T-board Teaching Pack

HELP Sheet A

PIC LOGICATOR

Moving commands
To move a command, click on it and drag it to its new position.

To move a block of commands, first select the block. Then select Edit>Cut. Click on the flowsheet where you want to reposition the block, and select Edit>Paste.

Deleting commands
To delete a command, click on it to highlight it; then press the Delete key.

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

Copying commands
To copy a command, click on it to highlight it. Then select Edit>Copy.

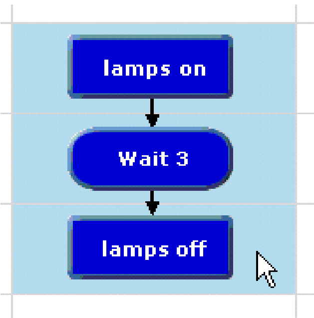
To copy a block of commands, first select the block. Then select Edit>Copy.

Click on the flowsheet where you want to place the copied command or block of commands. Then select Edit>Paste.

You can use the same method to copy blocks of commands from one Logicator program into another.

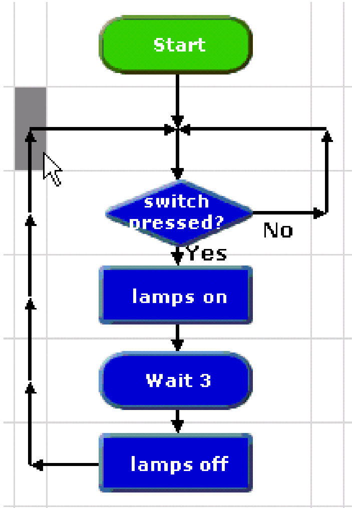
Remember that copied commands will still have their original settings.

Selecting a block
To select a block of commands, click on the first command in the block; then hold down the Shift key as you click on the last command in the block.

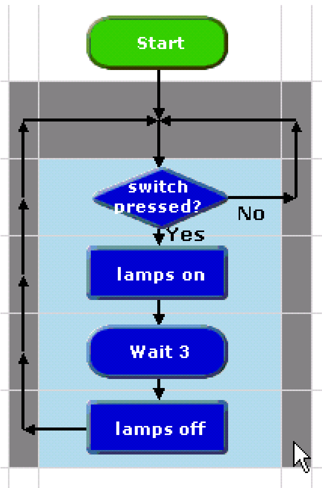


To select a block of commands with loops:

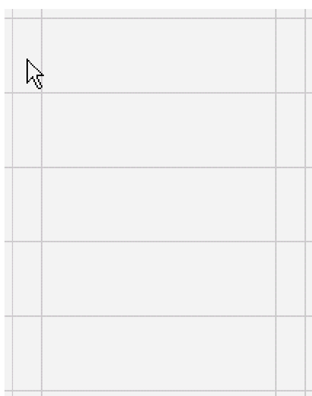
Click at the top left hand corner of the block........



.......Then hold down the Shift key as you click on the bottom right hand corner of the block.



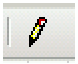
When you paste the block into its new position, select a starting point that is equivalent to the top left hand corner of the selected block.



HELP Sheet B

Drawing lines

Click the Drawing Tool icon on the menu bar to select it.

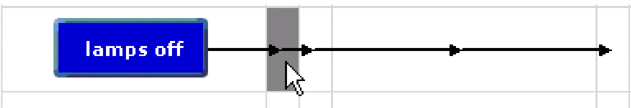


With the right-hand mouse button, click where you want the line to begin; then click where you want the line to end.

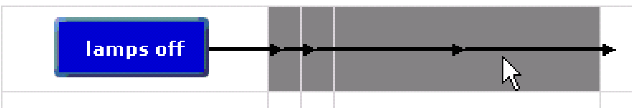
Click again on the Drawing Tool icon to deselect it.

Deleting lines

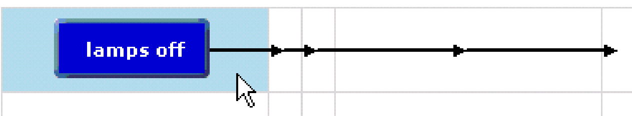
Click at the start of the line and press the Delete key:



To delete a series of lines, hold down the Shift key as you select the start of each line. Then press the Delete key:

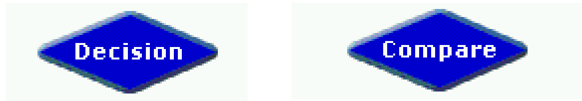


To delete the line coming out of a command, without also deleting the command, first click on the command to highlight it; then hold down the Ctrl key as you press the Delete key:

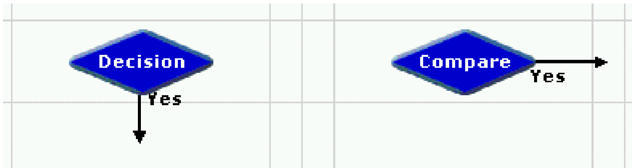


When you draw a new line from a command, the existing route from the command will automatically be deleted.

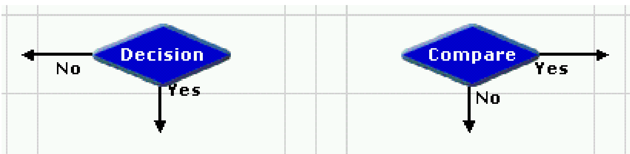
Drawing lines from Decision and Compare commands



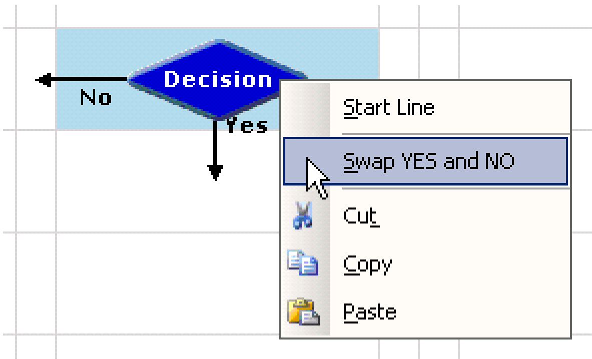
The first line that you draw from a Decision or Compare command is automatically the "Yes" route.



The second line is automatically the "No" route.



If you want to swap over the lines, click on the command with the right-hand mouse button and select "Swap YES and NO".

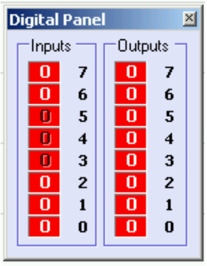


HELP Sheet C

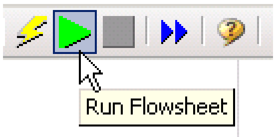
Test running a program

1. Select View> Digital Panel.

As the program runs, the Digital Panel shows the changing state of outputs and inputs as they will be when the program is downloaded to a PIC.



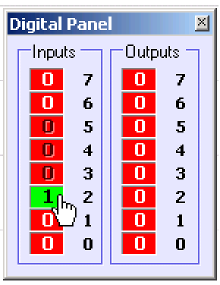
2. Click the green arrow icon on the Tool Bar to run the program.



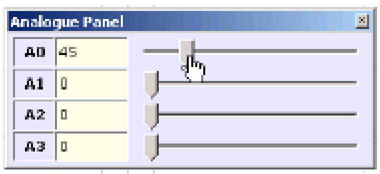
The program will automatically stop running when it reaches a Stop command. You can also stop the program running by clicking the red button on the Tool Bar.

Simulating inputs

To simulate the input from a **digital sensor** while the program is running, click on that input in the Digital Panel.



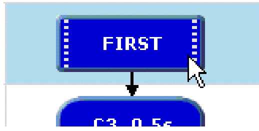
To simulate the changing input from an **analogue sensor** while the program is running, select View>Analogue Panel. Use the slider for the sensor to vary the reading.



Test running a procedure

Click on the Procedure command to highlight it.

Then click the green arrow icon to run it.

