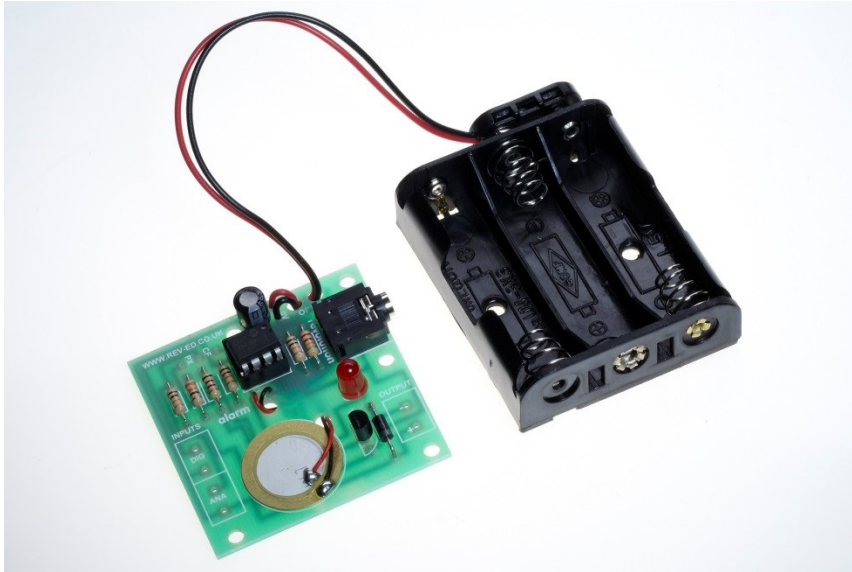


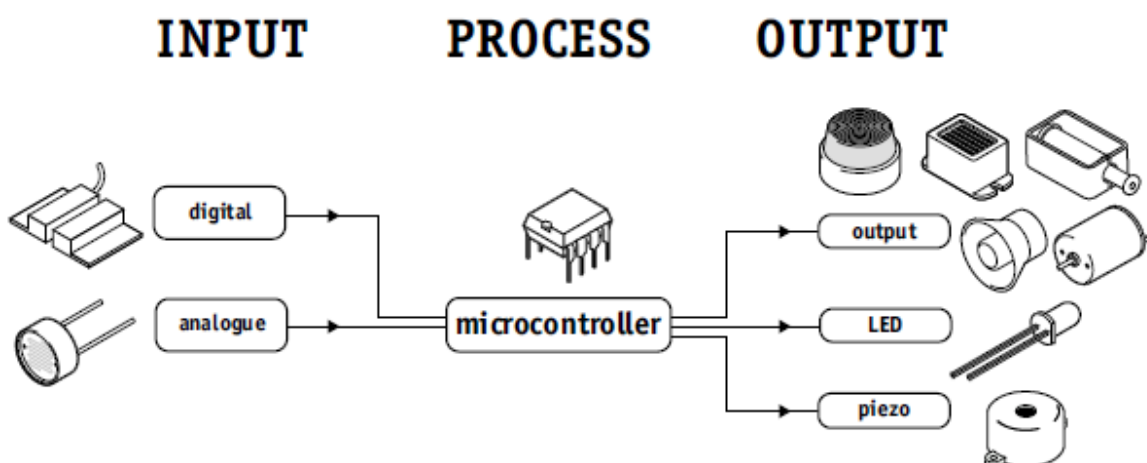
AXE102 PICAXE-08M2 Alarm / Steady Hand Kit



The alarm project kit uses a PICAXE-08M2 microcontroller with 3 available outputs, an LED a piezo sounder and a transistor driven spare output (e.g. buzzer, solenoid or motor).

The project also has two inputs – a digital input such as a switch and an analogue input such as a Light Dependent Resistor (LDR) so that the project can tell whether it is light or dark.

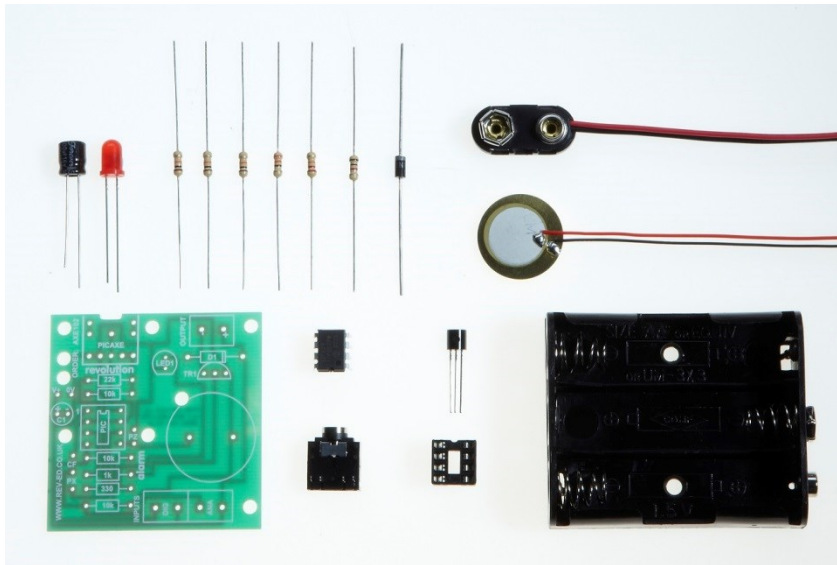
C.0 output	LED
C.1 input	analogue input e.g. LDR
C.2 output	piezo sounder
C.3 input	digital input e.g. switch
C.4 output	buffered output e.g. buzzer, motor or solenoid



Remember not to confuse the chip 'leg' number with the input/output pin number!

For the full datasheet please see www.picaxe.com/docs/axe102.pdf

1.0 Kit Contents



3	R1,R2,R6	RES-10K	10k resistor	(brown black orange gold)
1	R3	RES-22K	22k resistor	(red red orange gold)
1	R4	RES-330	330R resistor	(orange orange brown gold)
1	R5	RES-1k	1k resistor	(brown black red gold)
1	LK1		wire link	(use offcut resistor leg)
1	LED1	LED001	5mm red LED	
1	PZ	SPE003*	uncased piezo sounder	
1	TR1	TRT001	BC548B transistor	
1	D1	RES041	1N4001 diode	
1	C1	CAP006	100uF electrolytic capacitor	
1	IC1	ICH008	8 pin IC socket	
1	IC1	AXE007M2	PICAXE-08M2 microcontroller	
1	CT1	CON039	PICAXE download 3.5mm socket	
1	BT1	BAT016	battery clip	
1	BT1	BAT013	4.5V (3xAA) battery box	
1	PCB		alarm printed circuit board	

* The cased piezo SPE001 may be purchased separately if preferred to uncased SPE003.

Not supplied:

Loose wire (not supplied, if connecting LEDs and LDR by wires)

Digital input e.g. Switch

Ananlogue Input e.g. LDR (SEN002)

Tools required:

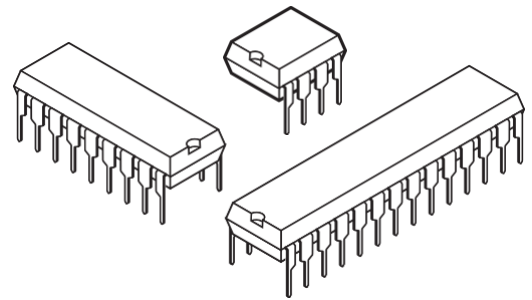
soldering iron and solder, side cutters

For the full please datasheet see www.picaxe.com/docs/axe102.pdf

2.0 What is a microcontroller?

A microcontroller is often described as a 'computer-on-a-chip'. It can be used as an 'electronic brain' to control a product, toy or machine.

The microcontroller is an integrated circuit ("chip") that contains memory (to store the program), a processor (to process and carry out the program) and input/ output pins (to connect switches, sensors and output devices like motors).



Microcontrollers are purchased 'blank' and then programmed with a specific control program. This program is written on a computer and then 'downloaded' into the microcontroller chip. Once programmed the microcontroller is built into a product to make the product more intelligent and easier to use.

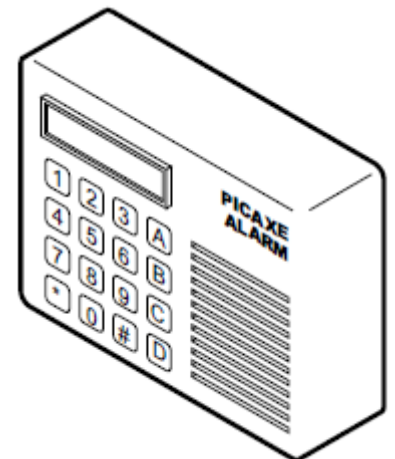
2.1 Example use of a microcontroller.

