PICAXE-28X1 Overview

Document Purpose

This document provides a brief overview of the new PICAXE-28X1 features for users already familiar with the existing 28X parts. Please see the revised PICAXE manual (part2 – BASIC Commands) for more detailed command information.

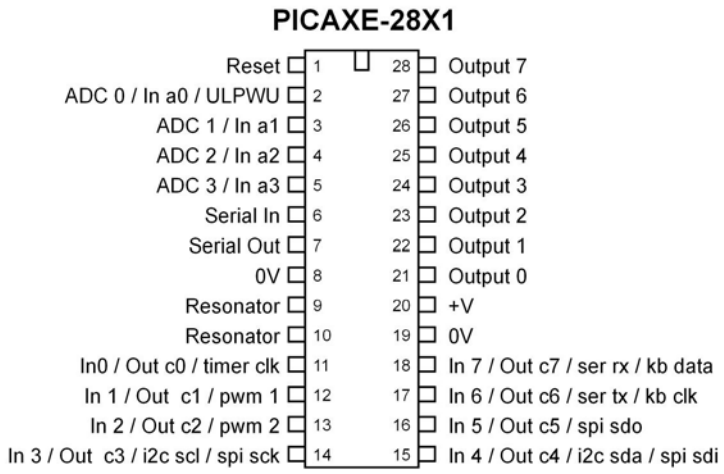
The 28X1 and 40X1 will be released on 1 May 2007.

All further references to the 28X1 in this document also apply to the 40X1.

All advance information provided is subject to change without prior notification.

28X1 overview

The 28X1 is pin compatible with both 28A and 28X devices. Therefore it is a simple drop-in replacement on existing PCB designs. However it has a large range of enhancements over previous parts. This is made possible via a recently released new generation of low-cost Microchip PIC microcontrollers, which have many additional features and larger memory capacity. The 28X1 is based upon the new PIC16F886 microcontroller.



Software Installation:

To program the 28X1 you must be using Programming Editor Version 5.1.0 or later. This can be downloaded from the www.picaxe.co.uk website after 1 May 2007.

New Features

Program / Data Memory

The 28X1 program memory capacity has been doubled to 4096 bytes. This equates to approximately 1000 lines of BASIC code. The data eeprom memory (for read/write commands) has also been doubled to 256 bytes.

Direct Variables

28 directly accessible variables b0-b27 (w0-w13) are now supported.
32 direct bit flags bit0-bit31 (within b0-b3) are also supported.

In addition there are 7 special function byte variables:

pins (pin0-pin7)	- state of the input pins (byte/bits)
outpins (outpin0-7)	- state of the output pins (byte/bits)
flags (flag0-flag7)	- general purpose / special hardware feature flags (byte/bits)
ptr (ptr0-ptr7)	- scratchpad pointer (byte/bits)
@ptr	- scratchpad value (virtual, not a real register) (byte only)
@ptrinc	- scratchpad value with post increment (byte only)
@ptrdec	- scratchpad value with post decrement (byte only)

Note that when assigning an output port byte you should now use
outpins = 255 rather than pins = 255.
Similarly the state of the outputs can be read directly (e.g. let b1 = outpins)

An additional set of 8 special word variables are also defined. Some of these have special hardware functions, but these variables may also be used as general word variables if that particular hardware feature is not used within a program. The individual bytes of the word are not directly accessible; the full word must always be read or written.

Name:	Special Name:	Special Function
s_w0	-	- *reserved for future use*
s_w1	-	- *reserved for future use*
s_w2	-	- *reserved for future use*
s_w3	-	- *reserved for future use*
s_w3	-	- *reserved for future use*
s_w5	hserptr	- hserin pointer
s_w6	hi2clast	- last i2c (slave mode only) written address
s_w7	timer	- time elapsed

An additional 96 bytes are available via the peek/poke commands, and a further 128 bytes via the scratchpad (put/get commands).

This provides an equivalent total of 28 + 8 + 96 + 128 = 260 user byte variables.

Scratchpad Variables
PUT, GET
ptr, @ptr, @ptrinc, @ptrdec

The scratchpad is a new separate, additional 128 byte RAM area for rapid saving of temporary variable data (i.e. it is separate to the directly accessed byte variables). A common example of use would be in saving a long received serial data stream from a GPS module or for storing arrays.

