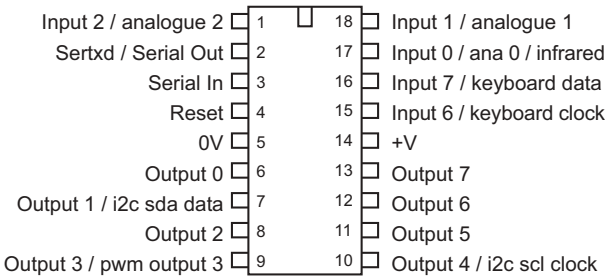
PICAXE 18X/28X/40X Extended Features...

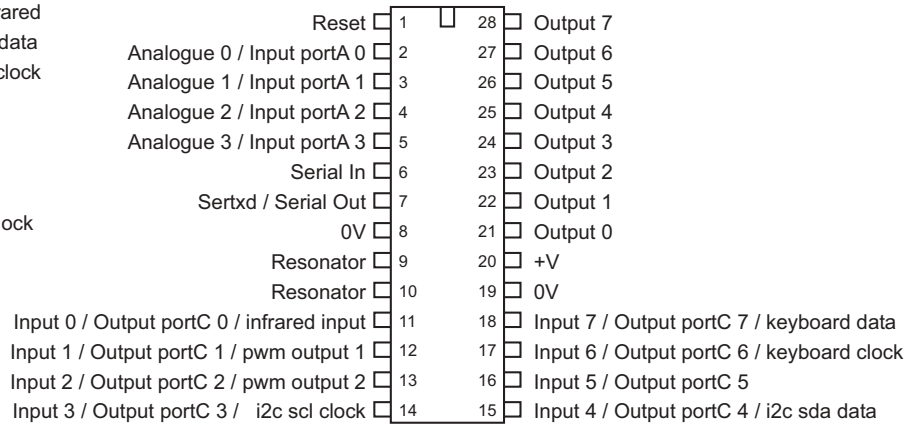
PICAXE Type	IC Size	Memory (lines)	I/O Pins	Outputs	Inputs	ADC (L =low)	Data Memory	Polled Interrupt
PICAXE-08	8	40	5	1-4	1-4	1L	128 - prog	-
PICAXE-18	18	40	13	8	5	3L	128 - prog	-
PICAXE-18A	18	80	13	8	5	3	256	Yes
PICAXE-18X	18	600	14	9	5	3	256 + i2c	Yes
PICAXE-28	28	80	20	8	8	4	64 + 256	-
PICAXE-28A	28	80	20	8	8	4	64 + 256	Yes
PICAXE-28X	28	600	21	9-17	0-12	0-4	128 + i2c	Yes
PICAXE-40X	40	600	32	9-17	8-20	3-7	128 + i2c	Yes