Almost all modern buildings are fitted with some type of alarm. For instance a fire detection system may have a number of smoke sensors to detect the smoke from a fire.



However many alarm systems are also safety systems - for instance an alarm system on an oil rig may monitor the temperature and pressure of the crude oil as it is being extracted and automatically shut the system down if a fault is detected. This ensures the safety of both the workers and the environment around the oil rig.

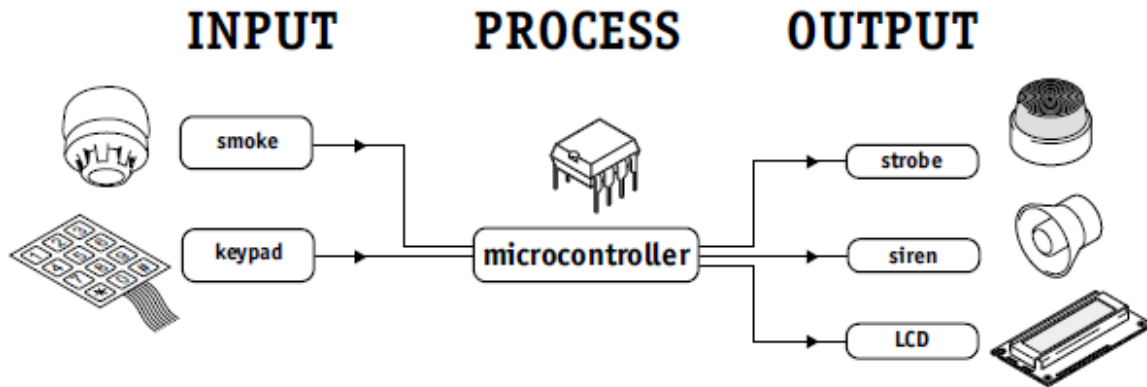
All systems are made up of input and output devices. Often these devices are connected to a microcontroller that interprets the information from the sensors and switches the outputs on and off at the correct time.



In the case of a fire alarm system the inputs may be smoke sensors and the keypad on the front of the control panel. The output devices are the display on the control panel as well as the external siren and strobe light.

The microcontroller is the 'brain' of the system. Microcontrollers are powerful electronic components that have a memory and can be programmed to switch things on and off in a special sequence. The microcontroller in the fire alarm, for instance, has been programmed to switch the siren on and off when the smoke sensor has detected fire.





In the fire alarm the smoke sensor and keypad provide information to the microcontroller, and so these are known as 'inputs'. The microcontroller then 'decides' how to behave and may then operate the outputs e.g. make the siren and strobe switch on or display a message on the Liquid Crystal Display (LCD).

3.0 Alarm Design Brief

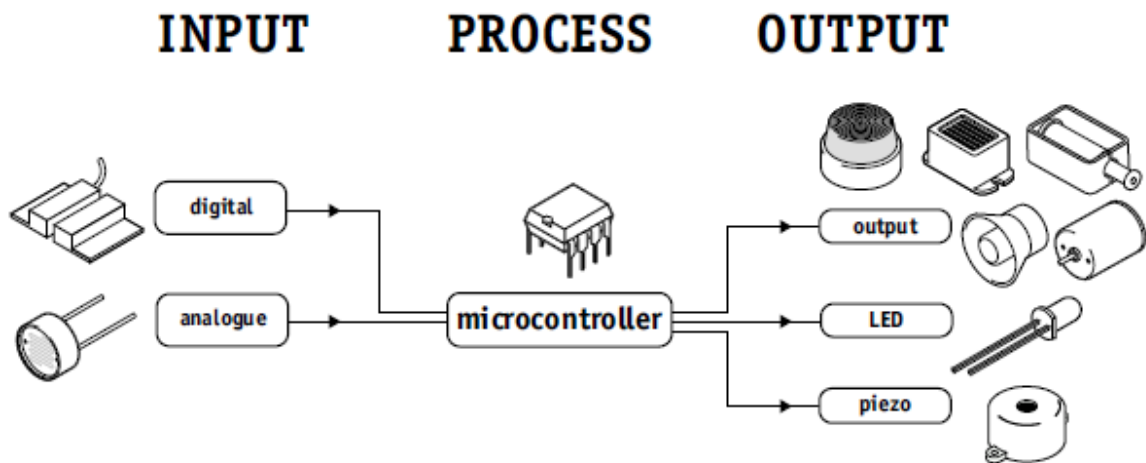
Design and make an alarm system. The alarm must be programmed to react to inputs and sensors.

3.1 Design Specification Points

- 1) The design will use a PICAXE-08M2 microcontroller as its brain.
- 2) The design will include an LED indicator, a piezo sounder to generate noises and an alarm output that could be a siren or motor.
- 3) The design will be also be able to react to analogue sensors such as light sensors.

3.2 Block Diagram

The block diagram for your project may look like this:

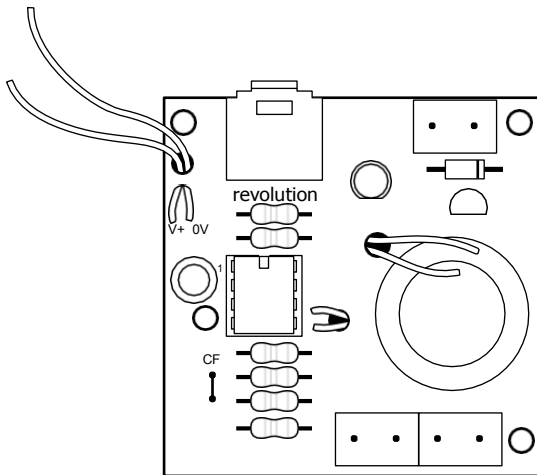


3.3 Designing Your Own Alarm

Your alarm may be for any purpose you choose. Some examples are given below.

- 1) A fire alarm. A light sensor is used to detect smoke. When smoke is detected a siren sounds.
- 2) A burglar alarm. When a tripwire is activated a flashing stobe light is activated. However the alarm is disabled during daylight by a light sensor.
- 3) A bank safety lock. When a 'panic' alarm switch is pushed an electronic solenoid bolt locks a bank safe door.
- 4) A baby monitor for a bedroom. When movement or sounds are NOT detected, a warning buzzer is activated.
- 5) A steady hand game. A 'switch' is made from a wire loop and a buzzer / LED activate when the handle touches the wire loop.

3.4 Assembly



How you build your Alarm PCB will depend on the shape and size of your design. You may solder all the components directly onto the board, or you may connect some of the components (e.g. the LED) by longer pieces of wire so they can be fitted inside your project. These instructions presume you are soldering all the components directly on the board. The instructions are identical if you are using longer wires to join some components, although you must be even more careful you get the wires around the correct way on the LEDs

- 1) Place the 1k (brown black red gold) resistor in position. Bend the legs to hold the resistors in position and then solder.
- 2) The bottom resistor marked 1k may be a 1k or 10k resistor depending on the type of analogue sensor (e.g. LDR) used. Use 10k (brown black orange gold) for a miniature style LDR (part SEN002). Bend the legs to hold the resistors in position and then solder.
- 3) Place the two 10k (brown black orange gold) resistors in position. Bend the legs to hold the resistors in position and then solder.
- 4) Place the 22k (red red orange gold) and 330 (orange orange brown gold) resistors in position. Bend the legs to hold the resistors in position and then solder.
- 5) Using an off cut resistor leg, make a wire loop over the letters PX marked beside the 330R resistors. Solder in position. (Ignore the hole above the holes marked CF).
- 6) Solder the 1N4001 diode in position D1, making sure the silver bar is pointing towards the right hand side of the PCB.
- 7) Push the PICAXE stereo download socket onto the PCB and make sure it clicks into position (so that it lies flat on the board). Solder the five metal square contacts (the five round plastic support post holes do not have to be soldered). Do not worry if the solder joins on the two metal contacts either side of the socket as they are supposed to be joined anyway.
- 8) Push the IC socket into position. Make sure the notch at one end points up towards the stereo download socket. Fold the legs over to hold the socket in position and then solder.
- 9) Solder the LED into position. Make sure the flat on one side of the LED aligns with the flat marked on the PCB.
- 10) Solder the BC548B transistor in position, making sure the flat side aligns with the marking on the PCB.
- 11) Solder the 100uF capacitor in position, making sure the + and – legs are the correct way round.
- 12) Thread the battery clip down through the large hole by the letters AXE. Thread it back up through the large hole by the letters 101. Then solder the black wire in the hole marked 0V and the red wire in the hole marked V+.



