

This programming guide is for use with:

- A programmable timer board.
- Crocodile clips - Yenka software.

Limitations of use:

- It must be used with kits from Kitronik.
- It can't be used for commercial gain.

Programming of your timer board has been split into four separate tasks. You might not have time to complete them all and your teacher will tell you how many they expect you to complete.

Task 1 - Basic timer

When the button is pressed the LED will light for 10 seconds, then the buzzer will sound for one second.

Task 2 - Early warning beeps

A few seconds before the one second out of time buzzer sounds, it will emit a few short beeps to warn the time is almost up.

Task 3 - User configurable delay

A special mode will be added to set how long the delay is so the timer can be reprogrammed during normal use without a PC.

Task 4 - Musical

The final task will be to replace the time out tone with a tune.

Start Yenka

- Press 'Start'.
- Select 'All Programs'.
- Select 'Yenka'.
- Run the application 'Yenka'.

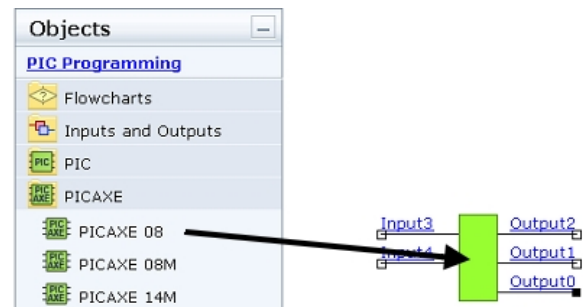
The following screen might be displayed:



If this is the case select the option to 'Start a new model'

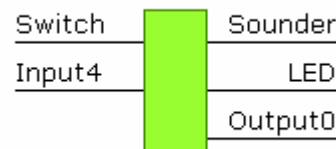
Before we can start looking at the software you will first need to put add processor:

- On the 'Objects' tool bar select 'PICAXE', (A list of PICAXE chips will be shown).
- Drag and drop a PICAXE 08 onto the main drawing area. (If you are doing all four tasks including the musical timer, use a PICAXE 08M)



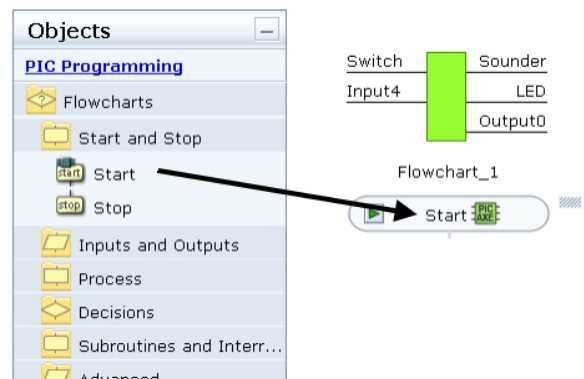
To make the software easier to write the names of the inputs and outputs can be changed to something more meaningful:

- Select Input3 (the cursor will change to a text editing cursor).
- Change the text to Switch.
- Now change Output1 to LED and Output2 to Sounder.



We can now start to add the flowchart:

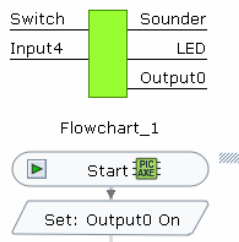
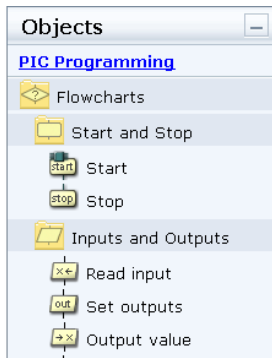
- On the 'Objects' menu, select 'Flowcharts'.
- From the extra options now shown select 'Start and Stop'.
- Drag and drop a 'Start' box onto the work area.



We are now ready to build up the flowchart. We are going to add to the 'Start' box.

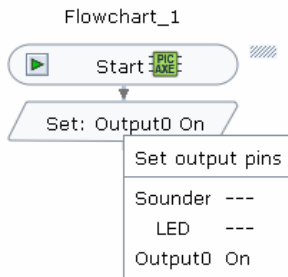
Firstly we need to ensure that the outputs are off.

- In the 'Flowcharts' section of the 'Objects' window, select 'Inputs and outputs'.
- Drag and drop a 'Set outputs' flowchart box under 'Start'.
- An arrow joining the 'Start' to the 'Output' box will appear to indicate that the two parts are joined together.



Click on the 'Output0 On' text in the outputs box that you have just added to the flowchart.

