

Appendix H - Frequently Asked Questions (FAQ).

Where can I purchase PICAXE microcontrollers?

All microcontrollers can be purchased from within the PICAXE section of the online store at www.tech-supplies.co.uk or from our distributors (see www.picaxe.co.uk)

Which cable - serial or USB?

Many modern computers do not have a 9 pin legacy serial port and so we always recommend the USB download cable part AXE027. However the AXE026 serial cable is a more economical option for multiple computers that still have serial ports - e.g. in a school IT room.

There appears to be two PICAXE serial download cables - which should I use?

The standard serial PICAXE cable (part AXE026) ends with a stereo style 3.5mm plug. If making your own board we recommend this stereo cable cheaper as it is cheaper, better quality, and our sample PCB files use this connector (part CON039). The original PICAXE-28 cable (part AXE025) ended with a 3 pin in-line connector, but this cable is no longer used on any of our project boards or sample pcbs.

I've built a second pcb (without the download circuit) and the PICAXE program will not run!

If you program a PICAXE chip in a different board, and then move the chip to a board without the download circuit, you must ensure that the 'serial in' pin is tied to ground (0V) on the second board for reliable operation.

I've bought some blank PICs and they don't work in the PICAXE system!

The PICAXE microcontroller is not a blank PICmicro! It is a microcontroller that has been pre-programmed with a 'bootstrap' program that enables the download via the direct cable link (the bootstrap program tells the microcontroller how to interpret the direct cable programming commands). Therefore you must buy 'PICAXE' microcontrollers, rather than blank microcontrollers, to use with the PICAXE system. However we sell PICAXE microcontrollers at approx. the same price as blank devices, so there is very little price difference for the end user, particularly if you purchase the multi-packs.

I've programmed a PICAXE microcontroller using a conventional programmer and it will now not work in the PICAXE system!

You have overwritten, and hence deleted, the PICAXE bootstrap program (see above). The microcontroller can no longer be used as a PICAXE microcontroller, but you can naturally continue using it with your conventional programmer.

Can you reprogram microcontrollers (that I have accidentally erased) with the bootstrap program?

No. We do not accept microcontrollers from unknown sources due to the correct storage/handling procedures required by these devices. We use gang programmers costing several thousand pounds to program the bootstrap code into the blank microcontrollers, and so must protect this expensive equipment from damage. It is also likely that if we did offer this service the handling cost would end up more expensive than new PICAXE microcontrollers anyway!

Can you supply the bootstrap program so that I can make my own PICAXE?

No. The small royalty made on each PICAXE chip sold is the only financial benefit to our company to support the PICAXE system - the software is free and the cables/development kits are sold at very low cost. Therefore we do not allow anyone else to manufacture PICAXE microcontrollers.

Can I see the assembler code that is downloaded into the PICAXE?

If you own a Revolution Serial PIC Programmer, you can convert PICAXE BASIC programs into assembler code, to program blank PICs or to just learn how assembler code works by 'disassembly'. However some of the more complex commands (e.g. serin) are not supported, and the assembler code program generated is optimised for sequential learning (not optimised for compactness as with the PICAXE system) and so the code is not 100% identical to that downloaded to the PICAXE.

Can you alter the input/output pin arrangement of the PICAXE microcontroller?

The PICAXE-08/08M and X2 parts have configurable pins. The other parts have mainly fixed i/o, although some pins can be changed - see the appendices at the end of Manual part 1 for more details.

How long a program can I download into the PICAXE microcontroller?

This varies on the commands used, as not all commands use the same amount of memory. As a general rule you can download about
40-110 lines of code into the PICAXE-08/18
80-220 lines of code into the PICAXE-08M/14M/20M/18A/18M/28/28A
600-1800 lines of code into the PICAXE-18X/28X/40X
2000-3200 lines of code into the PICAXE-20X2/28X1/28X2/40X1/40X2
However some commands, such as sound and serout use more memory and so will reduce this count. In our experience most educational programs that are too long to download are generally badly composed, and can be greatly reduced in size by use of sub-procedures etc.

Do I need to erase the device?

How do I stop a program in the PICAXE microcontroller running?

Each download automatically overwrites the whole of the previous program. There is generally no need to erase the memory at any point. However if you want to stop a program running you can select the 'Clear Hardware Memory' menu to download an 'empty' program into the PICAXE memory.

How often can the PICAXE microcontroller be reprogrammed?

PICAXE chips can be reprogrammed at least 100,000 times. Note these are minimum values and the actual values may be much greater.

How vulnerable to damage are the microcontrollers?