- 13) Use half a sticky pad to stick the SPE003 piezo sounder (brass side) to the top of the PCB. Thread the wires down through the hole below LED1 and back up through the hole marked PZ. Solder the red wire into the bottom hole and the black wire into the top hole. It does not matter if the red wire solder joint joins to pin5 of the IC socket as they are supposed to join anyway. However the black wire should NOT join pin 6 of the IC socket. *If using the optional SPE001 cased piezo instead simply solder it directly in place (either way around).*
- 14) Carefully check the board to make sure there are no missed joints or accidental solder bridges.

The board is designed to be used with a 4.5V (3xAAA) battery pack. Do **NOT** use a 9V PP3 battery.

3.5 Testing your circuit.

Step 1 - Check the solder joints.

Check that all the joints are connected to both the pad and the wire, and that the wire is held firmly so that it does not 'wobble' when pulled. Also check that the solder does not accidentally bridge between two pads. This is most likely to happen on the LEDs, the LDR and on the piezo. On the stereo socket the two square pads close together on each side can be joined as they are already joined by a track on the board. However they must not be joined to the central round hole.

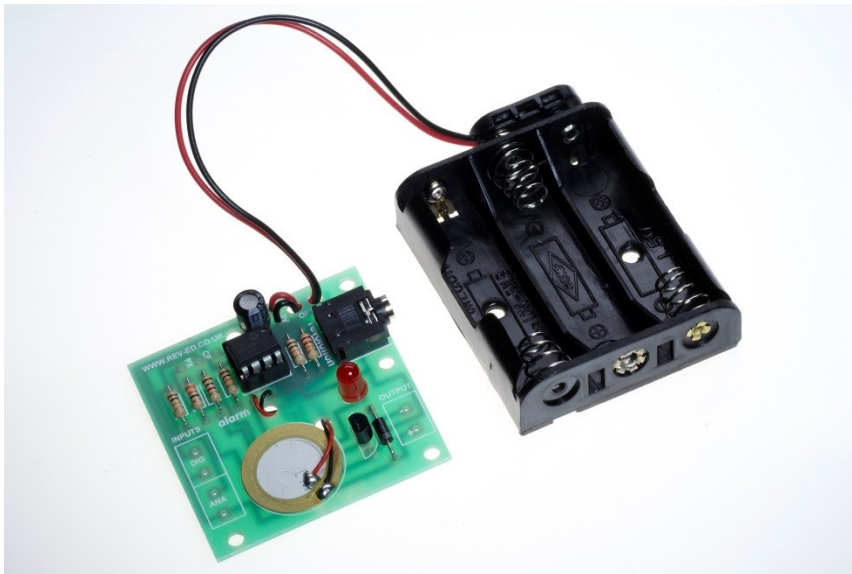
Step 2 - Check the components.

- 1) Check that the black battery clip wire is in the hole marked '0V' and the red battery clip wire is in the hole marked 'V+'
- 2) Check that the PICAXE-08M2 chip is in the socket correctly, with the dent (showing pin1) closest to the stereo download socket.
- 3) Check that the flat edge of the LED is connected to the correct hole on the PCB.
- 4) Make sure you have not forgotten the wire link over the holes marked PX at the bottom left of the board.
- 5) Make sure the brass side of the piezo is stuck down with a sticky pad.
- 6) Check that the socket is correctly soldered, including the middle square pad which is often forgotten by mistake.

Step 3 - Connect the battery.

Check the 3 AA batteries are in the battery box correctly. Connect the battery box to the battery snap and put your finger on the PICAXE chip. If it starts to get hot remove the battery box immediately as there is a problem – most likely that the chip or the battery snap wires are around the wrong way.

4.0 Example PICAXE programs

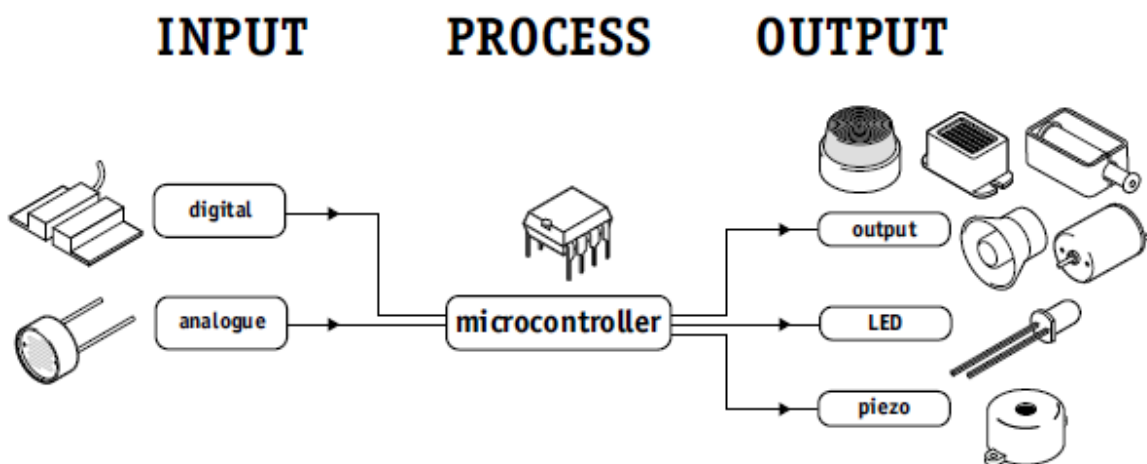


PICAXE-08M2 Input / Output Connections

C.0	output	LED
C.1	input	analogue input e.g. LDR
C.2	output	piezo sounder
C.3	input	digital input e.g. switch
C.4	output	buffered output e.g. buzzer, motor or solenoid

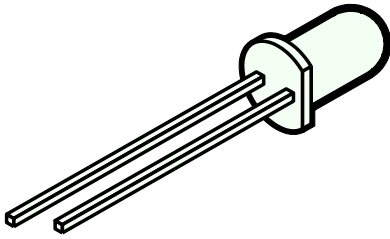
Examples are provided for Blockly, Flowchart and BASIC programming options.

To view the included samples files within 'PICAXE Editor 6' use the File>Open Samples menu and select 'AXE102 – Alarm'



4.1 Switching the LEDs on and off

4.1.0 What is an LED?



A Light Emitting Diode (LED) is an electronic component that gives out light when current passes through it. An LED is a special type of diode. A diode is a component that only allows current to flow in one direction.

Therefore when using a diode, it must always be connected the correct way around.

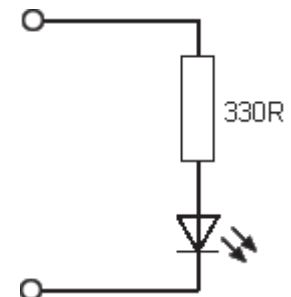
The positive (anode) leg of an LED is longer than the negative (cathode) leg (shown by the bar on the symbol). The negative leg also has a flat edge on the plastic casing of the LED.

4.1.1 What are LEDs used for?

LEDs are mainly used as indicator lights. Red and green LEDs are commonly used on electronic appliances like televisions to show if they are switched on or in 'standby' mode. LEDs are available in many different colours, including red, yellow, green and blue. Special 'ultrabright' LEDs are used in safety warning devices such as the 'flashing lights' used on bicycles. Infra-red LEDs produce infra-red light that cannot be seen by the human eye but can be used in devices such as video remote-controls.

4.1.2 Using LEDs.

LEDs only require a small amount of current to work, which makes them much more efficient than bulbs (this means, for instance, that if powered by batteries the LEDs will light for a much longer time than a bulb would). If too much current is passed through an LED it will be damaged, and so LEDs are normally used together with a 'series' resistor that protects the LED from too much current.



The value of the resistor required depends on the battery voltage used. For a 4.5V battery pack a 330R resistor can be used, and for a 3V battery pack a 120R resistor is appropriate.

4.1.3 Connecting the LED to a PICAXE microcontroller

Because the LED only requires a small amount of current to operate, it can be directly connected between the microcontroller output pin and 0V (with the series protection resistor).

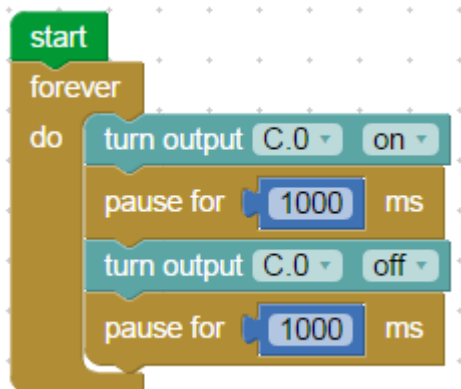
4.1.4 Testing the LEDs

This project board has an LED on: Output C.0

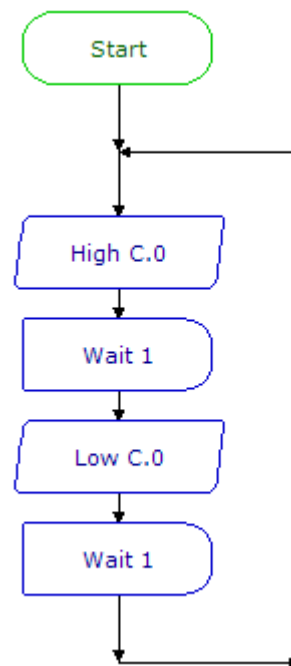
Therefore make sure the correct pin number is selected within each PICAXE command.