PICAXE-18X

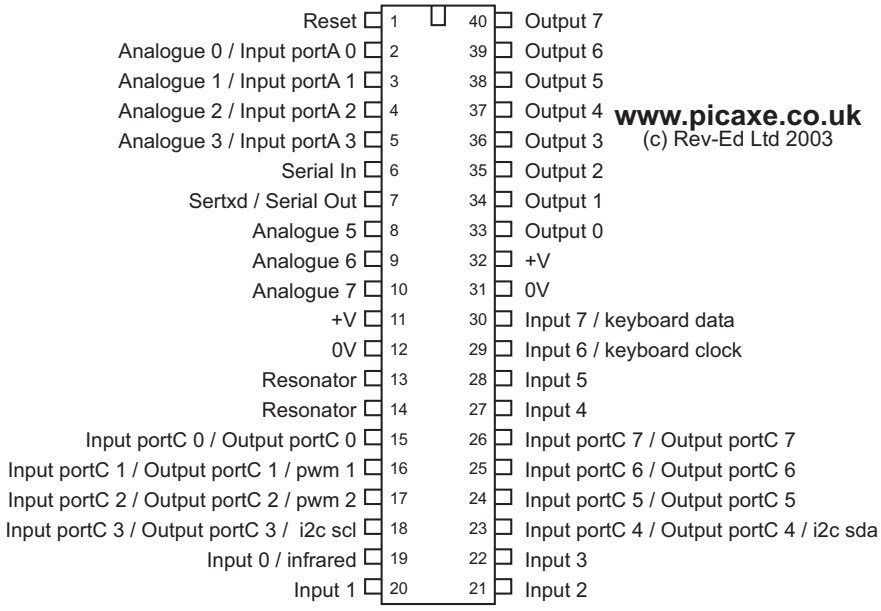


www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

PICAXE-28X



PICAXE-40X



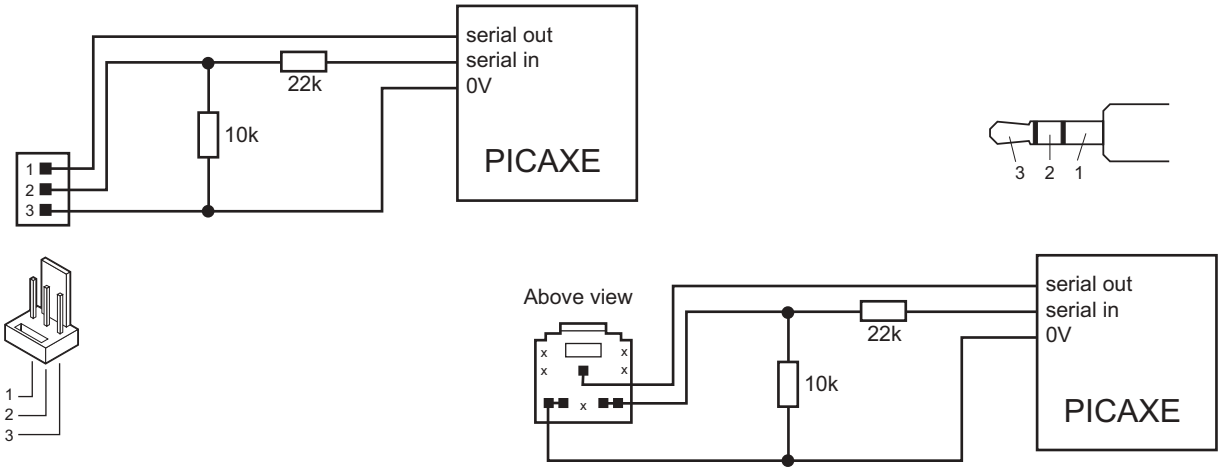
www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

www.picaxe.co.uk
(c) Rev-Ed Ltd 2003

In this datasheet...

Serial Download Circuit:

The serial download circuit for all PICAXE microcontrollers is (straight or 'stereo plug' cable connections):



SECTION 1 - PICAXE Commands: (new X part commands in bold)

Output -	high, low, toggle, pulsout, let pins =
Sound -	sound
Input -	if…then, readadc, **readac10**, pulsin, button
Serial -	serin, serout, **sertxd**
Program Flow -	goto, gosub, return, branch
Loops -	for…next
Mathematics -	let… (+, -, *, **, /, //, max, min, &, |, ^, &/, |/, ^/)
Variables -	if...then, random, lookdown, lookup
Data memory -	eeprom, write, read
Delays -	pause, wait, nap, sleep, end
Miscellaneous -	symbol, debug
RAM -	peek, poke
Servo Control -	servo
Infrared -	infrain
Interrupt -	**setint**
Temperature -	**readtemp, readtemp12**
Keyboard -	**keyin, keyled**
1-wire Serial No -	**readowsn**
I2C -	**readi2c, writei2c, i2cslave**
PWM -	**pwmout**
Counting -	**count**

Please see the BASIC Commands help file for more detailed syntax help and information about each command.

SECTION 2 - What's new in the PICAXE-18X, 28X, 40X?