The microcontrollers have a high level of static protection built into each pin and so handling them without any personal static protection in an educational environment is perfectly acceptable.

Can I control servos using the PICAXE?

Yes, many parts have a 'servo' command that allows control of up to 8 servos (one on each output).

Can I control an LCD display?

Yes, the PICAXE supports serial LCD modules (like the Serial LCD/Clock Module AXE033) via the serout command. Note that the AXE033 module can also be pre-programmed with up to 8 messages to reduce the memory usage of the PICAXE microcontroller.

How fast does the PICAXE operate?

The PICAXE-08/18 microcontrollers have an internal 4MHz resonator, and the PICAXE-28/40 uses an external 4MHz ceramic resonator. This means the microcontroller processes 1 million assembler commands a second, which equates to roughly about 1,000 BASIC commands per second.

The M and X parts can be overclocked to 8 or 16MHz (multiplies speed by x2 or x4).

Does the PICAXE support interrupts?

Yes. Many parts support a polled interrupt on the input port. Use the 'setint' command to setup the desired interrupt port setting.

How do I create time delays longer than 65 seconds?

The best way of creating long delays is to do minute delays with a loop, e.g. to wait an hour (60 minutes)

```
for b2 = 1 to 60 'start a for..next loop
pause 60000      'wait 1 minute
next b2          'next loop
```

The PICAXE microcontroller works at a nominal 4MHz, but due to device manufacturing tolerances there is likely to be a drift of a few seconds over long time periods (e.g. a day). Note that the Serial LCD/Clock module (AXE033) has a precision clock and 'alarm clock' function that can be used to trigger the PICAXE at predefined interval or at certain time/dates with much greater precision. The X parts can also be linked to the i2c DS13097 real time clock.

My program is too long! What can I do?

Tips for reducing program length (see BASIC Commands help file for more details):

- 1) Use 'let pins =' instead of multiple high/low commands
- 2) Use sub-procedures for repeated code
- 3) Try to reduce the use of sound and serout commands, which use a lot of memory
- 4) If using an LCD, store the messages in the AXE033 Serial LCD Module, rather than in the program
- 5) Use eeprom and read commands to store messages in data memory (see next page)
- 6) Restructure your program to reduce the number of 'goto' commands
- 7) Use a PICAXE chip with the largest memory (X1 or X2 parts)

You can use the 'PICAXE>Check Syntax' menu to test the length of your program without a download.

Do symbols increase the program length?

No, all symbols are converted back to 'numbers' by the computer software prior to download and so have no effect on program length. You can use as many symbol commands as you wish.

What notes are generated by the sound command?

The sound command generates different 'beep' sounds for the values 1-127. The tune and play commands on the PICAXE-08M are specifically designed to play tunes. See the tune command in section 2 of the manual for more details.

I need more outputs - what can I do?

Use the PICAXE-28X/28X1 or 40X/40X1 which can have up to 16 outputs. Or connect a single output (e.g. output7) from a first PICAXE chip to input0 of a second PICAXE-18 chip. Program the second PICAXE-18 chip with this simple program:

```
main: serin 0,N2400,b1
      let pins = b1
      goto main
```

The eight outputs of the second chip can now be controlled with a serout 7,N2400,(b2) command by the first chip, where b2 contains the 'pins' value (0 to 255) desired on the second chip. This gives you a total of 15 useable outputs.

I need more inputs - what can I do?

Use a PICAXE-28X1 or 40X1, which can be configured to have a large number of inputs. Remember that analogue inputs can also be used as digital inputs if required, just see if the 'readadc' value is greater or less than 100. In many applications switches can also be connected in parallel on a single input pin.

How do I test more than one input at once?

Use the following command to test two inputs together

```
if pin0 = 1 and pin1 = 1 then...
```

or either of two inputs

```
if pin0 = 1 or pin1 = 1 then...
```

Appendix I - Advanced Technical Information and FAQ

This appendix provides advanced technical data for users who wish to understand more advanced technical data about the PICAXE microcontrollers. This information is not required for normal PICAXE use.

These notes presume the user is familiar with PIC microcontrollers, their configuration fuse settings and programming in assembler code.

What is a PICAXE microcontroller?

A PICAXE microcontroller is a Microchip PIC microcontroller that has been pre-programmed with the PICAXE bootstrap code. The bootstrap code enables the microcontroller to be reprogrammed without the need for an (expensive) conventional programmer, making the whole download system a very low-cost simple serial cable!

The bootstrap code also contains common routines (such as how to generate a pause delay or a sound output), so that each download does not have to waste time downloading this commonly required data. This makes the download time much quicker.