The following program will flash one LED on and off every second.

Blockly



Flowchart



BASIC

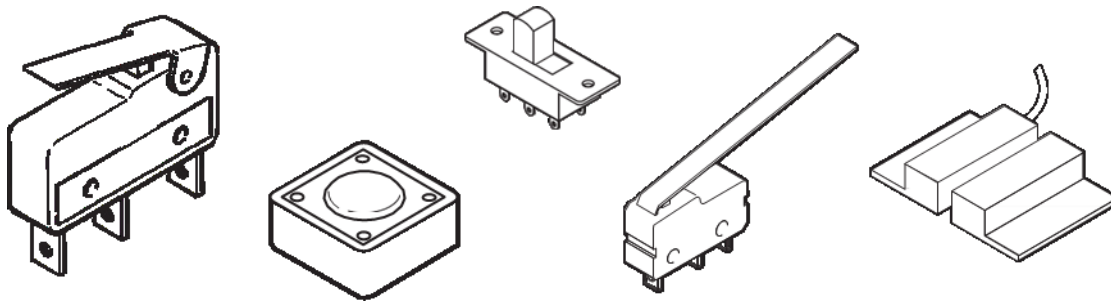
```

do
  high C.0      ; LED on
  pause 1000    ; wait 1 second
  low C.0      ; LED off
  pause 1000    ; wait 1 second
loop
  
```

If the LED does not work check:

1. the LED is connected the correct way around
2. the correct resistor is used
3. the correct output pin number is being used in the program
4. all the solder joints are good

4.2 Using the push switch



4.2.0 What are switches?

A 'switch' is a type of digital sensor, a sensor that can only be 'on' or 'off'. Switches detect movement.

There are a large number of different types of switches e.g.:

- push switches that detect a momentary 'push'
- micro-switches with long levers that detect small movements tilt-switches that detect jolting
- reed-switches that detect a magnet being moved

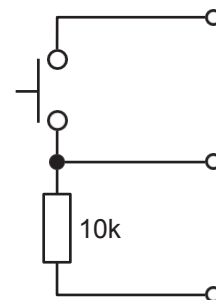
4.2.1 What are switches used for?

Push switches are commonly used on device like keypads. Micro-switches are used in burglar alarms to detect if the cover is removed from the alarm box. Reed switches are used to detect doors and windows being opened and tilt switches are often used to detect movement in devices such as toys, hair-dryers and tool-box alarms.

4.2.2 Using switches with a PICAXE microcontroller

A switch is often used with a resistor as shown in the diagram. The value of the resistor is not that important, but a 10k resistor is often used. When the switch is 'open' the 10k resistor connects the microcontroller input pin down to 0V, which gives an 'off' (logic level 0) 0V signal to the microcontroller input pin.

When the switch is activated, the input pin is connected to the positive battery supply (V+). This provides an 'on' (logic level 1) signal to the microcontroller.



4.2.3 Testing the switch

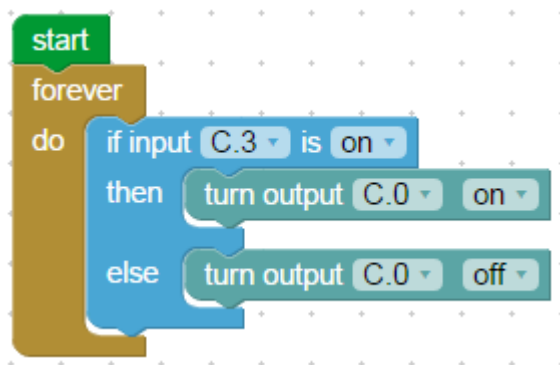
This project board has a digital input on: Input pinC.3

If you have not yet connected a switch you can simply use a loop of wire (or uncoated paperclip) to act as the switch.

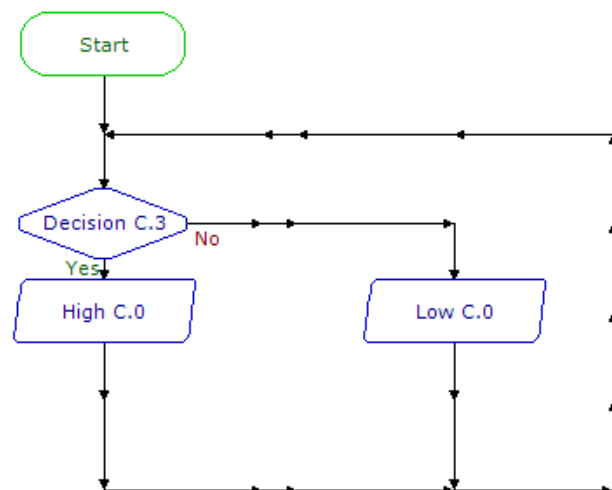
Therefore make sure the correct pin number is selected within each PICAXE command.

After connecting the switch it can be tested by a simple program like this. This program will switch an output on and off according to whether the switch is pushed or not.

Blockly



Flowchart



BASIC

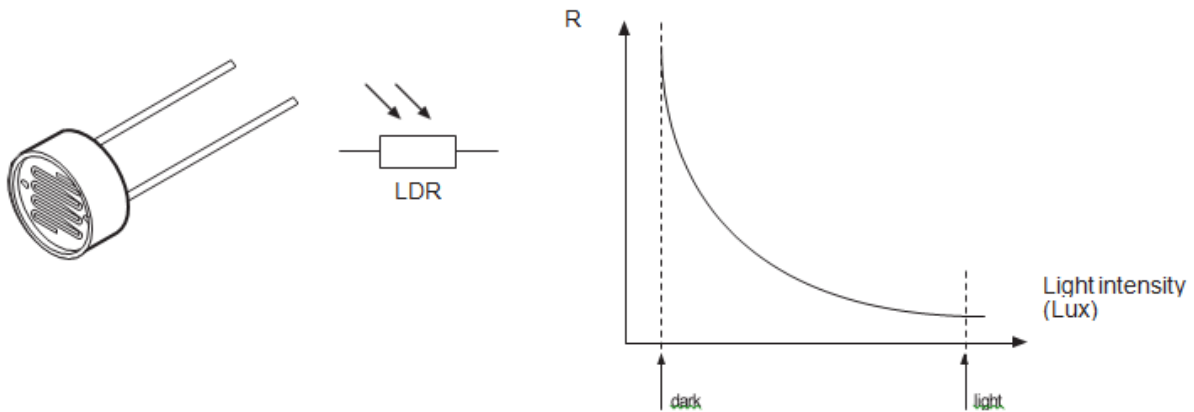
```

do
  if pinC.3 = 1 then      ; test pinC.3
    high C.0             ; LED on
  else
    low C.0              ; LED off
  end if
loop
  
```

4.3 Using the Analogue Input (e.g. using an LDR)

4.3.0 What is an LDR?

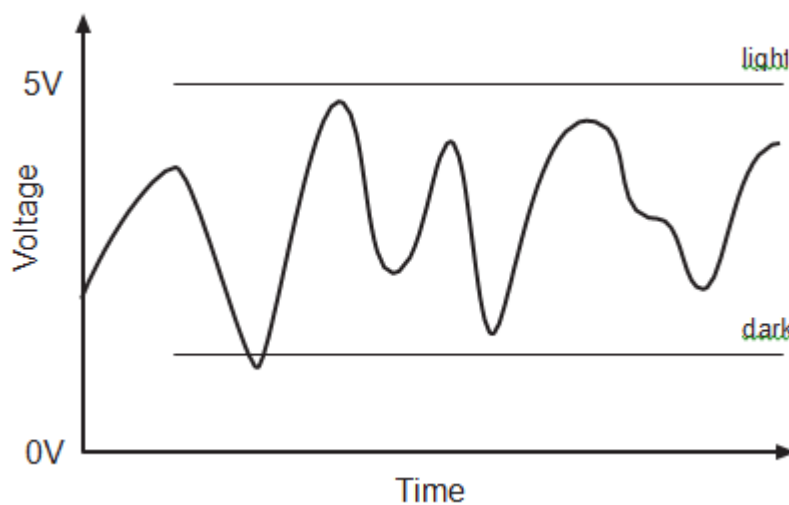
A Light Dependent Resistor (LDR) is a special type of resistor that reacts to changes in light level. The resistance of the LDR changes as different amounts of light fall on the top 'window' of the device. This allows electronic circuits to measure changes in light level.



