



CHECK THE FOLLOWING:

IF THE PICAXE WON'T RUN...

- **Power Supply is functioning;** The Picaxe requires 4.5 to 6 Volts. If you're using a battery box, check that the batteries aren't flat, the box isn't broken, and turned on if it has a switch.
- **Power is connected the right way around;** Reversing the battery connections may fry the chip, and will definitely prevent operation. An 08M chip should have Pin 1 connected to the positive, and Pin 8 connected to negative (Pin 1 is marked with an indentation and the pins are numbered anticlockwise).
- **No PCB Faults;** Given the nature of etching PCB's, there's a good chance a trace may have eroded away in the process. Solder a piece of wire over the gap to repair it.
- **Short circuits;** Sloppy soldering or leads touching on a breadboard can cause shorts. This could just make two LED's turn on at once, or it could blow up your battery. Look for wires touching or solder bridging two contacts, and rectify any issues.
- **Leads reach the contacts;** When using a breadboard, it's possible to cut a lead so it appears to be long enough, but doesn't quite reach the metal contacts inside. 5mm is an absolute minimum, 10mm is preferable. Look for loose wires, which are a dead giveaway that this is a problem.
- **Reset pin is tied high;** Larger Picaxe chips (18, 28 and 40 pin chips) have a reset pin which needs to be connected to the positive rail via a 4.7kΩ resistor for operation. If you've forgotten this, the chip won't respond.
- **Resonator is included;** Older 28 and 40 pin Picaxe Chips relied on an external resonator or crystal circuit for operation. Newer models have an internal resonator to make life easier, and the smaller chips have had internal resonators since they were introduced.
- **Electrical noise;** Large devices, especially motors and servos, will produce electrical noise during operation that will interfere with the chip. You can run your devices from a separate power supply or regulator, or add a suppression capacitor across motors to reduce noise.
- **The Chip works;** Some chips just won't work, especially if they've been fried by reversing the power connections, or in rare cases have been given a nasty static discharge. If it's bust, replace it.

IF THE PICAXE WON'T DOWNLOAD...

- **Download Circuit is correct;** You should have a diagram of the specific download circuit for your chip; if not, ask for one. The main errors are incorrect resistor values, the correct resistors but swapped around, the chip in the wrong way (which also messes up the power supply), or the download header or socket connected incorrectly.
- **The Programming lead is not reversed;** Circuits that use a 3 pin header for the download connection can be inserted the wrong way around (those that use a stereo headphones plug don't have this issue). Swap it round and see what happens.
- **The Programming lead works;** Over time and use, cables break. Ask to borrow one that's been tested and works, or buy a prebuilt one which will be better quality than the school ones made from mouse cables.
- **Programming Editor settings are correct;** The programming editor needs to be set to the right serial port, and finding the right one is usually a case of trial and error (fortunately there's usually only two options, and more often that not its connected to COM1). Additionally, you need to select the right type of chip (you will get an error telling you what chip is selected and what chip you're using if this is an issue). If you're using a chip that's different to the editors default, adding "#PICAXE" followed by a space and your chip (eg. 08M, 18X, 28X1) to the top of your code will automatically change the editor settings when you download.
- **No Errors in the code:** The Programming Editor won't let you download if you've made a syntax error. It will try and point this out to you, but it's up to you to find the issue and fix it.
- **Code fits on the chip:** The M-Series Picaxe Chips (08M, 14M, 18M, 20M) have a fairly severe memory limitation which will cause problems with longer codes. The editor will give you a heads up if this is an issue. A-Series (18A) and Number-Series (08, 18) have even tighter restrictions, whereas X-Series and later (18X, 28X1, 40X2 etc.) have a larger memory capacity and this shouldn't be an issue.
- **Chip doesn't need a reset:** In normal operation the Picaxe Chip will scan for incoming downloads. Sometimes a command may require that this scanning stops, which can cause a chip to appear faulty. To test this, turn the circuit off, start the download, and turn the circuit back on when the download window appears. This is known as a "Hard reset". You can use the reset switch if your chip supports one and it has been included.
- **Power supply is steady;** Picaxe Chips must be self powered, and if their power supply is disconnected or fluctuates during download, you will get a prompt on your computer and a corrupt program on your chip (don't worry, it will be overwritten if you try again). Fluctuating supplies can be fixed with a capacitor across the positive and negative rails.
- **Serout pin isn't busy:** Smaller Picaxe Chips (8, 14, and 20 pin chips) use Output 0 as the Serout pin for program download and serial communication. Sometimes, the devices attached to the pin will interfere with the comms, and need to be temporarily removed for the circuit to work. Some pre-built boards have a jumper that allows you to choose between the download cable and the output, and this needs to be moved over before downloading.
- **The chip still has it's bootstrap code;** Picaxe chips are actually a PIC chip with a special program called a bootstrap downloaded to it which allows it to interpret and run BASIC programs. If anyone's messed around with the chip in a PIC programmer, more than likely the bootstrap code will have been overwritten, and you'll need to find a new chip.

IF THE PICAXE WORKS, BUT NOT AS EXPECTED...

- **The code is correct;** Mistakes in the code happen, especially when you're experimenting. Make sure that the chip isn't just doing what you've told it to do. Using the simulator to compare the results of a simulated run of your code and your project will confirm if this is the issue.
- **Outputs are connected correctly;** LED's are polarised (have a positive and negative lead) and need to be connected with the positive lead to the output and the negative lead to the negative rail. Connecting the positive lead to the positive rail and the negative lead to the output will work, but the chip will light when you turn the output off and vice versa, so you will need to change your code. Any other configurations will result in the LED staying dark. All outputs will act in the same way regarding which rail they are connected to, but non polarised components don't have to be connected a specific way around.
- **Inputs use the right interface;** Many input components use a voltage divider interface to connect to the chip, such as LDR's, Thermistors, and technically Potentiometers (Variable Resistors). Switches also use a voltage divider to protect the chip. Check that these connections are correct, or your inputs may return unexpected results or no result at all.
- **Analog Inputs are connected to an ADC enabled pin;** Analog to Digital Conversion (ADC) is the process of taking an analog voltage reading between 0V and the supply voltage, and converting it to an 8 bit number (0 to 255). Not all input pins are connected to the ADC unit in the microcontroller, so these inputs can only be used with digital inputs like switches.
- **Output devices are not connected to inputs and vice-versa;** Although some chips have In/Out (I/O) pins that will operate as an input or output, most Picaxe chips have a set of inputs and a set of outputs. Connecting the wrong devices to Input-Only or Output-Only pins will stop the device from working. You should, however pick up these errors when you come to program your chip. The 08M uses I/O pins, with the exception of Pin 0 (Output only) and Pin 3 (Input only).
- **Multiple outputs operate at once;** This is most likely a short in your circuit, or you've made a blunder in your PCB or Breadboard design. This should be easy to spot and fix.
- **ADC results aren't linear;** Some analog sensors (such as Thermistors) don't have a linear output. You'll need to either use trial and error to get a good set of values, or graph the output and use maths to get a linear result (this will be second nature to physicists). The curve info should also be detailed in the sensors datasheet, which you will have to ask for.
- **High Power devices won't turn on;** The Picaxe is only capable of directly driving LED's, piezo buzzers, and similar low power devices. Heavier duty devices, like motors, relays, incandescent bulbs and solenoids require a transistor or FET interface to operate, or a dedicated driver chip (such as the ULN2083 for general digital outputs, or the L293D for bidirectional motor control).