

COMMOTION

Solutions for Education

Getting Started with CoCo+



A quick reference guide to using
CoCo + in the classroom

By Ron Allen

In association with



GETTING STARTED WITH

CoCo +

A quick reference guide to using control in the classroom

Written by Ron Allen

"CoCo +" works on any Archimedes computer running RiscOs 3.1 and above.

We would like to acknowledge and thank the following people for their support, suggestions, ideas, technical expertise, patience and co-operation during the development of this booklet.

Grace Woodford

Barry Payne

Ruth Darby

Purchasing institutions may copy all or part of this publication for use in that institution only.

Because our policy is to improve our products and services continually, we may make changes without notice. It is acknowledged that there may be errors and omissions in this publication for which responsibility cannot be assumed. No liability will be accepted for loss or damage resulting from the use of information contained in this manual or from use of the described product.

ISBN 1-873101-41-4



9 781873 101414

Commotion Limited • Unit 11 • Tannery Road • Tonbridge • Kent • TN9 1RF

©Commotion Ltd. 1997

CONTENTS

QUICK GUIDE

1. Loading CoCo+
2. Mouse control
3. Typing instructions
4. Building procedures
5. Using the inputs
6. Using sensors
7. Command summary 1
8. Command summary 2

TOPICS

1. Make a Working Lighthouse
2. Make a Clown Come to Life
3. Make an Exciting Clown
4. Traffic Lights
5. More with Traffic Lights
6. Home from Home
7. Lighting up Time
8. Build a Barrier
9. An "Are You There Barrier"
10. Make a Disco Light Show
11. Make a Digital Display
12. Make a Message Machine

GENERAL INFORMATION

QUICK REFERENCE GUIDE

The quick reference guide gives a brief introduction (copyright free) into using CoCo +. It is not as comprehensive as the full user guide. For the clearest explanation the two should be used together.

ACTIVITY SHEETS

The activity sheets are in two parts:

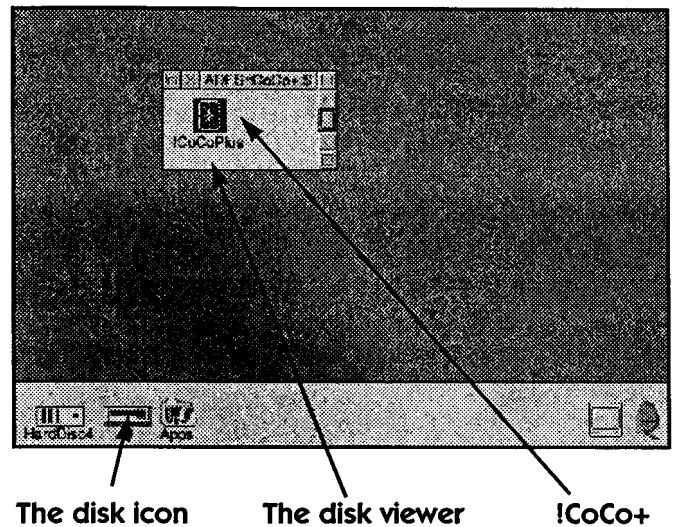
The first sheet sets the scene and the problem. At the bottom of the sheet is an extension activity that may be carried out by the pupil if required. The picture puts the task into a context and may also suggest a solution.

The second part of the activity offers a suggested solution to both designing and making as well as procedure building. Teachers may offer this to pupils if required. Further to this, a disk containing the suggested procedures may also be used.

Note: As some activity cards suggest the pupils carry out extension activities which build upon the basic procedures we suggest that you make backups of the procedures disk before pupils start programming. The procedures disk only contains the basic procedure, prior to any extension activity.

LOADING CoCo+

1. Place the CoCo+ disk in the drive and using the mouse point to the disk icon on the menu bar.
2. Click once with the <Select> button
3. Point your mouse button at !CoCo+ on the disk viewer and click twice with the <Select> button.



The program will begin to load.

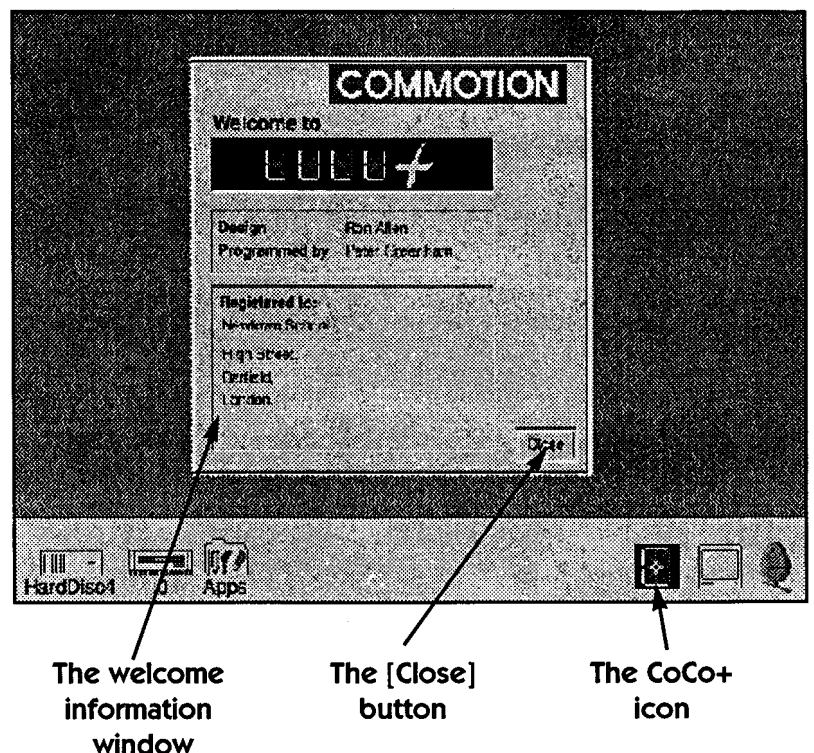
An information window will appear. This displays the name of the registered user.

You may hear CoCo+ loading and see the disk light come on.

4. To remove the Welcome window click on the [Close] button.

The CoCo+ icon will appear on the icon bar at the bottom of the screen to let you know CoCo+ is ready to go.

5. Click <Select> once on the CoCo+ icon.



The program will now appear and is ready to start.

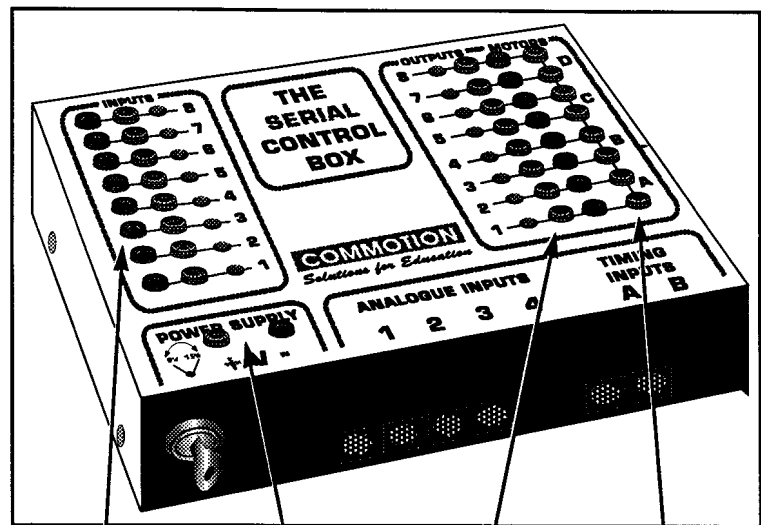
MOUSE CONTROL

The Serial interface and the Commotion control box have a row of red and black sockets numbered 1 to 8. These are the outputs and are used for switching electrical devices on and off.

There is a row of green and black sockets also numbered 1 to 8. These take input switches.

There is a row of blue sockets in pairs labelled A to D are motors.

The red and black sockets labelled POWER SUPPLY are always on and cannot be controlled by the computer. These are useful when checking that an output works prior to use within a control model.



Input Sockets Power supply Output sockets Motors Sockets

Things to do...to make a light bulb come on.

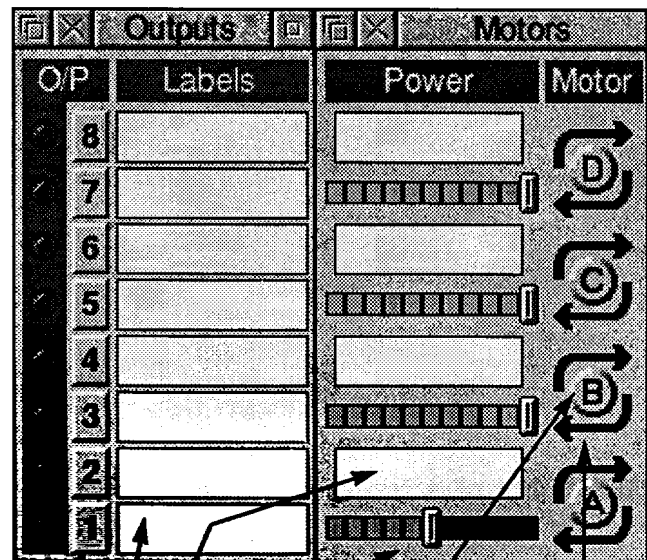
1. Connect a light bulb into the output socket number 1.
2. Place the pointer on button [1] in the output window and press <Select>. The bulb will come on.

And....to turn the bulb off click again.

3. Connect a motor to motor socket B. Now point to the button [B] on the Motors window and press <Select>. The motor will come on.

To reverse the motors point to the arrows and press <Select>.

4. Place the pointer on the slide bar opposite [B]. Hold down select and drag the bar to the left. The motor slows down.
5. Click on motor button [B] to turn it off.



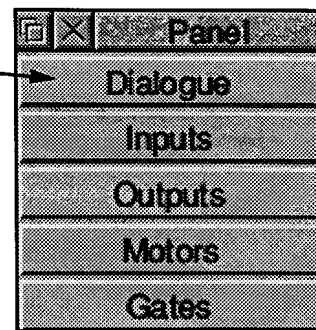
Output button 1 Slider bar Motor button [B] Reversing arrows

You can name the outputs and motors by typing a label in the white text box

TYPING INSTRUCTIONS

1. Click on the dialogue button in the panel window.

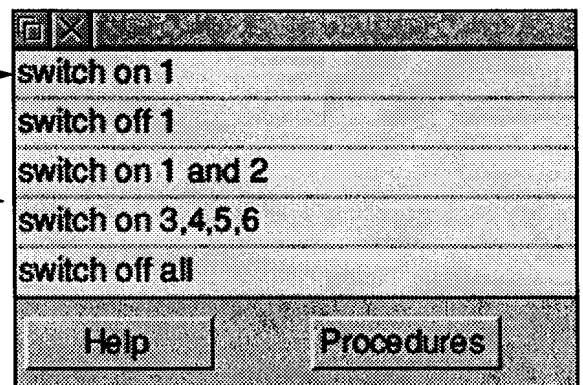
This will give you a dialogue box into which you can type your instructions.



2. Plug light bulbs into output sockets 1 to 6 and type in:

SWITCH ON 1 <Return>

Try some of these command lines and see what happens

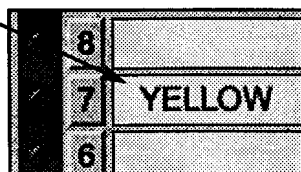


3. If you label output 7 YELLOW you can type in:

SWITCH ON YELLOW

4. To control motors plugged into the blue sockets type:

MOTOR A FORWARDS
MOTOR A BACKWARDS
MOTOR A OFF



5. Press <Select> on the [Commands] button on the dialogue window. Point to a word and press <Select>. This types the word for you. When the command is complete press [Return].

6. Type the following to see what happens:

ALARM [Return]
CLEAN [Return]

You may also press the red function keys for selected commands.