The extended X parts support all the standard commands and features, with the following enhancements:

- Program memory 8x as long (approx. 600 lines rather than 80), with intelligent download
- Continuously driven pwm motor drive outputs (pwmout command)
- Count high frequency pulses within a set time period (count command)
- Large data memory (128 or 256 bytes) (read/write commands)
- i2c bus support for EEPROMs and other devices (i2cslave/writei2c/readi2c commands)
- Interrupt feature on inputs (setint command)
- Accurate digital temperature sensor interface (readtemp/readtemp12 commands)
- 10 bit and 8 bit adc option (readadc10/readadc commands)
- User serial output via the serout pin / programming cable (sertxd command)
- 4800 baud rate option (and faster at higher clock frequencies) (serin/serout commands)
- Read serial number from any Dallas 1-wire device (e.g. iButton) (readowsn command)
- Computer keyboard interface on inputs 6 and 7 (keyin, keyled command)
- Software support for increased clock frequency (see section 3 of this datasheet).

See the BASIC Commands datasheet (v3.5 or greater) for further information on each command.

In addition the PICAXE-28X and 40X have a more **flexible i/o pin** layout to allow the user to select more inputs and/or outputs than the standard configuration. See section 2 of this datasheet.

Memory Size

The X parts have a memory size 8x larger than the A parts (2048 bytes rather than 256 bytes). This means it can store a program of approximately **500-700 lines** of BASIC code (depending on commands used).

To reduce download times the X parts only download the appropriate (used) pages of memory. Therefore a shorter program will download quicker than a longer program

PICAXE-40X

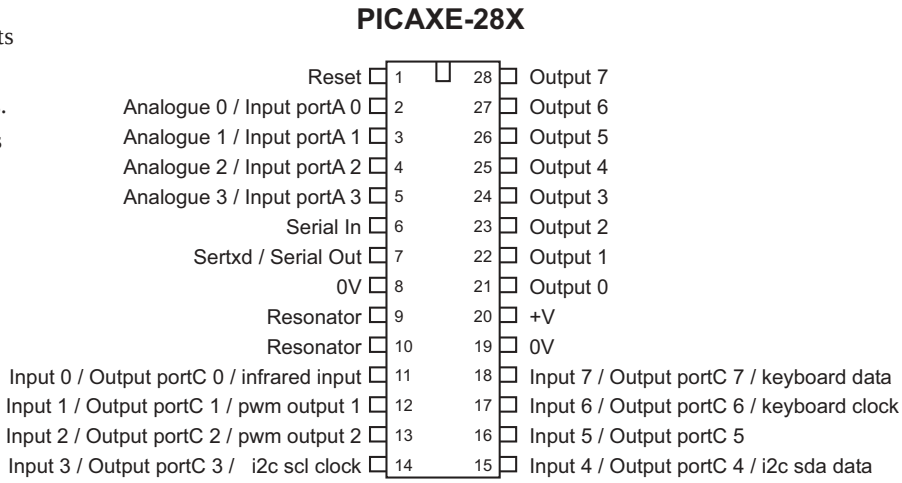
The PICAXE-40X is electronically configured as a 'special' version of the PICAXE-28X (with additiona l pins) Therefore when using the Programming Editor software the 'PICAXE-28X' mode is used for **programming both** the PICAXE-28X and the PICAXE-40X microcontrollers.

SECTION 3a - PICAXE-28X Input/Output Pins

To provide greater flexibility, the input/output pin configiration of the PICAXE-28X can be varied by the user. The default power up settings are the same as the other PICAXE-28 parts (8 in, 8 out, 4 analogue).

PORTA (legs 2 to 5) provide 4 analogue inputs (default) or up to 4 digital inputs.
PORTB (leg 21 to 28) provide 8 fixed outputs.
PORTC (leg 11 to 18) provide 8 digital inputs (default) or up to 8 outputs.

This gives a maximum of :
12 digital inputs
16 outputs
4 analogue inputs



PORTA Functions

Leg	Default Function	Second Function
2	analogue 0	porta input 0
3	analogue 1	porta input 1
4	analogue 2	porta input 2
5	analogue	porta input 3

PORTB Functions

PORTB pins are fixed as outputs and cannot be altered.