Why use the PICAXE instead of assembler / C?

The PICAXE uses a simple BASIC language (or flowcharts) that younger students can start generating programs with within an hour of first use. It is much easier to learn and debug than either C or assembler code.

The second advantage is the direct cable download method. The software is free and so the only cost per computer is a low-cost download cable. This enables students to buy their own cable and for schools to equip **every** single computer with a download cable. Other systems that require an expensive programmer are generally too expensive to implement in this way.

Finally as the PICAXE chip never leaves the board, all leg damage (as can occur when the chip is moved back and forth from a programmer) is eliminated.

How is the program stored within the microcontroller?

The program is stored in either data or program memory depending on the microcontroller type. The following table shows how program, read/write/eprom data and readmem/writemem data is stored.

	Program	Read/Write	Readmem/Writemem
PICAXE-08	Data	Data	N/A
PICAXE-08M	Data	Data	N/A
PICAXE-14M	Data	Data	N/A
PICAXE-20M	Data	Data	N/A
PICAXE-18	Data	Data	N/A
PICAXE-18M	Data	Data	N/A
PICAXE-18A	Program	Data (256)	N/A
PICAXE-18X	Program	Data (256)	N/A (use i2c)
PICAXE-28A	Program	Data (64)	Program (256)
PICAXE-28X	Program	Data (128)	N/A (use i2c)
PICAXE-28X1	Program	Data (256)	N/A (use readtable or i2c)
PICAXE-40X	Program	Data (128)	N/A (use i2c)
PICAXE-40X1	Program	Data (256)	N/A (use readtable or i2c)

The program and read/write memory is overwritten with every download. Use the eeprom command to preload data (within the program) for the read/write commands. The readmem/writemem memory is not changed during a download.

How many times can the microcontroller be reprogrammed?

PICAXE chips can be reprogrammed at least 100,000 times. Note these are minimum values and the actual values may be much greater.

How is a download started?

When the computer starts a download an interrupt is generated on the serial input pin on the PICAXE. This interrupts the main program and puts the PICAXE into a state for a new download to be received. Therefore you must ensure that the 'serial in' pin is tied to ground (0V) via the 22k/10k resistors on ALL project boards for reliable operation of the microcontroller (to prevent unwanted 'floating pin' interrupt signals).

What are the electrical characteristics of the PICAXE (e.g. operating voltage range etc.)?

The electrical characteristics of the PICAXE microcontroller is dependent upon the base PIC microcontroller that is programmed with the PICAXE bootstrap code to create the PICAXE microcontroller. Therefore see the Microchip datasheet (from www.microchip.com) for the appropriate microcontroller characteristics. The lowest recommended operating voltage from these datasheets is 3V (Note this is the 'operating voltage' only. You may require a higher voltage (minimum 4.5V recommended) whilst doing the actual serial download from the computer to ensure accurate memory programming of the chip). X2 parts are also available in special 1.8V to 3.3V variants.

Does the PICAXE set the watchdog timer fuse?

Yes, the watchdog timer is set and used within a number of commands such as sleep and nap. The user cannot alter it's settings.

Does the PICAXE set the power-up timer fuse?

Yes.

Does the PICAXE set the brown-out fuse?

Yes for the M, X1 and X2 parts, no for other parts. An unfortunate side effect of the brown-out fuse on the other parts is that it restricts the lowest operating voltage of the micro-controller to about 4.2V. As many users wish to use 3V battery packs, the brown-out fuse is not set on the PIC microcontrollers with a 4.2V brown-out.

The enablebod/disabledbod command can enable/disable the brown out function on M, X1 and X2 parts.

How does the PICAXE do ADC (analogue-to-digital) conversions?

The PICAXE-08 and PICAXE-18 use the internal comparator to do a low resolution ADC step comparison, providing 16 discrete analogue values. The other PICAXE microcontrollers all use the internal ADC to do a full 256 step (8 bit) conversion. Although the microcontrollers are technically capable of 10 bit conversions, this is converted by the readadc command into byte (8 bit) values for ease of use via the byte (b1 etc.) variables, which makes the maths easier for students. This gives a resolution of about 0.02V (at 5V supply) which is adequate for almost all educational projects. Most parts also have a separate 10 bit adc read option (1024 steps), via the readadc10 command.

Can you supply the bootstrap program so that I can make my own PICAXE?

No. The small royalty made on each PICAXE chip sold is the only financial benefit to our company to support the PICAXE system - the software is free and the cables/development kits are sold at very low cost. Therefore we do not allow anyone else to manufacture PICAXE microcontrollers.