Commands										
ALARM	ALL	AND								
AT	BACKWARD	BUILD								
CHANGE	CLEAN	END								
FOREVER	FORWARD	IF								
LIST	MOTOR	OFF								
ON	POWER	PROCEDURE								
REPEAT	SAY	SWITCH								
THEN	WAIT	WHENEVER								
1	2	3	4	5	6	7	8	9	0	Return
A	B	C	D	.	,	Spc	Del			

BUILDING PROCEDURES

To write a simple procedure that will

- turn on a light bulb
- wait for 10 seconds
- turn the light bulb off again

- Type BUILD LAMP <Return>.
- In the procedure window type the commands as shown.
- Click on the [OK] button when you have finished.

To carry out the procedure type:
LAMP <Return>

Procedures can be changed. To change the waiting time to just one second type:

CHANGE LAMP <Return>

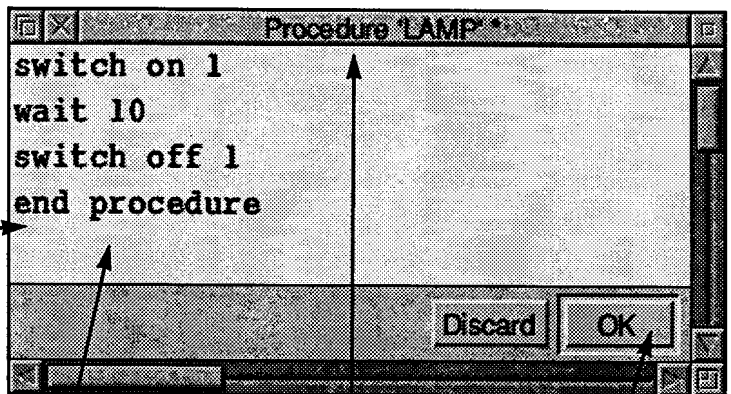
- Move your cursor to the 10, delete and replace with 1.

Click [OK] and it is changed.

You can repeat the instructions or procedures.

- Build the procedure FLASH that will flash the lamp 10 times.

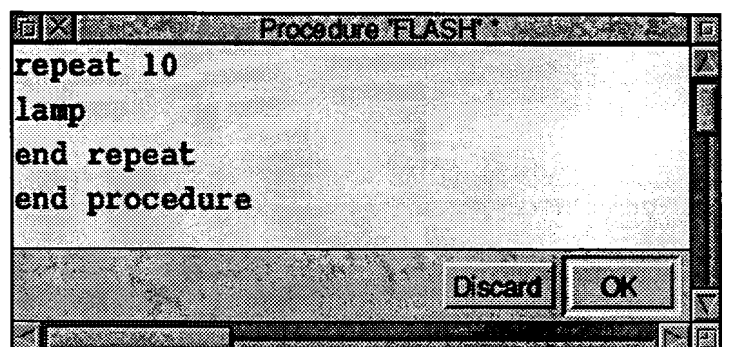
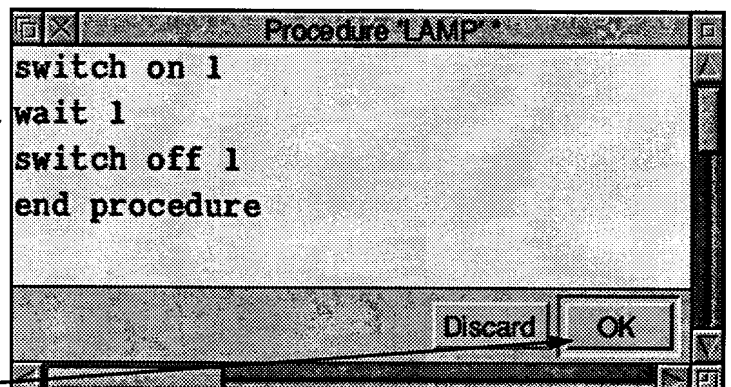
You can add to the list of commands CoCo+ knows by teaching the computer new ones. You can build new procedures.



Procedure window

Procedure name

OK button



USING THE INPUTS

The green and black sockets on the left of the box are for plugging switches in. CoCo + is able to tell if the switch is on or off. You can use these in your procedures.

1. Plug in a press switch (push to make) into input socket number 1 and type the following:

Build Test1 <Return>

This procedure will wait until you press the switch before going to the alarm command. It will only check once.

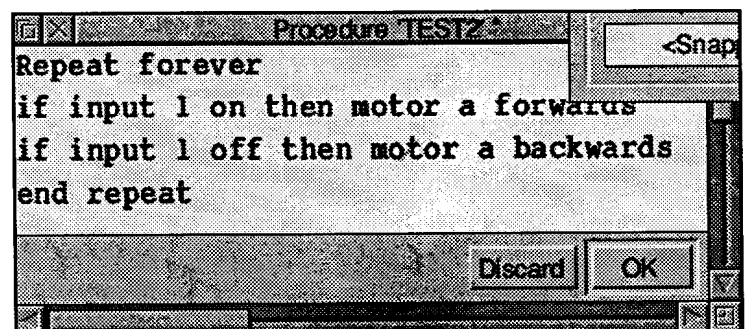
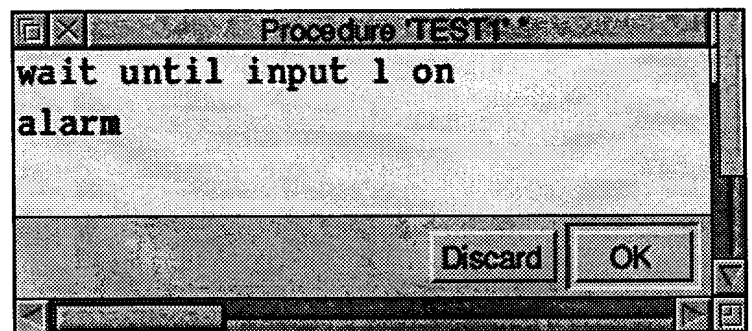
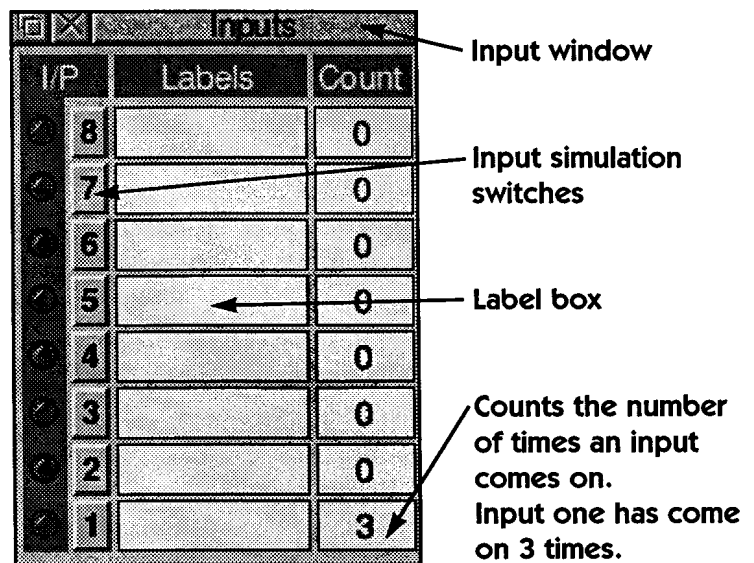
You can however keep checking if a switch is pressed.

2. **Type Build Test2 <Return>**

Plug a motor into socket A and run the procedure Test 2. You will need to press the <Escape> key to end it.

3. A third use of the input is to achieve the above with one command.

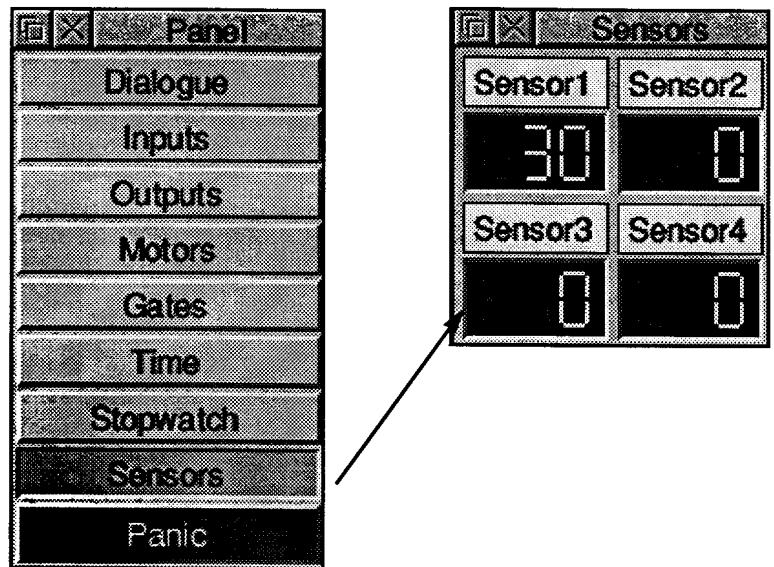
Type in the dialog box:
Whenever input 1 on alarm



USING SENSORS

On the front of the serial box (not on the standard control box) is a set of sockets into which you can plug in sensors. These can measure light, sound etc and give a value from 0 to 100.

1. Point to the [Sensor] button on the Panel window and press <Select>.
2. Plug a light sensor into Sensor socket 1, the value will be shown in the 1st window.

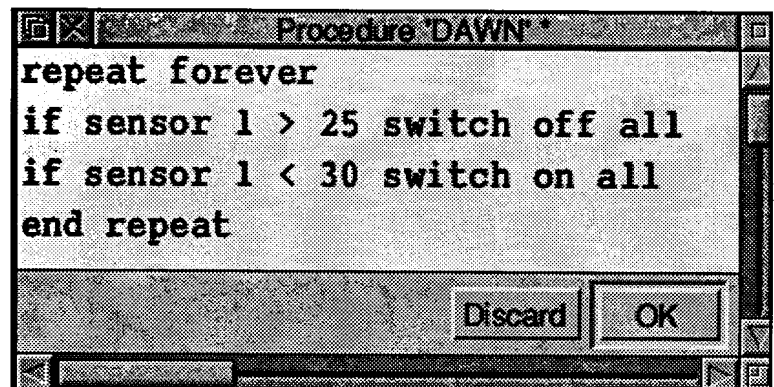


You can use this value in a procedure.

3. Plug a set of light bulbs into outputs 1 to 8.

Type Build Dawn <Return>

This procedure can test the amount of light in the room and switch on the outputs if it is too dark.

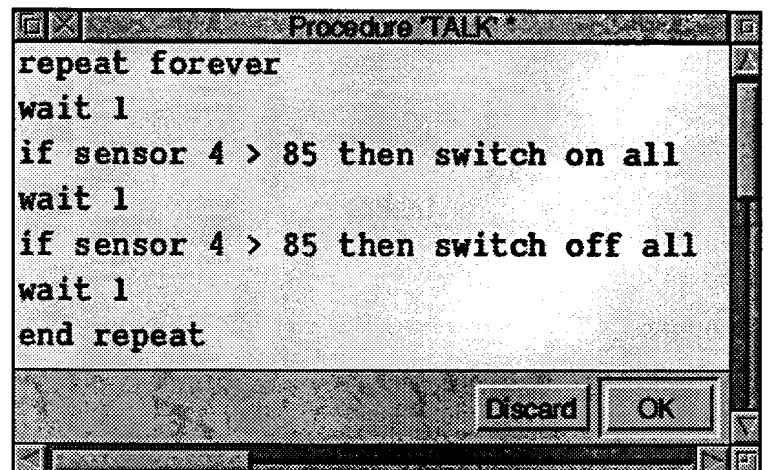


This procedure will turn the lights on when you shout ON and off when you shout OFF!