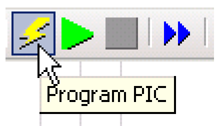


Downloading a program to the T-board

1. Connect the download cable between the USB port of the computer and the download socket on the T-board.

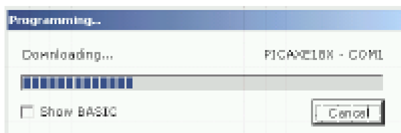
2. Use only the recommended power supply. Connect the power supply cable to the power socket on the T-board. Then plug the power supply into a mains socket.

3. Click the Program PIC icon on the Tool Bar.



4. It takes only a few seconds to download the program to the PIC on the T-board.

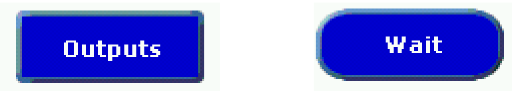
The Programming.. box shows the progress of downloading.



The program will run as soon as downloading is complete. To restart the program at the Start command, press the RESET switch on the T-board.

Switching Outputs (1)

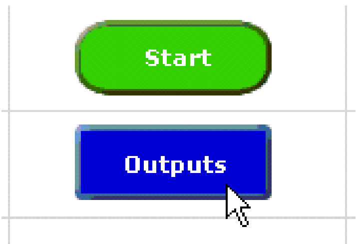
Commands and Techniques



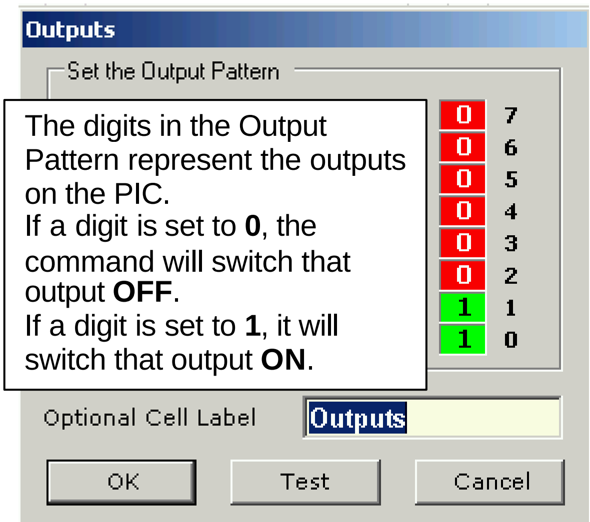
- Switching outputs (LEDs on the T-board) on and off in timed sequences.

Build a sequence

1. Drag an **Outputs** command from the Commands list and place it under Start:



Double click on the Outputs command to open it.



The digits in the Output Pattern represent the outputs on the PIC.
If a digit is set to **0**, the command will switch that output **OFF**.
If a digit is set to **1**, it will switch that output **ON**.

Click digits 0 and 1. When the program is downloaded to the PIC, this command will switch on LEDs 0 and 1 on the T-board.

Type a label to remind you what the command does: `leds 0+1 on`
Click OK.

2. Place a **Wait** command under the Outputs command. Double click on it.
Type 1.
Click OK.

3. Continue to build up the program as shown in Fig 1. Place a Stop command at the end.

You could copy and paste the Wait 1 command to save time.

Help Sheet A shows how to copy and paste commands.

4. Draw a line down from the Start command to the Stop command.

Help Sheet B shows how to draw lines.

5. Test run the finished program.

Then download it to the T-board.

Help Sheet C shows how to test run and download a program.

The program will run just once. Press the Reset button on the T-board to run it again.

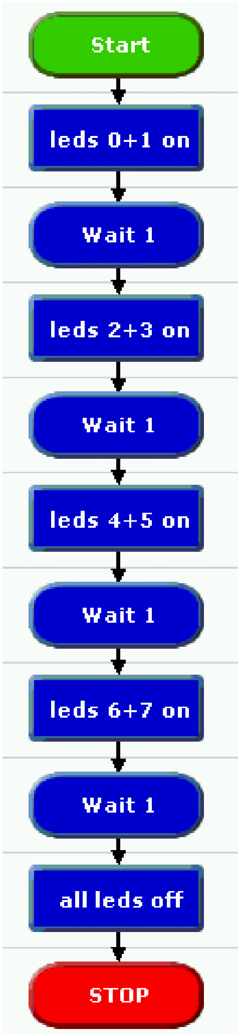


Fig 1. This program will switch on the LEDs on the T-board in pairs, at one second intervals; then switch them all off.

Switching Outputs (2)

Commands and Techniques

- Repeating a sequence.

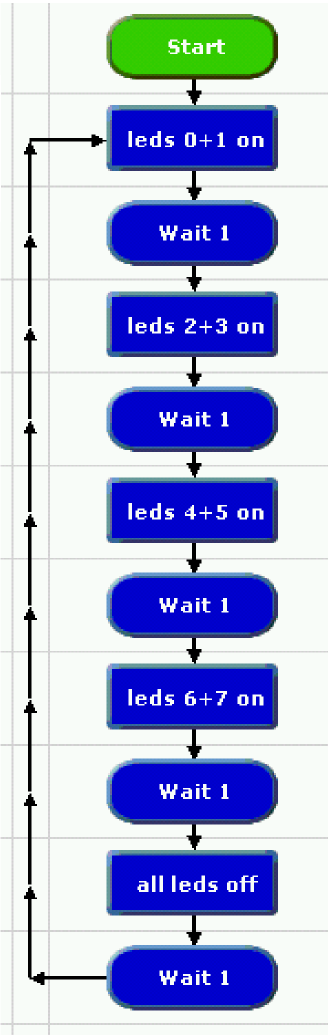
Repeat a sequence

1. Start with the program shown in Fig 1. Delete the Stop command.

Help Sheet A shows how to delete commands.

2. Add another Wait command at the end of the program. Set it to wait one second. Then draw a line from this command back to the start of the program as shown in Fig 2.

Fig 2. This program repeats the sequence automatically.



3. Test run the program. Then download it to the T-board.

Now the sequence will repeat itself continuously until you disconnect power from the board.

Extensions

Build more sequences

Use your program as a starting point to build one or more of the different sequences below.

If you want to save any of these sequences, remember to save the program as a different file name.

1. Change the Wait times to make a faster, slower, or irregular sequence.

2. Change the Output Pattern settings so that the first pair of LEDs stay on when the second pair are switched on; and so on so that all the LEDs are on by the end of the sequence.

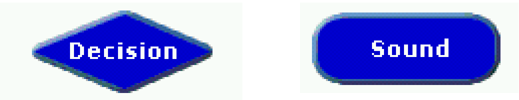
3. Change the sequence so that it begins with all the LEDs on; then switches them off one pair at a time.

Download your sequences to the T-board to try them out.

NOTE: When you power up the T-board to download the program, the program already in the PIC will start to run. It will stop as the new program downloads.

Digital Inputs (1)

Commands and Techniques

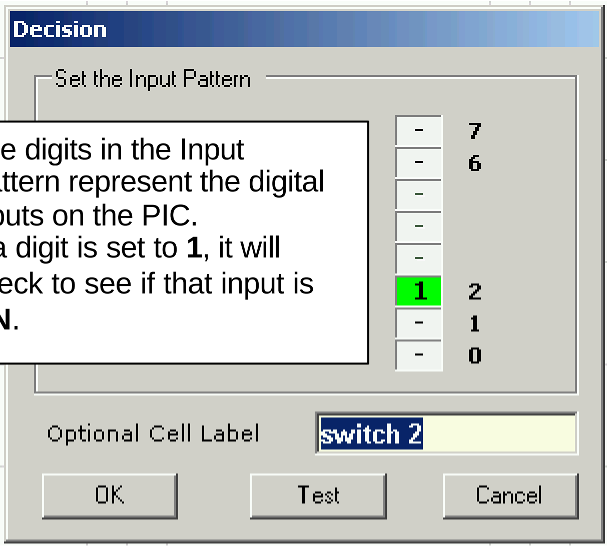


- Switching outputs on and off in response to input from digital sensors (switches on the T-board).
- Producing sounds from the piezo sounder on the T-board.

1. Select **File>New**

2. Place a **Decision** command two spaces under Start.

Double click on the Decision command.