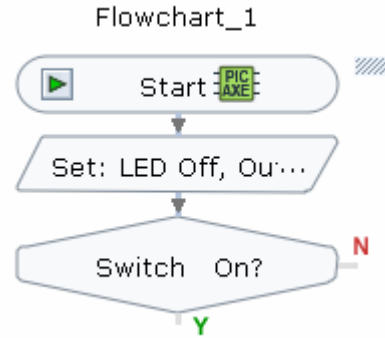
The following menu box will be displayed:



This allows you to decide if the output should be on, off, toggled or doesn't change ('---'). Try clicking on one of the outputs and you will see it step through each of these states. To start with we want every thing to be off, so make sure you set all three outputs to off before clicking off the menu box.

The first thing your program needs to do is wait until the switch has been pressed. To do this:

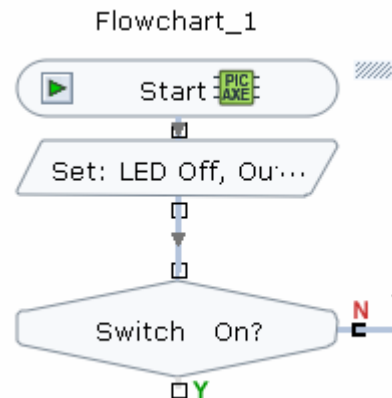
- Expand the 'Decision' section of the 'Flowcharts' options, in the 'Objects' menu.
- Drag and drop a 'Test input' box from the 'Objects' menu.



This decision box defaults to testing if the switch is on which happens to be what we need to test. You can change this by click on the text 'Switch' or 'On' and a drop down menu with a list of possible choices will be shown.

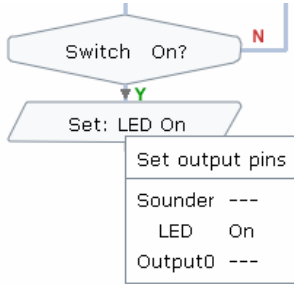
The program is going to wait until the switch is pressed, so if it is not pressed, it needs to loop back and check again:

- Drag the decision box a little further down the flowchart.
- Click on the line exiting the decision box by the red 'N'.
- Click again to extend a line right and up (as shown)
- Finally click once more this time on the interconnecting line between the 'Set' box and the 'Switch On?' box.



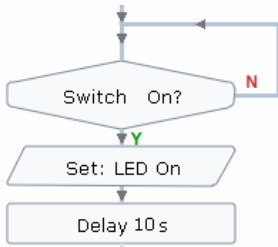
We need the LED to light once the switch has been pressed.

- Drag and drop a 'Set outputs' below the 'Switch On?'
- Click on the text to bring up the menu.
- Select the LED to be On and the other outputs to be unchanged, as shown.



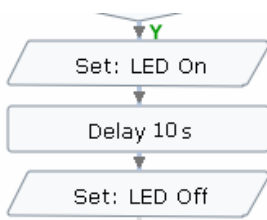
The next step is to make the processor wait. You probably will want a longer timer, but we will start with a ten second timer as it will make testing faster. You can always adjust this time later yourself.

- On the 'Processes' section of the 'Flowcharts' options, drag and drop a 'Delay' command, under the 'Set LED On' box.
- Click on the number after 'Delay' and enter 10 seconds.



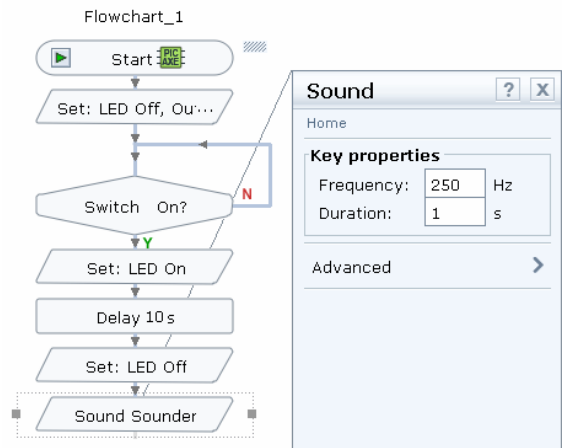
At the end of the ten seconds delay, the LED needs to go out.

- Drag and drop a 'Set outputs' box from the menu.
- Click on the output box.
- Set the LED to 'Off'.

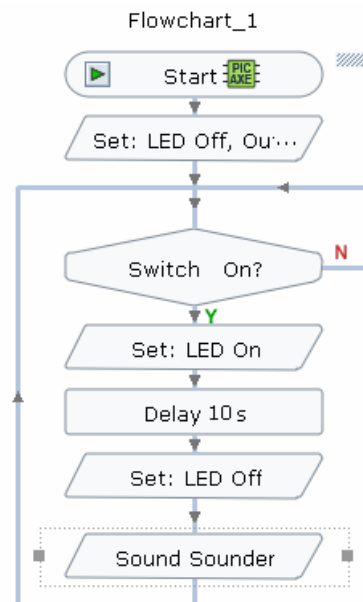


The final operation in the process is to play a sound.

