This programming guide is for use with:

- A programmable timer board.
- PICAXE programming editor 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 bleeps

A few seconds before the one second out of time buzzer sounds, it will emit a few short bleeps 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.

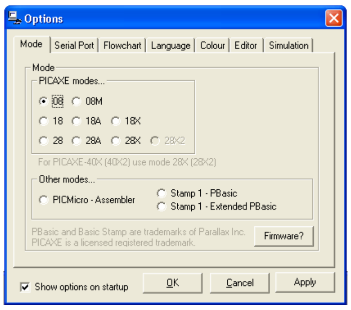
Task 4 - Musical

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

Start the PICAXE programming editor software

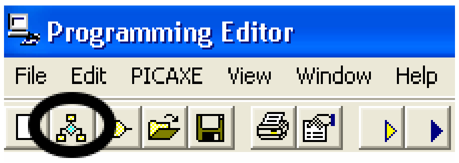
- *Press 'Start'.*
- *Select 'All Programs'.*
- *Select 'Revolution'.*
- *Run the application 'PICAXE programming editor'.*

When the software starts a menu is displayed allowing the PICAXE chip to be selected:


Options
Mode | Serial Port | Flowchart | Language | Colour | Editor | Simulation
Mode
PICAXE modes...
[✓ 08] 08M
18 18A 18X
28 28A 28X 28X2
For PICAXE-40X (40X2) use mode 28X (28X2)
Other modes...
PICMicro - Assembler
Stamp 1 - PBasic
Stamp 1 - Extended PBasic
PBasic and Basic Stamp are trademarks of Parallax Inc.
PICAXE is a licensed registered trademark.
Firmware?
[✓] Show options on startup
OK Cancel Apply

Make sure that a 08 chip is selected. (If you are doing all the programming tasks including the musical timer you will need to select a 08M chip).

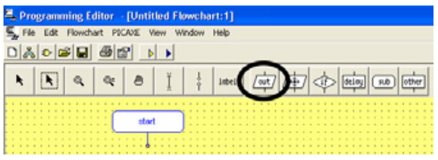
You are going to write the software using a flowchart. To do this you will need to select the flowchart button:



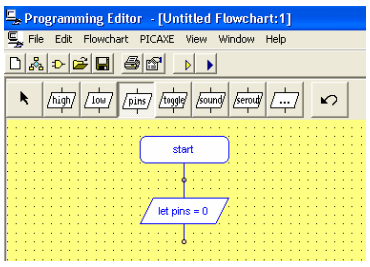
We are now ready to build up the flowchart. The application loads with a start box at the top of the screen. We are going to add to this.

Firstly we need to ensure that the outputs are off.

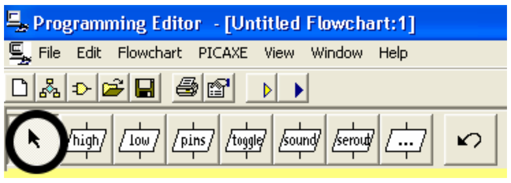
- *On the menu bar at the top of the screen select 'out'.*


Programming Editor - [Untitled Flowchart:1]
File Edit Flowchart PICAXE View Window Help
label
out
if
delay
sub
other
start

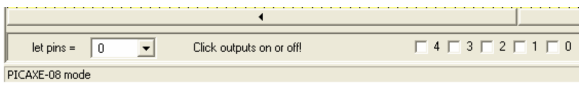
- *The menu changes to show the output flowchart options.*
- *Select the 'pins' flowchart box.*
- *Click on the chart below the start box.*


Programming Editor - [Untitled Flowchart:1]
File Edit Flowchart PICAXE View Window Help
high
low
pins
toggle
sound
serout
...
start
let pins = 0

To configure the outputs, you need to select the mouse pointer. Either click on the mouse pointer button or right click anywhere in the flowchart window:


Programming Editor - [Untitled Flowchart:1]
File Edit Flowchart PICAXE View Window Help
high
low
pins
toggle
sound
serout
...

Now when you click on the let pins = 0 box that you have added the tool bar at the bottom of the screen changes.


let pins = 0

Click outputs on or off!

4 3 2 1 0

PICAXE-08 mode

This allows you to decide if the output should be on or off. You can either set all the pins with a binary value or put a tick in the outputs you want to be on. Try clicking on one of the outputs and you will see a tick appear and the value change. To start with we want every thing to be off, so make sure you remove all the ticks and the let pins shows 0.

Each IO pin on the processor has been given a number so that it can either be tested or turned on or off.

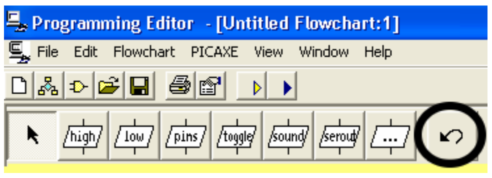
This number is different to the pin number on the chip.