PORTC Functions

Leg	Default Function	Second Function	Special Function
11	input 0	output portc 0	infrared (input)
12	input 1	output portc 1	pwm 1 (output)
13	input 2	output portc 2	pwm 2 (output)
14	input 3	output portc 3	i2c scl clock (input)
15	input 4	output portc 4	i2c sda data (input)
16	input 5	output portc 5	
17	input 6	output portc 6	keyboard clock (input)
18	input 7	output portc 7	keyboard data (input)

The portC pins can be used as the default inputs, changed to outputs, or used with their special function via use of the infrain, keyin, i2cslave, or pwmout command as appropriate.

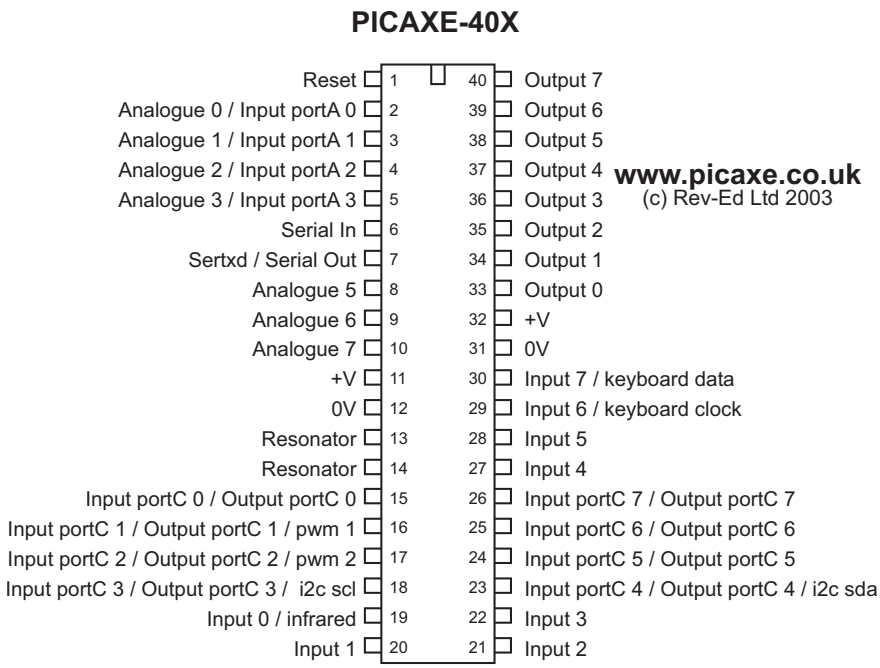
The second or special function of the pins are selected by modified commands as explained in the next section.

SECTION 3b - PICAXE-40X Input/Output Pins

To provide greater flexibility, the input/output pin configiration of the PICAXE-40X can be varied by the user.

PORTA (legs 2 to 5) provide 4 analogue inputs (default) or up to 4 digital inputs.
PORTB (leg 32 to 40) provide 8 fixed outputs.
PORTC (leg 15-18. 23-26) provide 8 digital inputs (default) or up to 8 outputs.
PORTD (leg 19-22, 27-30) provide 8 digital inputs
PORTE (leg 8 to 10) provide 3 analogue inputs

This gives a maximum of :
20 digital inputs
16 outputs
7 analogue inputs



PORTA Functions

Leg	Default Function	Second Function
2	analogue 0	porta input 0
3	analogue 1	porta input 1
4	analogue 2	porta input 2
5	analogue	porta input 3

PORTB Functions

PORTB pins are fixed as outputs and cannot be altered.