- Drag and drop a 'Sound' box onto the flowchart.
- Click on the text and set the output pin to 'Sounder'.
- Double click on the 'Sound' box.
- Leave the frequency at 250Hz.
- Adjust the duration to one second.
- Close the menu.



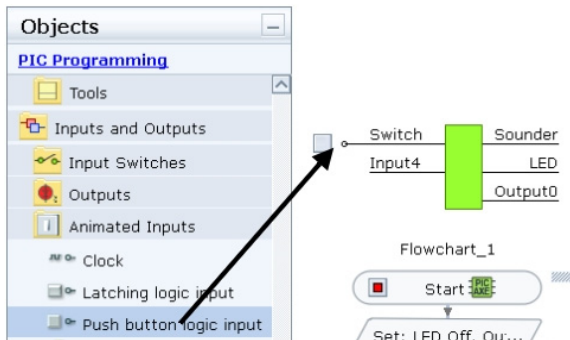
The final part of the flowchart is to get this repeated over and over. To do this the sound box needs connecting back into the point where we check the switch, as shown.



You are now going to simulate the software on the PC, before you program your board.

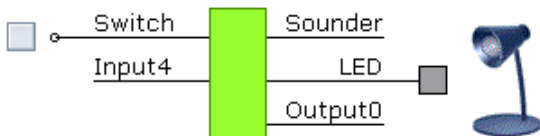
To help do this you will need to add a switch to the circuit to trigger the timer and a lamp to see when the LED turns on:

- On the 'Objects' menu select 'Inputs and Outputs'.
- Then select 'Animated Inputs'.
- Drag and drop 'Push button logic input' on to the switch line on the PIC hardware.

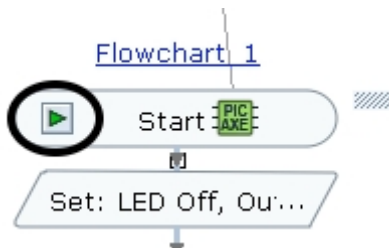


Now add a lamp to the LED:

- Select 'Animated Outputs'.
- Drag and drop a 'Lamp' onto the 'LED' pin on the PIC.



Now you are ready to simulate, by pressing the 'Play' button.



You will see the part of the flowchart that is running turn blue, in this case it will be waiting for the decision of whether the switch is pressed to be true.

When you click on the 'Switch' it will be set and the program will continue.



Check that this is the case, you should also see the lamp turn on as the LED goes on and if your PC has speakers connected to it you will hear a tone after 10 seconds. If your software didn't work as it should, double check the flowchart (previous page).

You are now ready to program the PIC. To do this make sure your board is powered up, it contains a PIXAXE® chip and that the programming cable is plugged into the board.

Double click on the start flowchart box (avoiding the 'play' / 'stop' button and the 'PICAXE' chip button. The following menu will be shown:



Press the 'Program real microcontroller' button. A programming box will appear and all being well a short while later the chip will be successfully programmed and the box will go away.

If there is a problem with the connection or power to the board, a box will pop up to tell you so.

You're now ready to check the timer works. Press the button, check the LED lights and then 10 seconds later as the LED goes out the buzzer sounds for a second. Check it works a 2nd time.

Well done you have completed the 1st programming task.

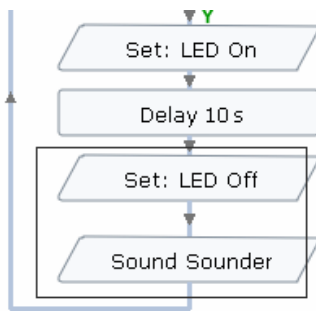
Task 2

In this task you will bleep the sounder 3 times just before the time period is due to run out.

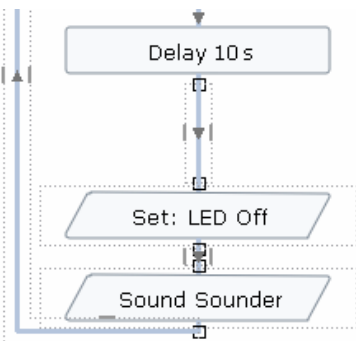
You are going to add to the flowchart you created in task 1.

You will need to make a space so that we can add the bleep buzzer functionality.

- Move the 'Set: LED Off' and 'Sound' box down:
- Select the last two boxes in the flowchart by left clicking to the side of the 'Set: LED Off' and drag down and right to select box boxes.