4. Plug a microphone into sensor socket 4. Talk into it and note the maximum value (say 90).

5. **Build Talk <Return>**

This procedure responds to your voice.



COMMAND SUMMARY

The following is a short list of the key words that CoCo+ knows.
For a more comprehensive list see the user guide book supplied with the software.

- SWITCH ON/OFF** <number> Switches ON and OFF specified outputs.
e.g. **SWITCH ON 1**
SWITCH ON ALL
SWITCH ON 1 AND 2
- MOTOR** <name> **FORWARDS/BACKWARDS/OFF**
Turns on a named motor socket in a given direction.
e.g. **MOTOR A FORWARDS**
MOTOR A OFF
- POWER 1/B** <level> This sets the outputs to a specified power level.
e.g. **POWER A 5**
- SAY** <message> The computer will say whatever follows in quotes.
e.g. **SAY "WHAT IS THE TIME?"**
- CLEAN**
Cleans the dialogue and write windows
e.g. **CLEAN**
- ALARM**
Makes an alarm noise through the computers own speaker.
e.g. **ALARM**
- BUILD** <procedure name> Allows you to add a sequence of instructions to the language. A procedure has to be given a name.
e.g. **BUILD FLASH**
- CHANGE** <procedure name> Allows you to make changes to a procedure.
e.g. **CHANGE FLASH**
- WAIT** <number> Waits a number of seconds before going to next command.
e.g. **WAIT 12**

COMMAND SUMMARY 2

The following is a short list of the key words that CoCo+ knows. For a more comprehensive list see the guide book supplied with the software.

REPEAT <number> or **FOREVER** Repeats the following actions the number of times specified. Needs to end with an end repeat

e.g. **REPEAT 10**
SWITCH ON 1
SWITCH OFF 1
END REPEAT

IF INPUT 1 ON/OFF THEN <statement>

Will carry out what follows THEN, if the condition is satisfied.

e.g. **IF INPUT 1 ON THEN SWITCH ON 3**

WHENEVER <condition> <task> This continues to check for a given condition until revoked.

e.g. **WHENEVER INPUT 1 ON SWITCH OFF ALL**

WAIT <number> or **UNTIL** <condition>

Waits a specified number of seconds or UNTIL a condition is met before going to the next command.

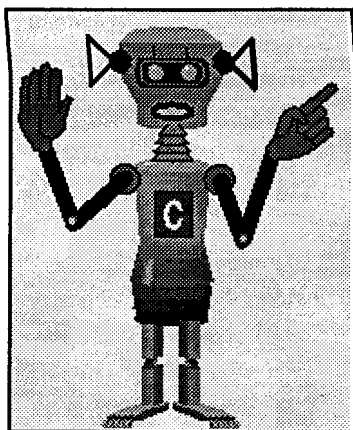
e.g. **WAIT 12**
will wait for 12 seconds
WAIT UNTIL INPUT 5 OFF

WRITE <message>

Displays in a special Write window any message you wish to display.

e.g. **WRITE "The barrier is now open!"**

Make a working Lighthouse...Using CoCo+



1

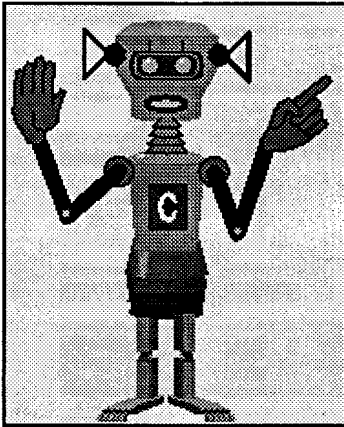
Why don't you?

Design and make a lighthouse that flashes its light every 2 seconds



- Did you know that each lighthouse has its own flash pattern?
- Can you make your lighthouse flash to your own distinctive pattern?
- During fog you may not be able to see the light. Can you make it buzz at the same time?

Here are some ideas...give them a try!



1

Make a model using re-cyclable material such as a drinks carton and a cardboard tube. Connect a 6V light bulb to output socket 1 on the control box and type the following:

How about this!
This light will now flash in a different sequence

Build Flash <Return>

```
Procedure FLASH*
Repeat forever
switch on 1
wait 2
switch off 1
wait 2
end repeat
end procedure
```

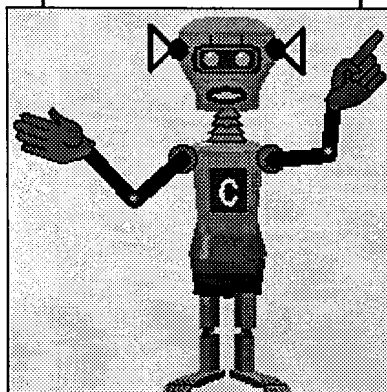
This will make the light flash every 2 seconds.

```
Procedure FLASH2*
Repeat forever
switch on 1
wait 1
switch off 1
wait 3
switch on 1
wait 0.5
switch off 1
wait 1
end repeat
end procedure
```

HELPFUL HINTS

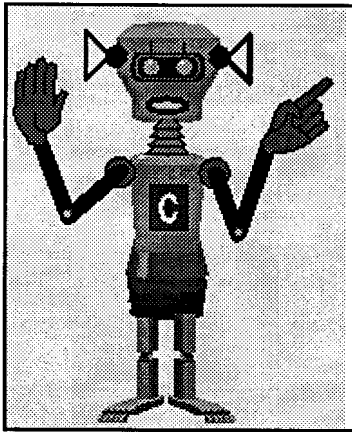
To make a buzzing sound at the same time, place a buzzer in socket number 2 and change your procedure FLASH to:(see opposite)

Remember buzzers must be connected the right way round: red to the red socket and black to the black socket.



```
Repeat forever
switch on 1 and 2
wait 2
switch off 1 and 2
wait 2
end repeat
end procedure
```

Make a Clown come to life...Using CoCo+



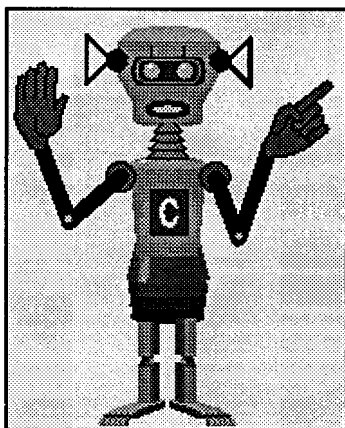
2 Why don't you?

Make a clown's face, with a nose and eyes that light up?
Can you make the clown talk and wink and flash the nose 5 times?



- Think of a way to make the bowtie spin round.
- First one way and then the other.

Here are some ideas...give them a try!



2

Why don't you?

Draw a clowns face on card and use two light bulbs for the eyes and another for the nose. Connect the eyes to the outputs 3 and 4 and plug the nose into output 5.

Build Wink <Return>

Build Nose <Return>

```
Procedure 'WINK'  
say "Look at my eyes I can wink"  
switch on 3 and 4  
wait 2  
switch off 3  
wait 1  
switch on 3  
wait 2  
say "Did you like that?"  
end procedure
```

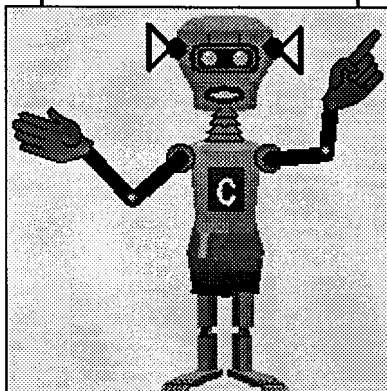
```
Procedure 'NOSE'  
Repeat 10  
switch on 5  
wait 1  
switch off 5  
wait 1  
end repeat  
end procedure
```

HELPFUL HINTS

Connect a motor to the blue motor socket labelled B. Make a bowtie and stick it on the end of the motor shaft.

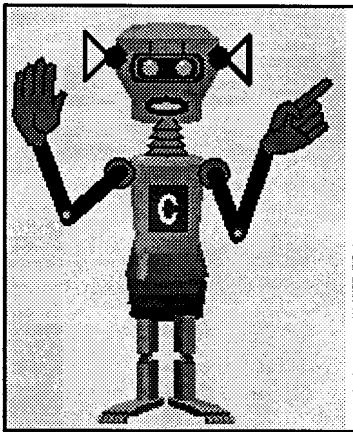
Type the following:

Build Tie <Return>



```
Motor b forwards  
wait 5  
motor b backwards  
wait 5  
motor b off  
end procedure
```

Make an exciting Clown...Using CoCo+



3

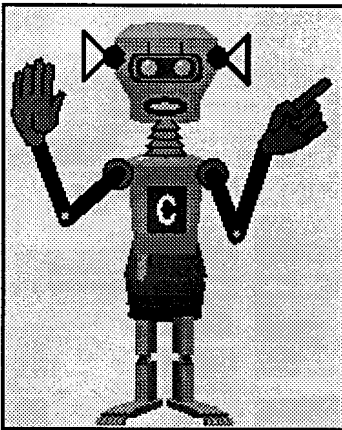
Why don't you?

Make a Clown's face do lots of things. You can make the clown talk, wink, flash the nose and spin the bowtie one after the other in a single procedure!



- You could add a buzzer in and make it buzz every time the nose flashes.
- Think up some funny ideas and put them into action.

Here are some ideas...give them a try!



3

Make a clown and build the procedures Wink, Nose and Tie as described on work card 2. Did you know that you can put all these procedures into one new procedure and add more new commands if you want.

Give it a go!

```
Procedure 'WINK'  
say "Look at my eyes I can wink"  
switch on 3 and 4  
wait 2  
switch off 3  
wait 1  
switch on 3  
wait 2  
say "Did you like that?"  
end procedure
```

```
Procedure 'NOSE'  
Repeat 10  
switch on 5  
wait 1  
switch off 5  
wait 1  
end repeat  
end procedure
```

```
Procedure 'TIE'  
motor b forwards  
wait 5  
motor b backwards  
wait 5  
motor b off  
end procedure
```

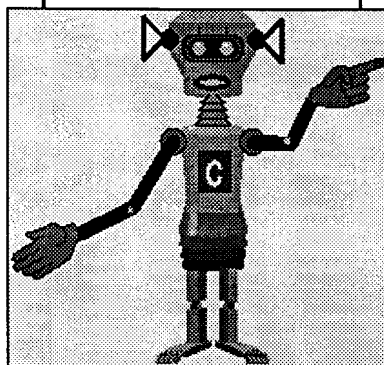
```
Procedure 'CLOWN'  
Say "I am a funny clown"  
Say "I can wink my eyes"  
wink  
say "I can flash my nose"  
nose  
say "I can also spin my tie"  
Tie  
say "Clever arn't I"  
end procedure
```

The procedure CLOWN will link all the actions together and have something to say.

HELPFUL HINTS

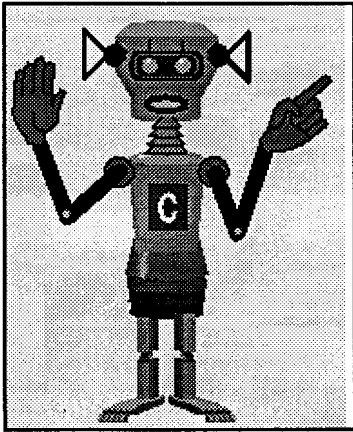
Plug a buzzer into output socket 6. Change the procedure Nose to switch on 6 at the same time as 5.