Your timer board has been wired up as follows:

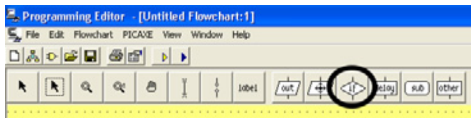
Connection	**Number**
LED	**1**
Sounder	**2**
Switch	**3**

So to reference the switch we would use 3.

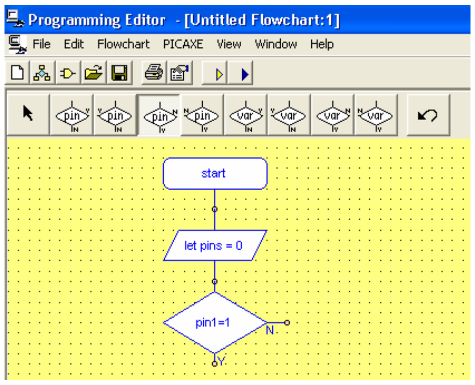
The first thing your program needs to do is wait until the switch has been pressed. To do this:


Programming Editor - [Untitled Flowchart:1]
File Edit Flowchart PICAXE View Window Help
high
low
pins
toggle
sound
serout
...

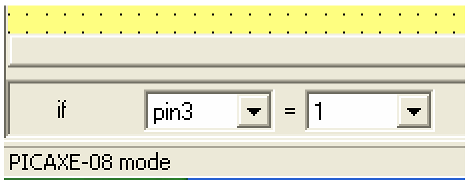
- ***Press the go back arrow.***
- ***Select the 'if commands' menu.***


Programming Editor - [Untitled Flowchart:1]
File Edit Flowchart PICAXE View Window Help
label
out
if
delay
sub
other

- ***Select the condition to test a pin with the 'Y' at the bottom and 'N' on the right.***
- ***Click on the flowchart below the 'let pins = 0' to add the 'if command'.***


Programming Editor - [Untitled Flowchart:1]
File Edit Flowchart PICAXE View Window Help
pin
var
start
let pins = 0
pin1=1
N
Y

The switch is on number 3 and we need it to wait until this becomes high (1). To do this:

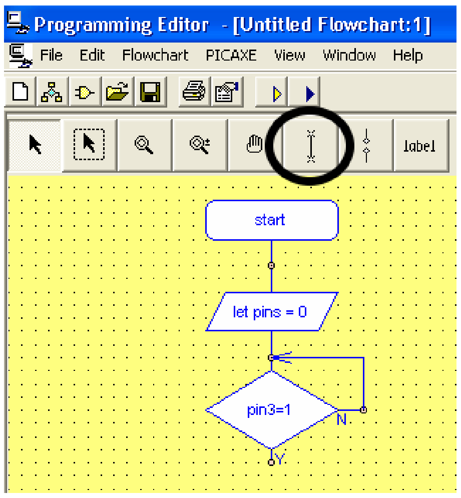

if pin3 = 1

PICAXE-08 mode

- ***Select the Pin1=1 if command. (Right click to deselect the 'if command', then left click on the box).***
- ***In the bottom toolbar select pin3 from the dropdown toolbar.***

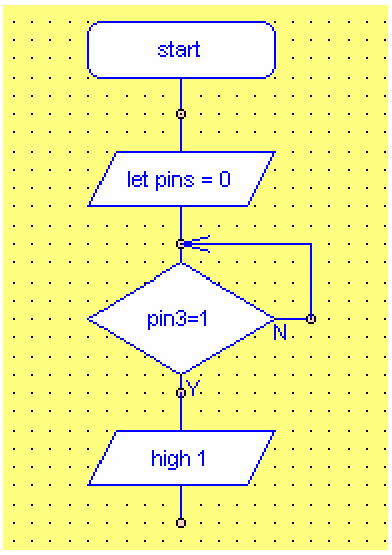
You can now connect up the ‘No’ route, going back into the 'if pin3 = 1' box.

- *Go back to the main toolbar buttons by selecting the back button.*
- *Select the draw lines button.*
- *Click on the ‘N’ on the ‘if command’ and then click on the connection between the ‘let pins = 0’ and the ‘if pin3=1’ box as shown:*


Programming Editor - [Untitled Flowchart:1]
File Edit Flowchart PICAXE View Window Help
label
start
let pins = 0
pin3=1
N
Y

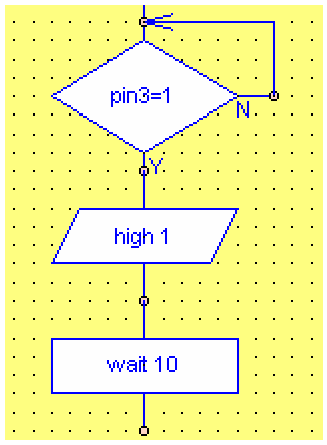
We need the LED to light once the switch has been pressed.

