TECHNOLOGY PLAN

ASHBURTON COLLEGE SCIENCE DEPARTMENT

ACHIEVEMENT OBJECTIVES:

A Knowledge and Understanding:
Develop an understanding of....

√√ 1. the use and operation of technologies (use);
√√ 2. technological (a);principles and (b) systems;
(a. considerations b. how things fit together)
√ 3. the nature of technological practice (people who carry out activities to help us);
√ 4. strategies for the communication; promotion and evaluation of technological ideas and outcomes (Promotion, Advertising)

B. Capability:
Students will...

√√ 5. identify needs and opportunities to provide information for possible technological practice (Needs, opportunities, preferences);
6. with reference to identified needs and opportunities.
√√ a. generate possible options and strategies, and select, develop, and adapt possible solutions (Planning);
√√ b. produce technological outcomes to agreed quality standards, managing time, and using human and physical resources skillfully, safely and effectively: (Making);
c. present and promote ideas, strategies outcomes throughout technological practice (Sharing, presenting, discussing;
√√ d. evaluate designs, strategies, and outcomes throughout technological practice in relation to their own activities and those of others (Evaluating).

C. Technology and Society:
Students- should...

7. develop awareness and understanding of the ways beliefs, values and ethics of individuals and groups (People influencing technology):
• promote or constrain technological development;
• influence attitudes towards technological development.
√√ 8. develop awareness and understanding of the impacts of technology on society and the environment (Technology affecting people):
• in the past, present and possible future;
• in local, national, and international settings.

Topic:

Lego Robotics

Year Level:

10 Science

Technological Areas:

Biotechnology
Materials
√ Electronics and Control
Production and Process
Food Technology
Structures and Mechanisms
Information and Communication

Contexts:

√ Personal
Energy
√ Community
√ School
Industrial
Recreational
√ Home
√ Business
Environmental

Essential Skills:

√ Communication
Numeracy
√ Social and Cooperative
√ Problem Solving
Physical
Information
√ Self Management and Competitive
Work and Study

Need / Opportunity:

Road Safety – there have been near misses on pedestrian crossing.
Design a series of light signals so students can cross the road safely.

√ Indicates coverage

√√ Indicates assessment coverage

Introductory Activity:

Brainstorm – what we have now; what we can do to make things better. Then teach the topic once the need to learn programming is identified.

Resources:

Lego sets in S8.

Learning Outcomes:

- recognise the use of technology in road safety, especially as related to pedestrian crossings. (AO A1)
- understanding the basics of the logo programming language. (AO A2 (a))
- combine programming procedures to achieve a workable solution. (AO A2 (b))
- complete a planning sheet for solving the pedestrian crossing problem. (AO B6 (a))
- write a set of procedures to operate the lights according to the specifications on the plan. (AO B6 (b))
- evaluate the effectiveness of the procedures in meeting the specifications of the plan. (AO B6 (d))
- students recognise the advantages of technology in road safety at pedestrian crossings. (AO C8)

Assessment:

1. Planning sheet (planning). (Technological knowledge)
2. Write a plan (making). (Technological capability)
3. Evaluate the result (evaluating). (Technological capability)

Unit Outline:

1. Introduce the issue of the pedestrian crossing.
2. Brainstorm
   - present vs past ideas.
   - possible solutions for the future.
3. Learn programming skills
   - context 1 (fan system).
   - context 2 (car).
4. Problem solving task – the pedestrian crossing.
   - discussion.
   - planning sheet.
   - design.
   - test.
   - evaluate.

Health and Safety *(refer to Appendices in "Safety and Technology Education".)*

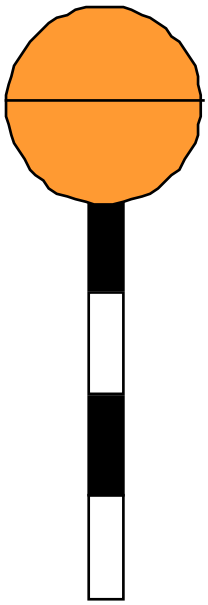
Possible hazards / risks:

- Traffic survey.
- use commonsense when using 230V outlets.

Safety Strategies:

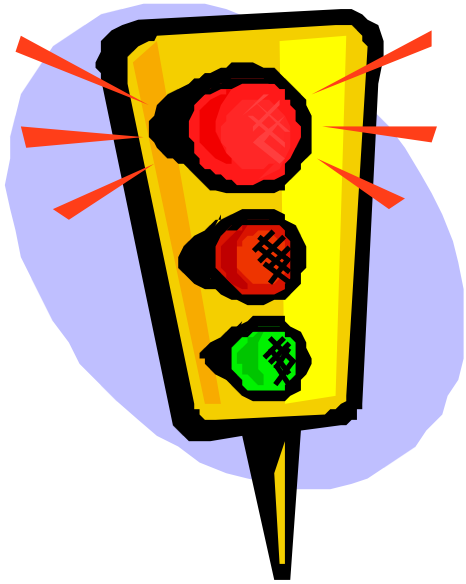
- Keep off the road during the survey. Remain on the footpath.
- Plug in before switching on.

ROBOTICS LEGO - UNIT OUTLINE:



1. Introduce the traffic safety problem at the Walnut Ave Primary School.
2. As a class, brainstorm relevant traffic safety issues, and develop options for a solution for the future. Use this discussion to focus on the need to learn programming skills, which can be applied to developing a solution.
3. Learn programming skills using context one - the fan model (Part A).
4. Problem solving task. Use the programming skills learnt to produce a solution to the problem related to crossing the road safely.
   - Discussion.
   - Complete the planning sheet.
   - Design the plan.
   - Test the plan.
   - Evaluate the solution.
5. Extension tasks (if time).
   - The battery powered car (Part B).
   - Optional extensions (Part D).

Possible hazards / risks:

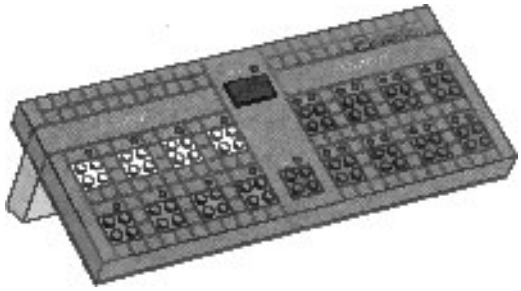


- Traffic survey.
- Common sense re use of 230V electrical points.
- Ensure the interface is connected before powering up.

Safety Strategies:

- Stay off the road. Stay on the footpath during the survey.
- All connections made and 230V plugs in before turning on the power.

CONTROL LAB LEGO



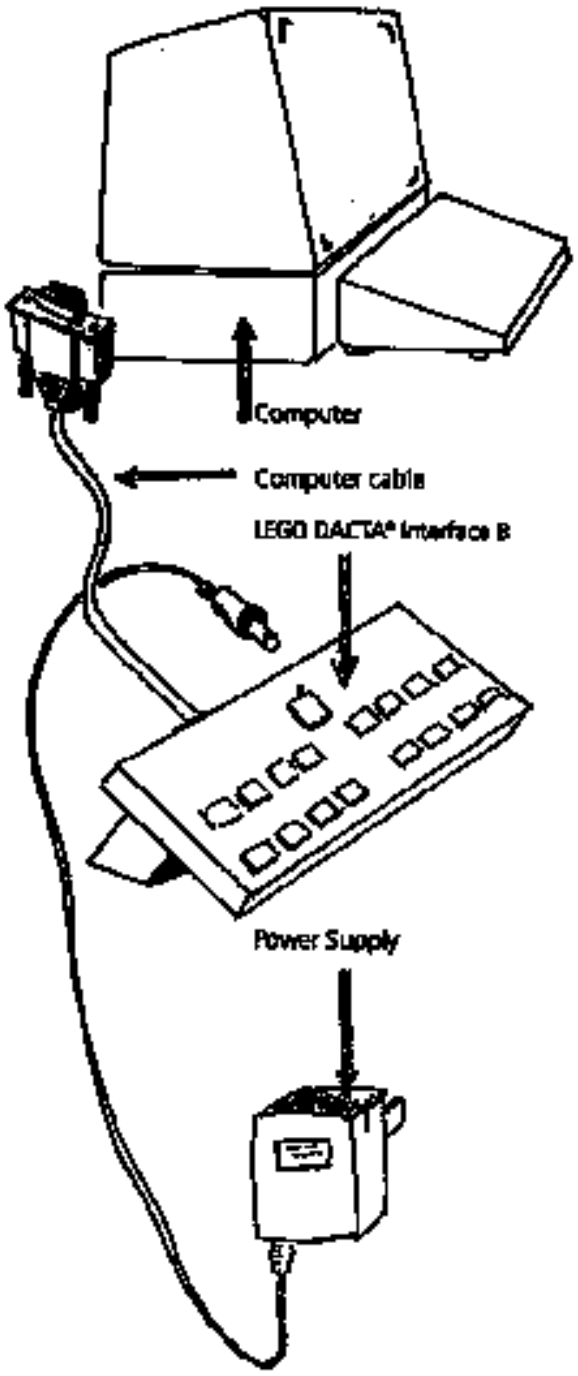
PART A LEARNING THE BASICS

INTRODUCTION

1. RoboLab is software which lets you use a computer to monitor or control lego models.
2. What you will learn using RoboLab, will help you to understand robotics and control technology in real life.
3. You will also develop problem solving and communication skills. You will be asked to construct and programme models, evaluate how they work and then design modifications to improve the performance of your model.
4. You will work in groups of two. It is important that you share the tasks and contribute your ideas so that you are both equally involved in the project. Your teacher will be assessing you on such qualities as participation, cooperation and ability to work as part of a group.
5. Please remove only the pieces you need from the boxes. Keep the bench top clear of all books, bags and pencil cases etc. Take care when packing up as it is easy for some of the small pieces to get lost. Check the floor and bench carefully.
6. When dismantling models, join up the parts to others of the same shape. **Stack the thin plates so one protrudes over the edge of the others** so that they are easier to separate. Storing parts in this way will make finding them easier, cut down on losses and make checking easier at the end.
7. Do not loan pieces to other groups. If you are short of something you need, see your teacher.