Type:
Change Nose <Return>



```
Repeat 10  
switch on 5 and 6  
wait 1  
switch off 5 and 6  
wait 1  
end repeat  
end procedure
```

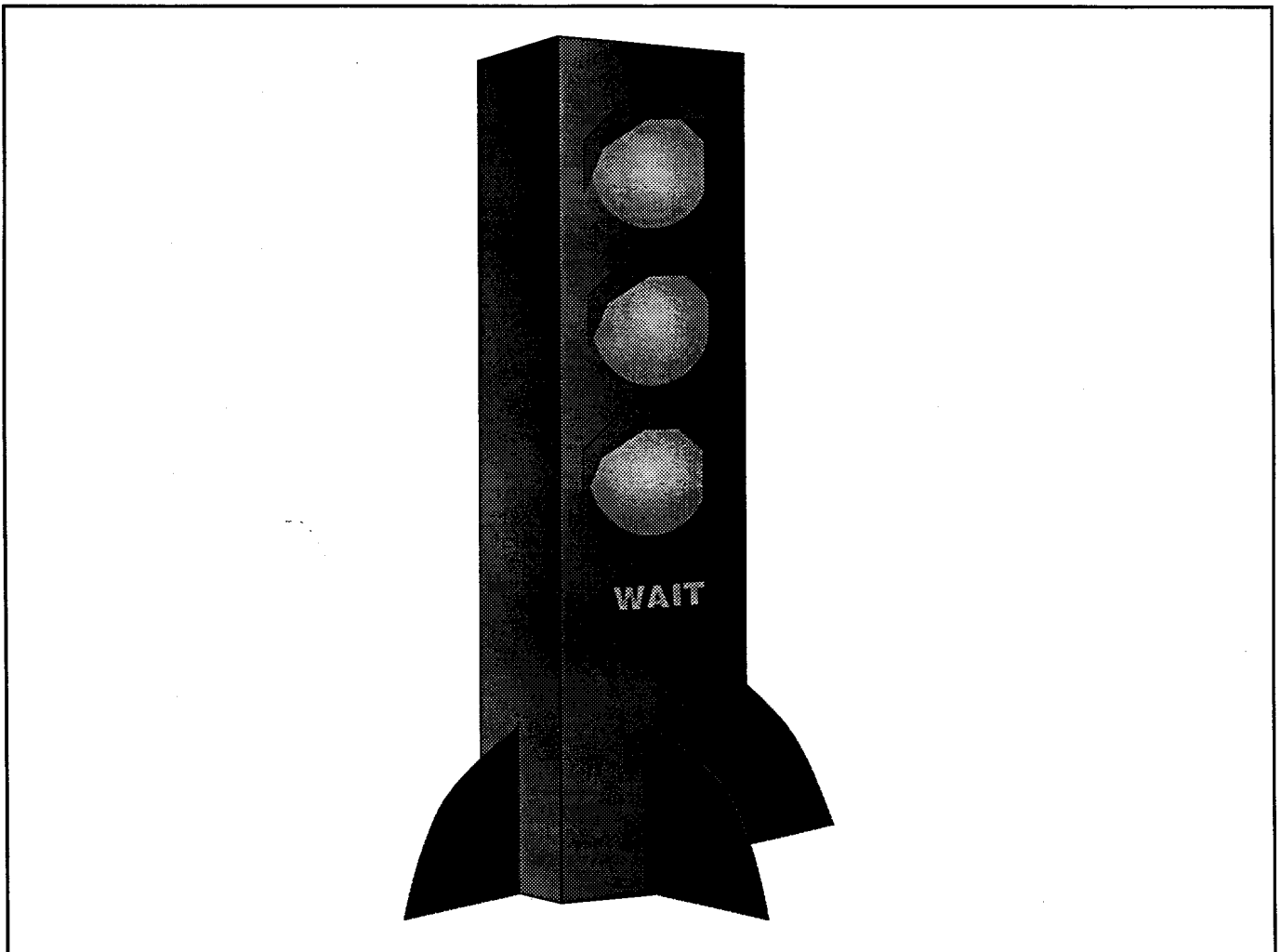
Traffic Lights...Using CoCo+



4

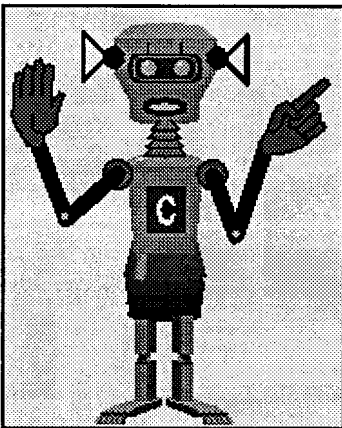
Why don't you?

Make a set of traffic lights. The traffic lights you see on the roads have a set pattern. Find out what this is and build in the same sequence for your lights.



- When the lights are red do they give pedestrians time to cross?
- Measure and then mark out the road width on the playground, then time how long it takes to cross.
- Change your procedure to allow for this.

Here are some ideas...give them a try!



4

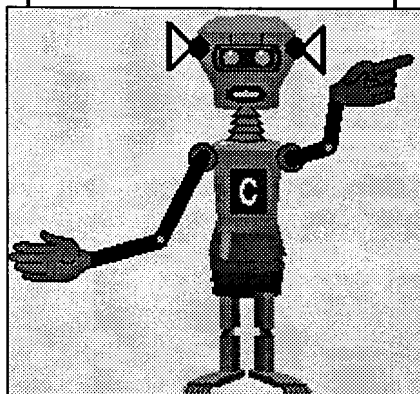
Plug a red bulb into output 8, an amber bulb into output 7 and a green bulb into output 6. If you don't have coloured bulbs try painting them or using coloured tissue paper. Indelible overhead projection pens are also ideal for this.

```
Procedure TRAFFIC**
Repeat forever
switch on 8
wait 5
switch on 7
wait 2
switch off 7 and 8
switch on 6
wait 5
switch off 6
switch on 7
wait 2
switch off 7
end repeat
end procedure
```

The wait command after each line determines how long the lights stay on for.

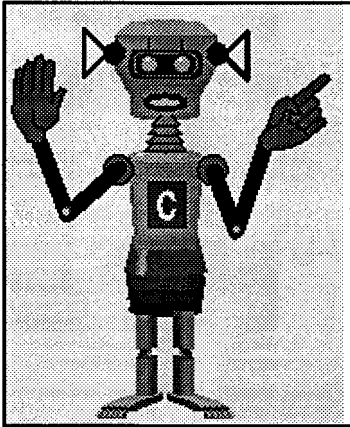
HELPFUL HINTS

Use traffic cones or markers to represent a road crossing on the playground. Time some of your friends to see how long it takes them to cross the road. If it takes 10 seconds, change the wait time after the red light comes on to 10.



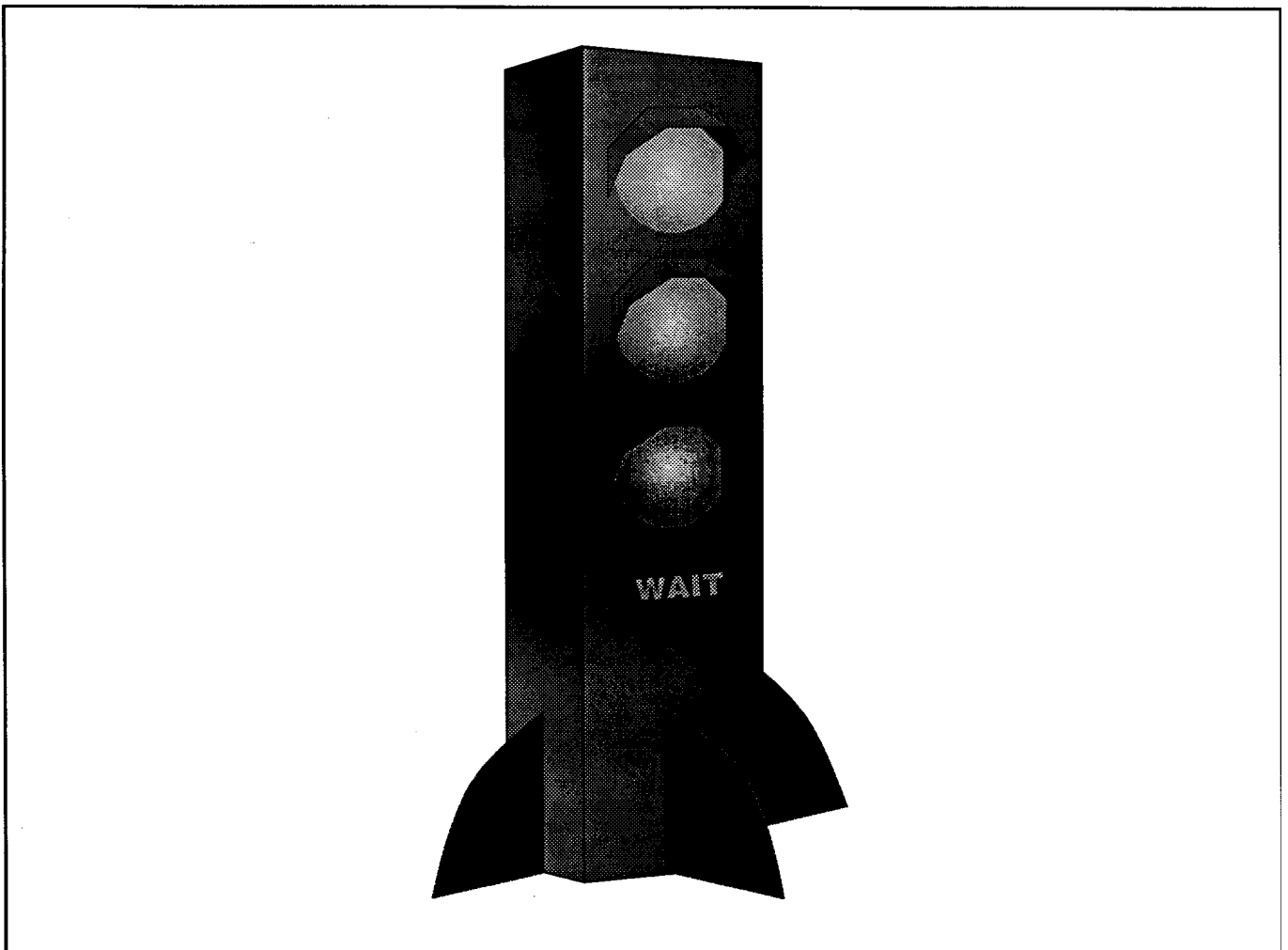
```
Repeat forever
switch on 8
wait 10
switch on 7
wait 2
switch off 7 and 8
```

More with traffic lights...Using CoCo+



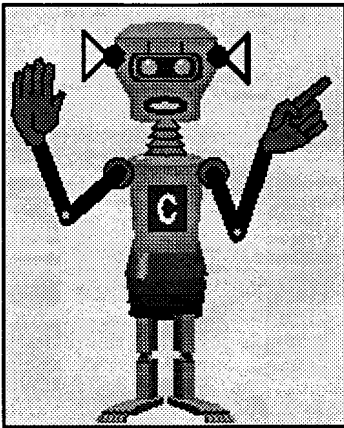
5 Why don't you?

Make a set of traffic lights so that when a button is pressed a WAIT sign telling pedestrians not to cross is lit up.



- Pedestrians have the right of way to cross the road when the lights are red or if the amber light is flashing.
- This is okay if you are able to see . . . what if you are blind? Can you think of a way to let blind people know when it is safe to cross?

Here are some ideas...give them a try!



5

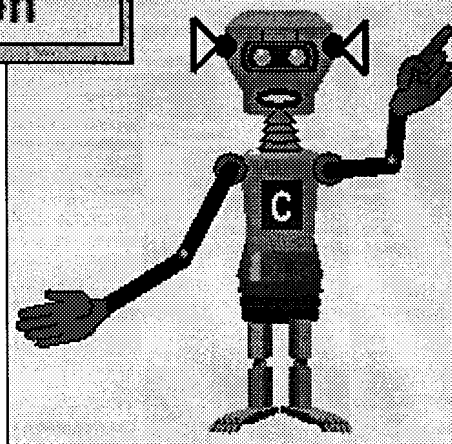
Plug a red light into output 3, an amber light into output 2 and a green light into output 1. A sign saying WAIT could be lit up with a bulb and plugged into output socket 4. Plug a push switch into input socket 1.