- *Bring up the ‘out’ buttons.*
- *Select the ‘high’ box.*
- *Add a ‘high’ box to the flowchart below the ‘if pin3=1’.*
- *Right click to deselect the ‘high’ box.*
- *Left click on the ‘high 0’.*
- *On the bottom toolbar select ‘1’ to set the LED to a high state.*



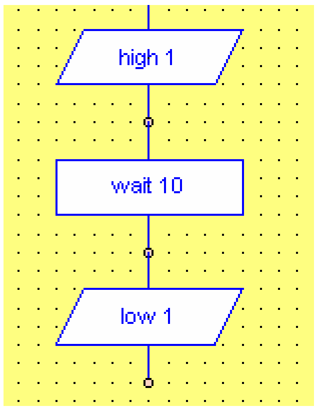
The next step is to make the processor wait. You probably will want a longer timer, but we will start with a ten second timer as it will make testing faster. You can always adjust this timer later yourself.

- *Go back to the main toolbar.*
- *Bring up the ‘delay’ buttons.*
- *Select ‘wait’.*
- *Add a ‘wait’ under the ‘high 1’.*
- *On the bottom toolbar configure this to 10 seconds.*



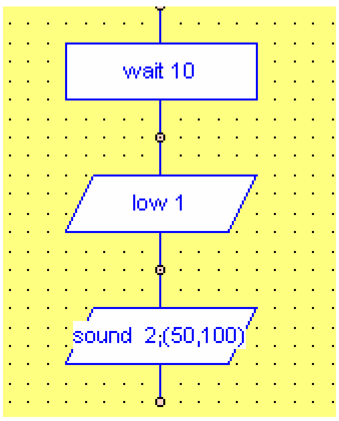
At the end of the ten seconds delay, the LED needs to go out.

- *From the main toolbar, select ‘out’.*
- *Then select and add a ‘low’ box to the flowchart after the ‘wait 10’.*
- *Configure it so that the LED (‘1’) is low.*

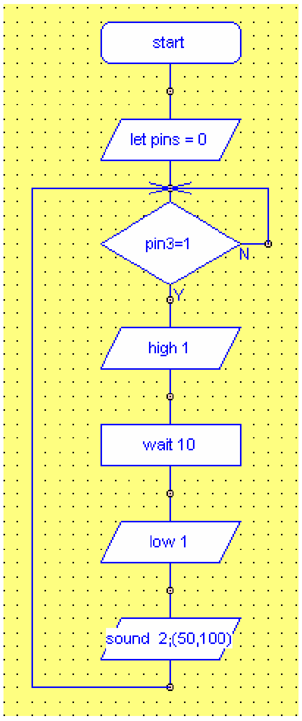


The final operation in the process is to play a sound.

- *In the 'out' buttons, select and add to the bottom of the flowchart a 'sound' box.*
- *On the bottom toolbar, on the first dropdown menu, select an output pin of 2.*
- *In the next box on the bottom toolbar select a note of 50 and duration of 100.*

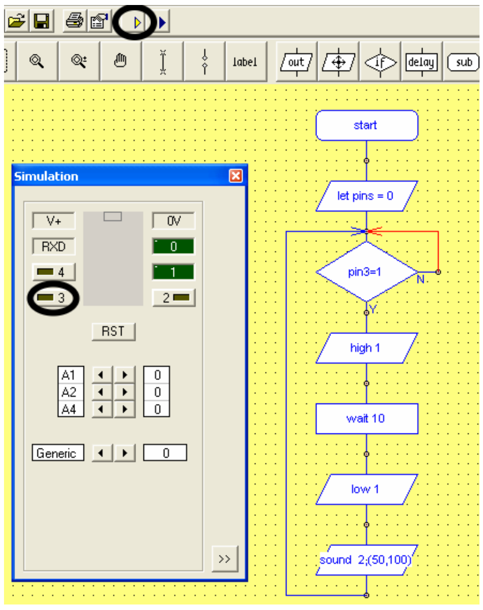


The final part of the flowchart is to get this repeated over and over. To do this the sound box needs connecting back into the point where we check the switch, as shown:



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

- *Press the yellow play button.*
- *The simulation box will open and the part of the flowchart that is running will be highlighted in red.*



- *In the simulation window clicking the button labelled 3 will simulate the switch being pressed, clicking a second time will release the switch.*

When you press the switch, you should also see output 1 change from 0 to 1 as the LED goes on and if your PC has speakers connected to it you will hear a tone after 10 seconds.
If you software didn't work as it should, double check the flowchart (left).

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.