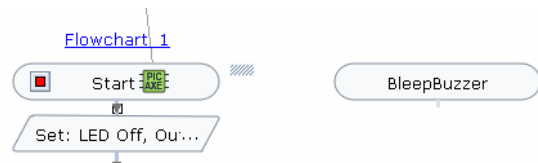


- Once selected drag them down to make space for an extra box:



We now have space for the bleep buzzer functionality. This is going to be added as a procedure. Procedures are good for two reasons:

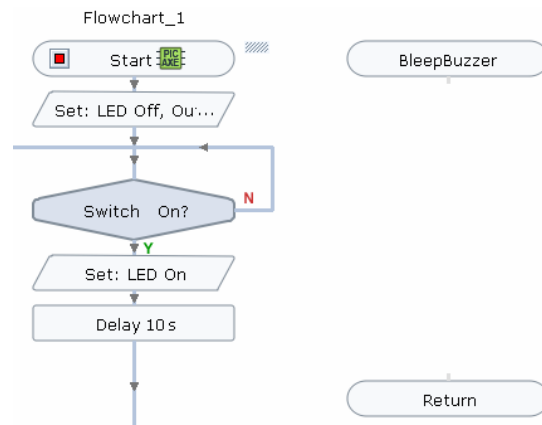
- They keep sections of the functionality together, making it easier to read.
- The same functionality can be run from more than one place.
- In the 'Objects' menu select 'Subroutines and Interrupts'.
- Drag and drop a 'Subroutine' box to the right of 'Start' in the main window.



- Click the 'Subroutine1' box to edit the text.
- Enter the name of the procedure in the menu. As this is going to bleep the buzzer let's call it 'BleepBuzzer'.

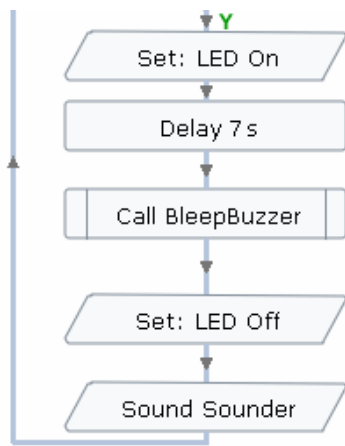
You will see that the box is now called 'BleepBuzzer'.

Whilst the 'Subroutines and Interrupts' part of the tool bar is open, let's drop the 'Return' onto the flowchart. We can put it in the right place later.



We will add the functionality to the subroutine shortly, first let's have the BleepBuzzer subroutine called at the end of the 10 second delay.

- Drag and drop a 'Call subroutine' box into the gap made in the main flowchart.
- The software will automatically connect the flowchart either side of the box.
- As only one subroutine has been put on the flowchart, this will be selected by default.



The timer is going to be set-up to bleep 3 times before the end of time sounds.

To do this we are going to use a variable to count how many bleeps there have been. This way you don't need to put a wait, sound, wait, sound, wait, sound command in the flowchart.

Not too bad with three bleeps but if you wanted ten warning bleeps it would get very tedious and use up lots of the memory in the processor.

On the 'Processes' section of the 'Objects' menu are the options to set and increment variables.

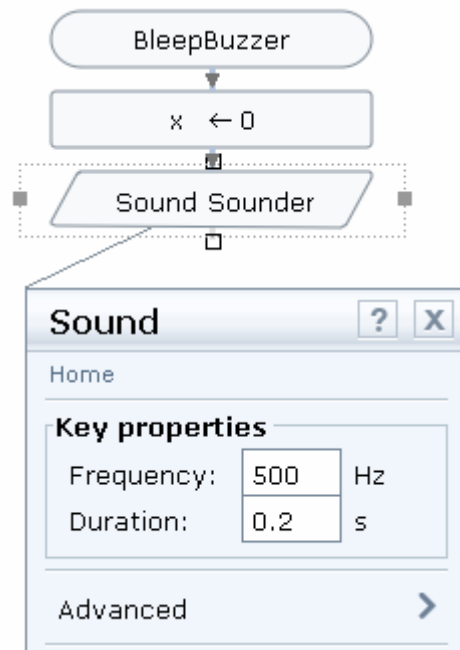
Before the three bleeps can be counted, a variable must be set to 0.

- Drag and drop a 'Set variable' box onto the flowchart.
- This defaults to set $X = 0$ which is what we need.

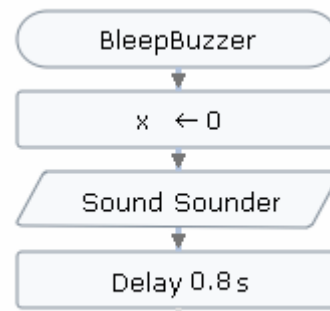


The next step is to sound a short beep.