The digits in the Input Pattern represent the digital inputs on the PIC.
If a digit is set to **1**, it will check to see if that input is **ON**.

Click digit 2. When the program is downloaded to the PIC, this command will check to see if switch 2 on the T-board is on.

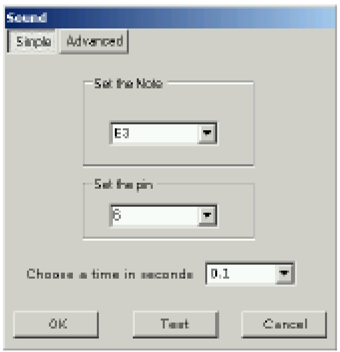
Type a label to remind you what the command does: `switch 2`
Click OK.

3. Place a **Sound** command under the Decision command.

Double click on the Sound command.

Select a note and time for the sound.

Set the pin to 6. The piezo sounder on the T-board is connected to pin 6 of the PIC.



4. Draw the lines on the program as shown in Fig 3.

Help Sheet B shows how to draw lines from a Decision command.

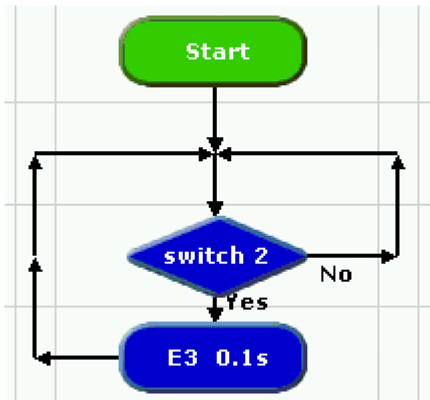


Fig 3. This program makes a sound whenever you press switch 2.

5. Download the program to the T-board. When you press switch 2, the note should sound. Note that LED 6 is also connected to pin 6 of the PIC, so it will switch on whenever the piezo sounder is on.

Digital Inputs (2)

An alarm system

Fig 4 shows a program for a simple burglar alarm system. Switch 2 on the T-board is used as the sensor that responds if a burglar treads on it.

Two Sound commands are used as the alarm. Each one is set to sound for 0.2 seconds.

The alarm continues to sound until you press the reset switch which is switch 7 on the T-board. When you press this switch, the program goes back to checking switch 2, the burglar sensor.

1. Start with your Fig 3 program; and edit it to be like Fig 4.

2. Test run the program; then download it to the T-board.

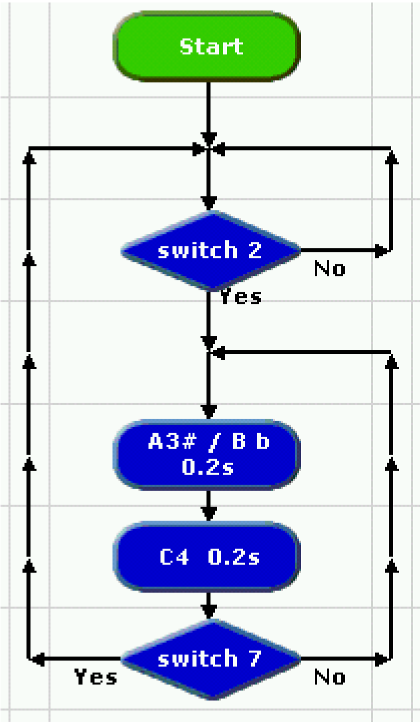


Fig 4. Alarm system with one sensor and a reset switch.

Extensions

1. Develop your system further so that it includes four different sensors that could be in different parts of the house. Fig 5 shows how to do it.

2. Develop your alarm system to use LEDs to indicate which one of the sensors has triggered the alarm.

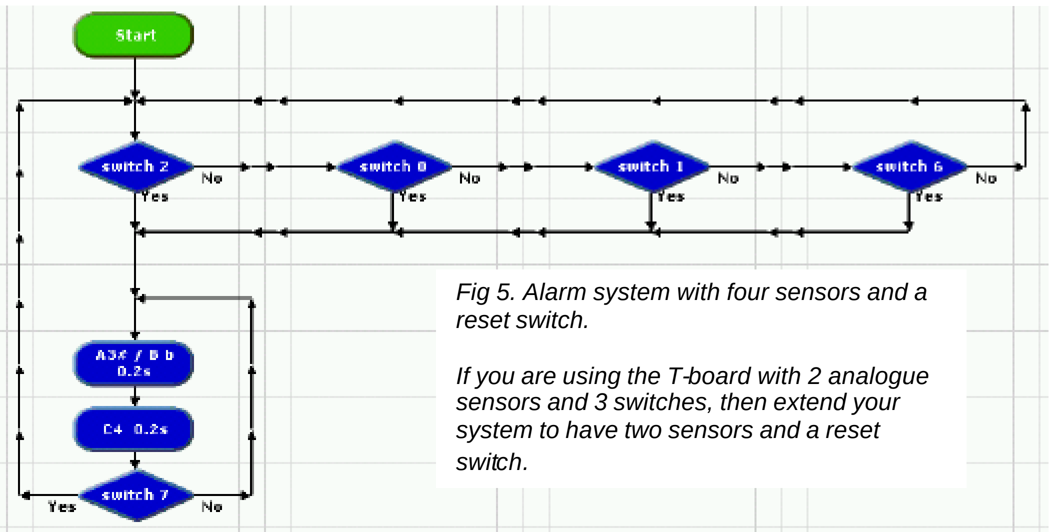
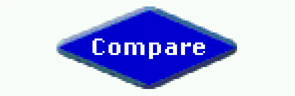


Fig 5. Alarm system with four sensors and a reset switch.

If you are using the T-board with 2 analogue sensors and 3 switches, then extend your system to have two sensors and a reset switch.

Analogue Inputs (1)

Commands and Techniques



- Switching outputs on and off in response to input from an analogue sensor (light sensor or variable resistor on the T-board).

Automatic Light

1. Select **File>New**

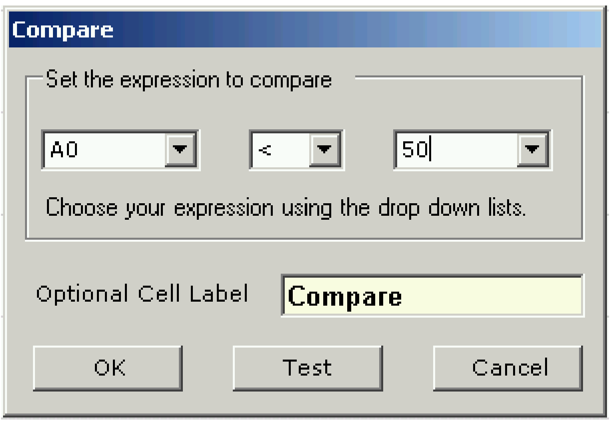
2. Place a **Compare** command two spaces under Start.

Double click on the Compare command

When the program is downloaded to the T-board, this command will check the light sensor which is analogue input 0, so select A0 from the first drop down box.

The command will check to see if the light level has dropped below 50, so select < ("less than") from the second box and type 50 into the third box.

Click OK.



3. Add two Outputs commands to the program as shown below. Set one to switch on LED 7. Set the other to switch off LED 7.

Complete the lines as shown in Fig 6.

Help Sheet B shows how to draw lines from a Compare command.

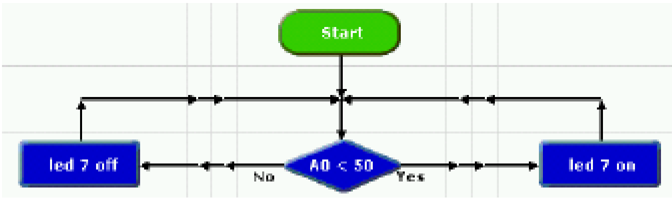
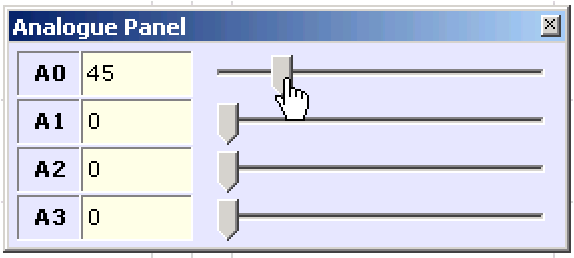


Fig 6. This program switches on LED 7 on the T-board when it gets dark, and switches it off when it is light.