If you are still simulating, you will need to stop the simulation (click anywhere on the flowchart window).

Press 'run' button (blue arrow). A 'Downloading program' box will appear and all being well a short while later the chip will be successfully programmed and the box will change for a box indicating success.

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.

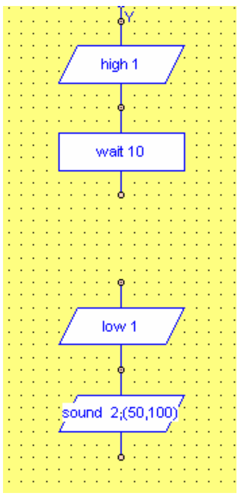
Task 2

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 'low 1' and the 'sound 2;(50,100)' down a bit, to make space for a new box.***



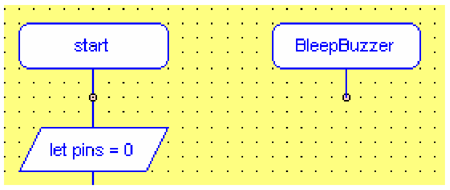
We now have space for the bleep buzzer functionality.

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.**

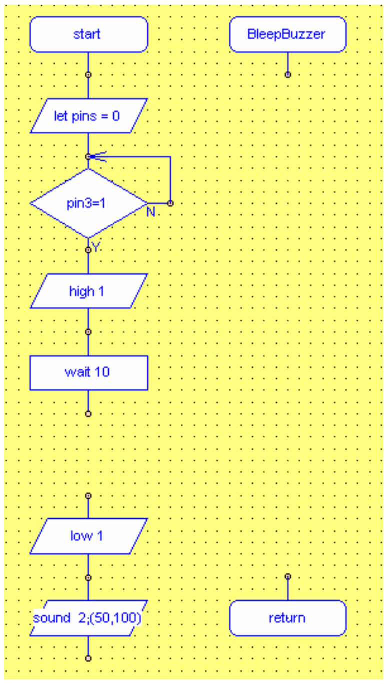
First let's add the procedure box.

- ***Select 'sub' from the main menu buttons.***
- ***From the subroutine options select 'sub'.***
- ***Add a subroutine to the right of the 'start' box.***
- ***In the bottom toolbar enter a name of 'BleepBuzzer'.***



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

Whilst the procedure tool bar is open, let's drop the 'return' onto the flowchart. We can put it in the right place later.

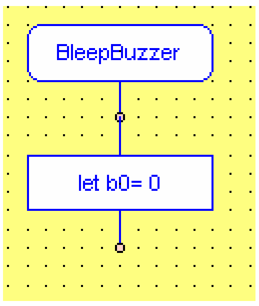


The timer is going to be set-up to bleep 3 times before the end of time sounds.

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.

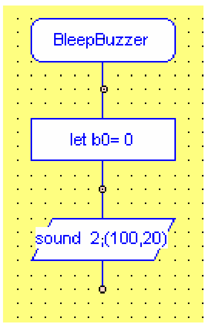
Not too bad with three bleeps but if you wanted ten warning bleeps it would get very tedious and use up lots of the memory in the processor.

- ***On the main tool bar select 'other' then select the 'let' command.***
- ***Put a 'let onto the flowchart below the 'BleepBuzzer' subroutine.***
- ***The let defaults to variable b0 being set to zero which is the functionality we require. The bottom toolbar could have been used to adjust this if required.***



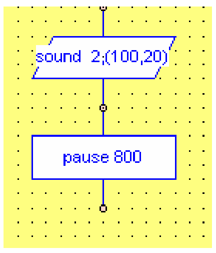
The next step is to sound a short beep.

- ***Go back onto the 'out' flowchart buttons.***
- ***Select 'sound' and add the box to the chart under the 'let b0= 0' .***
- ***Edit the sound on the bottom toolbar.***
- ***Set the pin to 2.***
- ***Set the note to 100 and the duration to 20.***



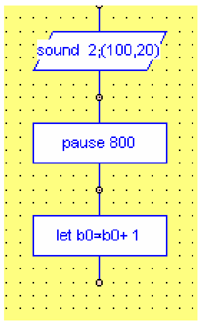
After the sound add a 'pause' of 800 mS:

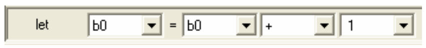
- ***On the 'delay' buttons, use 'pause'.***
- ***Enter 800 in the bottom toolbar.***



Now add an increment b0.