- Go back onto the 'Inputs and Outputs' section of the 'Objects' menu.
- Drag a 'Sound' onto the chart.
- Click on the text to set the output pin to 'Sounder'.
- Double click on the 'Sound' box.
- Set a duration of 0.2 seconds.
- You can use a higher pitch note, say 500Hz.

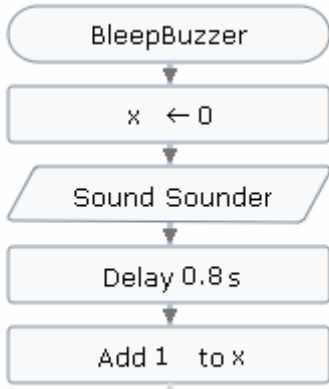


After the sound add a 'Delay' of 0.8 seconds:



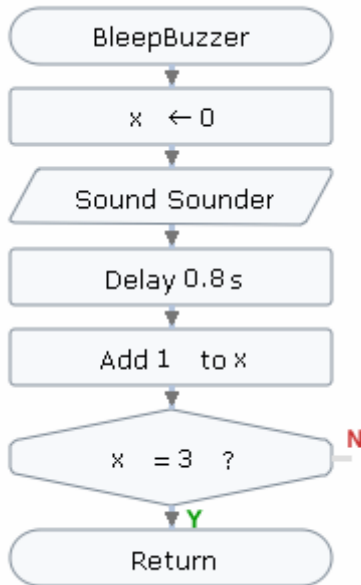
Now add an increment X.

- Select the 'Processes' part of the 'Objects' menu.
- Add an 'Increment variable' box, below the 'Delay'.
- This will default to 'Add 1 to x' which is what is required.

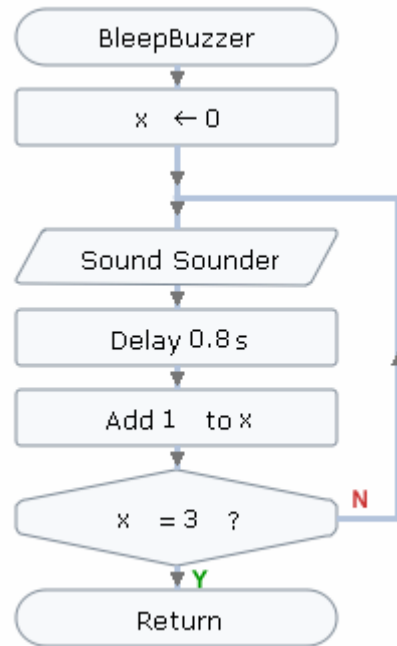


Now we need to check the value of 'X', if it is 3 then we are finished, otherwise it's back to the top for another bleep.

- Drag a 'Test variable' box onto the chart (in 'Decisions').
- This defaults to testing if 'X = 0' edit the number so that you are testing if 'x = 3?':

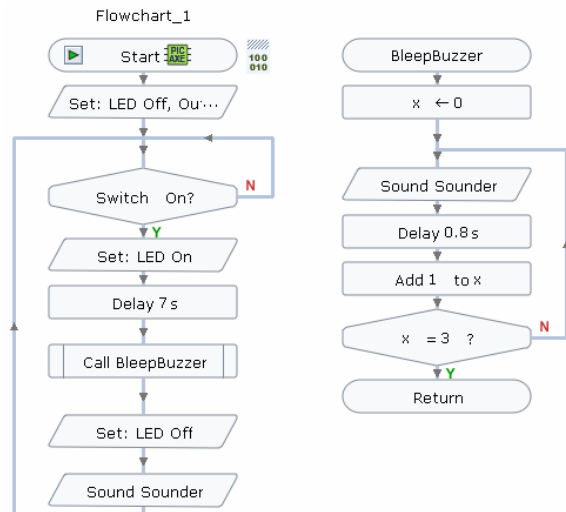


To finish the procedure you need to join the 'Return' box to the flowchart and connect to the 'No' path as shown:



Finally to keep the overall delay to 10 seconds adjust the 'Delay' in the main flowchart from 10 to 7 seconds.

Check your flowchart looks like this one:



You're ready to simulate.

- *Press 'Play' (on the 'Start' flowchart box).*
- *Press the 'Switch' input.*
- *Make sure your software waits 7 seconds, bleeps 3 times then after a total of ten seconds bleeps a longer time up sound.*

You can now program your board, just make sure its connected and powered before pressing the program button (Double click on the 'Start' flowchart box for the program menu).

Check the board works the same as the simulation.

Well done you have completed task two.