SETTING UP

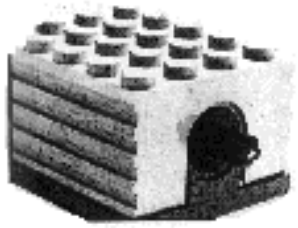
DO NOT TURN ON THE COMPUTER UNTIL YOU HAVE CAREFULLY READ AND FOLLOWED THE INSTRUCTIONS ON THIS PAGE.



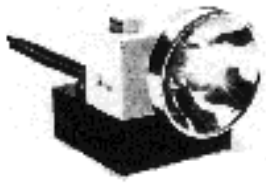
- The lego control lab interface (**CLI**), its power supply and the computer, should all be connected up ready to go.
- The CLI allows you to communicate with motors, lamps and sensors through your computer's serial port. Using RoboLab and the CLI, you can write programmes that can control up to 8 motors or lamps and take readings from up to 8 sensors.
- Just lift the interface and power supply out of their storage cupboard and place on the bench next to your computer.
- Plug the power supply to the 230 volt socket.
- Check that the lead is connected between the interface and the back of the computer, and that the power supply is connected to the back of the interface (refer to the diagram on the right of this page).
- Switch on the power to the power supply.
- **Now** you can turn on the computer and log on.
- When you are finished **DO NOT** disconnect the leads.
  Lock the interface and power supply away with all leads still connected. Take care to ensure that the lead fits through the space at the edge of the door and does not get pinched when the door is closed.

MORE ON SENSORS

1. The sensors are colour coded and must only be connected to the same coloured inputs on the interface.
2. The input devices (sensors) - connect to outputs A to H (blue or yellow).



9V Motor



9V Lamp

9V Sound

3. The output devices - connect to inputs 1 to 8 (black).

Touch Sensor

Temperature Sensor

Angle Sensor

Light Sensor

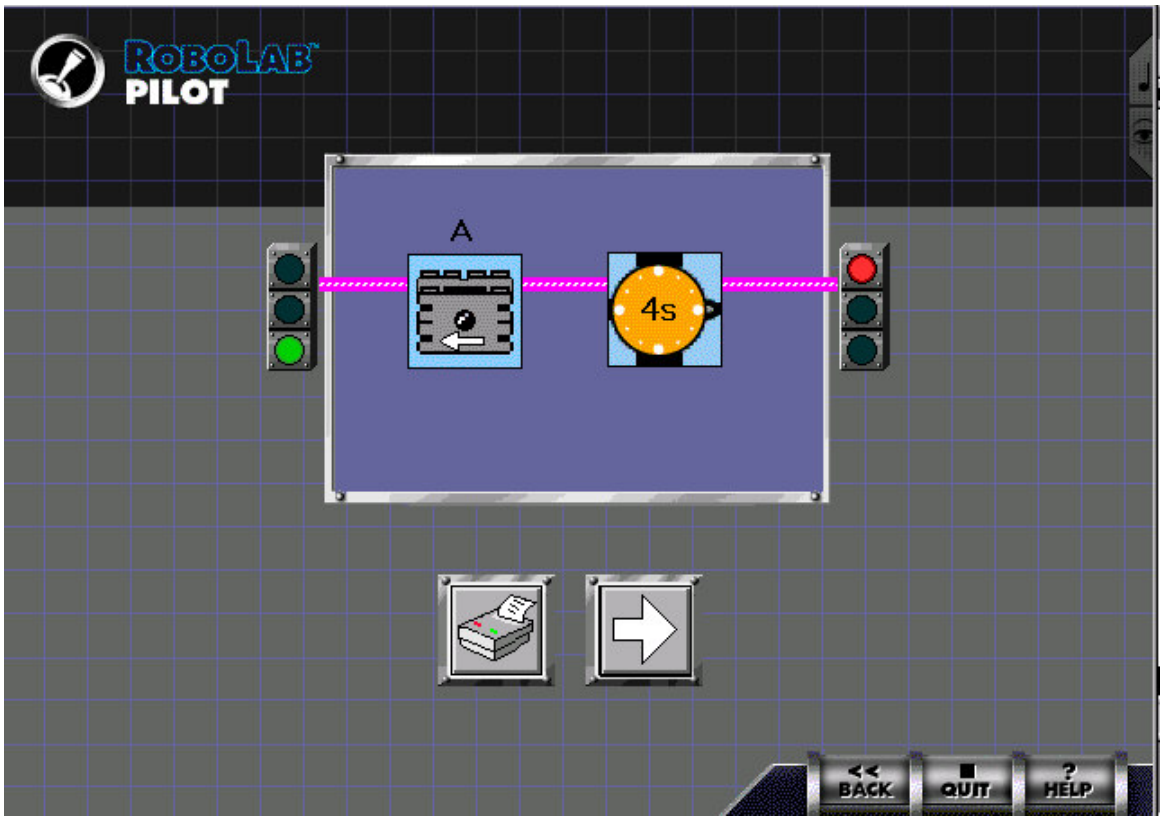
4. Some Hints:
   - If the sound emitter does not go, try rotating the top of it, a quarter of a turn.
   - If the sound sensor is reversed, you can get a different sound.
   - If the lights do not switch on, remove them and then re attach them at 90° to the previous direction.
5. The touch sensor can be used to switch a lamp or motor on or off when it is either pushed or released.
6. The light sensor measures the light intensity (brightness of the light). You can use it to turn on a light when it gets dark for example.
7. The angle sensor measures angles or rotation.

STARTING UP THE LEGO SOFTWARE

1. Log onto the computer in the usual way.
2. Double click on “RoboLab 2.5” icon. (LH button on mouse). You will get a screen from which you can enter the various parts of the Robolab programme. There are a number of demonstration (pilot) programmes which will enable you to learn some of the basics of programming using RoboLab.
3. Click once on the Programmer menu.
   You will now see a screen indicating a number of interactive demonstration (pilot) programmes that you can try, to introduce you to the basics of programming using RoboLab.

NAVIGATING THE PILOT PROGRAMMES

The Pilot programming is an introduction to programming RoboLab. Pilot has 4 levels, with level 1 being the most simple and level 4 having the most flexibility. The four levels build on each other, making it easy to advance to the next level as the you become familiar with the previous level’s options.



Programming Features

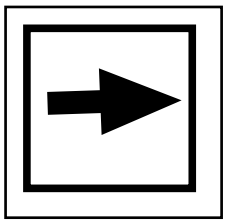
The two traffic lights on the program template represent the beginning and ending of the program. Between the two lights is a box that shows the program that will run. Clicking on the run arrow will start the programme running. The computer will send commands through the interface to the different output devices, or start to receive information from the various sensors attached to it.

Start program (green light)
This is the start of the program.

End program (red light)
This is the end of the program.

Pilot 1:

- First double click on Pilot level 1.
  On screen, you have a programme which will turn on a motor and allow it to run for a short time before stopping.
- Attach a motor to output A on the CLI.
- Click the arrow on the screen to run the programme.



- You can change the programme by clicking on either the motor or the timer icon and selecting (click and drag) one of the options available. if you choose the lamp, you will need to replace the motor attached to output A, with a lamp.
- Try different options and observe what happens.
  If you select the timer with the question mark, you can click on the box below and type in the number of seconds you want.
  Do you now understand what each programming icon does ?
- Click on the BACK button. On the screen which appears, double click on Pilot 2.

Pilot 2:

- You will first need to attach a motor to output A, a lamp to output C and a push sensor to input 1.
- Start the programme by clicking on the arrow button.
- What happens when you push the touch sensor ?
- Try changing the numbers below the motor and lamp icons. What do those numbers control ?
- Click on the BACK button. On the screen which appears, double click on Pilot 3.

Pilot 3:

- Connect two motors to outputs A and C, a lamp to output B and a touch sensor to input 1.
- Click on the arrow button to set the programme running. wait for about 15 seconds and then press the touch sensor.
- Explain what this programme is designed to do.
- You may now press the Back button to return to the pilot menu again. Click on Pilot 3 and then on Amusement Park. Select any of the four programmes which appear.
- You will need to decide which motors and sensors need to be connected to the CLI before you run the programme.
- Explain what this programme is designed to do.

Pilot 4:

- At this stage you will not progress to the pilot 4 level. You will now start some projects to practice some programming on your own, and return to the pilot 4 demonstrations once your programming skills have developed further.
- Press the Back button to return to the main RoboLab menu.

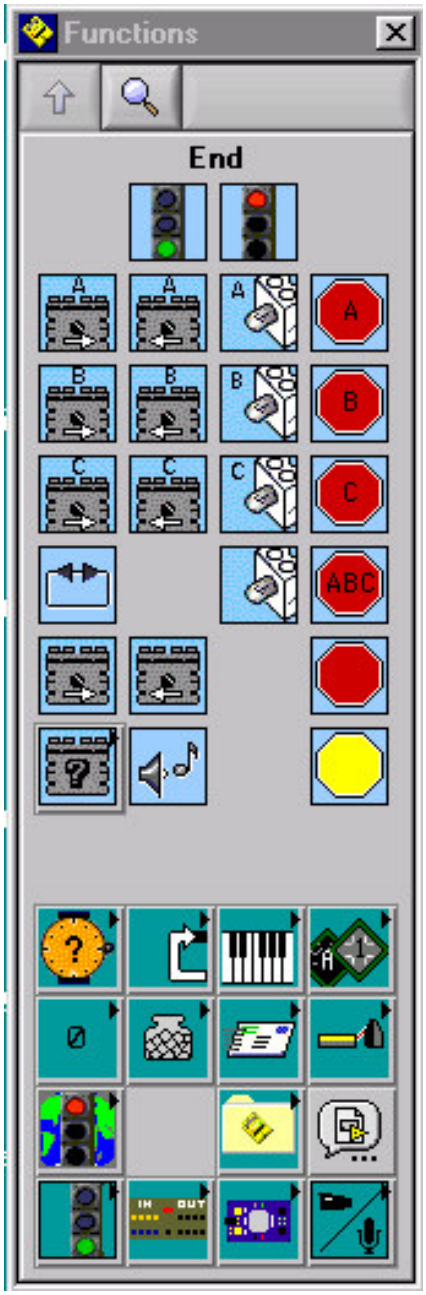
  Click on the Programmer icon.

  Double click on Inventor Level 4.

  You now have a screen, which enables you to write your own programmes.