Each scratchpad address can be directly accessed via the PUT and GET commands. However a more versatile use is via the ptr (scratch pad pointer), which is a unique virtual 'pointer variable'. The pseudo variable name '@ptr' can then be used in commands as if it was a normal variable, but in use actually 'reads/writes' to the current address on the scratchpad pointed to by ptr. Similarly if '@ptrinc' is used as a variable in a command, read/write is to the current pointer address and then the pointer (ptr) automatically increments to point to the next address. Therefore '@ptrinc' can be repetitively used in commands such as serin to save data in different scratchpad addresses e.g.

```
ptr = 10
serin 1,N2400, @ptrinc, @ptrinc, @ptr
```

In this example the three serial bytes received will be saved in scratchpad positions 10, 11 and 12 and can then be later accessed, for example, by GET commands.

The scratchpad is also used as the data exchange area within i2c slave mode and for temporary storage of received serial hardware (hserin) data.

Resonator
Internal resonator (8 frequencies from 31kHz to 8MHz)
Optional external resonator 4, 8, 16 or 20MHz

The 28X1 can use either an internal or external resonator. The SETFREQ command is used to switch between frequencies (internal and external frequencies may be used within the same program). To reduce PCB costs/simplify circuits the resonator may be totally omitted.

Please note the chip always starts in internal 4MHz resonator mode. You must use a setfreq command at the start of your program to switch to the external resonator.

e.g. setfreq em16 (set to external 16MHz resonator)

If the external resonator fails, operation will automatically drop back to using the 4MHz internal resonator. The maximum external resonator speed is 20MHz.

Some commands with specific timing (e.g. readtemp or irout) always use the internal resonator automatically, switching back to normal timing after the command completes.

Commands

The 28X1 supports all 28X command features. In addition to these commands there are a number of new commands and features:

Data Storage Table
TABLE
READTABLE

A 256 byte data lookup table can be defined when the program is written and automatically downloaded (as part of the program) into the 28X1. This is a very convenient and efficient way of storing data, e.g. for storing LCD menu text, calibration values etc.

Pause
PAUSE
PAUSEUS

The additional pauseus command creates a small time delay, in 10us multiples

16 bit Timer
SETTIMER
SETTIMER COUNT
timer
toflag

The 28X1contains a general use 16 bit background timer/counter. When in timer mode this timer runs in the background at a user defined frequency (e.g. every second). The timer variable can be read at any time. Timer overflow is used to set a flag bit (toflag).

When in counter mode the timer records the number of rising edge pulses on input 0. This occurs in the background, whilst the PICAXE program performs other tasks.

Low Power
ENABLEBOD, DISABLEBOD
HIBERNATE, SLEEP, NAP

The 28X1 has a 2.7V brownout that can enabled or disabled.

The new hibernate command enters a permanent low power sleep state. The 28X1 can then only be woken by a hardware interrupt. The hibernate/sleep/nap commands can all be interrupted by the hardware ULPWU pin or a hardware serial / i2c reception.

Sounds
SOUND
PLAY, TUNE

The 28X1 supports the mobile phone ring tone play and tune commands (as 08M) on any output pin.

ADC
READADC, READADC10
CALIBADC, CALIBADC10

The ADC reading within PICAXE chips is based upon the supply voltage, which can sometimes vary over time (e.g. as a battery runs down). The new calibadc command uses a fixed internal 0.6V source to provide a known voltage ADC reading. This enables users to calibrate their ADC readings to a known fixed voltage, regardless of battery state.

Reset
RESET

The reset command will reset the chip, clearing variable values and restarting the program.

Serial Communications
SERIN, SEROUT
DISCONNECT, SERRXD, SERTXD, RECONNECT
HSERSETUP, HSERIN, HSEROUT
hserinflag, hserptr

Serial communication can now be carried out in 3 different ways:

1) serin/serout act via the standard input/output pins. The serin commands now support a timeout feature.

2) serrxd and sertxd act via the PICAXE download cable. To prevent a clash between program downloads and user serial communication the disconnect/reconnect commands are used (disconnect prevents the PICAXE chip looking for new program downloads). This enables user program data to be both transmitted and received via the download cable.

3) hserin and hserout are a more advanced method of communication via the on-board serial hardware pins. This allows advanced features such as wake-up from sleep, background receive and much higher serial baud rates. The background receive allows serial data to be automatically saved into the scratchpad area as the PICAXE chip completes other tasks. A flag is set when background serial data is received.

INTERRUPT
SETINT, SETINT NOT

The polled interrupt now also has the option to operate when a condition is NOT true, as well as the original ’match condition’ option.

Peripheral interfacing

a) i2c
HI2CSETUP, HI2CIN, HI2COUT
hi2cflag, hi2clast

The 28X1 can now act as an i2c master or i2c slave.