Build Traffic <Return>

Label the outputs to match the colour of your bulbs.

4	
3	red
2	amber
1	green

Note that this line just makes sure that the wait sign is off when the lights are green

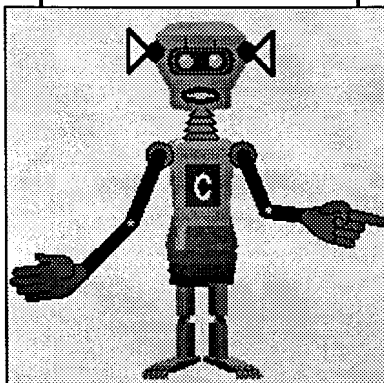


```

Procedure TRAFFIC
whenever input 1 on switch on 4
repeat forever
switch on red
wait 5
switch on amber
wait 2
switch off red and amber
switch on green
switch off 4
wait 5
switch off green
switch on amber
wait 2
switch off amber
end repeat
end procedure
    
```

HELPFUL HINTS

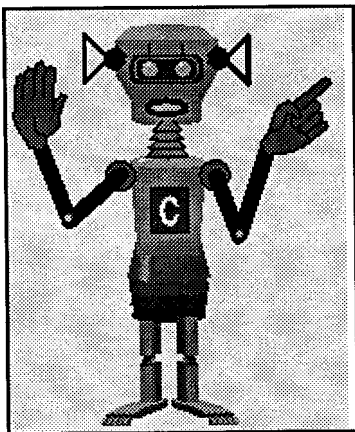
After the line that turns the lights green, insert a SAY command to tell people it is safe to cross. Just before they turn amber insert another to say "do not cross ..."



```

switch off 4
say "The lights are now green, you may cross"
wait 5
say "The lights are going red, do not cross"
switch off green
switch on amber
    
```

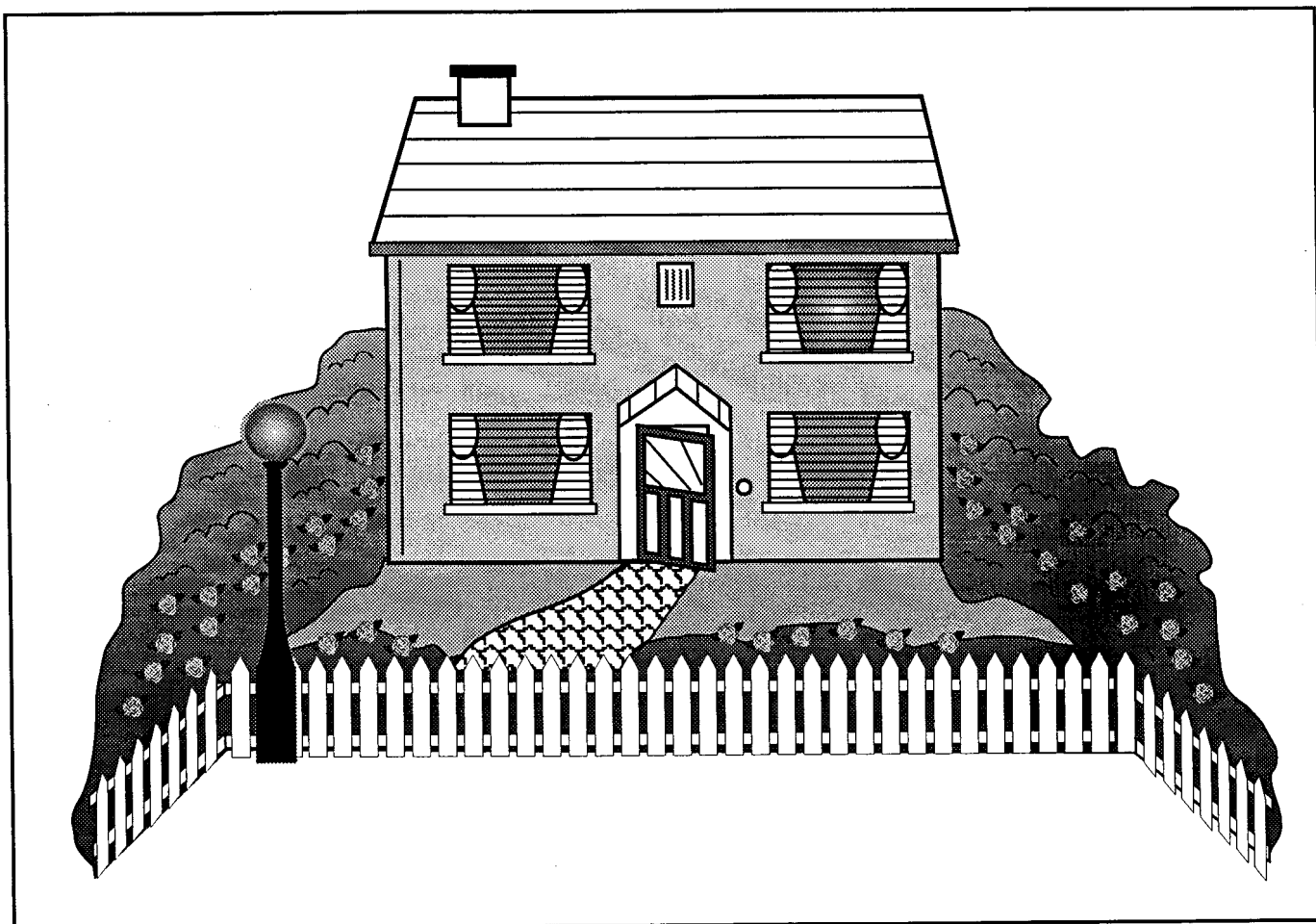
Home from Home...Using CoCo+



6

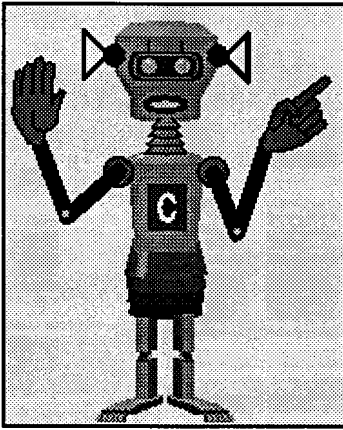
Why don't you?

Build a model house with rooms that light up. Make a door that opens and a bell that buzzes when a button is pressed. Write a short story about the house and make it come to life.



- Perhaps you can devise a burglar alarm for your house.
- Connect it to the front door.
- Write a procedure so that when the front door is opened the alarm will go off.

Here are some ideas...give them a try!



6

Make a model house from cardboard. Show 4 windows each with a light plugged into outputs 1, 2, 3 and 4. Cut a door in the front that opens and near it place a push switch and plug it into input 5. A buzzer can be fitted anywhere in the house but plug that into output socket 5.

Don't forget to label the output sockets

1 ... Lounge
4 ... Bedroom

2 ... Dining
5 ... Buzzer

3 ... Bathroom

THE LOST KEY

Late one day Simon came home from work (*no lights on in the house*). It was dark and there were no lights to be seen. Realising he had lost his key that morning Simon rang the door bell (*press button and buzzer sounds*). His wife, Shena, opened the door and said... (*Computer talks*). They went into the lounge (*front lights go on*) and sat watching TV for a while. When dinner was ready they left the lounge (*front lights go off*) and went into the dining room (*another light comes on*). After dinner and feeling very tired they decided to go to bed. Shena went upstairs and was first in the bathroom. (*Upstairs light goes on*). Simon soon followed (*Dining room light off bedroom light on*). Soon they were both ready for bed (*Bathroom light off*). They turned the lights out and went to sleep (*all lights off*). Say "Good night Everybody".

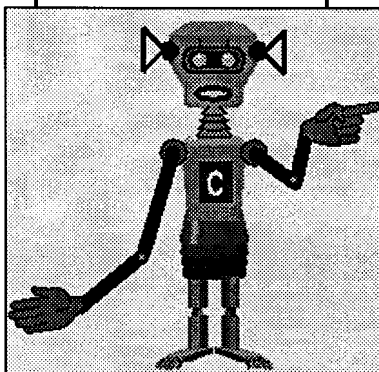
Build story <Return>

```

Procedure 'STORY'
whenever input 5 on switch on buzzer
whenever input 5 off switch off buzzer
say "You're late, forget you key again?"
say "Dinner will be ready soon"
wait 2
switch on lounge
wait 10
switch on Dining
switch off lounge
wait 10
switch on Bathroom
wait 5
switch on Bedroom
switch off Dining
wait 5
switch off bathroom
wait 3
switch off all
say "Good night everybody"
    
```

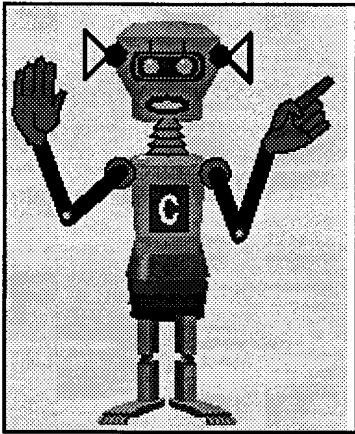
HELPFUL HINTS

Place a magnet sensor just behind the door and plug it into input socket 1. Now put a magnet on the door so that when the door is closed they are near, and sensor 1 shows as ON. Build procedure Burglar (see opposite)



Whenever input 1 off
alarm

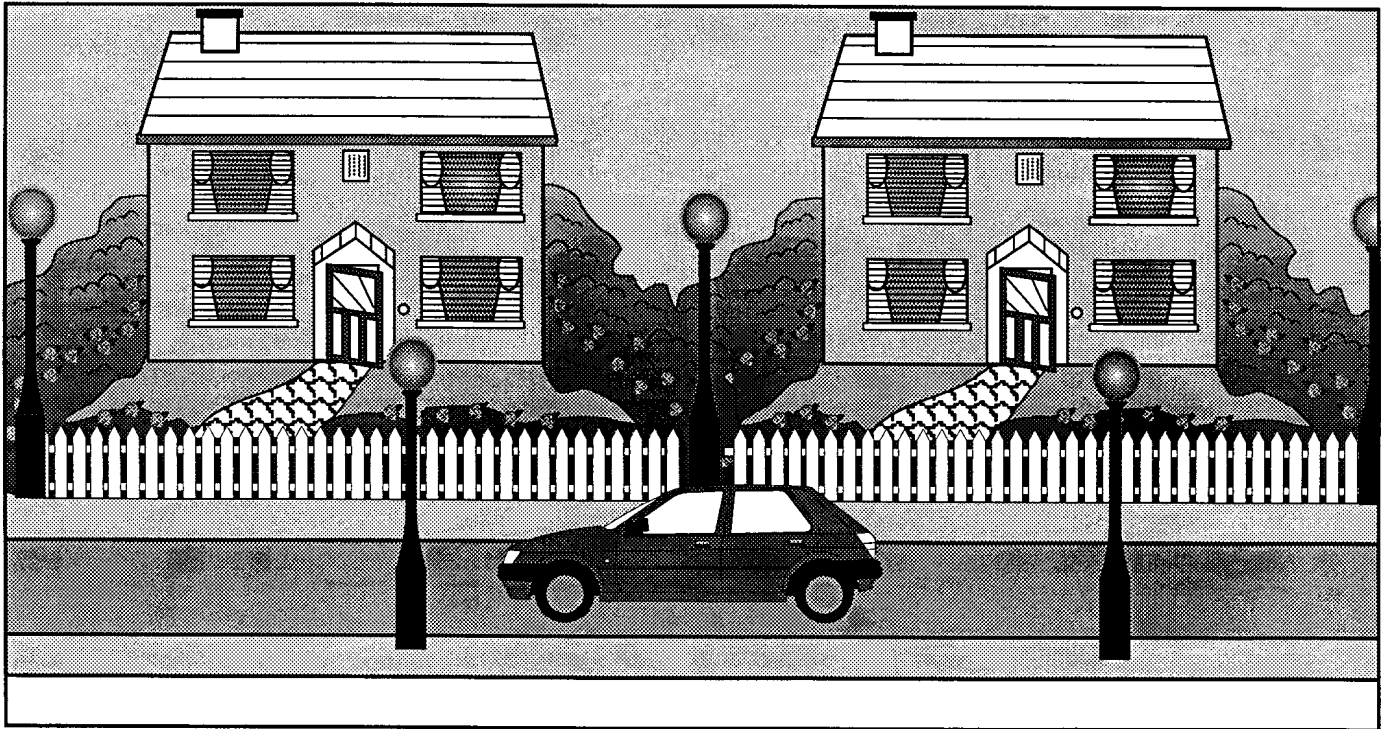
Lighting up time...Using CoCo+



7

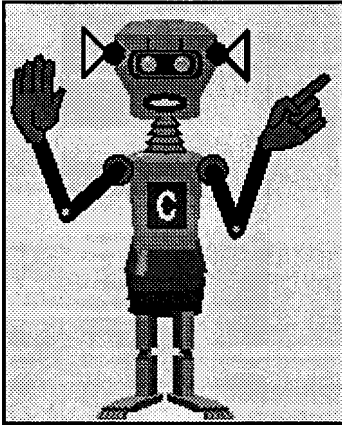
Why don't you?