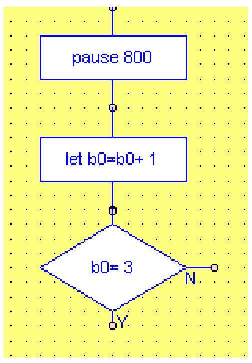
- ***Return to the 'other' flowchart buttons.***
- ***Add a 'let' box, below the 'pause'.***
- ***Configure this with 'b0 = b0 + 1' (using the drop down boxes on the bottom toolbar).***



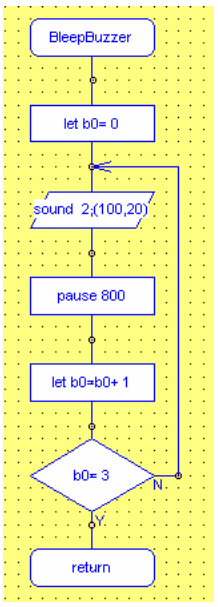


Now we need to check the value of 'b0', if it is 3 then we are finished, otherwise it's back to the top for another bleep.

- ***On the 'if' buttons select a 'var' with the 'Y' on the bottom and the 'N' on the right.***
- **Put this on the flowchart under the 'let b0 = b0 + 1'.**
- ***Use the three drop down menus on the bottom toolbar to set it up with if 'b0 = 3'***

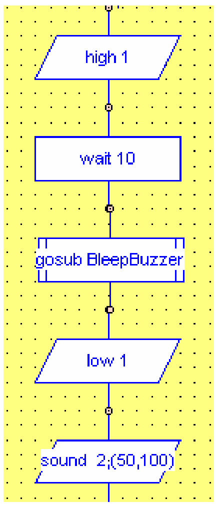


To finish the procedure you need to join the various boxes together.



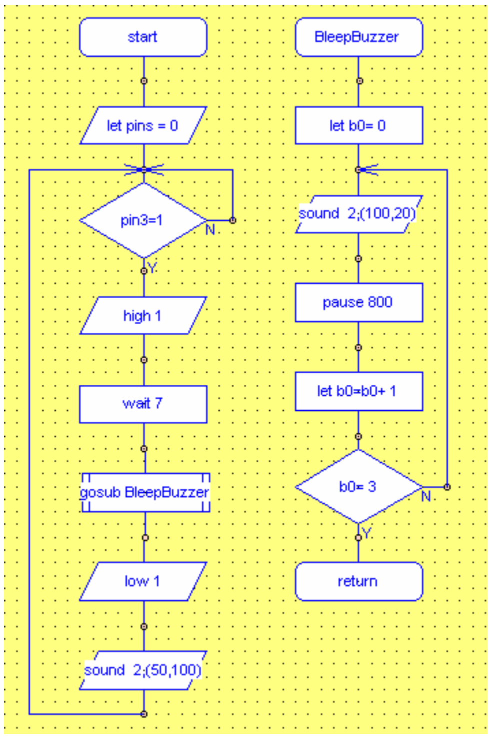
The procedure is now written, however it won't be run unless we tell the software to do so.

- ***Go back onto the 'sub' flowchart buttons.***
- ***Add a 'gosub' box into the gap made in the main flowchart.***
- ***On the bottom toolbar select 'BleepBuzzer'***



Finally you will need to reconnect the 'sound' back to the if 'pin3'. Also to keep the overall delay to 10 seconds adjust the 'Wait 10' in the main flowchart from 10 to 7 seconds.

Check your flowchart looks like this one:



You're ready to simulate.

- ***Press the yellow 'simulate' button.***
- ***Press input 3 to simulate the start switch being pressed.***
- ***Make sure your software waits 7 seconds, bleeps 3 times then after a total of ten seconds bleeps a time up sound.***

You can now program your board, just make sure its connected and powered before pressing the program button.

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.

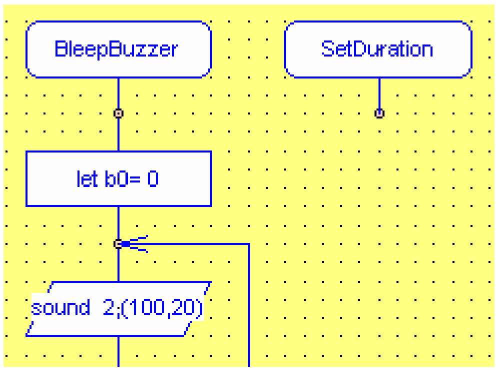
Whilst the switch remains pressed this part of the flowchart will sound a short beep every second and increment a count.

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.

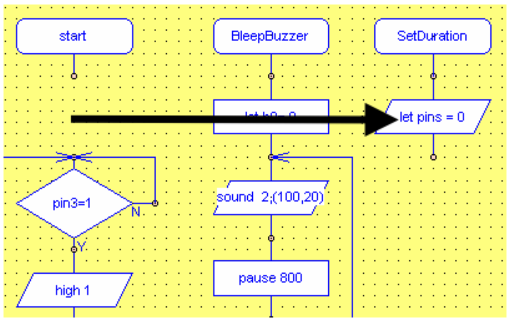
In place of the single 'Wait' a count of 'Wait 10' commands will be maintained, when this matches the stored count the buzzer can sound.