STARTING PROGRAMMING

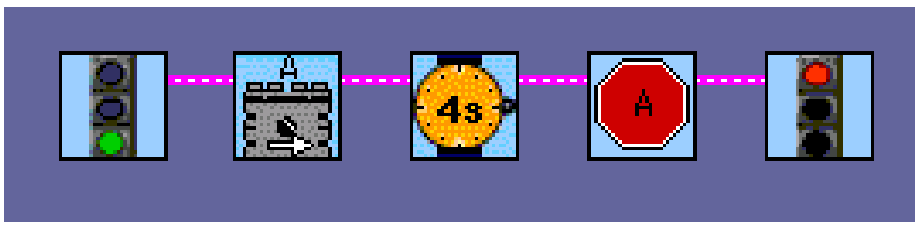
1. Click on the Programmer icon.
   Double click on Inventor Level 4.
   You will get the programming window on your screen.
   It will show a green traffic light on the left, and a red traffic light on the right.
   The commands needed to make a programme, must be placed between these icons and “wired” together.
2. You will need to use the functions palette and tools palette when you start programming. If they are not visible on your screen, go to the “Window” menu and select
   “Show Functions Palette”.
   “Show Tools Palette”.



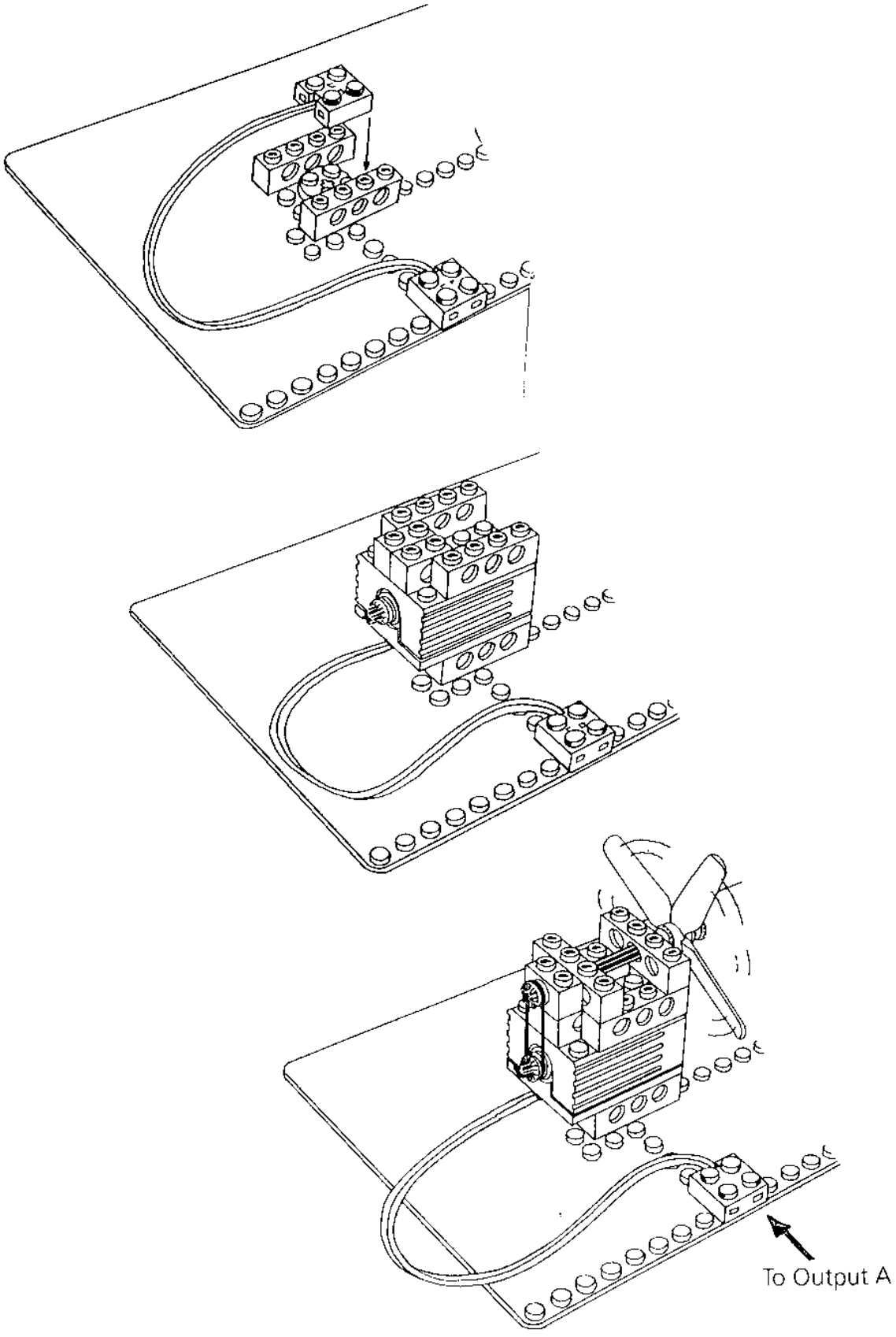
STARTING A PROJECT

TASK 1 Fan Model

1. Assemble the fan model on the next page of this manual. Connect a black extension lead from the model to output A on the interface.
2. Click on the arrow at the top of the tools palette. This allows you to choose and place icons on the programming screen.
3. Note: When you point the arrow at an icon on the functions palette, its name will appear at the top of the functions palette. When you point the arrow at an icon on the tools menu, a box will appear telling you what the tool does.
4. Click on Motor A on the functions palette and drag it onto the programming screen a few millimeters to the right of the green traffic light icon. A pink line should form between them.
5. Click on the orange “wait for” icon lower down on the functions palette. A new menu of wait for icons will open. Choose the icon with 4s, drag it onto the programming window and place it to the right of the motor icon.
6. Click on the ⇑ arrow at the top of the functions palette to return to the full functions menu.
7. Choose the red stop A icon and drag it next to the timer icon.
   All the icons should be joined by a pink wire connection.
8. Click on the red traffic light and move it next to the stop icon. It will have no pink connecting wire. To join it up, click on the connect wire icon on the centre left of the tools menu. Click once on the top right hand side of the stop icon, then on the top left hand side of the red traffic light icon, and a pink connection should appear.
9. Your programme is now ready to run. Your programming screen should look like the one on the diagram below.



FAN MODEL

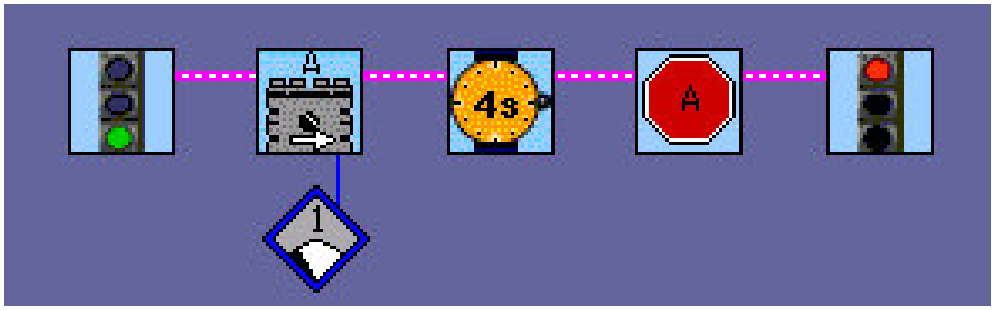


Click on the ⇒ arrow on the menu bar to start it running.
The motor should run for 4 seconds and then stop.
When you run any programme a window will pop up in the upper left hand corner of the screen. This screen is used to communicate between RoboLab and the CLI, and needs to be open when the programme is running. If you close this box, the programme will immediately stop running.

10. For the experts: When a programme is running, it is possible to open the CLI status window, to see what commands are being downloaded to the CLI and which sensors are being read. Do this by clicking on the View checkbox.

11. Changing the programme
    - Can you make the motor run for 10 seconds ?
    - Can you make the motor run in reverse ?

12. You can change the speed of the motor by clicking on the modifier menu button near the bottom of the functions palette, and selecting the appropriate blue speed icon. The speeds can vary on a scale from 1 to 5.
    Drag it onto the programming screen and use the wire tool to link it to the motor. Do this and check that it will work.
    Which is the fastest speed, level 1 or level 5 ?



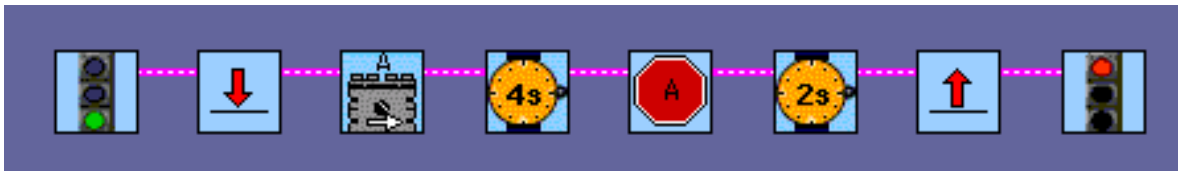
13. **Note:**

- All programmes must start with the green traffic light “begin” icon.
- All programmes must end with the green traffic light “end” icon.
- A pink connecting wire must join each programming icon, to determine the sequence of commands to be obeyed.
- If you cannot move or delete an icon on the screen, make sure you have selected the selection tool (arrow) on the tools menu.
- If you don’t get a pink connecting wire when you place an icon, move it a little closer to the one before it. If you still get no link, select the wire connecting icon on the tools menu the click the top corners on the two icons to be connected.

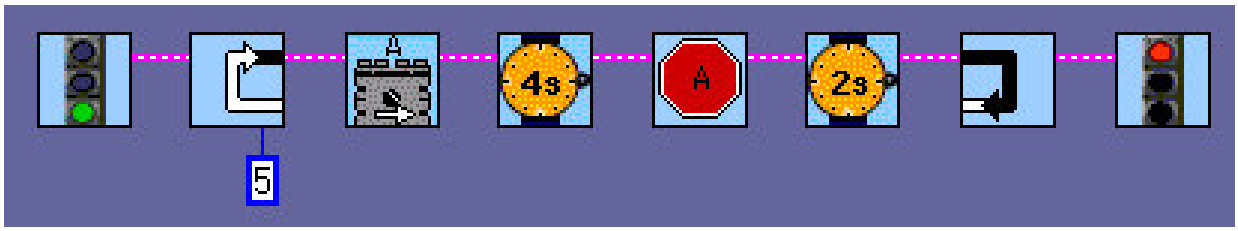
MODIFYING THE MOTION OF THE FAN

1. You should be able to write a programme which makes the fan go slowly for 4 seconds, stop for 5 seconds, then go faster in reverse for 10 seconds before stopping. Do this. If you have problems, see the teacher.
   Note: To select a time of 5 seconds from the “wait for” menu, choose the orange icon with the question mark. Then choose the numeric constant box from the “modifiers” menu and “wire” it to the wait for icon. Select the “edit text” button from the tools menu, then highlight the numeric constant box and type in the time you want.