Task 3

In task three the user will be able to set how long they would like the delay to be in multiples of 10 seconds.

To do this whenever the software is started with the switch pressed a special section of flowchart will run.

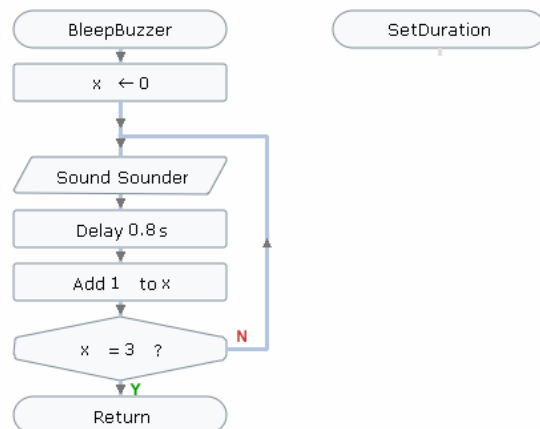
Whilst the switch remains pressed this part of the flowchart will sound a short beep every second and increment a count.

When the switch is released this count will be stored in secure memory. If the switch is not pressed on power up this count will be read from the memory.

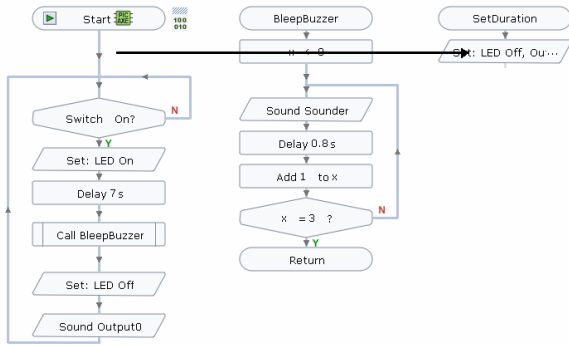
In place of the single 'Delay' a count of 'Delay 10s' commands will be maintained, when this matches the stored count the buzzer can sound.

The functionality to set the duration is going to be put in a subroutine, this can be put on the flowchart sheet to the right of the 'Start'.

- *Select 'Subroutines and Interrupts' on the 'Objects' menu.*
- *Drag a 'Subroutine' box onto the flowchart.*
- *Click to name the function 'SetDuration'.*

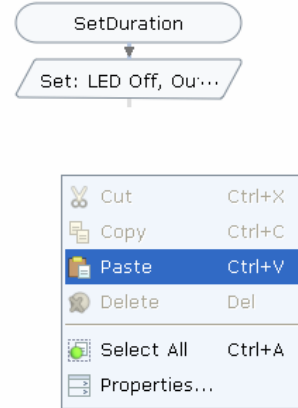


Move the 'Set LED Off, ...' command under the 'Start' to the right (underneath the set duration subroutine).

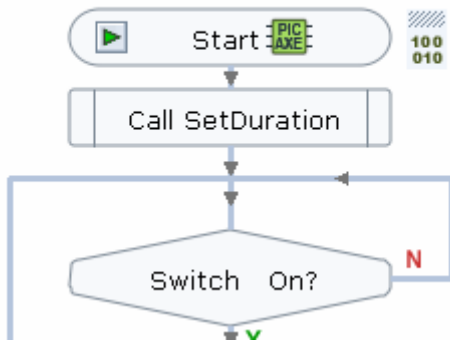


- Drag a 'Call subroutine' box onto the flowchart below 'Start'.
- Click on the text to set the subroutine to 'SetDuration'.

- Select the place where you want this box to go.
- Right click.
- Select Paste from the drop down menu.

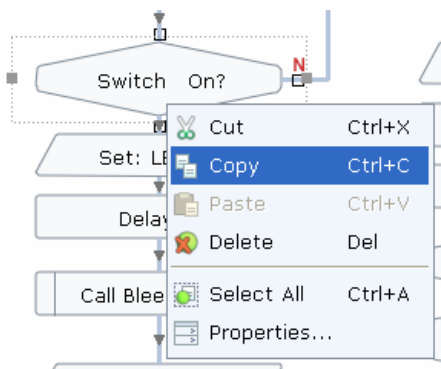


Flowchart_1



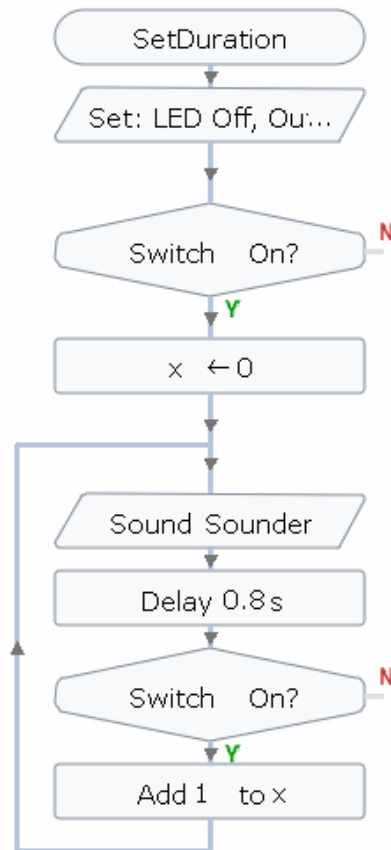
This function reuses existing functionality, which can be copied (rather than re-entered). To cut and paste the 'Switch On?':