PORTC Functions

Leg	Default Function	Second Function	Special Function
15	input portc 0	output portc 0	
16	input portc 1	output portc 1	pwm 1 (output)
17	input portc 2	output portc 2	pwm 2 (output)
18	input portc 3	output portc 3	i2c scl clock (input)
23	input portc 4	output portc 4	i2c sda data (input)
24	input portc 5	output portc 5	
25	input portc 6	output portc 6	
26	input portc 7	output portc 7	

The portC pins can be used as the default inputs, changed to outputs, or used with their special function via use of the i2cslave or pwmout command as appropriate.

PORTD Functions

Leg	Default Function	Special Function
19	input 0	infrared (input)
20	input 1	
21	input 2	
22	input 3	
27	input 4	
28	input 5	
29	input 6	keyboard clock (input)
30	input 7	keyboard data (input)

PORTE Functions

PORTE pins are fixed as analogue inputs and cannot be altered.

SECTION 3c - Using porta analogue inputs as digital inputs (28X, 40X)

The porta pins 0 to 3 (legs 2 to 5) are, by default, configured as analogue inputs.
However with the PICAXE-28X and -40X they can also be used as simple digital inputs.

The following syntax is used to test the input condition:

```
if porta pin0 = 1 then jump
```

i.e. the additional keyword 'portA' is inserted after the 'if' command.

to test if two (or more) porta inputs are on

```
if porta pin0 = 1 AND pin1 = 1 then jump
```

to test if either of two (or more) porta inputs are on

```
if porta pin0 = 1 OR pin1 = 1 then jump
```

Note the portA command is only required once after the 'if' command.
It is not possible to test inputs on two different ports within the same if…then statement.

It is not possible to access the portA pins with any other 'input' type commands (count, pulsin etc).
Therefore these pins should be reserved as simple on/off switches.

SECTION 3d - Using portc as inputs (40X)

On the PICAXE-28X portC are the standard input pins and addressed by the standard `if pin0 =` command.
On the PICAXE-40X **portD** are the standard inputs, and hence use the standard `if pin0 =` command. Therefore for **portC** inputs the extra keyword `portC` must be used (as in the `if portA pin0 =` example above).

SECTION 3e - Using portc as outputs (28X, 40X)

The portc pins are, by default, digital input pins.
However with the PICAXE-28X and -40X they can also be configured to be used as digital outputs.

To convert the pin to output and make it high

```
high portc 1
```

To convert the pin to output and make it low

```
low portc 1
```

To convert all the pins to outputs

```
let dirsc = %11111111
```

To convert all the pins to inputs

```
let dirsc = %00000000
```

Note that 'dirsc' uses the common BASIC notation 0 for input and 1 for output. (Advanced - If you are more familiar with assembler code programming you may prefer to use the command 'let trisc =' instead, as this uses the inverted assembler notation - 1 for input and 0 for output. Do not attempt to directly poke the trisc register (poke command) as the PICAXE bootstrap refreshes the register setting regularly).

To switch all the outputs on portc high

```
let pinsc = %11111111
(or)   let portc = %11111111
```

To switch all the outputs on portc low

```
let pinsc = %00000000
(or)   let portc = %00000000
```

To use portc 1 and portc 2 as pwm controlled outputs use the **pwmout** command (see the BASIC Commands help file for further information). The pwm output is maintained continuously in the background, making these pins ideal for controlling motors etc.

It is not possible to access the portc pins with any other 'output' type commands (sound, serout, pulsout etc). Therefore these pins should be reserved as simple on/off outputs (apart from the pwm control on 1 and 2).

When using the special input functions (infrared sensor (0), or an i2c device (3, 4), or a keyboard (6, 7)) you must take care to ensure that the appropriate pins are maintained as inputs. Converting these pins to outputs may damage the external device and/or the microcontroller.

SECTION 4 - Resonator Frequency and Overclocking.

All PICAXE functions are based upon a 4MHz resonator frequency. This is the only frequency recommended. However the user may choose to 'overclock' the X parts if desired, although this is not recommended unless absolutely necessary for a particular project (e.g. when using the count command).