Test run the program. Help Sheet C shows you how. When you run the program, move the A0 slider in the Analogue Panel to simulate changes in the light level.



Move it above 50 and check that output 7 switches off. Move it below 50 and check that output 7 switches on.

Download the program to the T-board. Test it by covering and uncovering the light sensor.

Analogue Inputs (2)

Light meter

The light sensor on the T-board sends information to the PIC about the level of light shining on it. The information is sent as a number between 0 and 255. When it is completely dark, the number is 0. In normal room lighting it will be about 125. If you shine a bright light onto the sensor, the number will be about 255.

Fig 7 shows a program for a simple light meter. When it is dark, no LEDs are lit. As the light level increases, more LEDs are switched on. As the light level falls again, the LEDs are switched off.

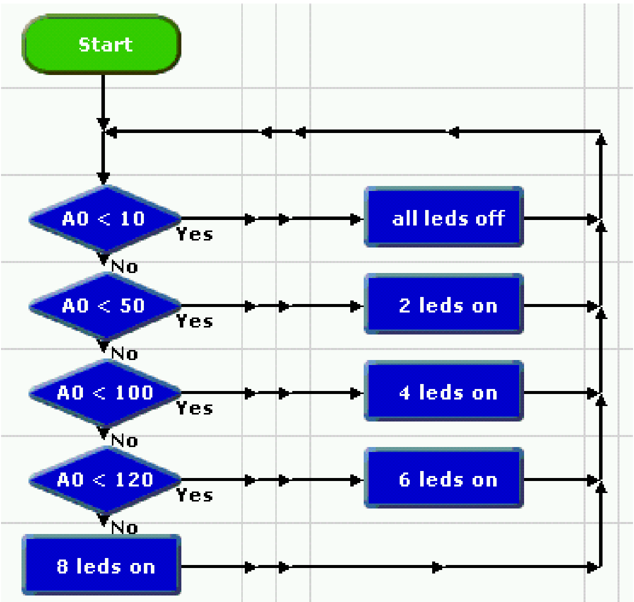
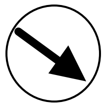
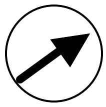


Fig 7. Simple light meter

Build the program shown in Fig 7. Test run it. Then download it to the T-board.

Variable resistor

Analogue sensor 1 on the T-board is a variable resistor. As you turn it, this sensor sends information to the PIC in the form of a number from 0 to 255.

Reading 0  *Reading 255* 

This sensor is useful for simulating other analogue sensors when you investigate project ideas.

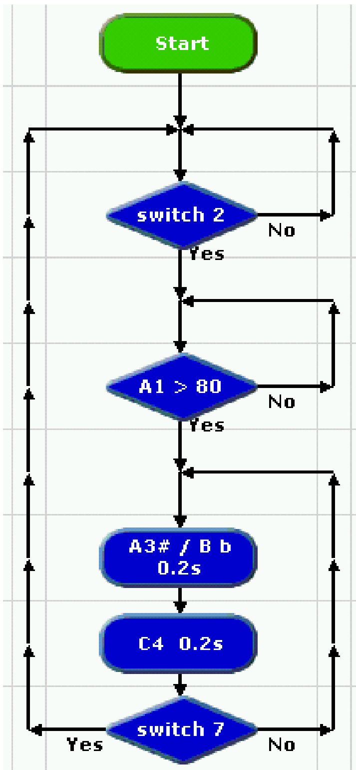


Fig 8. Temperature alarm

Fig 8 shows a program for a system that sounds an alarm if a machine overheats to a temperature over 80º. It uses the variable resistor to simulate a temperature sensor.

Switch 2 is used to set the alarm system. When the alarm sounds, switch 7 is used to switch it off.

Build the program shown in Fig 8. Test run it. Then download it to the T-board.

Remember to press switch 2 to set the alarm, before you turn the variable resistor.

Extensions

1. Build a program that uses the variable resistor to simulate a temperature sensor in a system in which three LEDs are used to indicate water temperature – green for cold (<10), orange for medium (<60), and red for hot (>60). You can base this system on Fig 7.

2. Edit Fig 8 to make a system that sounds an alarm if the light level falls below 20. Extend the system so that the alarm can be switched off by switch 7, but also switches off automatically if the light level rises above 20.

Procedures (1)

Commands and Techniques

- Developing a large or complex system as a series of procedures
.
- Playing a tune on the piezo sounder.

First line of the tune

1. Select **File>New**

2. Build the program shown in Fig 9. Take care to set the note, pin and time boxes of all the Sound commands.

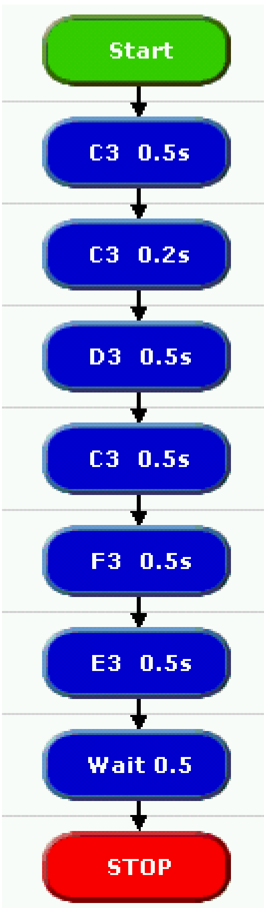


Fig 9. This program plays the first line of a well known tune.

3. Test run the program. Then download it to hear how it sounds played on the T-board piezo sounder.

The complete tune

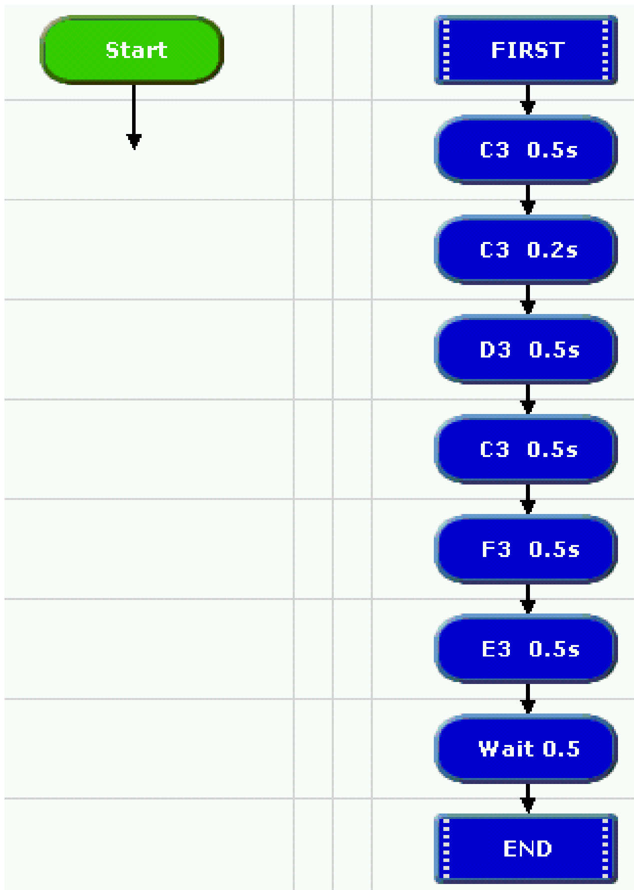
The complete tune has four lines. The best way to build the program to play it all is to write each line as a separate procedure. Follow the instructions below to develop your program to play the complete tune.

1. Place a **Procedure** command beside the Start command. Double click the Procedure command. Type: `first`
Click OK.

2. Cut all the commands under Start ....and paste them under the Procedure command.

3. Delete the Stop command. Drag an **End** (Return) command into its place.

Your program should now look like this:



Continued on Procedures (2)

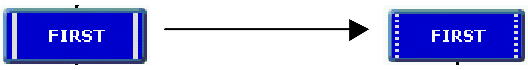
Procedures (2)

4. Build the other three procedures called SECOND, THIRD, LAST, as shown in Fig 10. You can save yourself time by copying commands (See Help Sheet A).