The i2c data rates have been upgraded to include 100k, 400k or 1M operation.

The new i2c slave mode is fully automatic and is the recommended method of networking multiple PICAXE-28X1 chips. The slave features are completely automatic, so the PICAXE chip acting as a slave can still run a normal program, with the i2c slave read and writes occurring in the background without program intervention. When a write to the slave occurs a flag is set and the PICAXE program can then poll, and act upon, this flag signal. The i2c interface operates in the same manner as the popular 24LCxx EEPROM chips, with the 128 scratchpad bytes as the read/write memory area.

This method theoretically allows one 28X1 master to control over 120 28X1 slaves, all connected via a common two wire i2c bus.

b) SPI (3 wire)
SPIIN, SPIOUT
HSPISETUP. HSPIIN, HSPOUT

The 28X1 supports 3-wire (SPI) communications on input/output pins or via the dedicated hardware pins (which provide a faster transfer rate). Normal input/output pin support is included to allow both i2c and SPI on the same system (as hardware SPI and hardware i2c use the same pins).

c) IR
IRIN, IROUT

The 28X1 supports infrared input (with timeout) and output. Sony SIRC format.

d) One Wire
OWIN, OWOUT
READTEMP, READTEMP12, READOWSN

The 28X1 fully supports the Dallas/Maxim 1-wire protocol, allowing direct control of all 1-wire parts.

Mathematical Statements
Brackets are now supported within mathematical statements:
e.g. let b4 = b1 + (b2*5)

The following additional operators are now available

<<	shift left
>>	shift right
*/	multiply (middle word)

The following unary operators are now available

sin		sin (degrees)
cos		cos (degrees)
sqr		sqr (N-R on val/2, 10 iterations)
inv	~	invert
ncd		2^n decoder
dcd		2^n encoder
bintobcd		convert binary to BCD
bcdtobin		convert BCD to binary

The following variable operators are also available:

rev	reverse a number of bits
dig	find the BCD digit of a variable
swap	exchange two variable values

The following pseudo helper commands are also available:

bintoascii	converts a single variable value to multiple ascii bytes
bcdtoascii	converts a single bcd value to multiple ascii bytes

Other changes

Stack
The gosub nesting stack has been doubled to 8 levels.

Readtemp / readtemp12
These commands have been modified to make them faster in operation.

Command Renaming.
Some X commands have been renamed to add additional features. Although the original command may still function, it is recommended that users edit their X1 program to use the new command, as this may provide additional enhanced features.

Old Command	***New Command***
i2cslave	hi2csetup
readi2c	hi2cin
writei2c	hi2cout
infrain, infrain2	irin
infraout	irout
keyin	kbin
readoutputs	*not required* (read 'outpins' variable directly)

Software Simulation
The 28X1 project board can be connected 'online' (via the download cable) so that 'real life' inputs and outputs react on the project board as the simulation steps through program lines on screen.

Serial to TCP/IP
The Programming Editor software can now use the serial port on a networked computer as if it was the local serial port, by transmitting the serial data over TCP/IP. Therefore a new program, or data from a running program, can be transmitted to / received from a remote computer. The software runs as a service on the remote computer, and so can be configured to automatically start when the computer starts.

FAQ

Can I use a 28X1 in my 28A / 28X PCB?
Are existing programs compatible?
Yes, all 28 pin PICAXE chips are pin compatible and so interchangeable. Almost all 28A / 28X programs will work directly on the 28X1 without any modification, although a small number of commands have been renamed (see above).

Will there be an 18X1?
At present there is no suitable base 18 pin PIC microcontroller available. If a suitable device is released in the future we will naturally develop an 18X1.

What is the difference between the 28X1 and the 28X2?
The 28X2 is a future product not yet available. As it is based upon the 18Fxx series microcontrollers the 28X2 is more expensive, but provides additional variables and more flexible input/output pin control. The supported command features are almost identical.

The main differences are as follows (original 28X also shown for comparison):

	28X	**28X1**	**28X2**
PIC Technology	16Fxxx	16Fxxx	18Fxxx
I/O Pin Layout	Standard	Standard	Fully customisable
Direct Variables	14 bytes	28 bytes	48 bytes
Indirect Variables	96 bytes	96 bytes	208 bytes
Scratchpad	n/a	128 bytes	256 bytes
Program Slots	1 x 2048	1 x 4096	2 x 4096
Max. Speed	16MHz	20MHz	40MHz

All advance information provided is subject to change without prior notification.