Make a small model town with houses and a road. Make some street lamps for the town that light up. Can you write a procedure that will switch the street lights on when it gets dark and turn them off when it becomes light again?



- Did you know that you can sense the amount of light in any place?
- You could use a sensor to tell the computer to turn the street lights on when the room light goes below a certain level.
- Write a procedure that uses a light sensor to switch your street lights on and off.

Here are some ideas...give them a try!



7

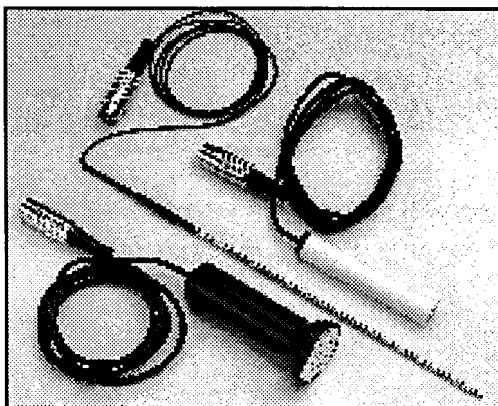
Make a model town from cardboard boxes and a street lamp using some doweling and a light bulb. Connect it up to output socket 6. Connect a light sensor into input socket 6 to check if it is on when the room is bright and off when the room is dull.

Build LIGHT <Return>

```
Procedure "LIGHT"  
Whenever input 6 off switch on 6  
whenever input 6 on switch on 6  
end procedure
```

Hint!

You may be able to improve the sensitivity of the light sensor by placing it in a tube. Put just enough cotton wool in the end so that when the lights are on the sensor is on but when the lights are off the sensor shows off.

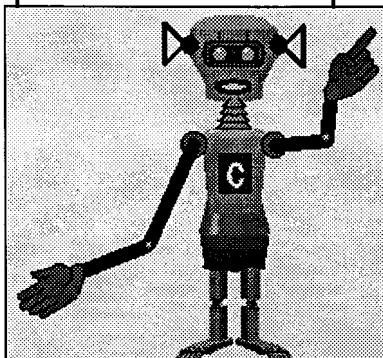


If you have a procedure story (see 5)
Change STORY <Return>

```
Procedure "STORY"  
Light  
whenever input 5 on switch on buzzer  
whenever input 5 off switch off buzzer  
.....
```

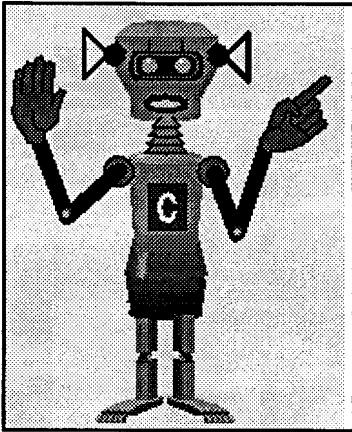
HELPFUL HINTS

If you have a light sensor, plug it in to sensor socket 1 in front of the control box. Show the sensor window by clicking on [Sensors] in the Panel window. Read the amount of light sensor 1 shows. Now turn the room lights off and read sensor 1 again. Write a procedure called SUN.



```
whenever sensor 1 < 50  
switch on 6  
whenever sensor 1 > 50  
switch off 6  
end procedure
```

Build a Barrier...Using CoCo+

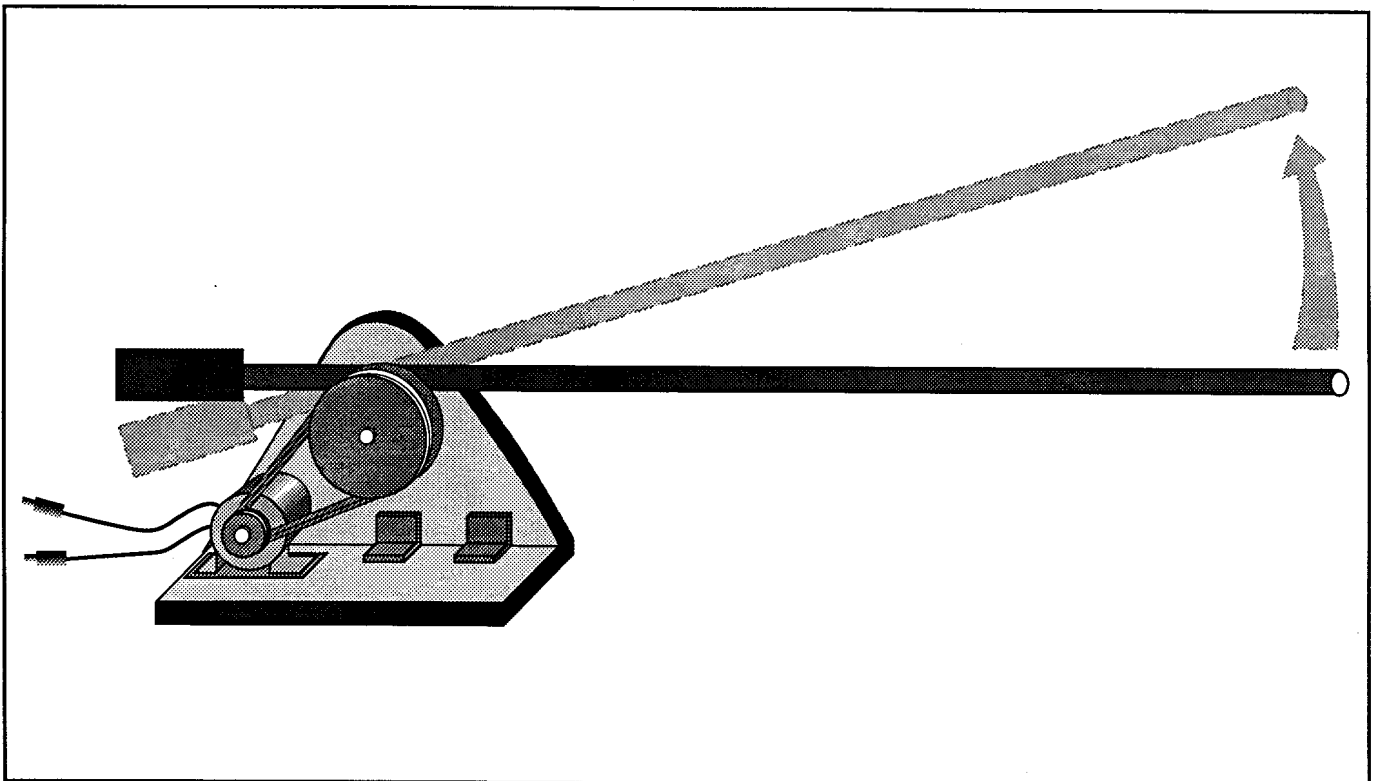


8

We see all around us barriers of different types, in supermarkets, car parks and at train crossings etc.

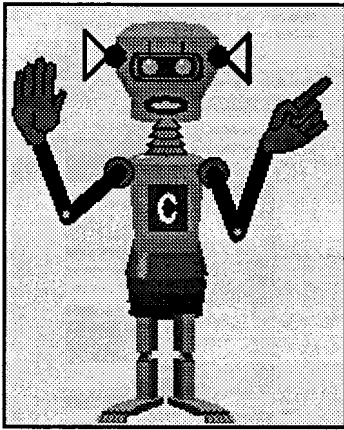
Why don't you?...

Build a barrier that can go up and down.



- Using a tilt sensor you can detect when the barrier is fully upright or horizontal.
- Choose a sensor you think is best for this.

Here are some ideas...give them a try!



8

To build a barrier that goes up and down, you could use a construction kit or perhaps make it from wood and glue. Connect a motor to pull it up and lower it. Plug the motor into motor socket A and write some procedures to make it go up and down.

Build UP <Return>

```
Procedure 'UP'
power a 5
motor a forwards
wait 1
motor a backwards
wait 1
power a 10
end procedure
```

Build DOWN <Return>

```
Procedure 'DOWN'
power a 5
motor a backwards
wait 1
motor a off
power a 10
end procedure
```

Build Barrier <Return>

```
Procedure 'BARRIER'
repeat 5
up
wait 3
down
wait 3
end repeat
end procedure
```

Help!

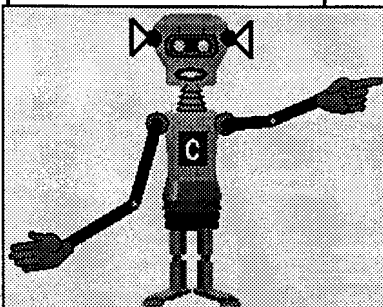
It may happen that your barrier goes down on the UP command and up on the DOWN command. If so, just reverse the plugs in socket A and all should be well.

Help again!

If the barrier goes up too fast or too slow alter the power command to between 1 and 10

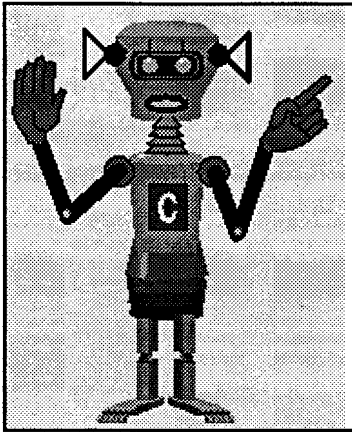
A tilt switch is a good idea. Connect one to your barrier and plug it into input socket 1 so that when the barrier is down it will show ON.