You can test run each procedure when you have built it. Help Sheet C shows you how to do this.

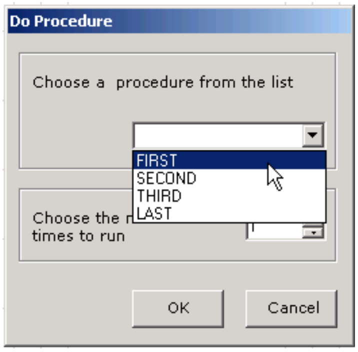
5. Finally, build the main routine under the Start command, as shown in Fig 10. It begins with a Decision command set to check switch 2. Then there are four **Do Proc.** (short for "Do Procedure") commands.

A Do Proc. command is used to tell the program to run the procedure with the same name.



To set a Do Proc. command, double click on it and select the name of the procedure from the drop down box.

Click OK.



6. Test run the program. Then download it to the T-board to hear how it sounds played on the piezo sounder. Remember to press switch 2 on the T-board to start the tune.

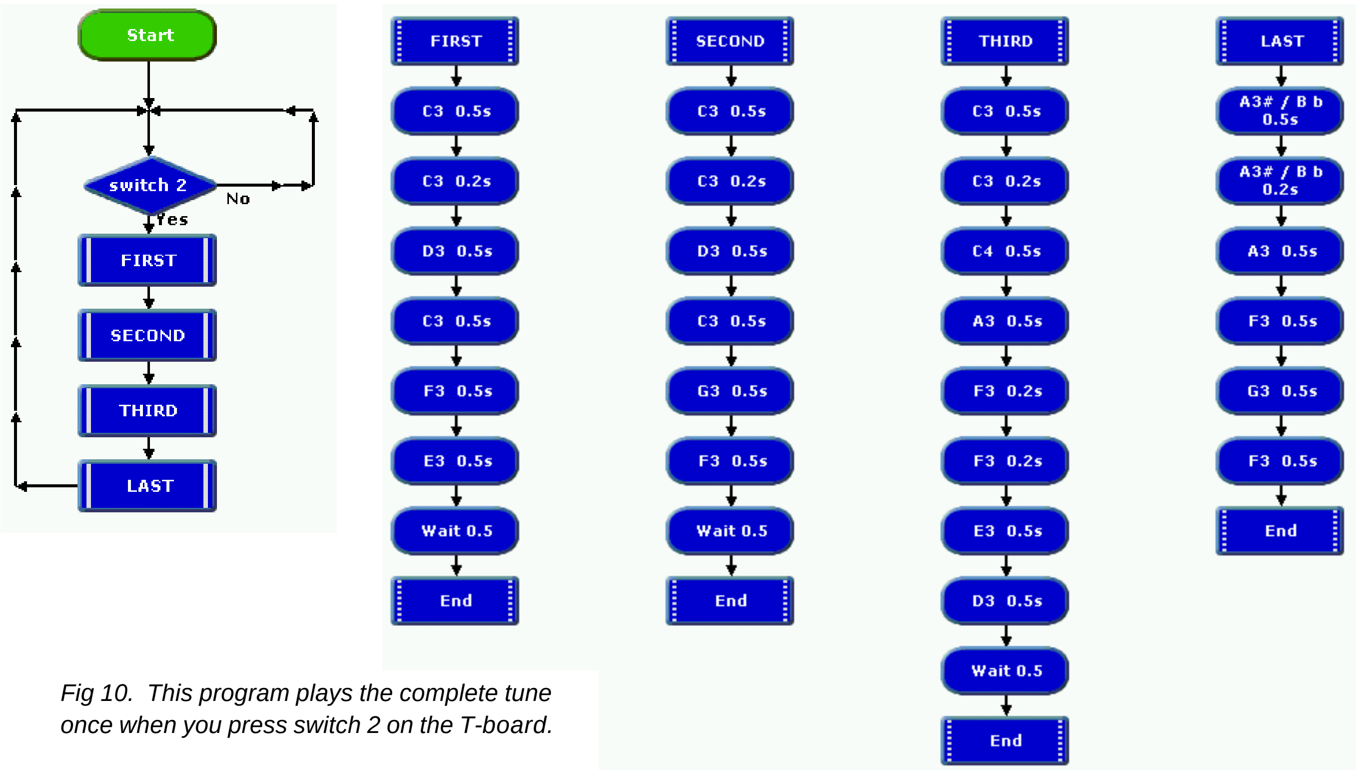


Fig 10. This program plays the complete tune once when you press switch 2 on the T-board.

Extension

Develop a system for a musical greetings card that automatically plays a tune when the card is opened.

Timing

Commands and Techniques

- Timing events.
- Repeating a sequence a number of times.

A games timer

Timers are often used in games to set a maximum time that each player can take for his or her turn.

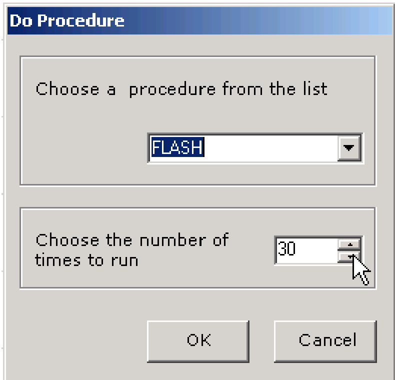
Fig 11 shows a program for a simple timer that times 30 seconds; then signals that time is up, by flashing all the LEDs on the T-board for one second while two notes are played on the piezo sounder.

The timer is started by pressing switch 2 on the T-board.

The FLASH procedure switches an LED on for half a second, and off for half a second; so the procedure takes just one second to run.

The program times 30 seconds by running the FLASH procedure 30 times.

When you open the Do Proc command to name it, select 30 from the "number of times to run" box.



Build the program shown in Fig 11. Test run it; then download it to the T-board and check that it signals 30 seconds after you press switch 2.

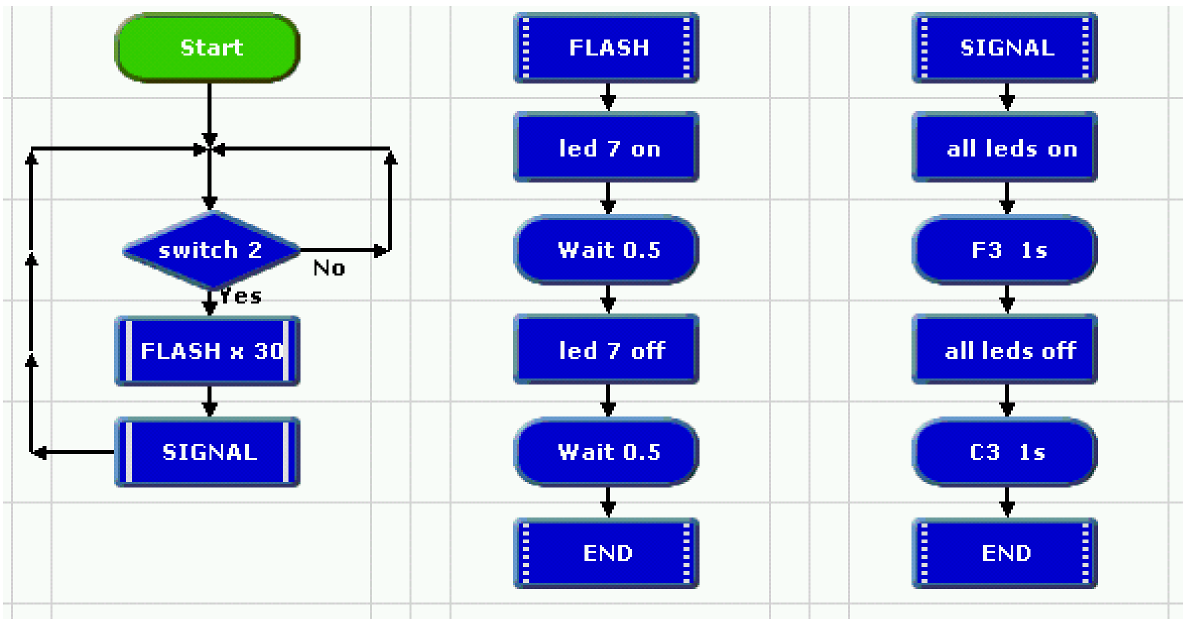


Fig 11. A simple games timer that signals when a time of 30 seconds has passed.