4.3.1 What are LDRs used for?

LDRs are used in automatic street lamps to switch them on at night and off during the day. They are also used within many alarm and toys to measure light levels.

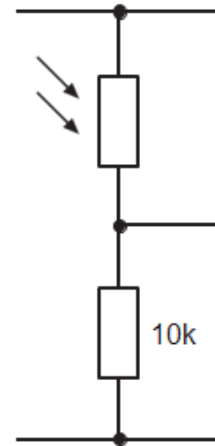
The LDR is a type of analogue sensor. An analogue sensor measures a continuous signal such as light, temperature or position (rather than a digital on-off signal like a switch). The analogue sensor provides a varying voltage signal. This voltage signal can be represented by a number in the range 0 and 255 (e.g. very dark = 0, bright light = 255).



4.3.2 Using an LDR with a PICAXE microcontroller.

The electronic circuit for using the LDR is known as a 'potential divider'. A suitable value for the lower resistor is 10k. The voltage at the centre position of the potential divider varies as the resistance of the LDR changes in different light conditions.

The microcontroller contains an 'analogue to digital converter (ADC)' which converts the varying voltage signal into a number between 0 and 255. This number can then be stored in a variable and compared against a desired 'threshold point'.



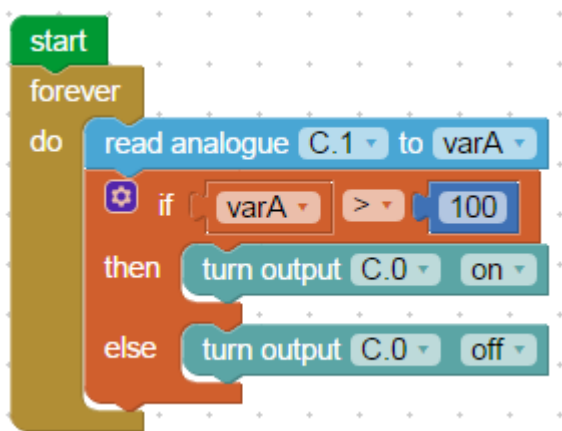
4.2.3 Testing the LDR

This project board has an analogue input on: Input C.1

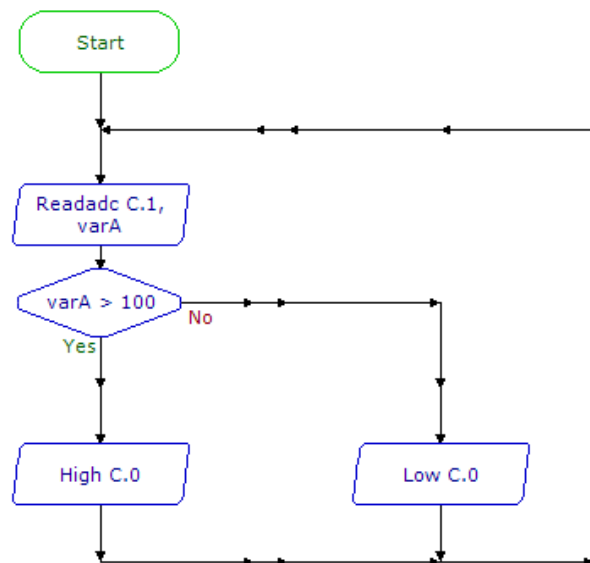
Therefore make sure the correct pin number is selected within each PICAXE command.

This program will switch an output on and off according to whether the light level is greater than 100. But how do you know 100 is the correct threshold value to use? See the next section...

Blockly



Flowchart



BASIC

```

symbol varA = b1
do
  readadc C.1, varA      ; read ADC value on C.1
  if varA > 100 then    ; test value
    high C.0           ; LED on
  else
    low C.0            ; LED off
  end if
loop
    
```

4.2.3 Calibrating the analogue sensor

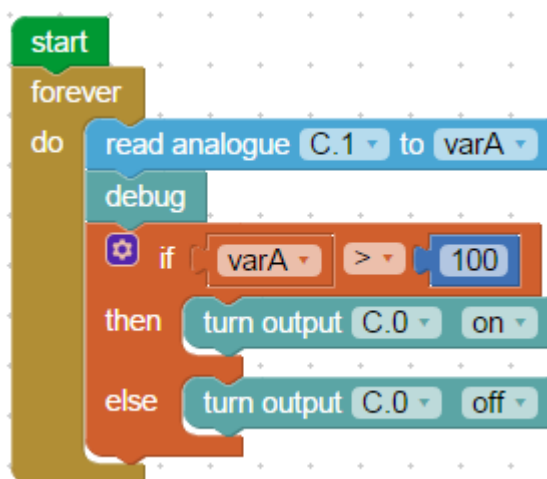
When using an analogue sensor you may not always know what the correct threshold value to use. The threshold value is normally a number that is half way between the lowest (darkest) and highest (brightest) analogue values possible.

Fortunately the PICAXE system has a special 'debug' command which can transmit the value of the variable to the computer screen. Using this command you can therefore run an experiment – for instance using your hand to cover the LDR to calculate the highest and lowest possible values.

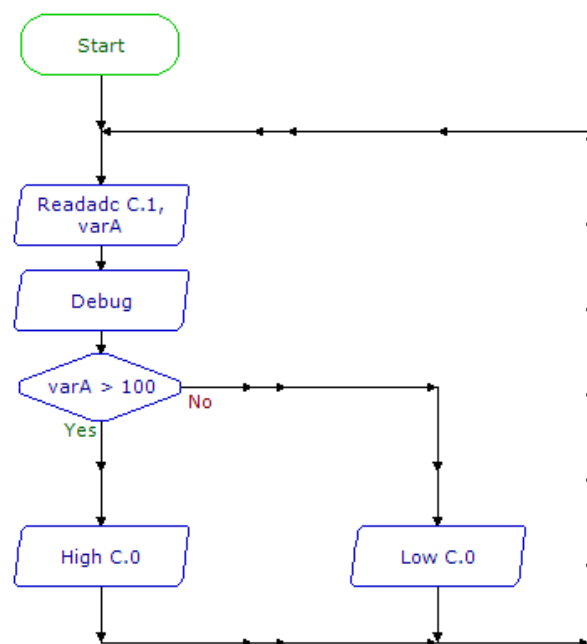
The threshold value to use is the value that is half way between these maximum and minimum values.

To try this out modify the previous program by inserting an extra 'debug' command (found in the Serial Commands section) after the readadc command.

Blockly



Flowchart



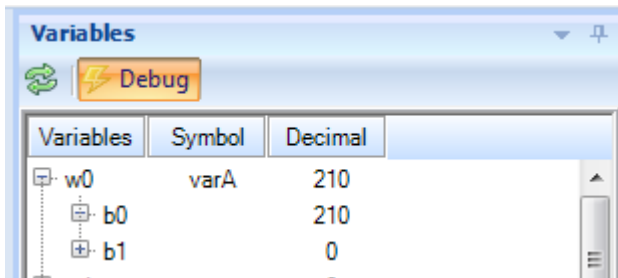
BASIC

```

symbol varA = b1
do
  readadc C.1, varA      ; read ADC value on C.1
  debug                  ; send value to computer
  if varA > 100 then     ; test value
    high C.0             ; LED on
  else
    low C.0              ; LED off
  end if
loop
  
```

After downloading this new program open the Debug panel on the computer screen. The Debug panel will then show the value of the light reading.

In **PICAXE Editor 6** the debug panel is found in the Code Explorer on the right hand side of the screen.



In the **Blockly Chrome App** the Debug panel can be opened from the PICAXE menu.

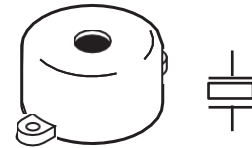
Blocks	PICAXE BASIC			Debug X	Javascript	XML
b0	211	\$D3	%11010011	w0	00211	\$00D3
b1	000	\$00	%00000000			
b2	000	\$00	%00000000	w1	00000	\$0000
b3	000	\$00	%00000000			

Once the experiment is complete select a new threshold value half way between the maximum and minimum values found. Then modify the original program to use this value rather than the value 100 that was originally used.

4.4 Using the Piezo Sounder

4.4.0 What is a piezo transducer?

A piezo transducer is a low-cost 'mini-speaker' that can be used to make sounds. The sound that the piezo makes can be changed by altering the electronic signals provided by the microcontroller.

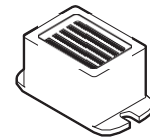


4.4.1 Where are piezos used for?

Piezos are used in many different consumer goods to provide 'feedback' to the user. A good example is a vending machine which will 'beep' whenever a keypad switch is pressed to select a drink or snack. The 'beep' provides the user with feedback to tell them their switch push has been successful. Uncased piezos are also often used in musical birthday cards to play a tune when the card is opened.