Now change DOWN



```
Power A 5
motor A backwards
wait until input 1 on
motor A off
end procedure
```

An "Are you there" Barrier...Using CoCo+

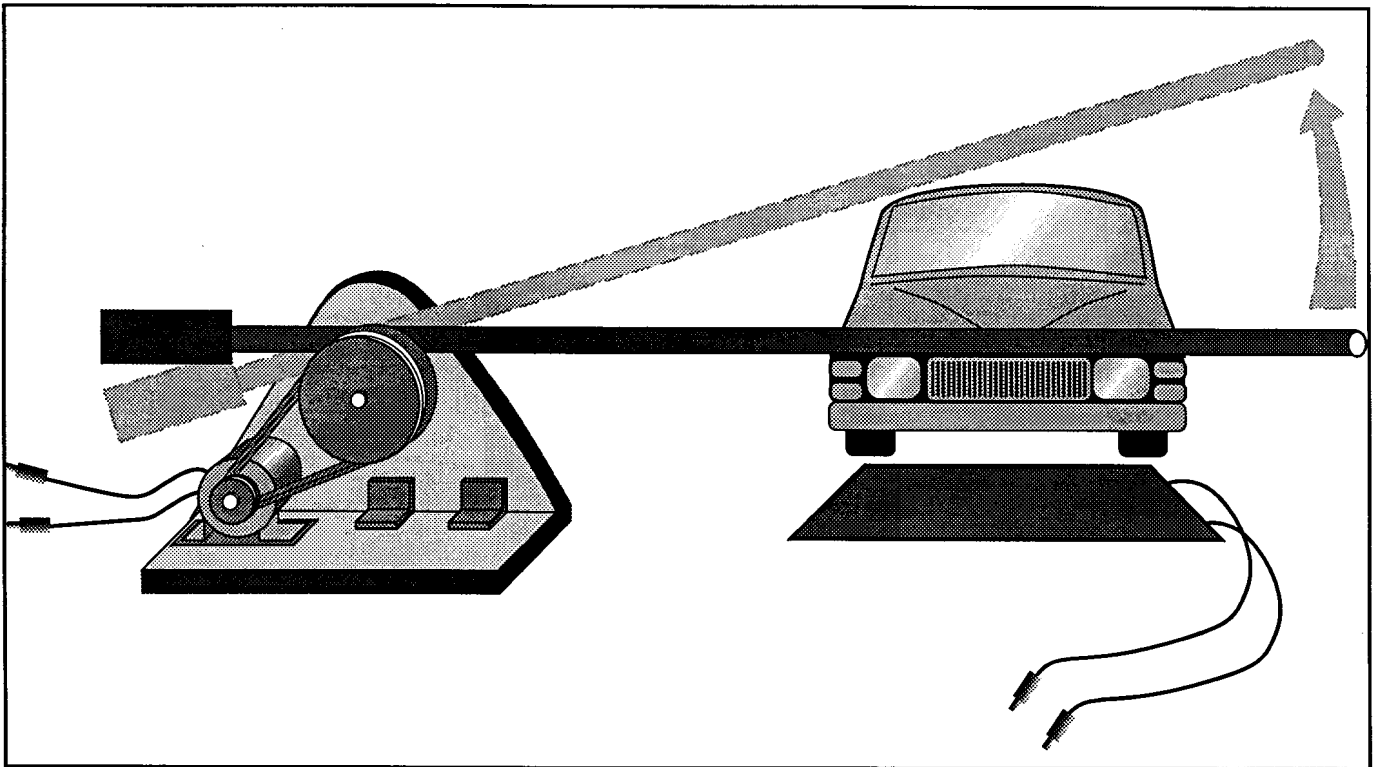


9

A car park barrier knows when to raise because a ticket is removed

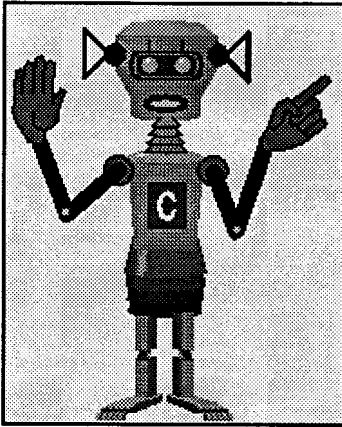
Why don't you?

Build a barrier with a device that can detect if a ticket is taken.



- A barrier also knows when to come down after the car has passed through. Can you make a barrier that can do all this?
- Add a feature so that when you have collected your ticket and passed a safe distance beyond, the barrier will close.

Here are some ideas...give them a try!



9

Near your barrier place a light sensor and a light bulb within close proximity. Connect the sensor into input socket 3 and the bulb into output socket 3. You will need to make certain that when the light is on the sensor shows as ON, and when the light is off the sensor shows as OFF

Build your procedures to make the barrier go up and down

Build Barrier <Return>

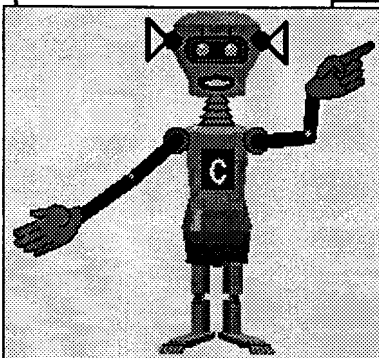
```
Procedure 'BARRIER'  
switch on 3  
repeat forever  
wait until input 3 on  
up  
wait 5  
down  
wait until input 3 off  
end repeat  
end procedure
```

Notice that the light in output 3 is always on. To avoid this, plug a light sensor into output socket 4 and place before the barrier making certain that when a vehicle passes near, it shuts out the light and the sensor turns OFF.

Change Barrier <Return>

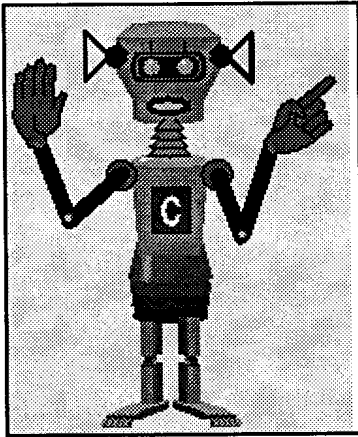
```
Procedure 'BARRIER'  
repeat forever  
wait until input 4 off  
switch on 3  
wait until input 3 on  
up  
wait 5  
down  
wait until input 3 off  
switch off 3  
end repeat  
end procedure
```

You could enhance the barrier by placing a light sensor just after the barrier connected to output 5. When the light beam is broken the barrier will come down and the sequence will be repeated again.
Change Barrier <Return>



```
up  
wait until input 5  
off  
down
```

Make a Disco Light Show...Using CoCo+

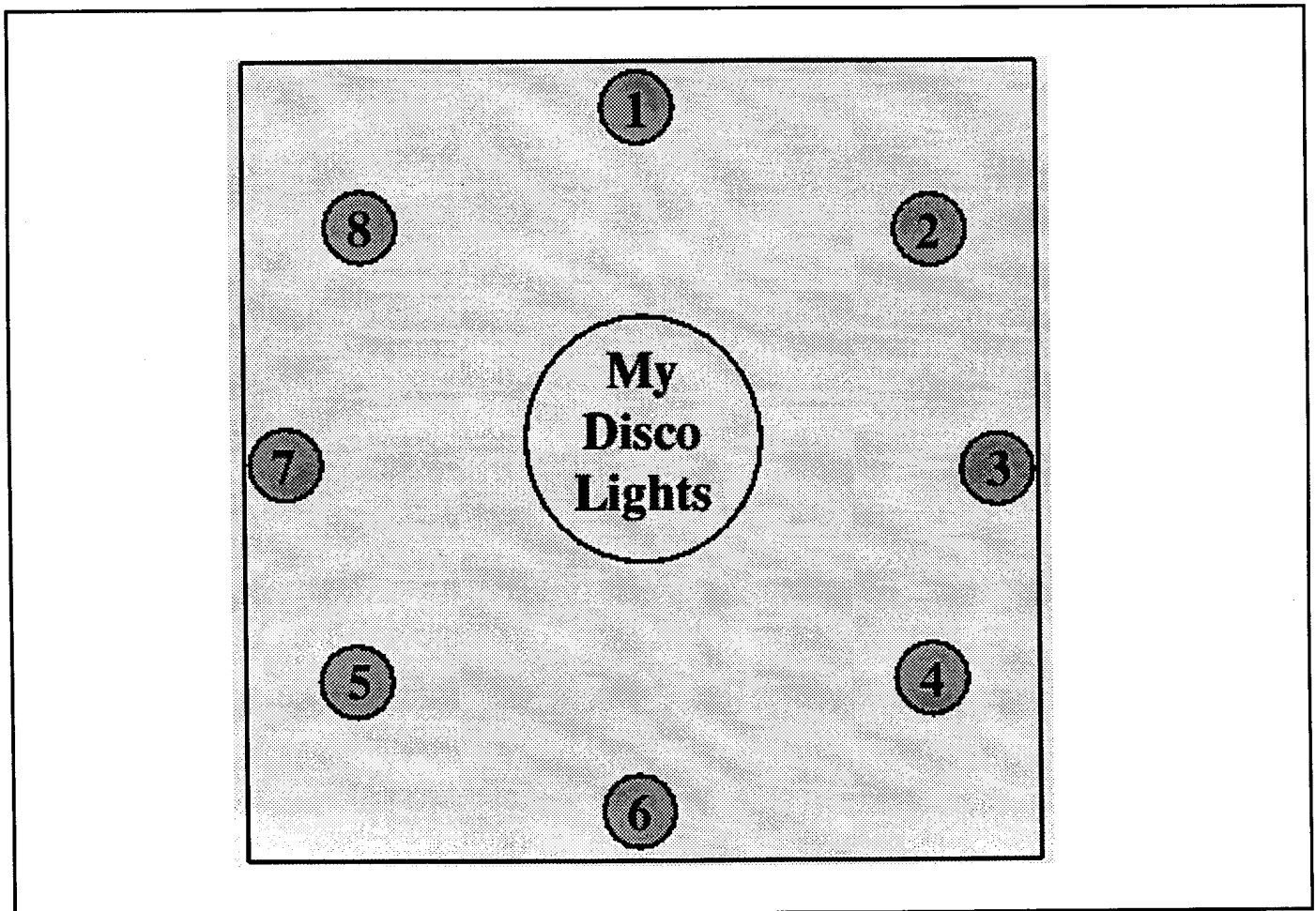


10

If you have lots of light bulbs ...

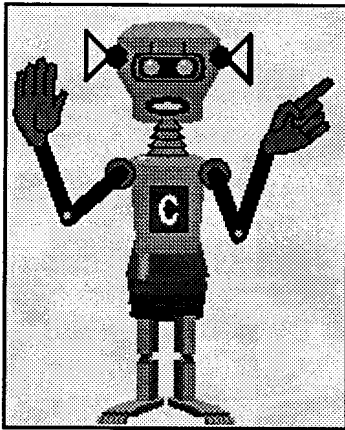
Why don't you?

Connect some to the control box and write some procedures to make different patterns of flashing disco lights.



- Can you link some of your procedures into one single procedure?
- Play some music and see if you can make light patterns to match the rhythm.
- Change your procedures to keep in time with the music.

Here are some ideas...give them a try!



10

Get a piece of stiff card and fix 8 lights in the shape of a circle. Plug these into output sockets 1 to 8. Now make the lights go on and off in turn to make it look like it is going round in a circle...first one way, then the other.

Build Spin 2 <Return>

```
Procedure 'SPIN1' *
repeat 10
switch on 1
switch off 1
switch on 2
switch off 2
switch on 3
switch off 3
switch on 4
switch off 4
switch on 5
switch off 5
switch on 6
switch off 6
switch on 7
switch off 7
switch on 8
switch off 8
end repeat
```

Build Spin 2 <Return>

```
Procedure 'SPIN2' *
repeat 10
switch on 8
switch off 8
switch on 7
switch off 7
switch on 6
switch off 6
switch on 5
switch off 5
switch on 4
switch off 4
switch on 3
switch off 3
switch on 2
switch off 2
switch on 1
switch off 1
end repeat
```

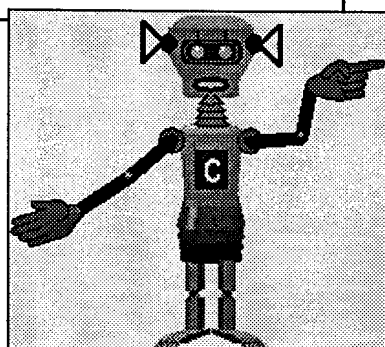
```
Procedure 'FLASH1' *
repeat 10
switch on 8 and 2
wait 1
switch off 8 and 2
switch on 5 and 4
wait 1
switch off 5 and 4
end repeat
```

```
Procedure 'FLICKER' *
repeat 20
switch on all
switch off all
end repeat
```

You can put all the procedures into one and repeat them in any order and any number of times.