2. **Feedback Loops - Perpetual Motion.**
   Construct a programme to make the fan go for 4 seconds, stop for two seconds then start again.
   It is possible to amend the programme to do this continuously. Amend your programme by inserting the two arrows to create a loop in your programme. Every time the second arrow is reached, the programme loops back and starts again at the first arrow.



   (The arrows are found in the structures menu - the button next to the orange wait for button. Click on the “jump” button (red arrow) and select the up and down arrows). You will need to remove the wire where you wish to insert the “jump” arrows. To do this, click on the wire with the arrow tool and then press the delete key on the keyboard.
   To stop the programme, close the start box.
   The programme below can be used to make the programme do a fixed number of loops, rather than run continuously.
   How many loops does it do ?
   Amend the programme to do just 3 loops.



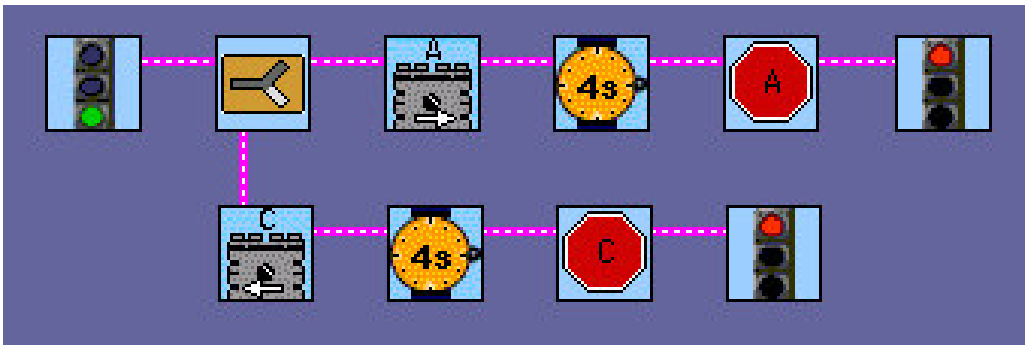
3. Adding Sound - Two Tasks at Once.

It is possible to write a programme which performs two different operations at once. Amend your programme as follows to make the sound come on when the fan is running.

You will need to connect the sound block to output C on the interface.

There is no icon for the sound block on the "functions" menu, so use a motor symbol instead. This means the interface will provide power to output C, which will make the sound operate.

Note: The icon for the branch is found in the "structures" menu.

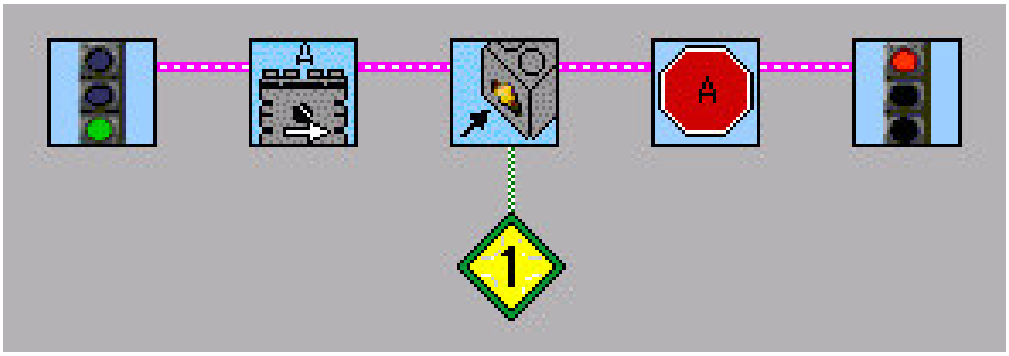


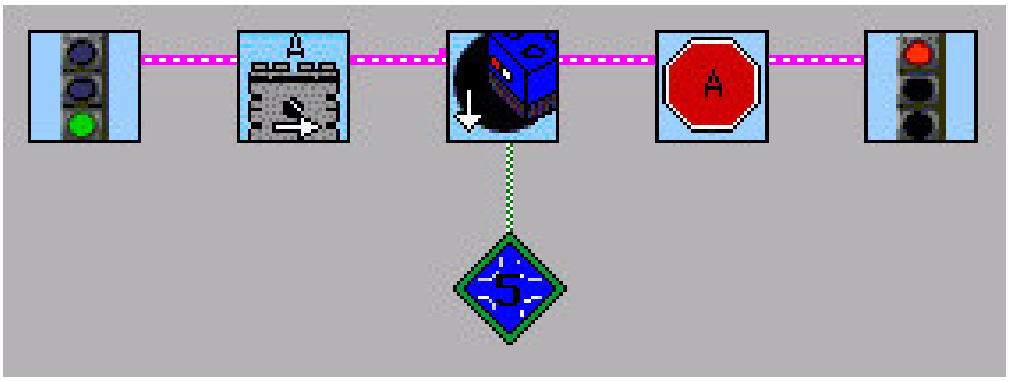
4. You can use different sensors to turn the motor on or off.

Construct and run the following programmes to make the fan run until you push the push sensor or put your hand over the end of the light sensor to reduce the light intensity.

Note: The yellow push sensor must be connected to a yellow (1 to 4) input port and the blue light sensor to a blue (5 to 8) input port.

Note: You will find the input port numbers on the "control lab interface" menu at bottom of the functions palette.


A
A
1


A
A
5

SAVING YOUR WORK

1. If you wish to keep your work so you can use it again in future periods, you will need to save it.
2. To save your work, follow these steps:
   - Click on the “File” menu and select “Save As”.
   - Type in a file name and press the “Save” button.

   Your file will be stored on network in a space only accessible by you via your log in ID.

3 To save future amendments to your work, just click on the “File” menu and select “Save”.

OPENING A FILE

Carry out the following instructions to open a file you have saved on a previous occasion.

- Open the Control Lab programme and proceed to the programming menu.
- Click on the “File” menu and select “Open”.
- Double click on the name of your file to open it.

STARTING A NEW FILE

1. Click in the “File” menu and select “New”.
2. A start screen will appear, with the green traffic light on the left and the red traffic light on the right. Drag the various icons onto this screen and connect them between the traffic lights to write your programme.

SHUTTING DOWN THE COMPUTER

1. Select “Exit” from the “File” menu to exit the programme.
2. Click on the “Quit’ button until the programme screens disappear.
3. Go through the usual procedures for logging off.
4. Do not go through the shut down process or turn the computer off.

PACKING UP

1. Turn off the power supply to the interface.
2. Disconnect all wires attached to the interface.
2. Unplug the interface power supply and place both the interface and its power supply in the cupboard. **Do not** unplug the wire leading to the computer. There is a space for the wire to fit through at the corner of each door. Please take care to ensure that it does not get pinched when the door is closed.
3. Place your lego model in the grey tray ready for use next time.
4. Put all pieces of lego back in your box. Please check the bench and floor carefully to ensure that all bits are found.

TROUBLESHOOTING

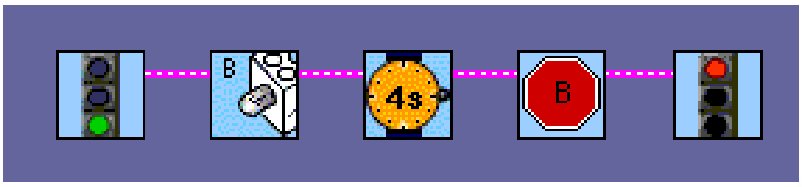
1. If you have made an error in your programming, the programme will not run. You can tell that there is an error because the “run” button will be broken.
2. To find out the nature of your programming error, click on the “window" and then on “Show error list”.