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 'BleepBuzzer' subroutine.

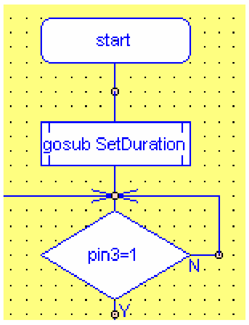
- *Select 'sub' off the 'sub' buttons on the top toolbar.*
- *Place a 'sub' box onto the flowchart.*
- *Edit the text in the bottom menu bar to 'SetDuration'.*



Move the 'let pins = 0' command under the 'Start' to the right (underneath the SetDuration subroutine).



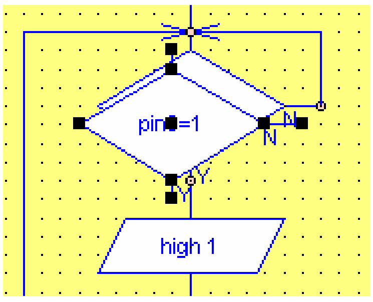
- *Add a 'gosub' Box onto the flowchart below 'Start'.*
- *On the bottom tool bar select the subroutine to call of 'SetDuration'.*



This subroutine reuses existing functionality, which can be copied (rather than re-entered). To cut and paste the 'if command' that checks the state of pin 3:

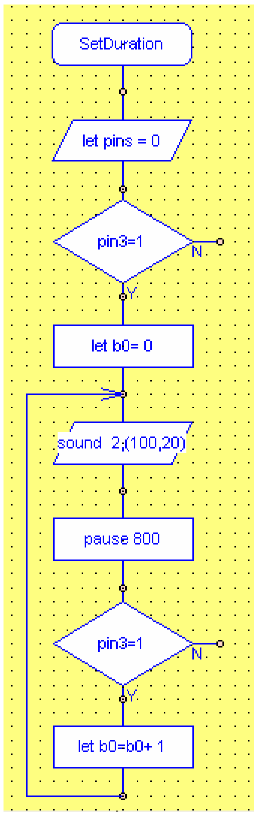
- *Press and hold 'Ctrl'.*
- *Left click on the box to be copied.*
- *Move the duplicated box to the right place.*

(Note if you already have a flowchart box selected, the duplicated item is placed on top of the current flowchart box.)



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

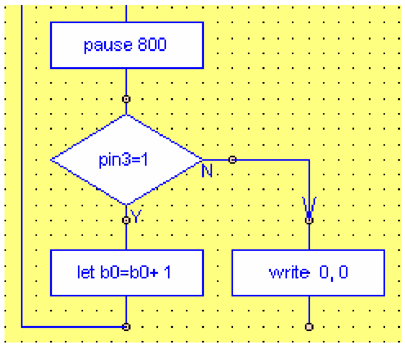
You will need the if pins 3 = 1 decision twice, as well as variable operations, sounds and a pause.



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

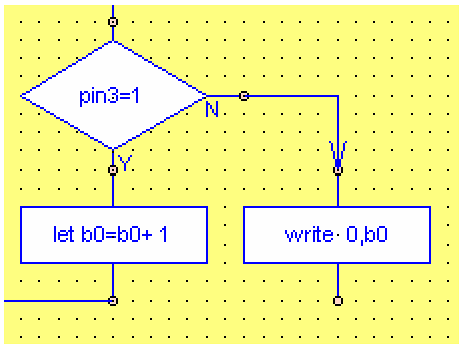
The next step is to store this in the secure memory (called EEPROM):

- ***Select the 'other' buttons on the top toolbar.***
- ***Select 'write'.***
- ***Add a 'write' to the bottom right of the flowchart.***
- ***Connect this to the 'if' command (as shown).***



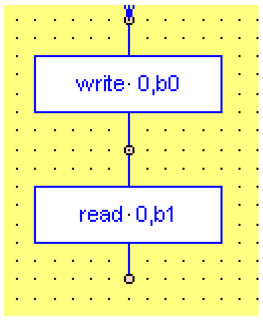
To configure the 'write':

- ***Select the write box and the details will be shown on the bottom toolbar.***
- ***The first box is the EEPROM location and the second box is the variable location.***
- ***Set the first box to 0 and the 2nd box to b0.***



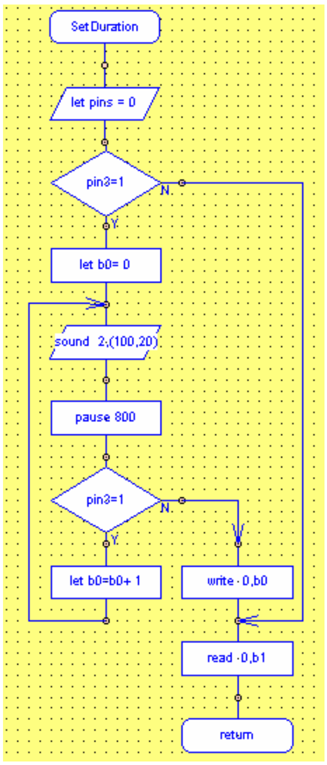
Because the variable b0 is used in other parts of the program we are going to read the EEPROM back into variable b1.