- Right click the item to be copied.
- Select copy from the drop down menu.



Using the boxes already on your flowchart cut and paste to build up the following chart:

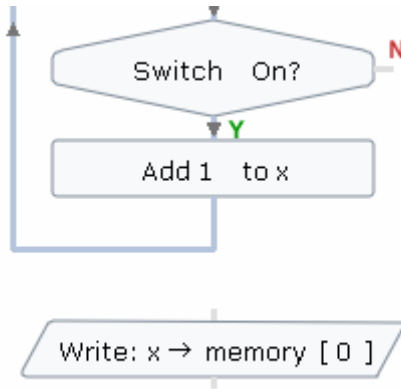
You will need the switch decision twice, as well as variable operations, 0.2 second sounds & delays.



So far the 'SetDuration' function has counted the number of seconds the switch has been pressed from program start.

The next step is to store this in the secure memory (called EEPROM):

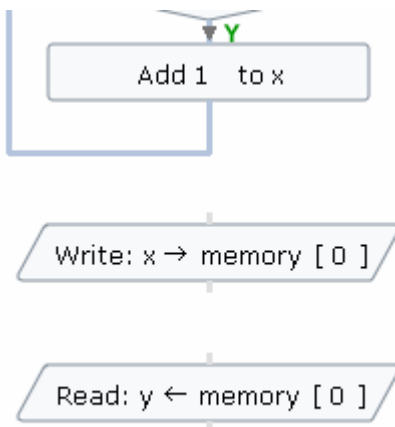
- Select the 'Advanced' section of the 'Objects' menu.
- Drag and drop a 'Write memory' onto the flowchart.



To configure the 'Write memory', you can edit the text indicating the variable and the address where this is stored. As the box defaults to storing X in memory location 0 it doesn't need to be changed.

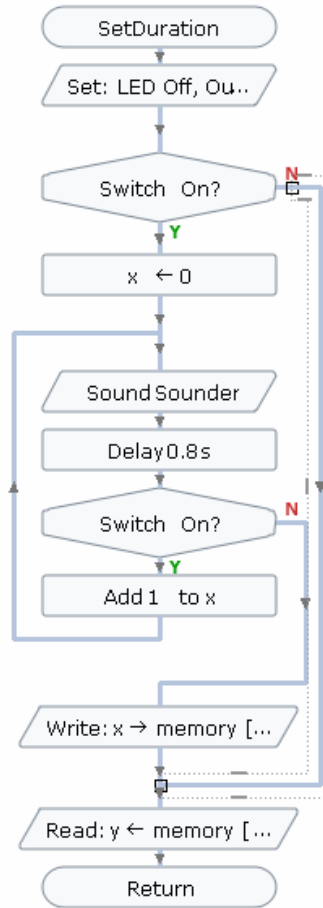
Because the variable X is used in other parts of the program we are going to read the EEPROM back into variable Y.

- Drag a 'Read memory' box onto the chart, leaving a gap between the 'Write memory'.
- Configure this to read from EEPROM 0 and store in variable Y.



This is the last part of the function.

- Add a 'Return'.
- Connect up the flowchart as shown:



Let's check this part works. To see what is happening we are going to view the variables X and Y:

- In the 'Flowcharts' menu expand the 'Tools' section.
- Drag and drop a 'Monitor Box' onto the work area.

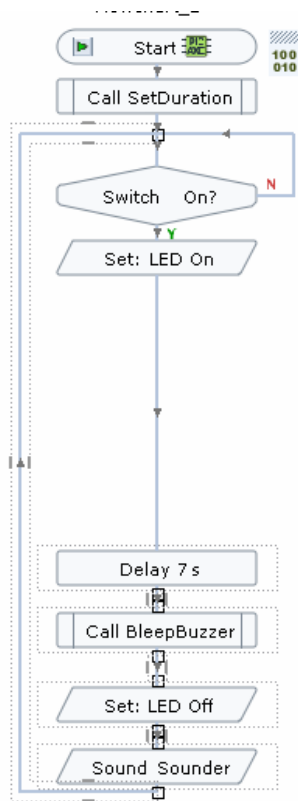
Press 'Play' and immediately press and hold the 'Switch' Input.