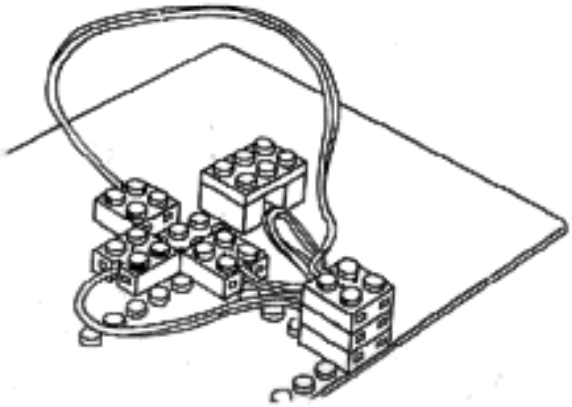
TIDYING UP YOUR PROGRAMMES

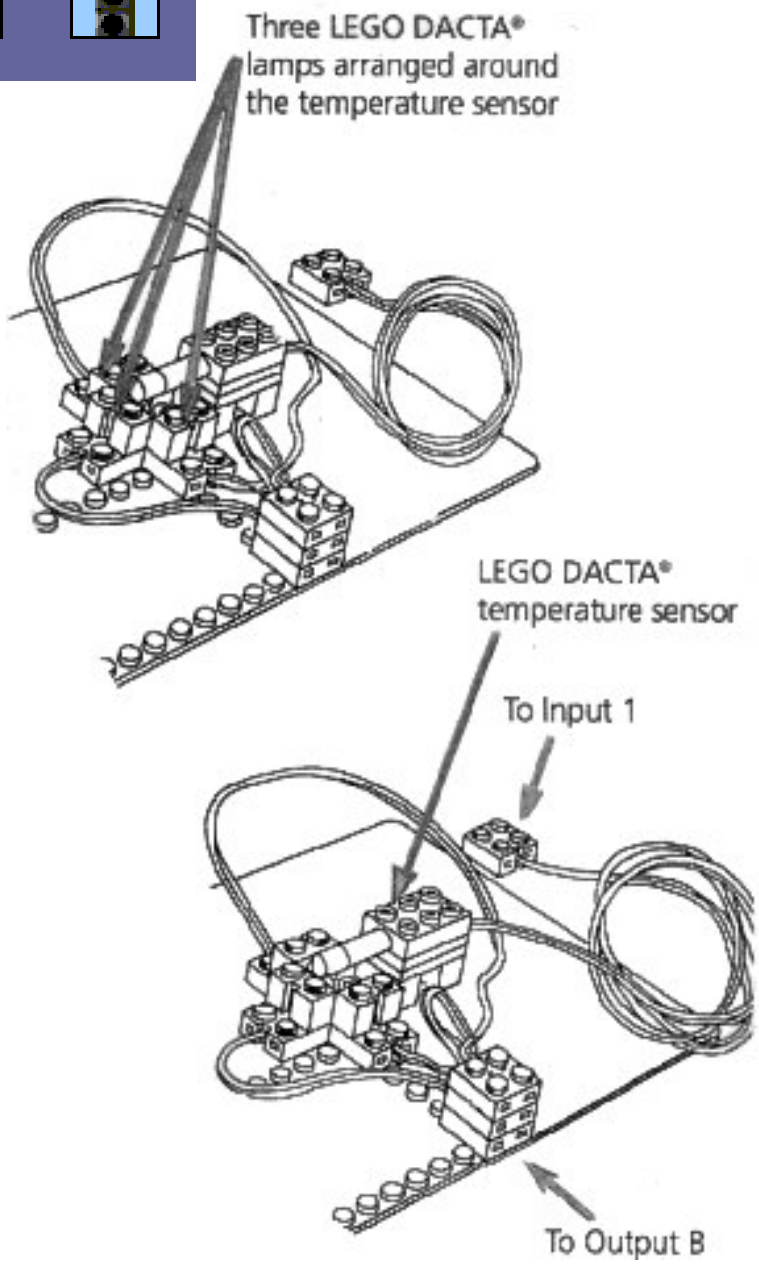
1. To space your icons evenly and have them all lined up, use the align objects and distribute objects menus (next to the “application font” button).
2. The diagrams show the various types of alignment and spacing available.

INTRODUCING LIGHTS

1. Connect up three lamps, arranged around a temperature sensor as shown on the diagram at the right.
2. Connect the three lamps to output B on the interface.
3. Go to the file menu and select “New”.
4. Construct and run the following programme.







5. Now add the necessary commands to this programme to do the following:

(a) Make the lights flash on for 4 sec and stay off for 2 sec continuously.

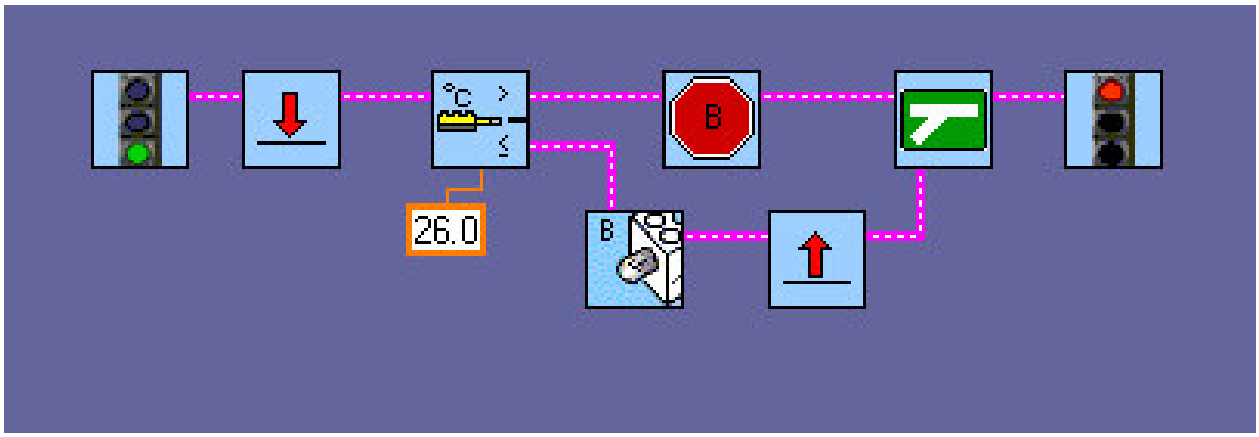
(b) Make the lights go only dimly for 5 seconds and then go off and stay off.

(c) Make the lights flash alternately bright and dim for 5 sec at a time.

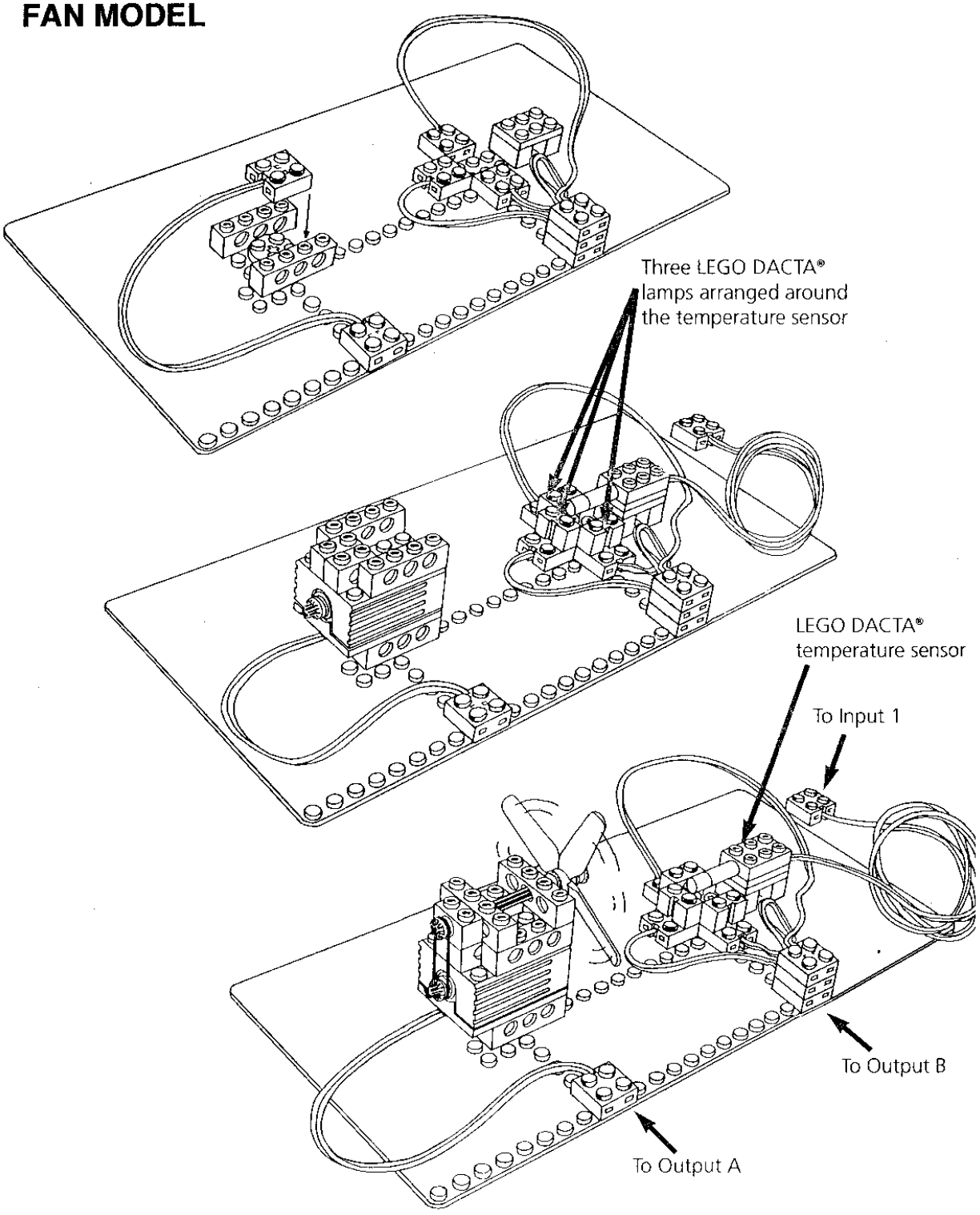
(d) Make the lights flash on (2 sec) and off (1 sec), five times.

INTRODUCING THE TEMPERATURE SENSOR

1. Combine the fan model, the lights and the temperature sensor set up as shown on the diagram on the next page.
2. Attach the temperature sensor to yellow input 1 on the control lab interface.
   The lamp should remain connected to output B.
   Connect the push sensor to input 2.
3. The teacher will tell you the temperature in the room.
   Write and run the following programme to make the lamps turn on, and stay on until the temperature is about 2 degrees above room temperature. (Hold the end of the temperature sensor between your fingers to make it warm up, or immerse the end of it in a beaker of warm water).
   Note: You will need to type the appropriate temperature in the box below the temperature sensor – the value of 26 degrees shown there, will be too high.