With the -08, -18, -18A the internal resonator is fixed at 4MHz and cannot be altered.
With the -18X the internal resonator has a default value of 4MHz. However it can be increased by the user to 8MHz via use of the 'setfreq' command.

With the -28 and -28A an external 4MHz resonator must be used.
With the -28X / -40X an external 4MHz 3 pin ceramic resonator is normally used, but it is also possible to use a faster resonator (8 or 16Mhz), although this will affect the operation of some of the commands.

NB PICAXE-28X firmware version 7.0 can be used at 4 or 8 MHz
PICAXE-28X or -40X firmware version 7.1 (or greater) can be used at 4, 8 or 16 MHz

The Programming Editor software supports resonator frequencies of 4, 8 and 16MHz only. No other frequencies are supported. If any other frequency is used it will not be possible to download a new program into the PICAXE microcontroller.

To change the frequency:

PICAXE-18X
Download a program containing the command `setfreq m4` (for 4 MHz) or `setfreq m8` (for 8Mhz). If no setfreq command is used in a program the frequency will default to 4MHz. Note the new frequency occurs immediately after the command is run. When downloading new programs, you must ensure the correct frequency (View>Options>Mode) is used to match the last program running in the PICAXE-18X chip. If in doubt perform a 'hard-reset' at 4Hz.

PICAXE-28X and PICAXE-40X
Solder the appropriate external 3pin ceramic resonator into the project board.

Downloading programs at 4, 8, 16MHz

After changing frequency you must select the correct frequency via the View>Options>Mode software menu. If the wrong frequency is selected the program will not download.

Commands affected by resonator frequency.

Many of the commands are affected by a change in resonator frequency. A summary of the important commands affected are given below (see BASIC Commands datasheet for detailed command syntax).

count

The base unit of count is

- 1ms at 4MHz
- 0.5ms at 8 MHz
- 0.25ms at 16 MHz

The pin is checked every

- 20us at 4MHz (max. 25kHz pulse rate)
- 10us at 8MHz (max. 50kHz pulse rate)
- 5us at 16MHz (max. 100kHz pulse rate)

i2slave

The bus speed within i2slave must be adjusted by use of the appropriate frequency keyword

- i2cfast / i2cslow at 4Mz
- i2cfast8 / i2cslow8 at 8Mz
- i2cfast16 / i2cslow16 at 16Mz

If the incorrect keyword is used the i2c function may not work.

pause / wait

The base unit of pause is:

- 1ms at 4MHz
- 0.5ms at 8 MHz
- 0.25ms at 16 MHz

The base unit of wait is:

- 1s at 4MHz
- 0.5s at 8 MHz
- 0.25s at 16 MHz

pulsout / pulsin

The base unit of pulsout/pulsin is:

- 10us at 4Mhz
- 5us at 8Mhz
- 2.5us at 16Mhz

pwmout

The period and duty cycle should be calculated using 4MHz, 8MHz or 16Mhz as appropriate.

serin / serout / sertxd

Due to the sensitive nature of serial communication no guarantee is given that serin or serout commands will work at any frequency other than 4MHz. However the theoretical baud rates at the higher clock frequencies are as follows:

Baudmode	4MHz	8MHz	16MHz
600	600	1200	2400
1200	1200	2400	4800
2400	2400	4800	9600
4800	4800	9600	19200 (also sertxd baud rate)

A maximum of 4800 is recommended for complicated serial transactions.

sound

The note of sound will be multiplied by 2 (8Mhz) or 4 (16MHz).
The duration of sound (12ms at 4MHz) will be reduced to 6ms (8MHz) or 3ms (16MHz)

Commands that do not work at 8 or 16MHz

The following commands will not work at 8 or 16MHz due to timing issues with the external device listed:

- infrain (infrared remote)
- keyin (keyboard)
- keyled (keyboard)
- readtemp / readtemp12 (DS18B20 temperature sensor)
- readowsn (1-wire device)
- servo (servo)

Commands that are not affected by frequency changes.

The following timing commands are NOT affected as they use a separate internal r/c timer:

- nap and sleep