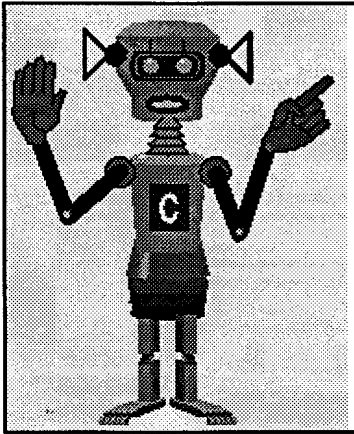
Try this:-

Build Disco <Return>



```
repeat 10
spin1
flash1
spin2
flash1
flicker
end repeat
```

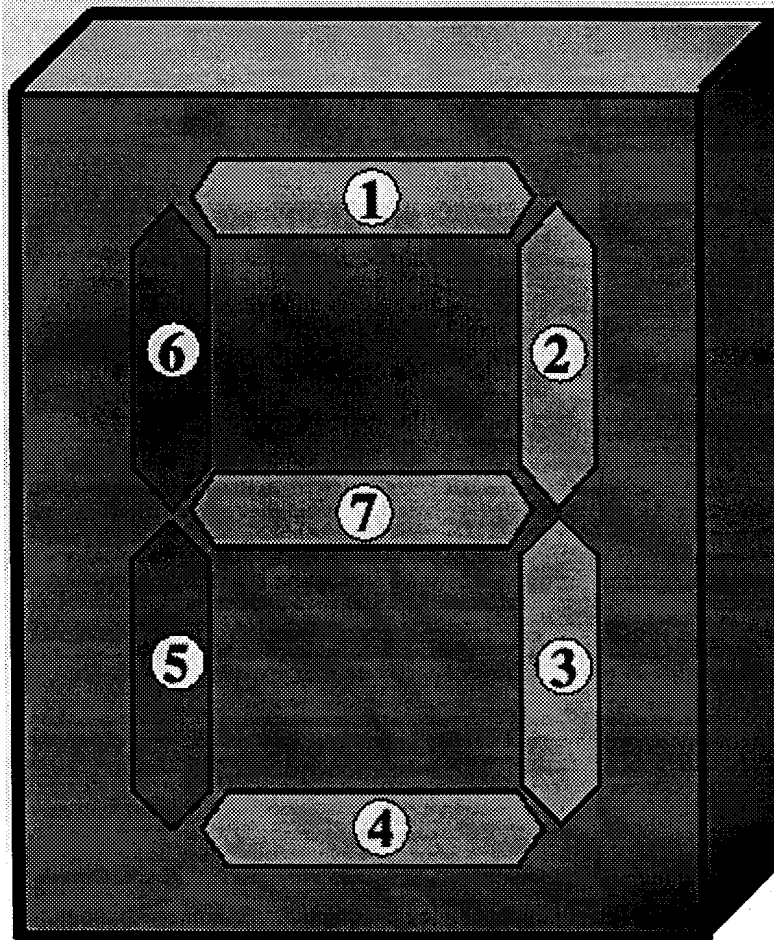
Make a Digital Display...Using CoCo+



11

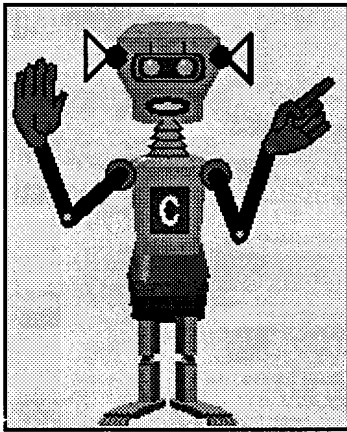
Why don't you?

Make a seven segment display that can be used to make all of the different digits from 0 to 9. You will need a light bulb in each of the seven segments.



- Build a procedure that can make it into a clock that will count down 10 seconds.
- Why not make it talk at the same time?

Here are some ideas...give them a try!



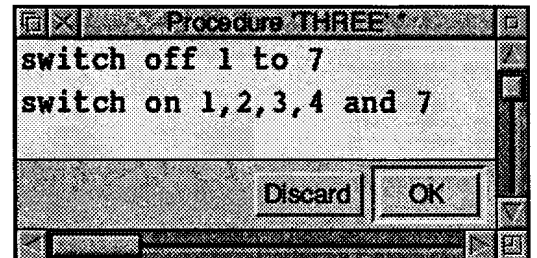
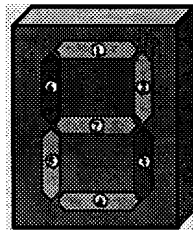
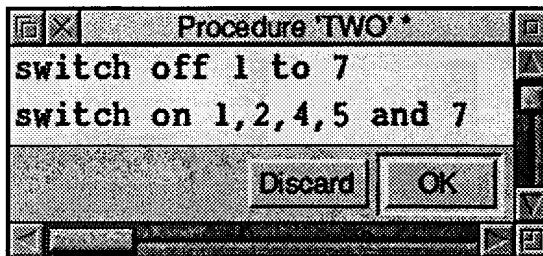
11

Build your display from some card and plug in 7 light bulbs behind each segment. Plug these into output sockets numbered 1 to 7. Switch on each light making certain that it only lights up a single segment. Now build your procedures.

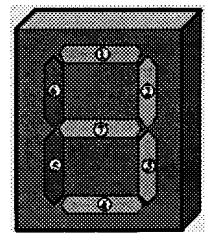
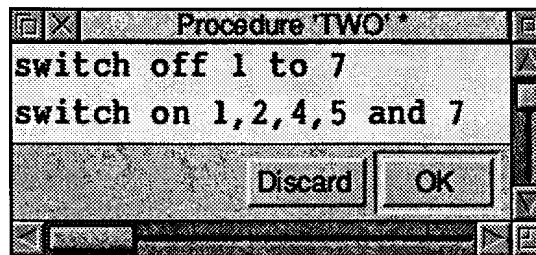
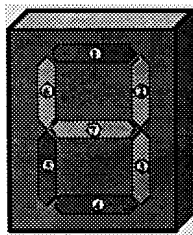
To make the display show numbers build a procedure for each separate number. Here are some examples for you.

Build two <Return>

Build Three <Return>



Build Four <Return>

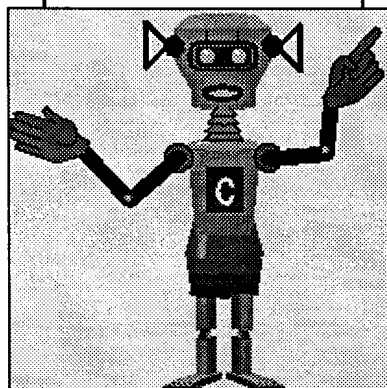


HELPFUL HINTS

When you have made all the numbers you can put them into one single procedure with a pause between each digit.

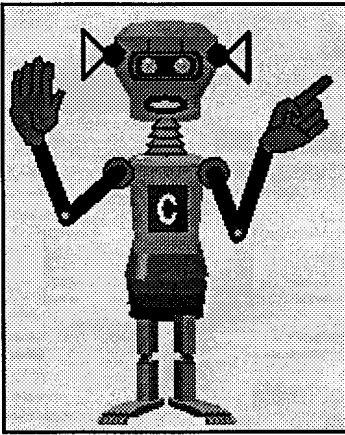
Build Clock <Return>

Help: Use the stopwatch to check how accurate your count down is.



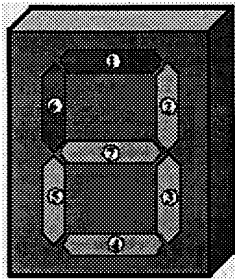
```
wait 1
nine
wait 1
eight
wait 1
seven
wait 1
end procedure
```

Make a Message Machine...Using CoCo+

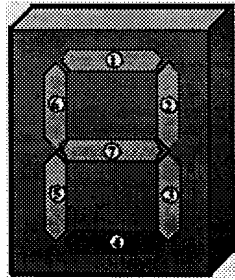


12 Why don't you?

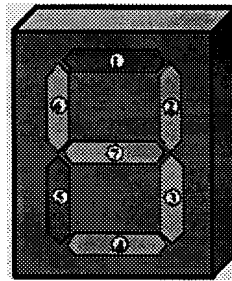
Using a seven segment display, see how many letters of the alphabet you can make. How many names of people you know can you display?



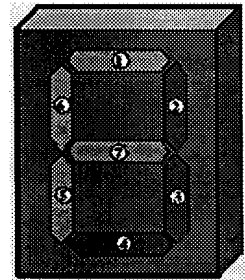
d



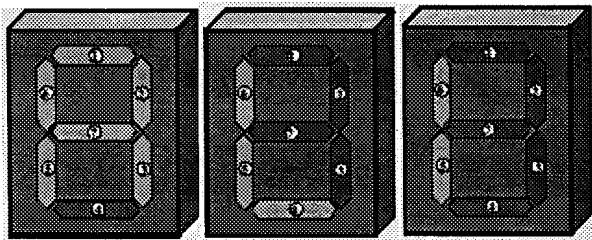
A



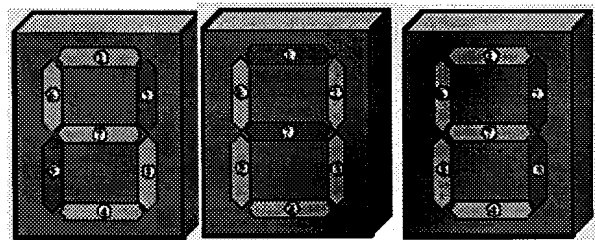
y



F



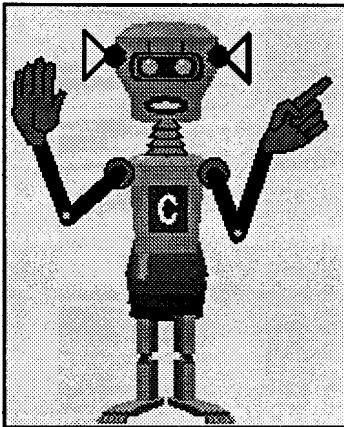
A L I



S U E

- Can you devise a system to send a message using your segment display?
- You can use the letters you have already built in procedures.

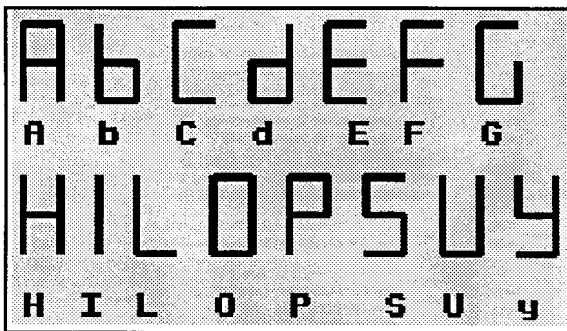
Here are some ideas...give them a try!



12

Work out on paper all the letters you could make. Then build a different procedure for each letter. All the different procedures can go into one single procedure called Message.

Here are some examples for you.



Build Sue <Return

```

switch off all
s
u
e
end procedure
    
```

Build Ali <Return

```

switch off all
a
l
i
end procedure
    
```

Build Message <Return

```

h
wait 1
e
wait 1
l
wait 1
l
wait 1
o
wait 3
s
wait 1
u
wait 1
e
wait 1
switch off all
    
```

Message
reads

HELLO SUE

Build countdown

nine
say "nine"
eight
say "eight"
seven
say "seven"
...
zero
say "zero"

To send a message build a procedure to flash each letter in turn... the wait 1 will keep it on for 1 second.

Note: In building your letters don't forget to switch off all before you display the letter.