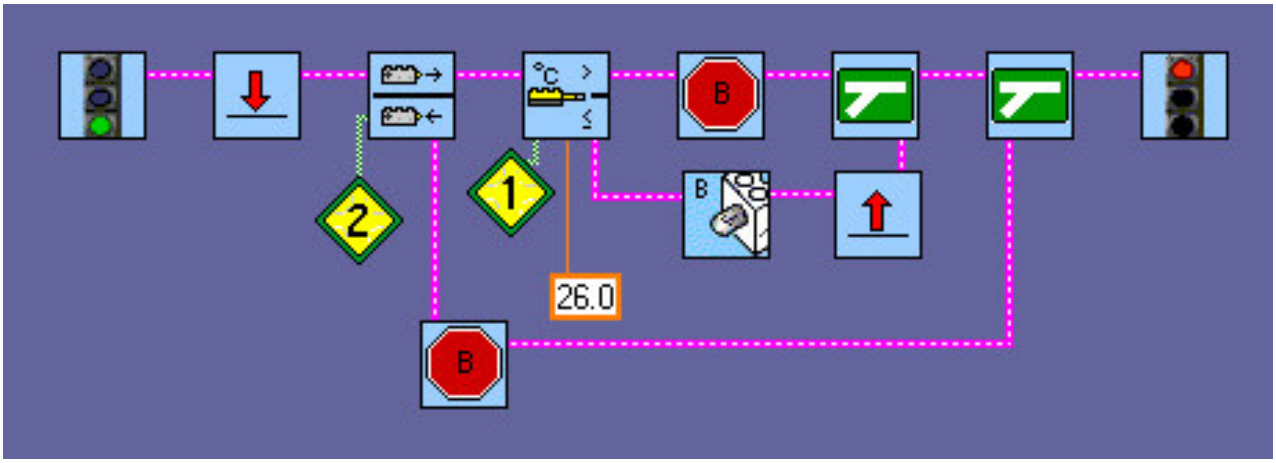


FAN MODEL

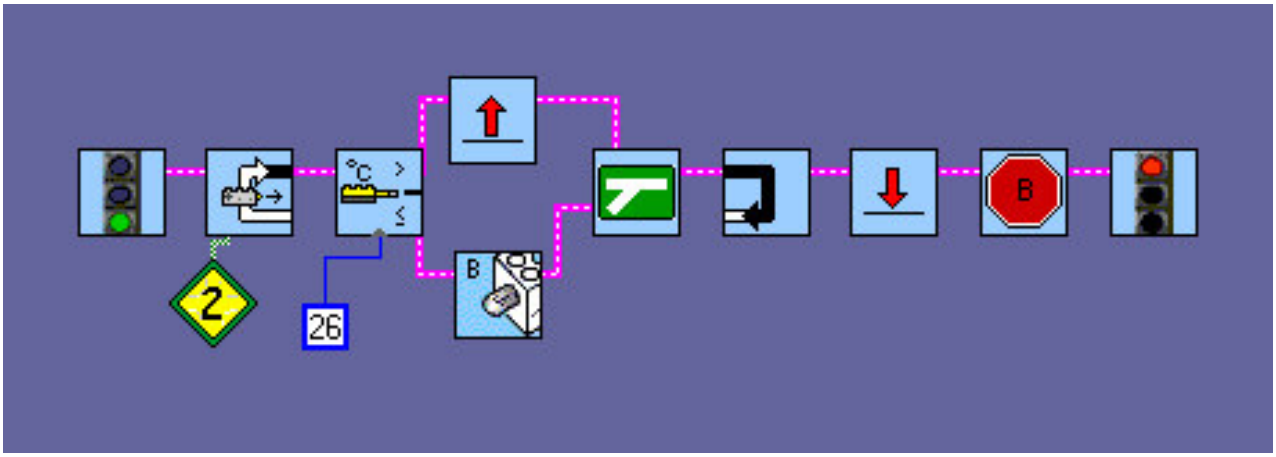

Three LEGO DACTA®
lamps arranged around
the temperature sensor
LEGO DACTA®
temperature sensor
To Input 1
To Output B
To Output A

4. You can amend the programme to allow you to use the push sensor to stop the programme at any time you wish.

Every time the programme jumps back to the start. It checks whether the switch is in or out, and follows the appropriate programme path.



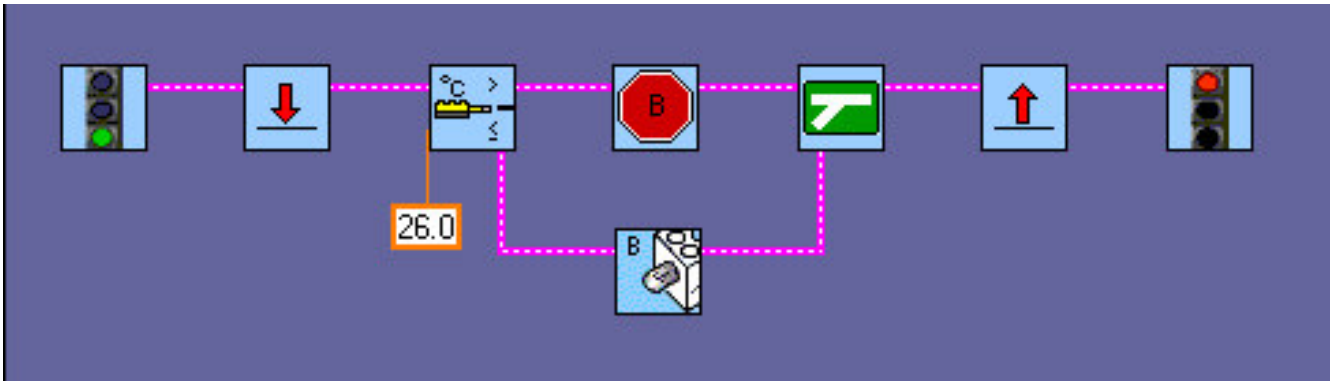
Another, neater way of writing a code for this is by using a “while” loop for the touch sensor. The loop will only operate while the touch sensor is out.



5. You are now to change the programme to get the lamps to come back on when the temperature drops below your set temperature.

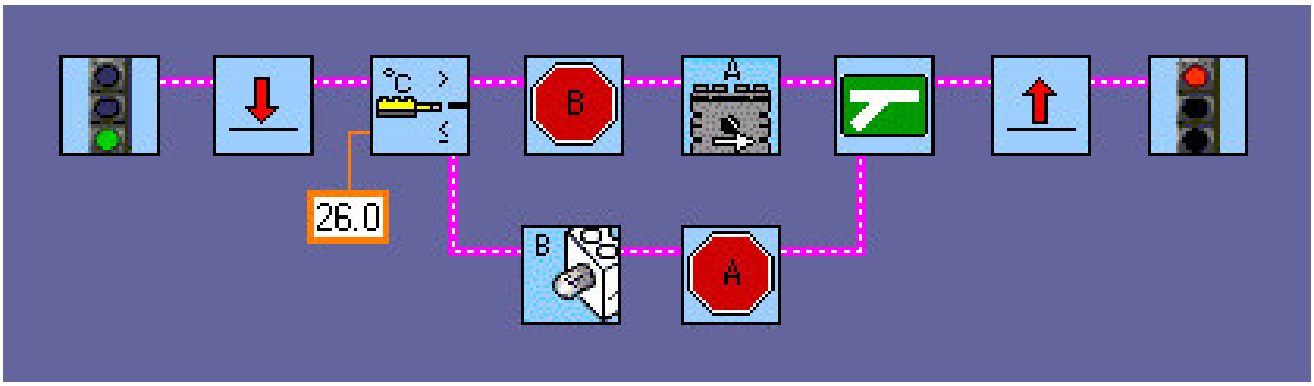
   This programme keeps the temperature constant. As soon as it drops the lights come on and the temperature rises. When the temperature rises, the lamps go off and it starts to cool again.

   Can you modify the programme to include the touch sensor to stop the programme running when the sensor is pushed ?

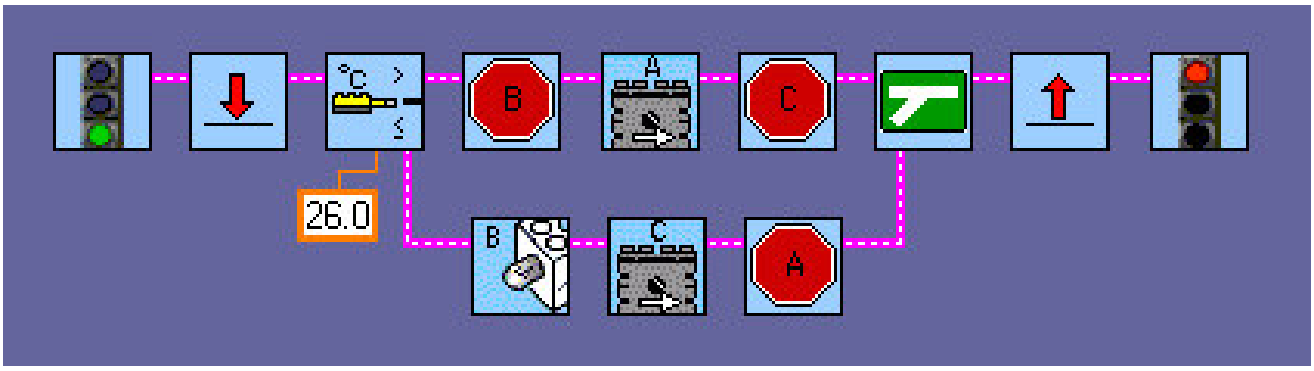


PUTTING IT ALL TOGETHER

1. Combine the fan model, lights and temperature sensor set up as shown on the diagram three pages back.
2. Check that the temperature sensor is connected to yellow input 1, the motor to output 1, and the lamps to output 2 on the control lab interface.
3. You want to construct a programme, which will use the lamps as heaters and the fan as a cooler to keep the temperature fixed.



4. You can add sound as well, to indicate when the lamps (heaters) are on.
5. Make sure the sound brick is connected to output C on the CLI.



You have now learned many of the basics of programming, and are ready to start Part B, where you will write your own programmes to control the motion of a car.

PART B THE BATTERY POWERED CAR

By now, you will have met most of the sensors and output devices and will have learnt most of the basic programming ideas. You are now ready to move on to some more challenging tasks.

You will not be given detailed instructions to follow. Instead you will be given a problem, and using your knowledge, you will have to provide a solution.

- For each of the following tasks, write the appropriate procedures, test them out, then modify them if necessary.
- Record each of the procedures on your worksheet.

Task 1. Constructing a Car.

Design and build a car which is powered by an electric motor.

The car must have 2 headlights and a horn.

Keep your design simple. Credit will be given **only** for what the car does.

Task 2. Forward and Reverse.

Write a simple programme to make the car move slowly forward for 4 seconds, stop and then move back to where it started.

Task 3 Change Speed.

Write a simple programme to make your car move forward slowly for 3 seconds and then travel faster for 2 seconds.

(You cannot use longer times because you are restricted by the length of the wires from the interface to the car).

Task 4 Night Travel.

Write a programme to switch the headlamps on when the car is running.

Task 5 Breakdown.

Write a programme to make the headlamps both flash on and off together when the car has an emergency breakdown (ie when the motor stops).

Task 6. Reversing safely.

Write a programme to make the sound sensor (horn) beep on and off when the car is reversing.

Task 7. Putting It together.

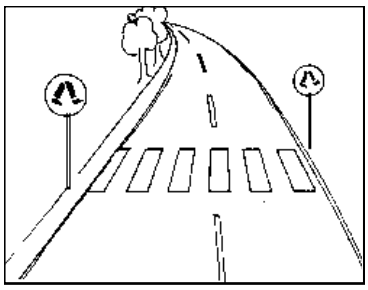
Write a programme which enables your car to carry out the following sequence of events:

- give a short toot on the horn
- move quickly forward for 2 seconds.
- stop for 5 seconds.
- turn on the lights.
- move slowly backwards for 5 seconds, with the horn sounding.