Extension

Edit the program so that it times for 15 seconds. Then edit it again so that it times for a minute.

Counting (1)

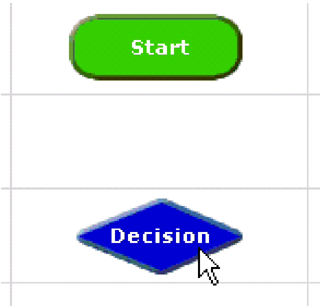
Commands and Techniques

- Counting signals from a digital sensor (a switch on the T-board).
- "Debouncing" a switch.

Counting switch presses

1. Place a **Decision** command two spaces under Start.

Double click on the Decision command.

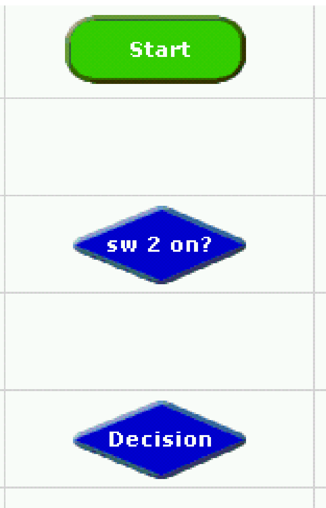


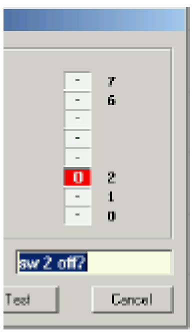
Set the command to check if switch 2 is pressed.

2. Place another Decision command two spaces under the first one.

Double click the second Decision command.

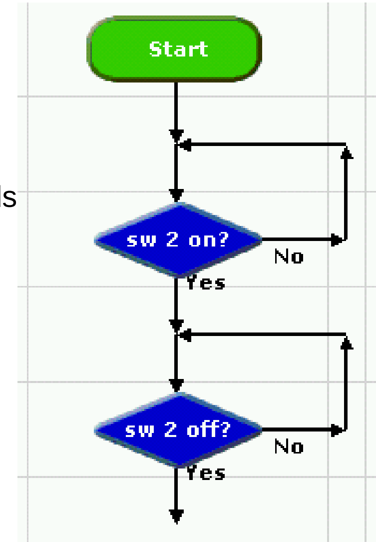
Click digit 2 to a 0, as shown below. The command is now set to check if switch 2 is **not** pressed.





3. Draw the lines from the two Decision commands as shown below.

These two commands together "debounce" the switch to make sure that only one switch press is counted at a time.



4. Add an **Inc** command to the program as shown in Fig 12. Double click on the Inc command and select **A** from the drop down box. Click OK.

Draw a line from the Inc command back to the start of the program.

When the program runs, one is added to the counter A each time flow passes through the Inc command.

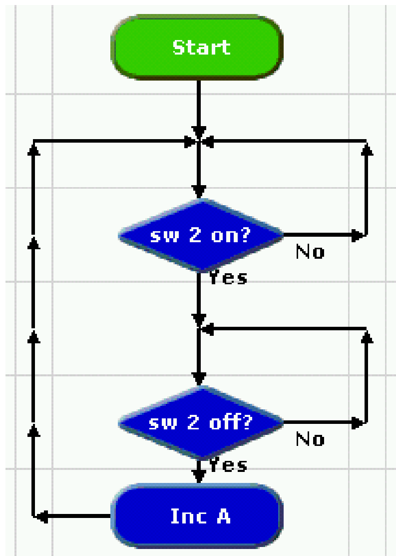
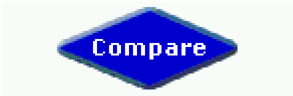


Fig 12. This program counts single switch presses.

5. Select View>Variables Panel from the Menu, and then test run the program. As you simulate the switch presses, you will see the value of A change in the Variables Panel.

Counting (2)

Commands and Techniques



- Displaying a changing count (using LEDs on the T-board).

Displaying the count

Fig 12 provided a program to count switch presses, but before you download it to the T-board you need a way of displaying the count.

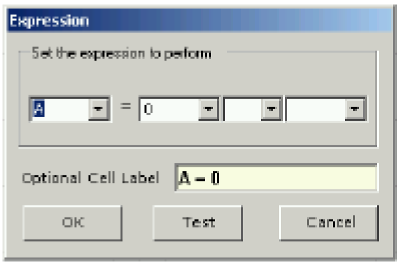
The program shown in Fig 13 includes a procedure called PROCESS which uses six of the LEDs on the T-board to display the count. The procedure is made up of a series of Compare commands that check the value of A, and a series of Outputs commands that switch on the corresponding number of LEDs.

The main routine also includes a Compare command that checks if A=6. When the count reaches six, it is reset to zero by an Expression command. Set this command as shown below.

Select A from the first box.

Select 0 from the second box.

Click OK.



Build and test run the program shown in Fig 13. Then download it to the T-board.

Extension

Develop the system so the user can also hear the count. Each time the count changes, it could make a higher pitched sound. When the count reaches six, it could play a sequence of sounds.

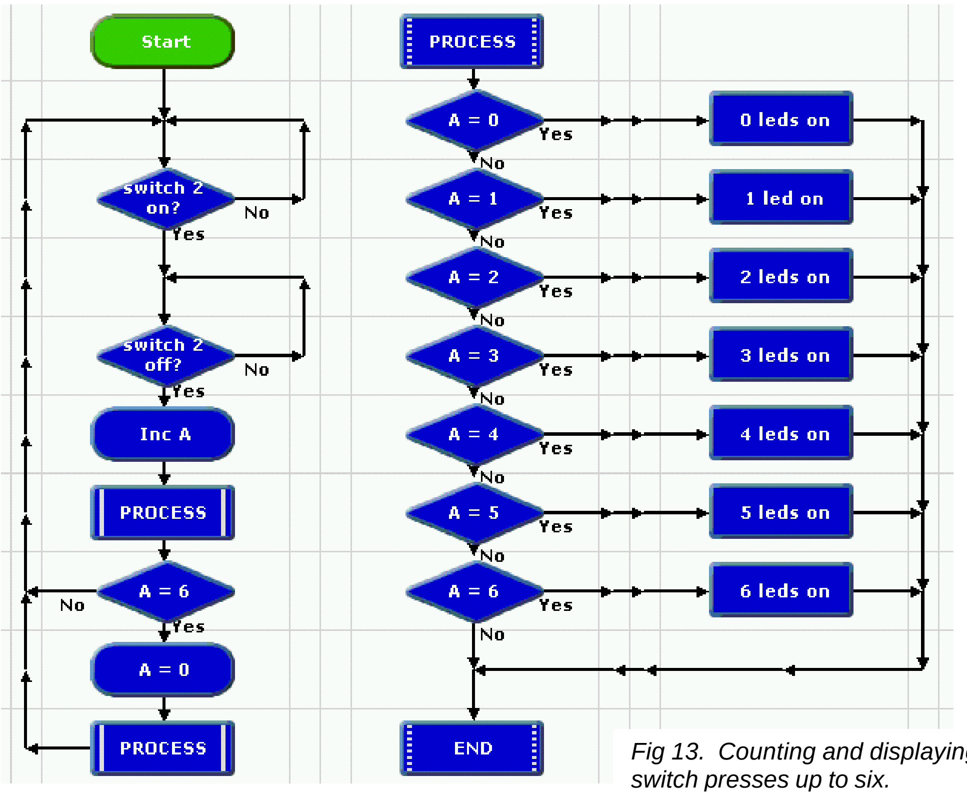


Fig 13. Counting and displaying switch presses up to six.