Watch the 'Monitor' box. In this you will see that variable X goes up every time the 'Add 1 to X' box is run.

Release the 'Switch', as the function finishes you will see:

- Variable X is stored in EEPROM.
- The EEPROM is copied to variable Y.
- Stop the simulation and delete the monitor box.

You need to make some space to add the new functionality to the main part of the flowchart. Move the last four boxes down:



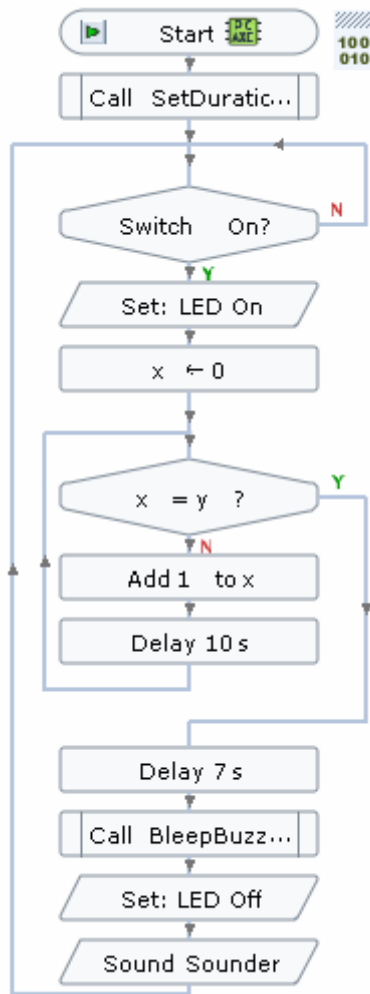
Add the following functionality in the space you have made, below the 'Set LED On' command.

- Set variable X = 0.
- Compare X to Y.
 - If equal run the 'Delay 7s' box.
 - Otherwise, Increment X, Delay 10 seconds, loop back to comparing X to Y.

If you need to swap the 'Y' and 'N' on a decision box around, just click on the 'Y' or the 'N'.

Try to finish this part of the flowchart on your own before you turn to the next page.

Hopefully you should now have a flowchart that looks like this one:



Now you can check that your user programmable timer works.

The EEPROM value you stored when you last simulated it will still be stored.

Run the software and check that it waits for the delay you set, before the buzzer sounds.

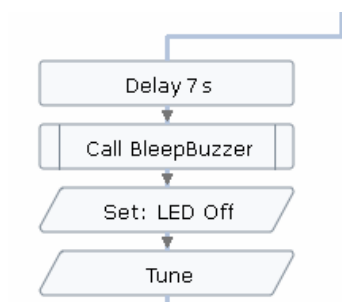
Once you are happy download the software into your board and test.

Well done you have completed task 3!

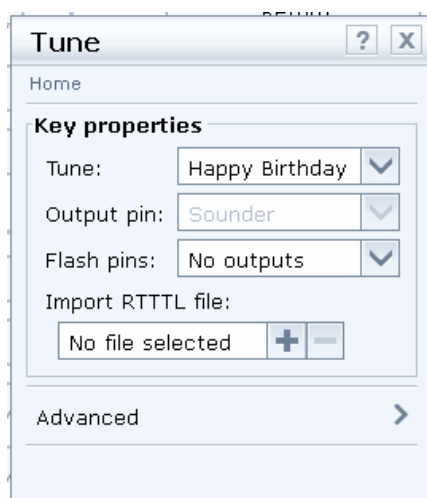
Task 4

In this task you will get your timer to play a short tune when the time is up. To do this you will need to be using a PICAXE08M chip.

- *Remove the 'Sound sounder' at the end of the main part of the flowchart.*
- *On the 'Inputs and outputs' section of the 'Flowcharts' toolbar, drag a 'Tune' into its place.*



When you double click on the 'Tune' a menu is shown:



On the Tune drop down menu you can select a pre-written tune.

Alternatively, select 'Imported RTTTL file' at the bottom of the Tune drop down menu, then press the '+' to open a RTTTL file.

You can now use any mobile phone ring tone that has been saved in the 'Ringing Tones Text Transfer Language (RTTTL)'.

Once you have selected a tune close the menu box.

You can simulate the tune or just programme the board to test the tune.

Well done - you have completed all of the tasks.

Thank you for using this guide, which has been produced by Kitronik in collaboration with Crocodile clips Ltd, developers of Yenka.

PICAXE® is a registered trademark licensed by Microchip Technology Inc. to Revolution Education Ltd.