- ***Add a 'read' box onto the chart.***
- ***Configure this to read from EEPROM 0 and store in variable b1 (read 0, b1).***



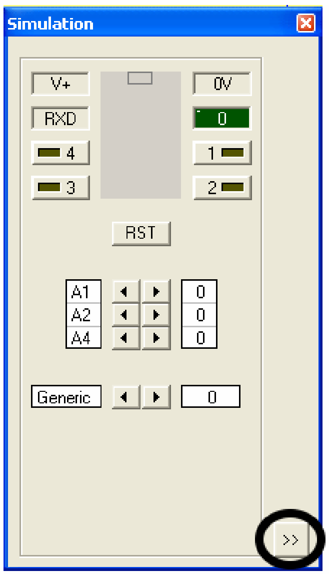
This is the last part of the function.

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



Let's check this part works:

- ***Press the yellow simulate play button.***
- ***On the simulation window expand the options that are shown, by pressing the '>>' button.***



We need input 3 to be set when the program starts:

- ***Press 'input 3', to set the input high.***
- ***Press 'RST', to reset the program.***

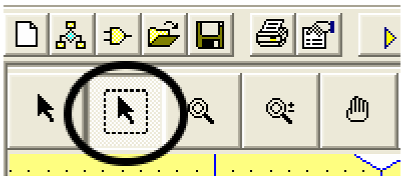
Watch the 'Variables' panel. In this you will see that variable b0 goes up every time the 'let b0 = b0 + 1' box is run.

Release 'Input 3', as the function finishes you will see:

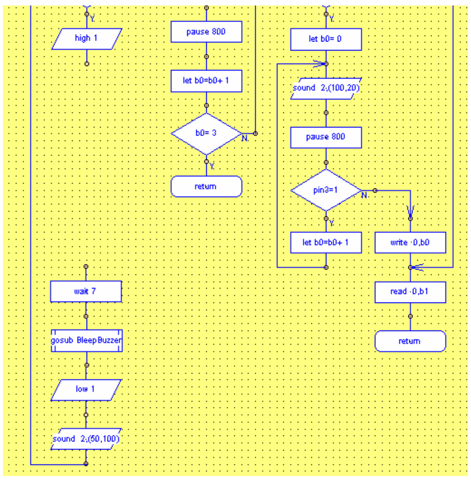
- **Variable b0 is stored in EEPROM.**
- **The EEPROM is copied to variable b1.**
- **You can now stop the simulation.**

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

- ***On the main toolbar select the select area button.***



- ***Drag a rectangle around the bottom four boxes on the main flowchart (wait 7 to the sound 2).***
- ***The four boxes with be highlighted with little squares on the corners & connections.***
- ***Drag and drop all four boxes so that the 'wait 7' is just below where the sound box was.***

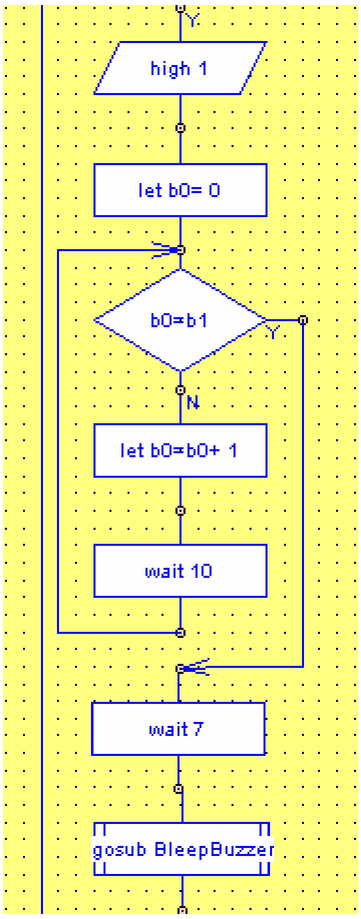


Add the following functionality in the space you have made, below the 'Outputs' command where the LED is turned on.

- **Set variable b0 = 0.**
- **Compare b0 to b1.**
  - **If equal run the ‘wait 7’ box.**
  - **Otherwise, add 1 to b0, wait 10 seconds, loop back to comparing b0 to b1.**

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, so you don't need to run that part of the program.