4.4.2 What is the difference between a piezo and a buzzer?

A buzzer contains a small electronic circuit that generates the electronic signal needed to make a noise. Therefore when a buzzer is connected to a battery it will always make the same sound. A piezo does not contain this circuit, and so therefore needs an external signal. This signal can be supplied by the output pin of a microcontroller. A piezo also requires less current to operate and so will last longer in battery powered circuits.



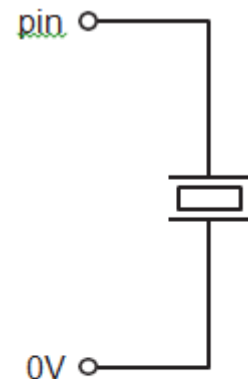
4.4.3 Using piezos.

A piezo is very simple to connect. Simply connect the red wire to the microcontroller output pin and the black wire to 0V (ground).

Note that the cheapest piezos do not have a plastic casing to them. In this case it is necessary to mount the piezo on a piece of board (with a sticky pad) to create a noise that can be heard. The board acts as a 'sound-box' to amplify the sound made by the piezo. For instance on a musical birthday card it is actually the whole card that vibrates and acts as the speaker, not just the piezo!

More expensive piezos have a round plastic case that acts as this sound-box. An example of this type of device is order code SPE001.

Make sure the sticky pad is stuck on the correct side of the piezo (the brass side without the wires!).



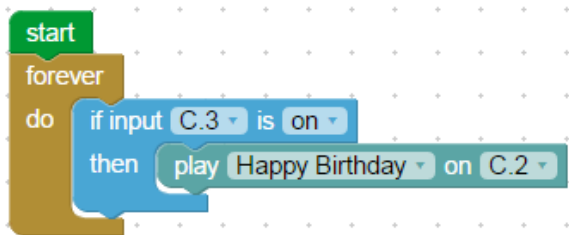
4.4.4 Testing the Piezo

This project board has a piezo on: Output C.2

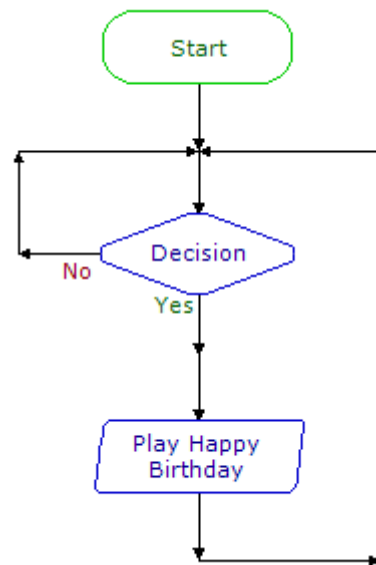
Therefore make sure the correct pin number is selected within each PICAXE command.

This program will play the tune 'Happy Birthday' on the piezo sounder when the switch is pressed.

Blockly



Flowchart



BASIC

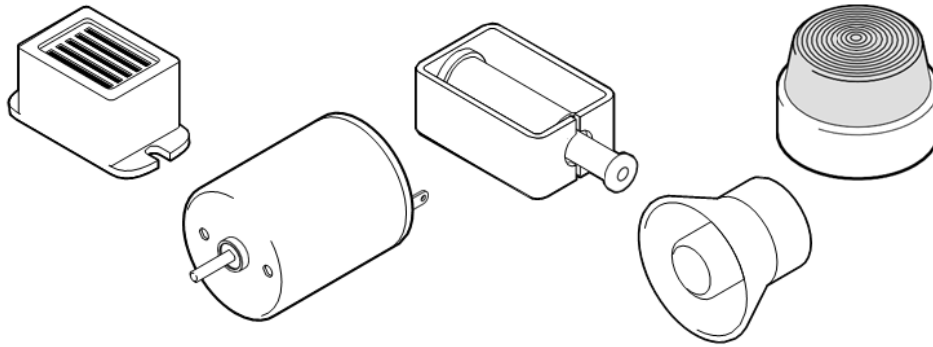
```

do
  if pinC.3 = 1 then
    play 0, 0
  endif
loop
    
```

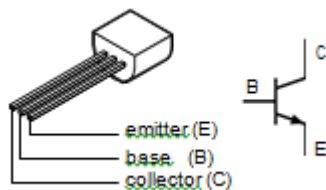
Other commands that can be used with the piezo sounder are **'sound'** to create beep type noises and **'tune'** to play other user defined RTTTL ringtone tunes. See the main PICAXE manuals for more details about these two extra commands.

4.5 Switching the Transistor on and off

4.5.0 What is a transistor?

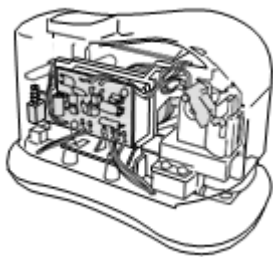


AXE102



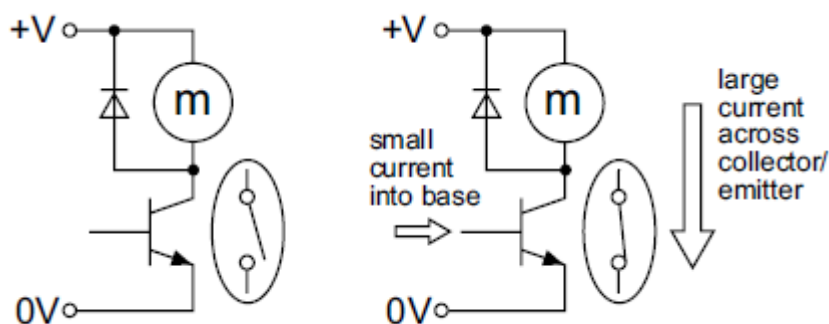
A transistor is a component that controls current flow in a circuit. A transistor acts as an 'electronic switch' so that a small current can control a large current. This allows low-current devices, like a microcontroller, to control large current devices (like motors).

4.5.1 What are transistors used for?



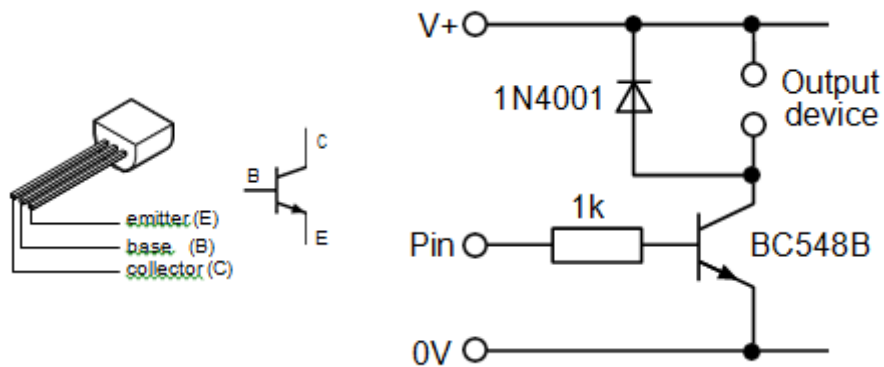
Transistors are used in radios and electronic toys and many other devices such as this electronic stapler. The transistor switches the motor inside the stapler on and off.

4.5.2 Using transistors



A transistor has three legs. These are labelled base, collector and emitter. The base connection is the leg that is used to activate the 'electronic switch'. When a small current is passed through the base connection, it allows a much larger current to flow down between the collector and emitter. This larger current can be used to switch on devices such as motors, lamps and buzzers.

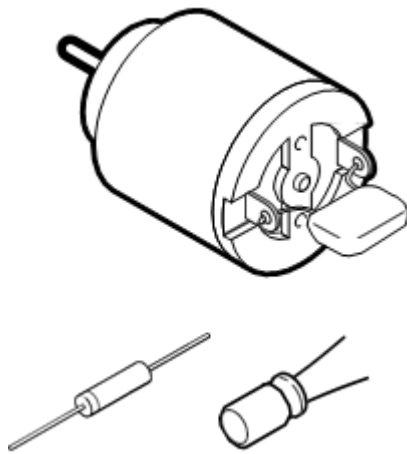
4.5.3 Connecting the transistor to a PICAXE microcontroller



A common transistor is the BC548B type. This has a plastic can with a flat edge. The flat edge enables the base, collector and emitter legs to be correctly identified.

4.5.4 Using output devices with a transistor

A transistor can drive many output devices such as a buzzer, motor or solenoid.



Motors can generate 'electrical noise' as they turn. This is because the magnets and electric coils inside the motor generate electrical signals as the motor rotates. These signals (the electronic noise) can disrupt the operation of the microcontroller. Some motors, like solar motors, produce very little noise whilst others can create a lot of noise.

To prevent electrical noise affecting the micro-controller circuit a 220nF capacitor should always be soldered across the motor terminals before it is used.