Task 8. Extension - Smart car

Design a car that 'sees' objects and stops before hitting them. It should include a light sensor located between the front lights. When you start the car and drive it towards an object. It should stop before it hits anything. The light sensor should detect light reflected from the object and when the reflected light gets bright enough, (ie when the car gets close), the car should automatically stop.

PART C THE PEDESTRIAN CROSSING (Assessment Activity)



Students leaving Walnut Primary School must cross a busy street on their way home from school. Warning lights are needed to alert drivers of the school crossing.
Your job as a traffic engineer is to design and construct a set of lights to control this crossing.

Your Brief:

- Write a programme for lights to stop the traffic and allow time for the students to cross the street safely.
- Two warning lights are required
  - one to control the cars. It should be off when the cars can go, and red when they must stop.
  - another to indicate that it is safe for the pedestrians to cross. It must be off when they are to wait and green when they can cross.
- There should be a delay between the traffic light going red and the pedestrian light going green.
- The students should be allowed a certain time to cross the road before the traffic light signals the cars to go again.

Extra For Experts:

- Programme a flashing orange light to warn cars that a red light is coming soon.
- Programme a sound to indicate to visually impaired students that it is safe to cross.
- Programme a button for pedestrians to press to activate the lights when they want to cross.

Planning Sheet:

Collect a planning sheet from the teacher. You have to complete some planning tasks for HW before starting this project. Full details are on this sheet.

PART D OPTIONAL EXTENSIONS

1. TANK

- Design and build a tank which has a flashing light and a siren.
- Programme it so that sounds its siren for 5 seconds before moving slowly forward, with its light flashing, for 10 seconds.
- Then it is to sound its siren again for 5 seconds before reversing slowly, light flashing, to its starting point.
- It is to repeat this over and over without stopping.

2. ENVIRONMENTAL ROOM

Modern homes will be controlled by computers which regulate the temperature in each room according to the tastes of the occupant and will automatically turn on the lights when the light level gets too low.
Bill Gates (who owns Microsoft) has a house that detects where everyone is and tailors each room to the particular needs of the occupant.

Your task:

1. Build a room with a light sensor and a light inside.
2. Outside the room you must have a temperature sensor and a fan to cool it down, plus two lights positioned to heat it up.
3. Program the room so that when you shade the top, the light comes on, and when you make it brighter, the inside light turns off.
4. Program the temperature sensor so that the lights to heat it up, come on at a specified temperature and then turn off at some specified higher temperature. At this point, the fan should come on to cool the sensor down until the specified lower temperature is reached.

3. GREENHOUSE

Construct and programme the Greenhouse project. Refer to the Lego instruction sheets.

4. YOUR CHOICE

If you have a good idea for a project, write up a design brief and show the teacher. When you have permission, you can proceed with the project.

PEDESTRIAN CROSSING PLANNING SHEET

Name: _____

- You are to design a programme for operating lights at a pedestrian crossing.
- Before you can start to write the programme, you must do some research so that your programme will allow pedestrians to cross safely.
- Choose a convenient road to study - It may be the road outside your home, where you wish to site the pedestrian crossing.
- Find the answers to the following questions:

(1) Approximately how fast is the traffic travelling ? _____

(2) How long would it take a car to stop at this speed ? _____

(3) What delay would you need between the red light (for the car) coming on, and the green light (for the pedestrian), to give the cars time to stop before the pedestrian starts to cross the road ? Allow a margin for safety. _____

(4) How wide is the road ? _____

(5) How long would it take a slow person (elderly) to walk this distance ? _____

(6) How long should the green light for the pedestrians stay on ? _____

Reason _____

(7) When should the red light for the car go off ? _____

Reason _____

(8) For added safety for visually impaired people, you want a buzzer to sound when it is safe for a person to start to cross. For how long should the buzzer sound? _____

Reason _____

(9) Another safety feature is to have a flashing orange light warn the cars that the red light is about to come on. How long should the orange light flash for ? _____

Reason _____

When you have completed this survey, hand it to the teacher for marking. This should be done at least 2 days before you want to start the project, to allow time for marking and discussion.

Now you can write your procedure. For marking, you will need to print off a copy for the teacher.

PEDESTRIAN CROSSING - A POSSIBLE SOLUTION

- You are to design a programme for operating lights at a pedestrian crossing.
- Before you can start to write the programme, you must do some research so that your programme will allow pedestrians to cross safely.
- Choose a convenient road to study - It may be the road outside your home, where you wish to site the pedestrian crossing.
- Find the answers to the following questions:

(1) Approximately how fast is the traffic travelling ? _____**50 kph**_____

(2) How long would it take a car to stop at this speed ? _____**6 s**_____

(3) What delay would you need between the red light (for the car) coming on, and the green light (for the pedestrian), to give the cars time to stop before the pedestrian starts to cross the road ? Allow a margin for safety._____**8 s**_____

(4) How wide is the road ? _____**15 m**_____

(5) How long would it take a slow person (elderly) to walk this distance ? ____ **20 s**_____

(6) How long should the green light for the pedestrians stay on ? _____**10 s**_____

Reason **To stop more people coming on to the crossing without time to cross.**

(7) When should the red light for the car go off ? _____**32 s**_____

Reason **Allows a 2 sec margin for error after the last person has got over the road.**

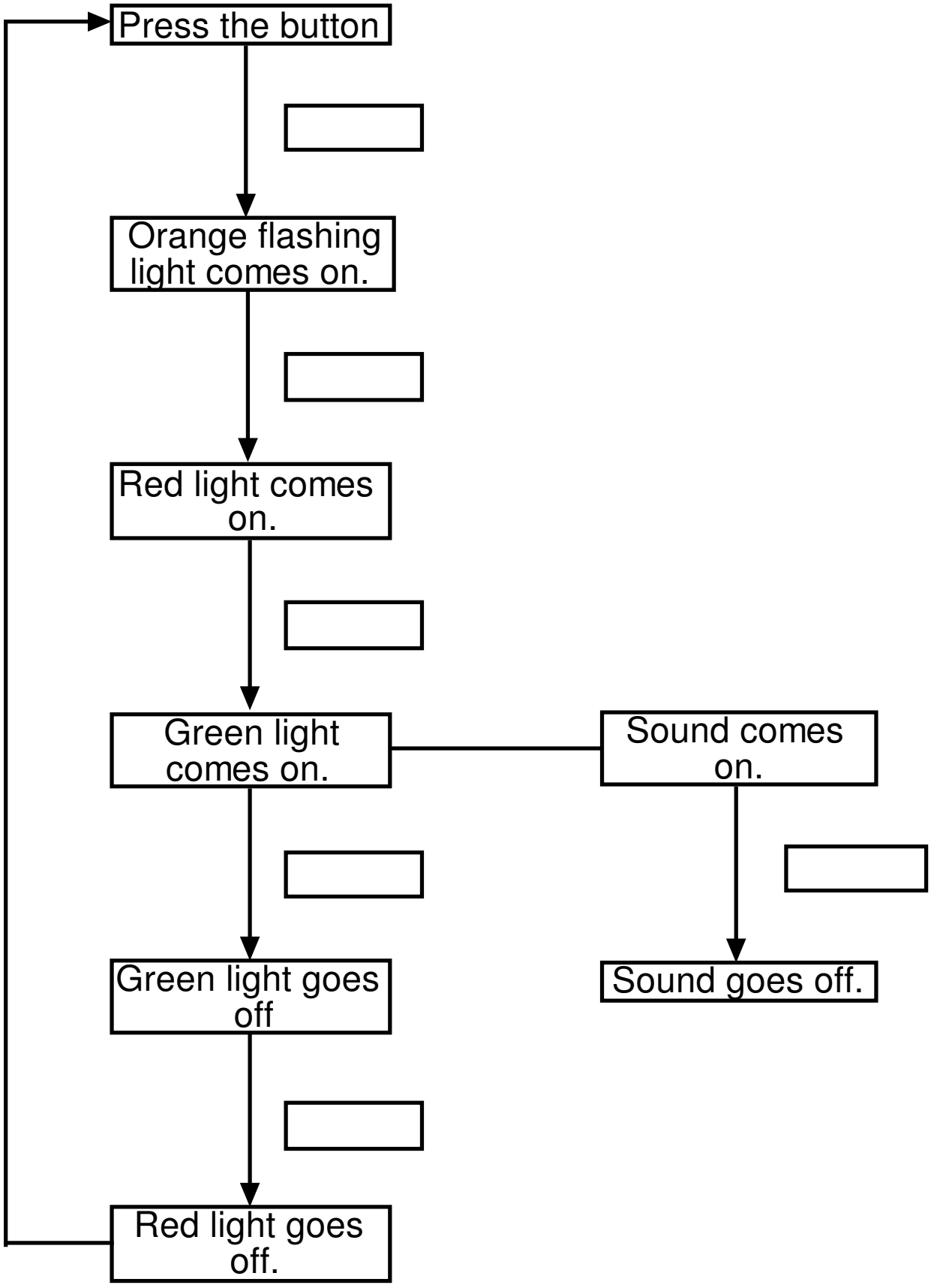
(8) For added safety for visually impaired people, you want a buzzer to sound when it is safe for a person to start to cross. For how long should the buzzer sound ? _____**10 s**_____

