensors
forever
```

Procedure: checksensors

```
if sensor 3 is on then down
if sensor 4 is on then up
```

Controlling Movement 2: Packing System

Extending the System

The system shown below includes both the indicator lights (extension 1) and the additional counting and displaying element (extension 2).

Procedure: main

```
zero
repeat
step
seal
forever
```

Procedure: seal

```
switch off 6
switch on 7
forward b
wait until sensor 6 is off
halt b
wait 1
backward b
wait until sensor 7 is off
halt b
let a = a + 1
print "box number" a
switch off 7
switch on 6
```

Procedure: step

```
switch off 6
switch on 7
forward a
wait until count 0 = 17
halt a
switch off 7
switch on 6
```

Procedure: zero

```
backward a
wait until sensor 1 is on
halt a
let a = 0
```

Controlling Movement 3: Shooting Gallery

Extending the System

The system below allows each player five shots and displays the score (extension 2). It also includes the use of lamps to indicate a hit (extension 1).

Procedure: shoot

```
let a = 0
let b = 0
repeat
wait until sensor 3 is on
switch on 3
wait 0.5
check
switch off 3
let a = a + 1
until a = 5
print "Game Over"
print "Your score is" b
```

Procedure: check

```
if sensor 7 is on then cam
if sensor 6 is on then crank
```

Procedure: cam

```
switch on 4
backward b
wait until sensor 1 is off
wait until sensor 1 is on
halt b
let b = b + 1
switch off 4
```

Procedure: crank

```
switch on 5
backward a
wait until sensor 0 is off
wait until sensor 0 is on
halt a
let b = b + 1
switch off 5
```

3. The new version of the "crank" procedure which makes the figure pause for 2 seconds when it is out, is shown below. The "cam" procedure should be edited in the same way.

Procedure: crank

```
forward a
wait until sensor 0 is off
wait until sensor 0 is on
halt a
wait 2
backward a
wait until sensor 0 is off
wait until sensor 0 is on
halt a
```

Controlling Movement 4: Theme Ride

Extending the System

A procedure of the kind shown below could be added to the end of each of the destination procedures.

Procedure: announce

```
switch on 7
print "Nature World"
wait 2
switch off 7
cls
```

Controlling Movement 5 : Analogue Sensors

Varying the Output

1. A procedure for using a Smart position sensor to control the speed of a single motor buggy is shown below.

Procedure: buggy

```
repeat
let p = position
if p < 50 then halt a
if p >= 50 and p < 100 then forward a speed 4
if p >= 100 and p < 150 then forward a speed 8
if p >= 150 then forward a
forever
```

2. The procedure shown below switches on the lamps one by one as it gets darker, and switches them off in the same way as it get lighter.

Procedure: lamps

```
repeat
if light < 100 then switch on 0 else switch off 0, 1, 2
if light < 50 then switch on 0, 1 else switch off 1, 2
if light < 5 then switch on 0, 1, 2 else switch off 2
forever
```

This procedure can easily be extended to make use of the eight green LEDs on Smart Box as a light meter.

Project File - Car Park Barrier

The system shown meets the main specification for one demonstration, using a motor driven car. It requires one lamp and light sensor in front of the barrier to sense the approach of a car, and another behind the barrier to sense when a car is safely through. Use a switch to simulate taking a ticket.

Procedure: demo

```
switch on 1,2
forward a
wait until sensor 1 is off
halt a
ticket
lift
forward a
wait until sensor 2 is off
switch off 4
lower
halt a
switch off 1,2
```

Procedure: ticket

```
switch on 3
print "Stop. Take ticket"
wait until sensor 4 is on
switch off 3
switch on 4
cls
```

Procedure: lift

```
forward b speed 5
wait 0.5
wait until sensor 3 is on
halt b
```

Procedure: lower

```
backward b speed 5
wait 0.5
wait until sensor 3 is on
halt b
```

Extending the System

A system for counting cars and displaying appropriate messages is shown below. Use two switches to simulate pressure pads at the entrance and exit to the car park.

Procedure: main

```
let a = 0
spaces
repeat
repeat
checkbuttons
until a > 9
full
repeat
checkbuttons
until a < 10
spaces
forever
```

Procedure: spaces

```
cls
print "spaces"
```

Procedure: full

```
cls
print "full"
```

Procedure: checkbuttons

```
checkin
checkout
```

Procedure: checkin

```
if sensor 1 is off then end
wait until sensor 1 is off
let a = a + 1
```

Procedure: checkout

```
if sensor 2 is off then end
wait until sensor 2 is off
let a = a - 1
```

Project File - Controllable Vehicle

2. Avoiding Obstacles

A system for moving round an obstacle in the vehicle's path is shown below. The wait times will vary according to the speed of the vehicle and the size of the obstacle.

Procedure: main

```
repeat
forward a,b
wait until sensor 1 is on
halt a,b
avoid
forever
```

Procedure: avoid

```
backward a,b
wait 4
forward b
wait 4
forward a,b
wait 4
backward b
wait 4
```

Economatics (Education) Ltd.
Epic House, Darnall Road,
Attercliffe, Sheffield S9 5AA.
Telephone: (0114) 281 3311
Fax: (0114) 243 9306

L30/040/2