In addition, a diode (e.g. 1N4001 diode) should be connected alongside the motor. This prevents damage to the transistor as the motor slows down after the transistor switches off (for a short period of time (as it slows down and stops) the motor is acting as a 'dynamo' and generating electric current!). When connecting a diode make sure the 'band' is connected the correct way around. It is also a good idea to connect a 100uF electrolytic capacitor across the battery supply to help 'suppress' the electrical noise.

4.5.3 Testing the transistor

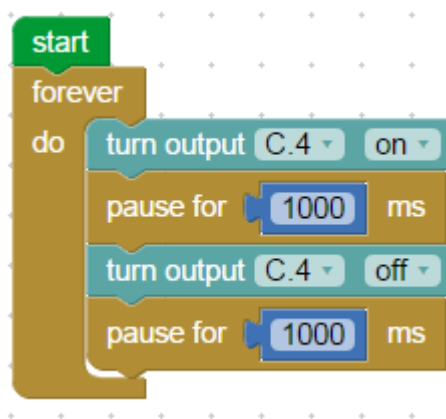
This project board has a transistor on: Output C.4

You may connect and output device such as a motor or buzzer to the output. If using a buzzer check that the red and black wires are around the correct way.

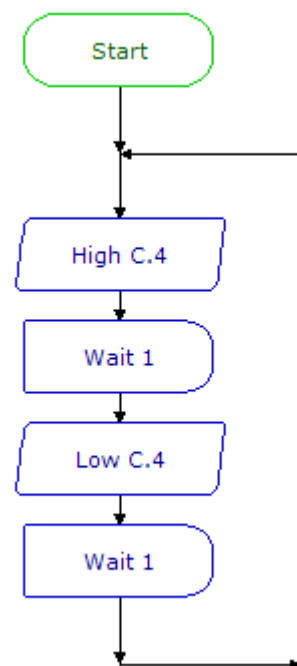
Therefore make sure the correct pin number is selected within each PICAXE command.

The following program will switch the transistor on and off every second.

Blockly



Flowchart

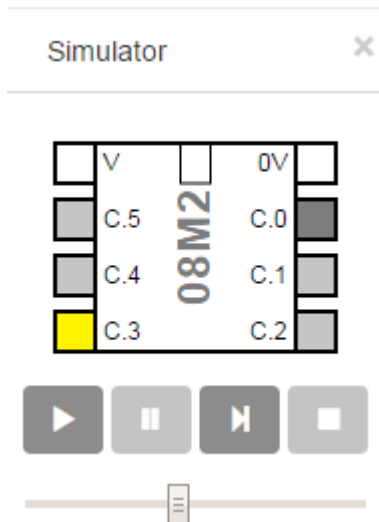


BASIC

```

do
  high C.4      ; transistor on
  pause 1000   ; wait 1 second
  low C.4      ; transistor off
  pause 1000   ; wait 1 second
loop
  
```

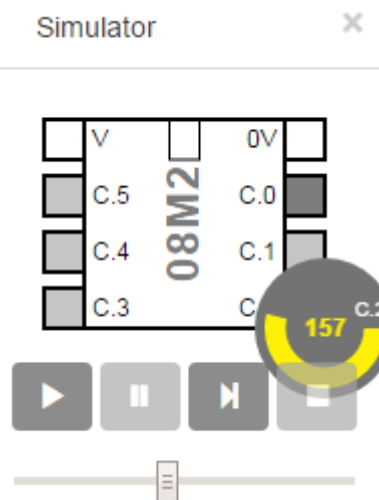
5.0 On-Screen Simulation



Bloky and PE6 both contain a simulation feature that allows you to 'try out' your programs on screen. The simulated chip leg changes colour when outputs are switched on and off, and you can also click on an input pin to change it from on to off and vice versa.

To start the simulation click the 'run' button in the Simulation tab/panel.

The speed that the simulation runs can be adjusted by the simulation speed slider (in PE6 this is at the bottom right of the screen).



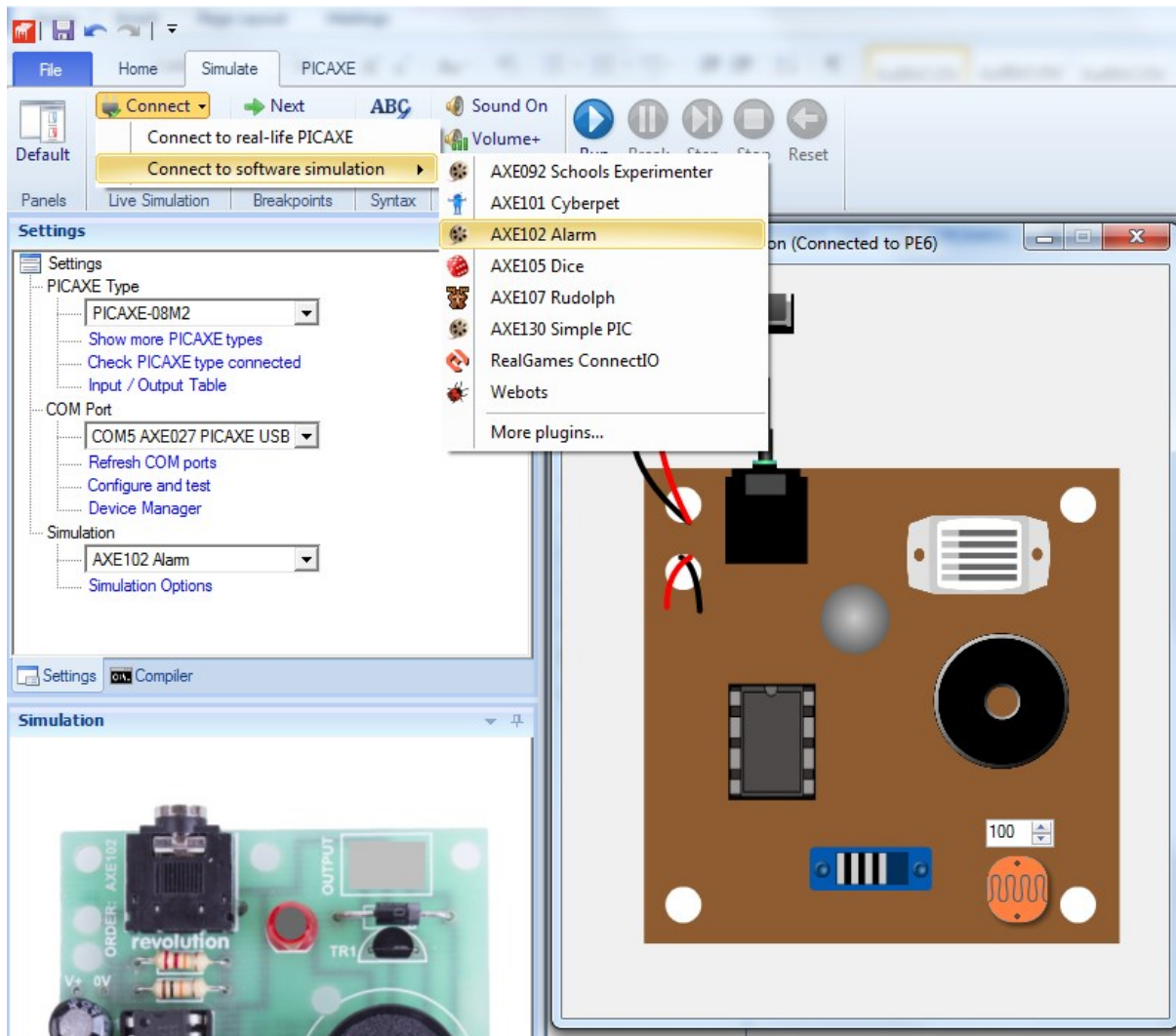
To simulate an analogue input like an LDR value right click over the pin and use the 'radial slider' to change the analogue value used within the program.



In PE6 it is also possible to change the default simulation layout (08M2 chip) to a picture of the project board instead. To do this change the simulation setting within the Workspace Explorer as shown here.

In PE6 it is also possible to 'connect' to a larger simulation model of the project kit. To do this select Simulate > Connect > Connect to software simulation > AXE102 Alarm

When the program is simulated you can then interact with the larger simulation model.



6.0 More Complex Example

This general purpose program has a main loop which flashes the LED on and off, and also checks the analogue sensor (fitted with a LDR) and digital input (fitted with a switch). When the push switch is pressed an alarm is sounded for two seconds.

If the LDR light sensor is covered the piezo will make a warning beep sound until the light level rises again.

In PE6 use **File > Open Samples > AXE102 Alarm** to open these sample files.