Reason **Only while the light is green ie safe to enter the crossing.**

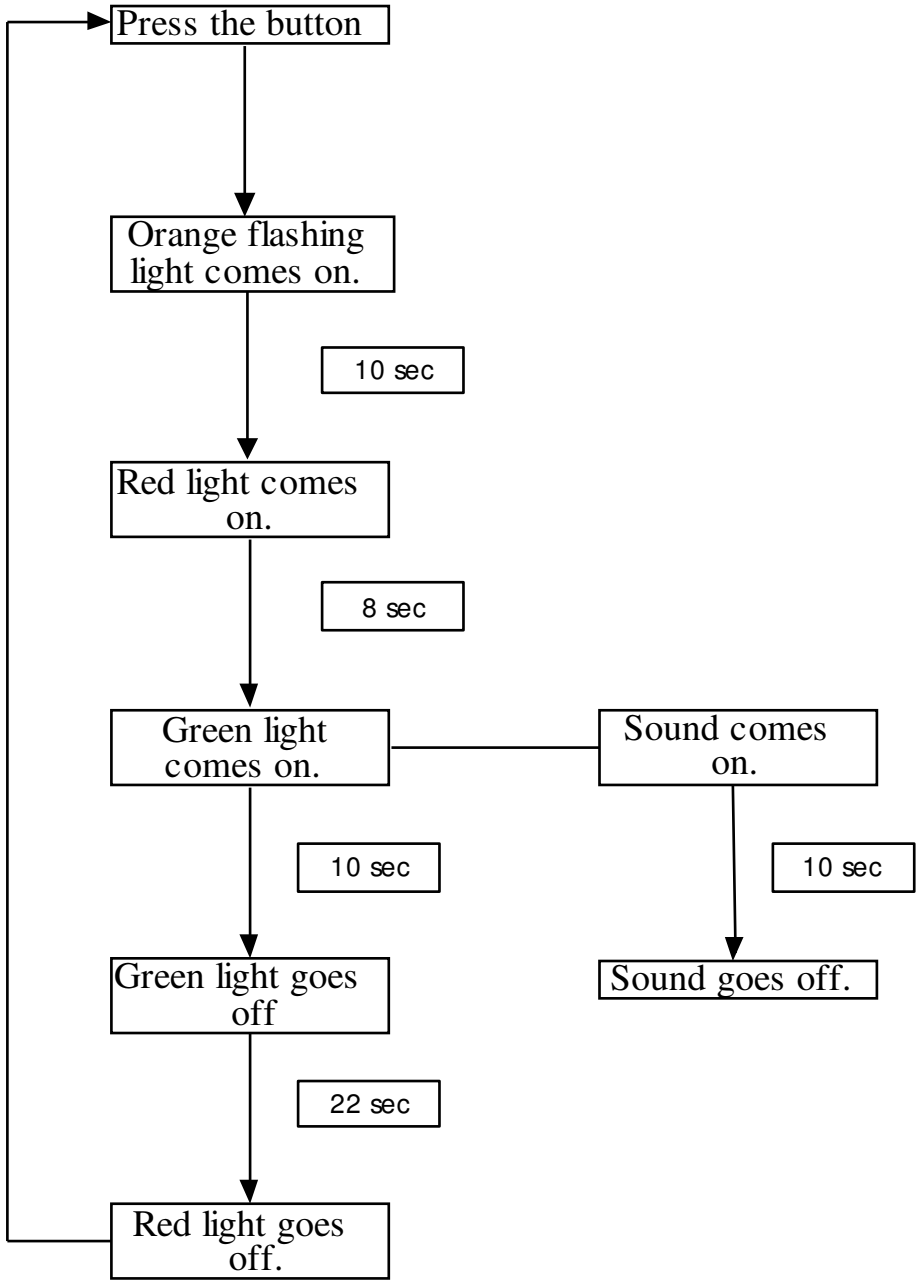
(9) Another safety feature is to have a flashing orange light warn the cars that the red light is about to come on. How long should the orange light flash for ? _____**10 s**_____

Reason **There is plenty of time for a driver to see the light and to stop.**

This flow chart may help you in your planning and in the writing of your programme code.



The flow chart with the times from the suggested solution added.



PEDESTRIAN CROSSING SOLUTION

NOTE: A = orange flashing light (for cars).

B = red light (for cars).

C = green light (for pedestrians).

D = sound brick (for pedestrians).

Name:

EVALUATION

How well does the performance of your model meet the specifications required ?
Complete the following table:

Objective	Performance Expected	Performance Achieved
Flashing orange light operates		
Flashing time correct		
Red light comes on		
Delay time before green on		
Time light remains green		
Sound comes on when green		
Time after sound stops till red		
Activation using touch sensor		

How do you rate the performance of your traffic lights ?

_____

_____

_____

_____

_____

What are two advantages of using modern technology (eg traffic lights at pedestrian crossings) for improving road safety.

_____

_____

_____

_____

_____

ASSESSMENT and COVERAGE OF OBJECTIVES

Technological Knowledge and Understanding (coverage):

• Recognises the use of technology in road safety. (AO A1)	Yes / No
• Robotics worksheet completed with only a few errors. (AO A2)	Yes / No

Technological Capability (assessment):

• Planning sheet completed to a satisfactory standard. (AO B6 (a))	Yes / No
• Procedure for lights completed - specifications met. (AO B6 (b))	Yes / No
• Satisfactory evaluation of procedures. (AO B6 (d))	Yes / No

Technology in Society (coverage):

• Recognises advantages of technology for road safety. (AO C8)	Yes / No

OVERALL LEVELS OF ACHIEVEMENT

- **Level 1:**

All tasks completed with a minimum of assistance.

Independent completion of the basic pedestrian crossing problem; all criteria achieved, including the sound, flashing light and activation by the button (touch sensor).

- **Level 2:**

All tasks completed, but with some guidance.

Independent completion of the basic pedestrian crossing problem; flashing light and sound also operate (some guidance allowed here).

- **Level 3:**

All tasks completed, but with teacher support.

A workable solution to the basic pedestrian crossing problem (red and green lights only) is achieved with some guidance.

- **Level 4:**

Attempts all tasks, but workable solution not achieved.

- **Level 5:**

Tasks incomplete, poor effort made, workable solution not achieved.

TECHNOLOGY CRITERIA RECORDING SHEET – ROBOTICS LEGO ASSESSMENT TASK

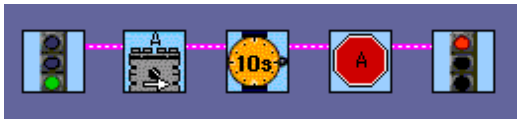
Technological Knowledge and Understanding (completion).																													
• Recognises use of technology in road safety.																													
• Robotics worksheet completed with only a few errors.																													
Technological Capability. (Assessment focus)																													
• Planning sheet completed to an acceptable standard.																													
• Procedure for lights completed and specifications met.																													
• Satisfactory evaluation of procedures.																													
Technology in Society. (coverage)																													
• Recognises advantages of technology for road safety.																													

TEACHER INFORMATION

Solutions to Tasks in Part A – Learning The Basics

Starting a Project – Task 1 – Fan Model

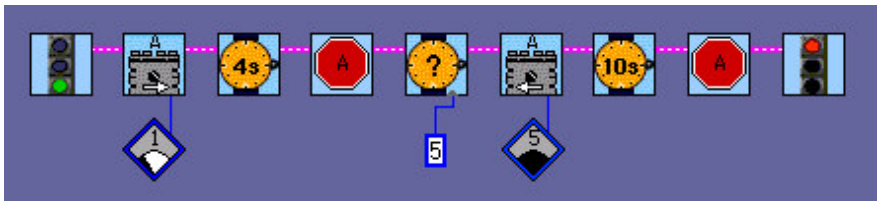
Making the motor run for 10s.



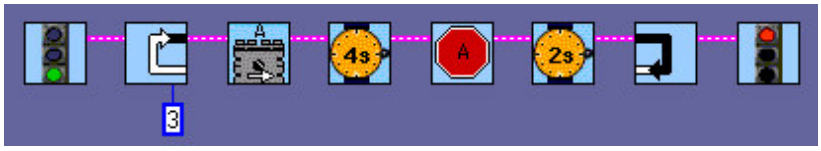
Making the motor run in reverse.

Modifying the Motion of the Fan.

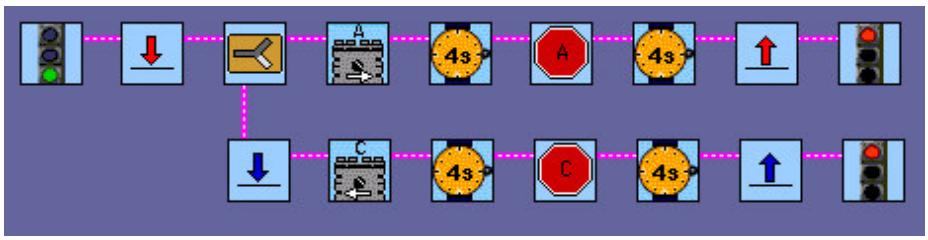
Slower for 4s, stop for 5s, faster in reverse for 10s, stop.



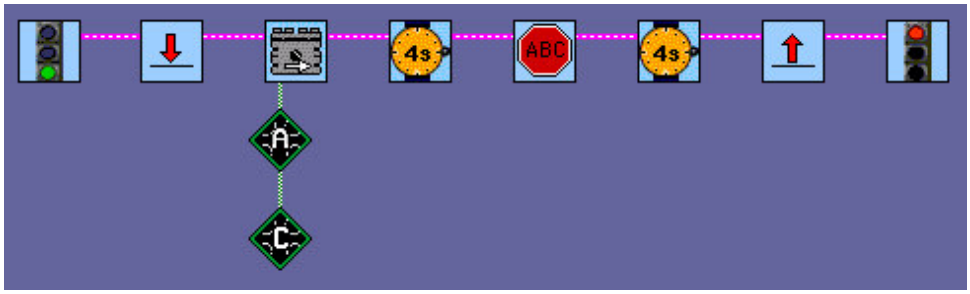
Amending the programme to do 3 loops.



Other alternatives to get sound when the fan is running – on a continuous basis rather than just once like the answer given.



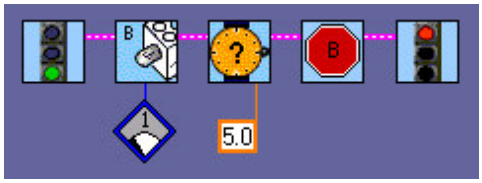
A neater option.



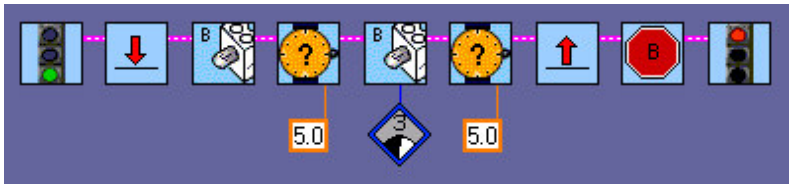
Introducing Lights.

On for 4s and off for 2s continuously.

Lights dim for 5s then off.



Lights flash alternatively bright and dim.



Lights flash on and off 5 times.