Interrupts (1)

Commands and Techniques

- Responding to a digital sensor while the program carries out other actions.

Decisions and Interrupts

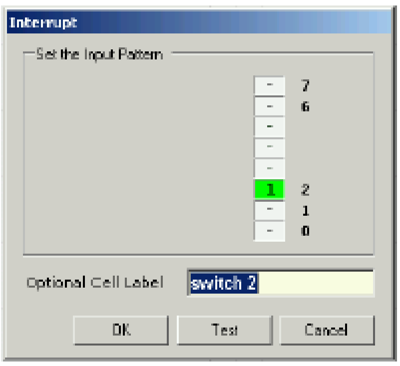
When you use a Decision command to check a digital input, the program stays in a loop around the command until the sensor responds as set.

However, an Interrupt checks a digital input continuously while the program carries out other actions. For example, the program shown in Fig 14 flashes the LEDs on the T-board in a continuous sequence. If you press switch 2 at any time, the sequence stops and the program waits in a Decision loop until you press switch 7 to start the sequence flashing again.

1. Build the program shown in Fig 14. Use the information below to help you.

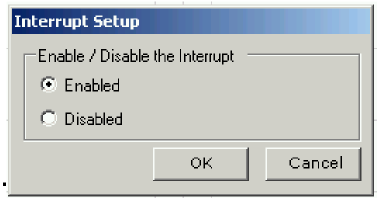
Setting the Interrupt
An Interrupt is similar to a procedure. It is a separate part of the program that begins with an Interrupt command and finishes with an End command.

Set the **Interrupt** command to check a digital input in the same way as you set a Decision command.



Whenever the digital input responds as set in this command, the program jumps to the Interrupt and carries out the commands below the Interrupt command. When it reaches the End command, the program goes back to the point it jumped from.

Enabling the Interrupt
The program has to be set up to respond to the Interrupt.
This is done by placing an **Interrupt Setup** command (set to "Enable") at the start of the program.



Whenever the Interrupt is triggered it needs to be enabled again, using the same command.

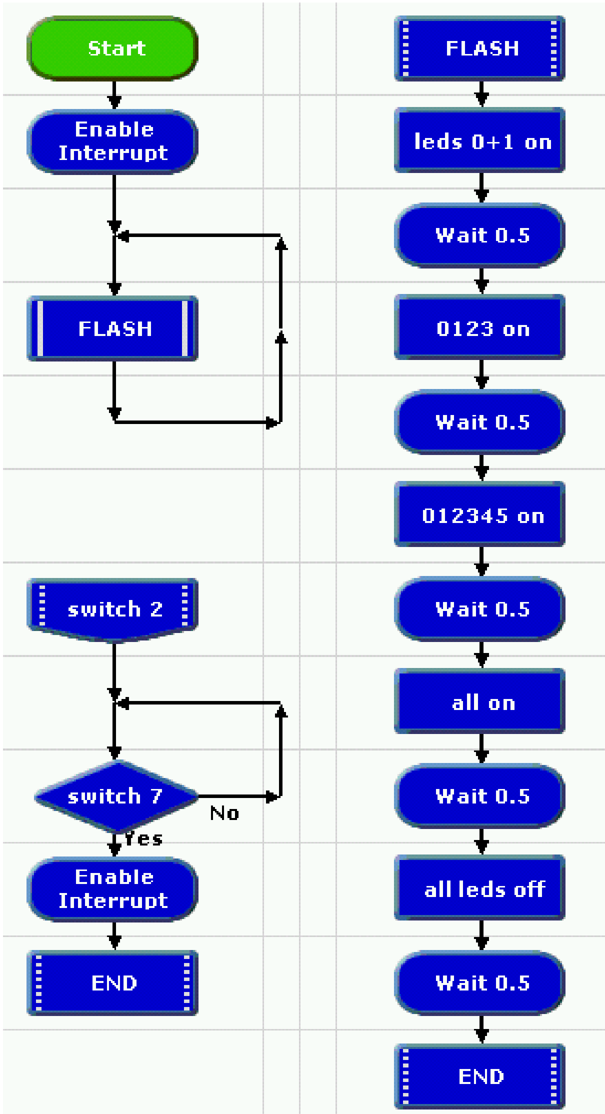
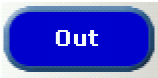


Fig 14. Using an Interrupt to freeze a flashing light sequence at any point.

2. Test run the program; then download it and try it in the T-board. If you reduce the Wait times, it can be used as a fast-reaction tester.

Interrupts (2)

Commands and Techniques



- Responding to a digital sensor while the program carries out other actions.
- Timing events.
- Displaying information in binary form, using the LEDs on the T-board.

Simple stopwatch

The program shown in Fig 15 turns the T-board into a simple stopwatch. Timing begins when you press switch 2. Timing is done by repeating the Inc A command every second. You can press switch 7 at any time to pause the timing and display the number of seconds elapsed in binary form using all eight LEDs on the T-board. Timing restarts (from the point at which you stopped it) when you press switch 2 again.

1. Build the program shown in Fig 15. Test run it; then download it and try it in the T-board.

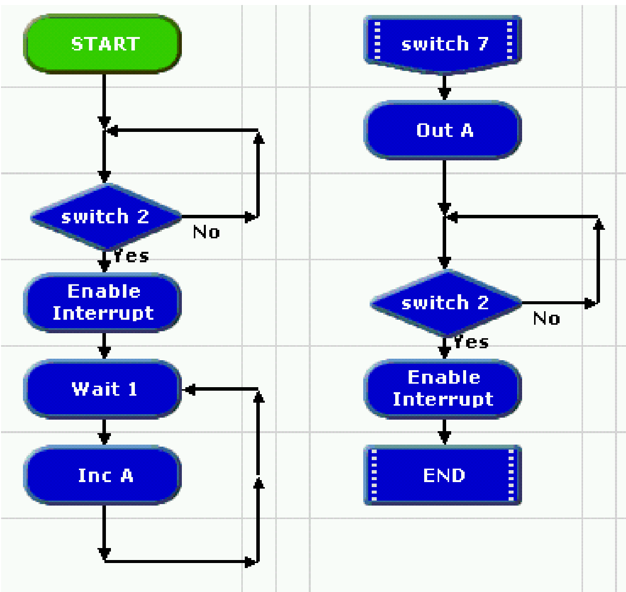
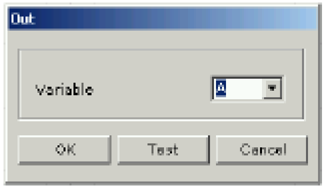


Fig 15. Simple stopwatch using an Interrupt to pause the timing, and displaying time elapsed in binary form.

Binary display using the Out command

In this program, the Inc A command is used to count the number of seconds elapsed.

So, when you set the Out command, select A. Click OK.



The Out A command sets the Output Pattern to the value of A, in binary. The table below shows the binary equivalent of each LED on the T-board. Add up the values of the LEDs that are lit. The total is the value of A. In this program, this is the number of seconds elapsed.

LED number:	0	1	2	3	4	5	6	7
Binary value:	1	2	4	8	16	32	64	128

Stopwatch with pause and reset features

The program shown in Fig 16 is a development of the simple stopwatch program. After you have pressed switch 7 to display the time, you have the choice of either: pressing switch 2 to restart the timing from the point at which you stopped it – or, pressing switch 6 and resetting the time and the display to zero. The command which sets A=0 is an Expression command.

2. Edit your program to that shown in Fig 16. Test run it; then download it and try it in the T-board.

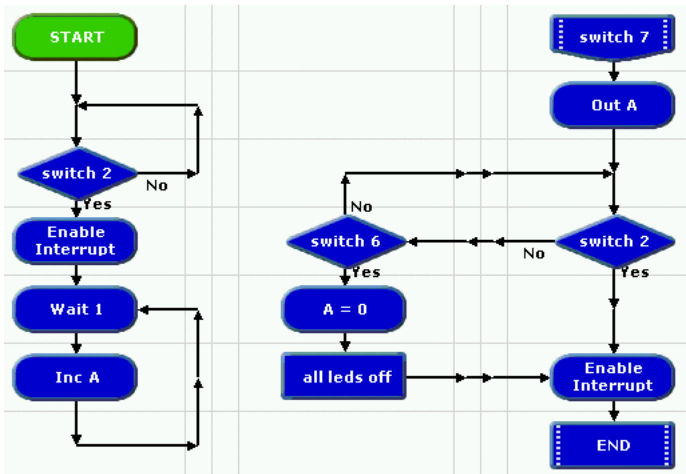


Fig 16. Stopwatch with pause and reset features.