Blockly

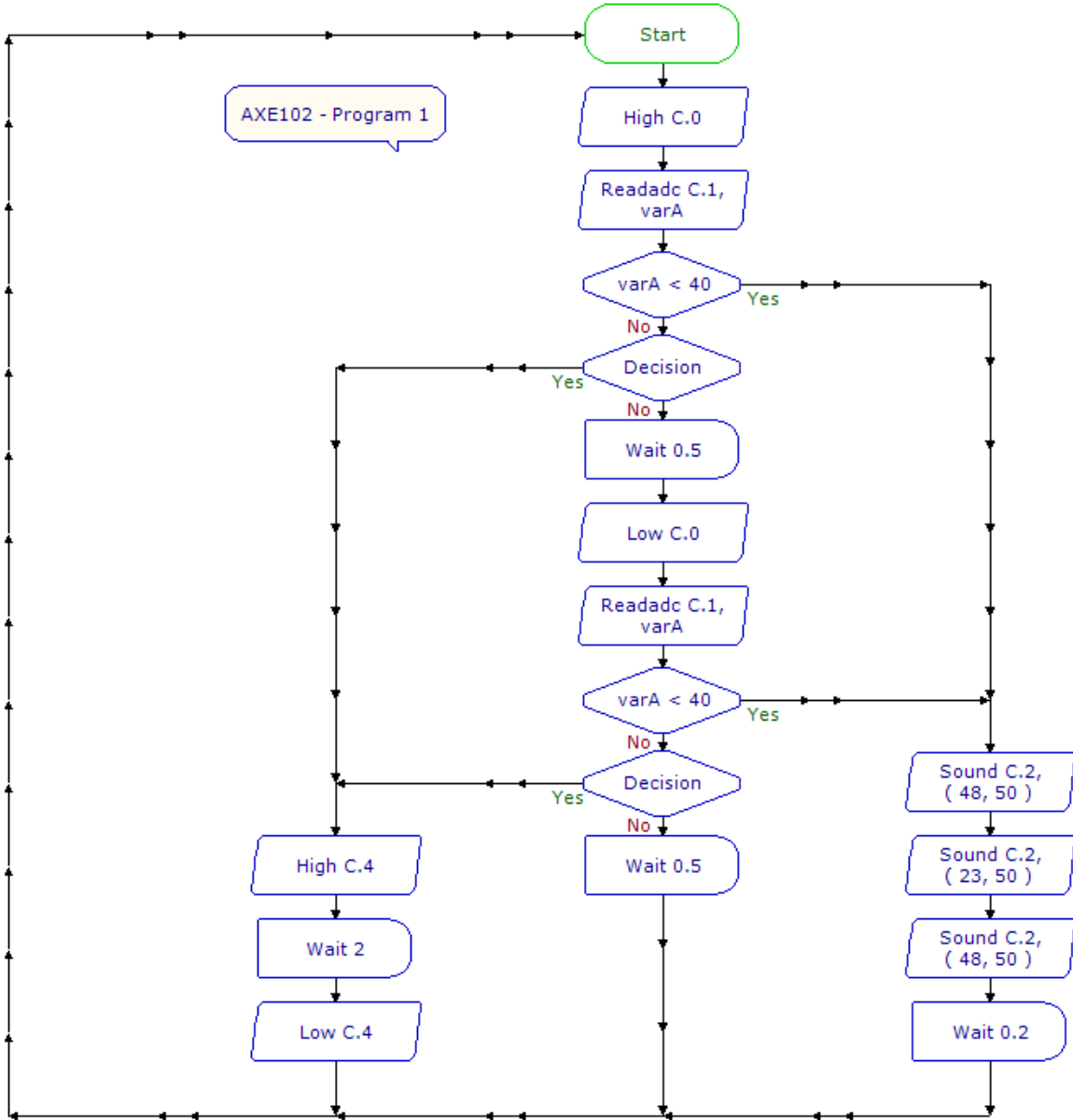
```

start
forever
do
  turn output C.0 on
  read analogue C.1 to varA
  if varA < 40
  then
    call do beep
  else
    if input C.3 is on
    then
      call do alarm
    else
      pause for 500 ms
      turn output C.0 off
      read analogue C.1 to varA
      if varA < 40
      then
        call do beep
      else
        if input C.3 is on
        then
          call do alarm
        else
          pause for 500 ms

to do beep
  play note 120 for 50 on C.2
  play note 80 for 50 on C.2
  play note 120 for 50 on C.2
  pause for 200 ms

to do alarm
  turn output C.4 on
  pause for 2000 ms
  turn output C.4 off
  
```


Flowchart



BASIC

```

symbol light = b1

' ***** main loop *****

' loop here flashing lights
' and checking sensors

main:

' LEDs full on and read light value
  high 0
  readadc 1,light

' if analogue value low then make sound
  if light < 40 then beep

' if switch pushed do alarm
  if pin3 = 1 then alarm

' do a delay
  pause 500

' LEDs full off and check sensor again
  low 0
  readadc 1,light

' if analogue value low then make sound
  if light < 40 then beep

' if switch pushed do alarm
  if pin3 = 1 then alarm

' do a delay
  pause 500

'loop back to start
  goto main

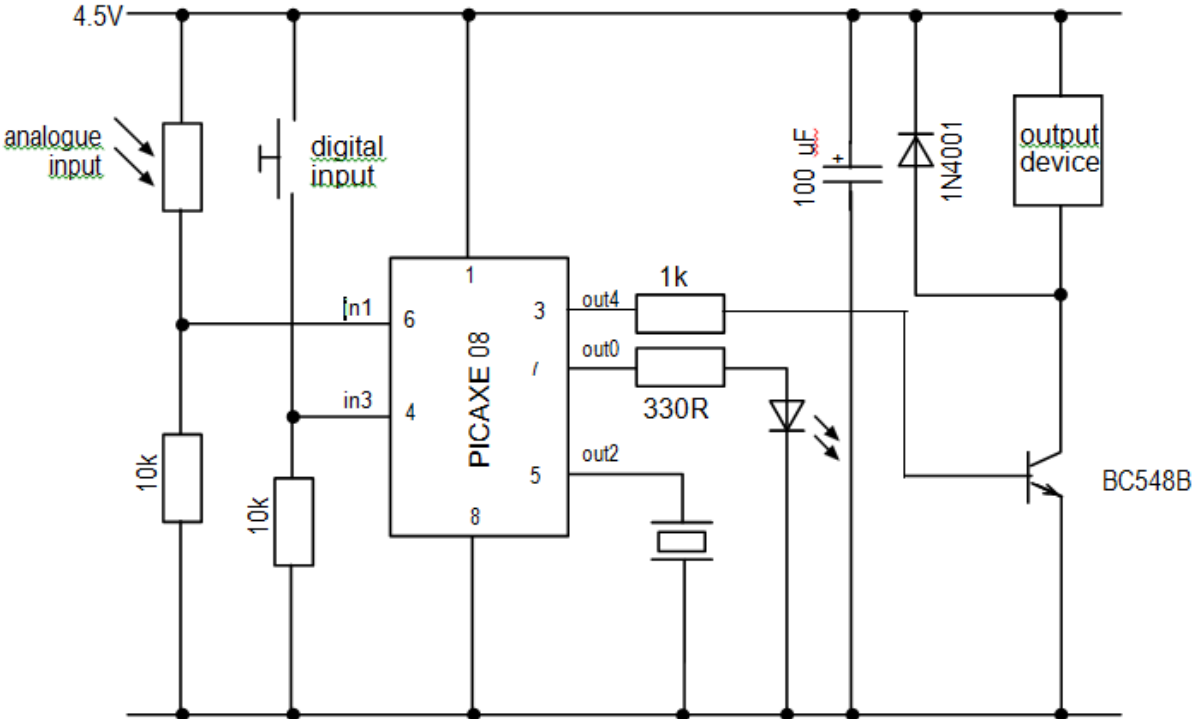
' ***** make sound *****
beep:
  sound 2, (120,50,80,50,120,50)
  pause 200
  goto main

' ***** alarm output on ****
alarm:
  high 4
  pause 2000
  low 4

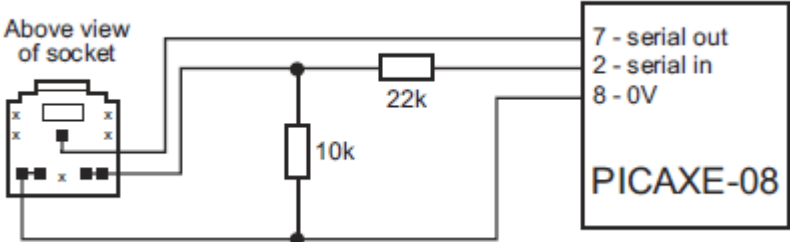
  goto main

```

7.0 Circuit Diagram



Download circuit:



Note that output C.0 is connected to both the LED and to the download circuit, so the LED will flicker during the programming process.