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

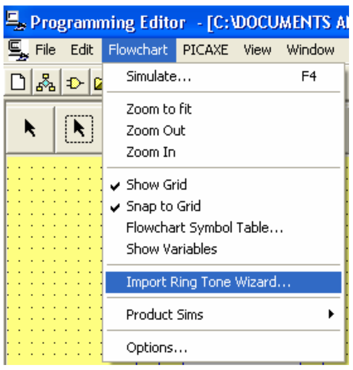
Once you are happy download the software into your board and test.

Well done you have completed task 3!

Task 4

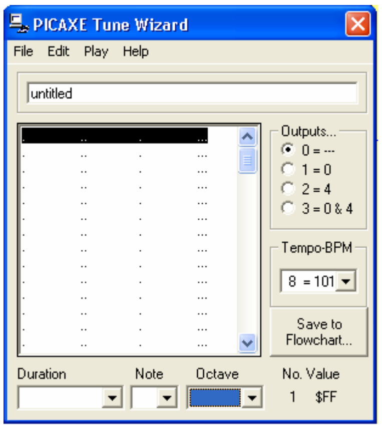
In this task you will get your timer to play a short tune when the time is up. To do this you will need to be using a PICAXE08M chip.

Before you can change the flowchart you will need to import the ring tone you would like to use. You can use any mobile phone ring tone that has been saved in the 'Ringing Tones Text Transfer Language (RTTTL)'.


Programming Editor - [C:\DOCUMENTS A
File Edit Flowchart PICAXE View Window
Simulate... F4
Zoom to fit
Zoom Out
Zoom In
Show Grid
Snap to Grid
Flowchart Symbol Table...
Show Variables
Import Ring Tone Wizard...
Product Sims
Options...

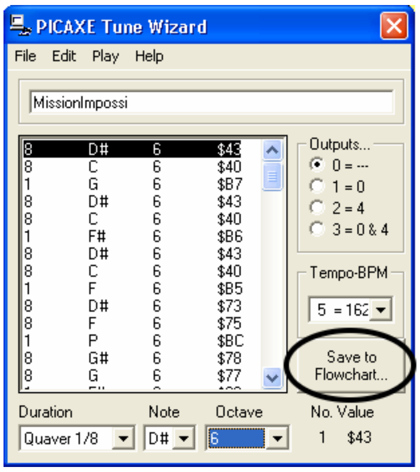
- ***Select 'Flowchart' on the top menu.***
- ***Then select 'Inport Ring Tone Wizard…'***

The following tune wizard will be shown:



- ***Select 'File'***
- ***Select 'Import ringtone…'***
- ***Navigate to the ring tone you want to use.***

The Tune wizard now displays the details of your ring tone.



MissionImpossi

8	D#	6	$43
8	C	6	$40
1	G	6	$B7
8	D#	6	$43
8	C	6	$40
1	F#	6	$B6
8	D#	6	$43
8	C	6	$40
1	F	6	$B5
8	D#	6	$73
8	F	6	$75
1	P	6	$BC
8	G#	6	$78
8	G	6	$77

Outputs...
[✓ 0 = ---]
1 = 0
2 = 4
3 = 0 & 4

Tempo-BPM: 5 = 162

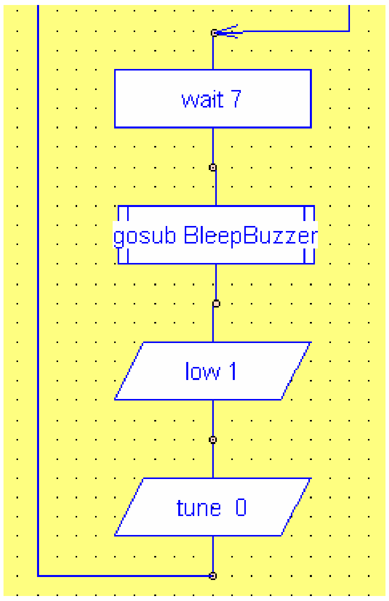
Duration	Note	Octave	No.	Value
Quaver 1/8	D#	6	1	$43

- ***Finally press the 'Save to Flowchart…' button.***

Your tune is now associated with this flowchart and ready to play. You need to remove the sound and add a tune:

- ***Remove the 'sound 2;(50,100)' at the end of the main part of the flowchart.***
- ***On the 'Out' flowchart toolbar, select 'Tune'.***
- ***Add a tune where the sound used to be.***
- ***The options in the bottom toolbar control if any of the outputs flash whilst the tune plays. This will be set to zero and doesn't need changing.***
- ***Finally reconnect the route from the 'tune' box to the top of the program (just after go sub SetDuration)***

Your flowchart should now look like this one:



You can now simulate the software or just go straight to programming up 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 Revolution Education, developers of the PIC programming editor software.

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