Monitoring Analogue Input

Commands and Techniques

- Continuously checking input from 2 analogue sensors and switching outputs on and off in response.
- Using the “ignore” setting in the Outputs command.

Environment monitoring system

This system represents those used in tropical plant houses to keep light and humidity automatically at the levels required by the plants.

1. Build the program shown in Fig 17. This checks just the light sensor. As long as the light level is above 20, the artificial lighting (represented by LED 7) is switched off. If it falls below 20, the lighting is switched on.

2. Test run the program; then download it and try it in the T-board.

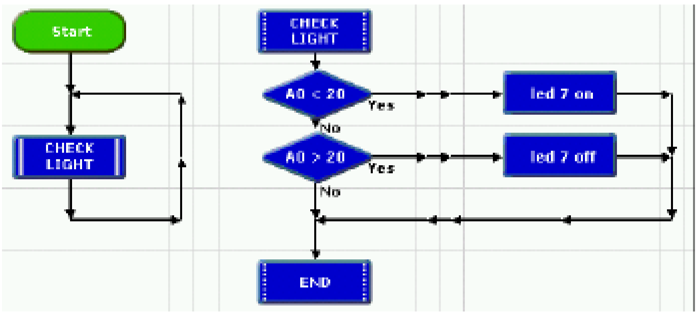


Fig 17. Light level monitoring system.

3. Add another procedure to the program, as shown in Fig 18, to check the humidity level.

The humidity sensor is represented by the variable resistor on the T-board. A sprinkler system, represented by LED 1 is switched on or off according to the reading from the sensor.

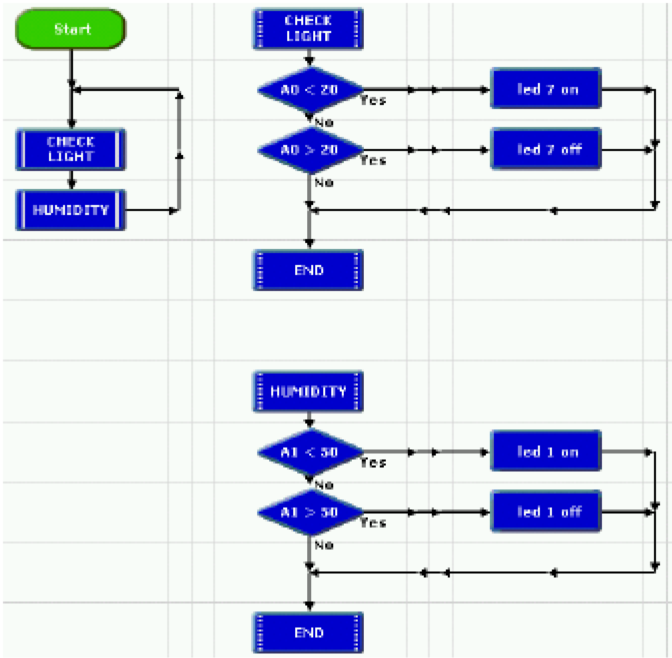


Fig 18. Light level and humidity monitoring system.

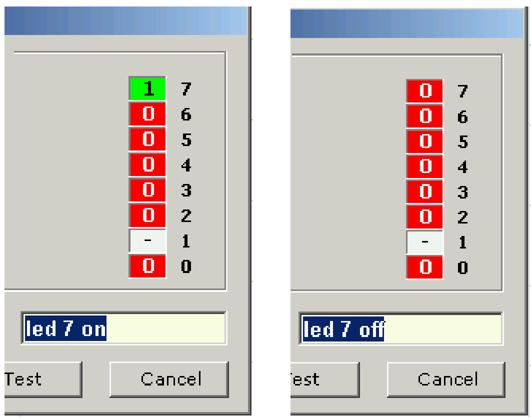
Outputs command “ignore” state

Each digit in the Output Pattern of an Outputs command can be clicked to a third setting:

This means “ignore this output; leave it as it was set in the last Outputs command.”

You will need to use this setting in the Outputs commands in the Fig 18 program, to prevent them from switching off an LED that you may want to keep switched on.

Set the LED 7 on and off Outputs commands as shown below. Then do the same for the LED 1 on and off commands.



EEPROM

Commands and Techniques

- Storing information in the PIC's memory – and retrieving it.

A counter without memory

The program shown in Fig 19 counts switch presses and displays the count in binary form using the LEDs on the T-board.

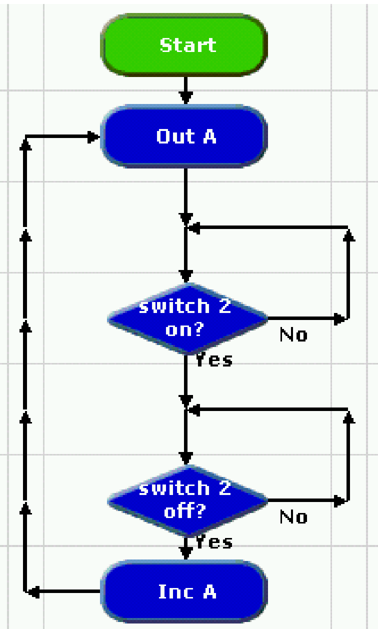


Fig 19. Counter with display but no memory.

1. Build the program shown in Fig 19. Test run it; then download it and try it in the T-board.

The program uses the variable "A" to store the count. When power is disconnected from the T-board, the value of all variables is reset to zero, so whenever power is reconnected to the T-board, the count begins again at zero. Try it.

It would be useful to be able to store the count in the T-board so that when you power it again the count resumes where it left off when power was disconnected. You can use **Read** and **Write** commands to access an area of the PIC's memory (EEPROM), so that you can do this.

A counter with memory

Fig 20 shows the program developed to memorise the count even when power is disconnected from the T-board.

2. Build the program shown in Fig 20. Use the information below to help you.

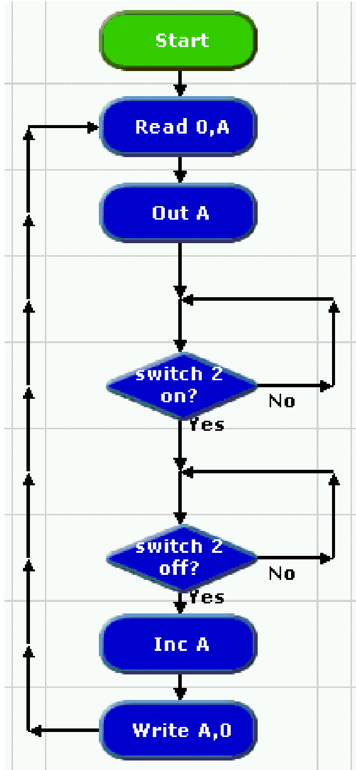
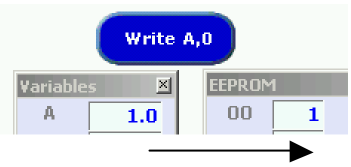


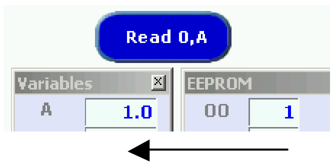
Fig 20. Counter with display and memory.

When the switch is pressed, one is added to the count by the Inc A command.

The **Write** command then copies the value of A into EEPROM address 0.



Before the count is displayed by the Out A command, the **Read** command copies the number stored in EEPROM address 0 into variable A.



When you set the Write and Read commands, select A from the Variable box, and 0 from the EEPROM box.

3. Test run the program with the Variables and EEPROM Panels displayed. Then download it into the T-board and check that the count is stored when power is disconnected.