Can I mix assembler in with the BASIC code?

No. The program and bootstrap code cannot be 'mixed' with assembler code, this is not good programming practice. However you can achieve the same goal by converting your BASIC into assembler code using the automatic conversion feature, and then editing the converted assembler code program (see below).

Can I see the assembler code that is downloaded into the PICAXE?

If you own a Revolution Serial PIC Programmer (part BAS800), you can convert PICAXE BASIC programs into assembler code, to program blank PICs or to just learn how assembler code works by 'disassembly'. However some of the more complex commands (e.g. serin) are not supported, and the assembler code program generated is optimised for sequential learning (not optimised for compactness as with the PICAXE system) and so the code is not identical to that downloaded to the PICAXE.

Can you alter the input/output pin arrangement of the PICAXE microcontroller?

The PICAXE-08 has 5 pins that can be configured as desired. The 28 and 40 pin PICAXE can also be altered to give more inputs or outputs. The 18 pin input/output pin arrangements are fixed and cannot be altered.

How long a program can I download into the PICAXE microcontroller?

This varies on the commands used, as not all commands use the same amount of memory.

There is no fixed 'byte' formula as to memory usage e.g. pause 5, pause 50 and pause 500 will all take different amounts of memory space! To calculate memory usage use the 'Check Syntax' option from the PICAXE menu. This will report the amount of memory used.

Do symbols increase the program length?

No, all symbols are converted back to 'numbers' by the computer software prior to download and so have no affect on program length. You can use as many symbol commands as you wish.

Do I need to erase the device?

How do I stop a program in the PICAXE microcontroller running?

Each download automatically overwrites the whole of the previous program.

There is generally no need to erase the memory at any point. However if you want to stop a program running you can select the 'Clear Hardware Memory' menu to download an 'empty' program into the PICAXE memory.

Why is an 'empty' program still 3 bytes long?

Each downloaded program contains some configuration data, and an 'end' command is always added automatically to the end of each downloaded program. Therefore an 'empty' program on screen will not generate a zero byte program.

How vulnerable to damage are the microcontrollers?

The microcontrollers have a high level of static protection built into each pin and so generally handling them without any personal static protection in an educational (non-production) environment is acceptable.

Can I use i2c EEPROMs with the PICAXE?

The X parts support all i2c parts via the i2cslave, readi2c and writei2c commands.

Can the PICAXE count pulses?

The M and X parts support the count command which can count the number of pulses in a defined period. All parts support the pulsinc command to measure the length of a pulse.

Can I control servos using the PICAXE?

Can I do PWM control of a motor using the PICAXE?

The M and X parts have a dedicated pwmout command which acts on one or two of the pins for full pwm control.

The A, M and X parts have a 'servo' command that allows control of up to 8 servos (one on each output). The servo command uses the internal timer and an interrupt, so that the pulses are maintained 'in the background' all the time that the PICAXE is running the main program.

The servo command produces a pulse of length 0.01ms to 2.55 ms approximately every 20ms. Therefore it can also be used as a simple background PWM output with PWM mark:space ratios between 1:2000 and 1:8 (approx).

How fast does the PICAXE operate?

Can I overclock the PICAXE?

The PICAXE-08/18/18A/18M/18X microcontrollers have an internal 4MHz resonator, and the PICAXE-28/40 family use an external 4MHz ceramic resonator. This means the microcontroller processes 1 million assembler commands a second, which equates to roughly about 1000 BASIC commands per second. Different commands take different times to execute depending on how complex their 'assembler code' is.

The M and X parts can be overclocked to 8 or 16MHz (see the Over-clocking Appendix for restrictions).

Why does the PICAXE only support up to 4800 baud rate on serout/serin commands?

Can I send and receive serial data via the download cable?

The maximum baud rates were originally selected for reliable operation with microcontrollers with internal resonator. The early internal resonators were not as accurate as an external device, and a slower baud rate ensures reliable operation. The X1 and X2 parts support much higher baud rates via the hardware EUSART using the hserout command.

Many parts can send data via the download cable via a 'sertxd' command and receive data via the 'serrxd' command.

Does the PICAXE support interrupts?

The PICAXE uses the internal microcontroller interrupts for some of its BASIC commands (e.g. servo). Therefore the internal interrupts are not available for general use. However the A, M and X parts all support a single 'polled' interrupt on the input port. Use the 'setint' BASIC command to setup the desired interrupt port setting to enable the polled interrupt. The polled interrupt scans the input port between every BASIC command (and constantly during pause commands), and so activates very quickly.