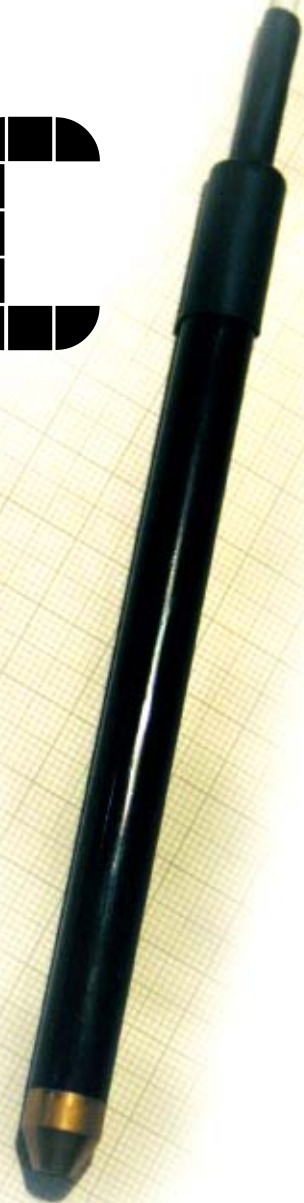


PENPIC

PENPIC is finally available to schools and colleges. PENPIC is TEPs latest word in providing a simple, accessible and innovative PIC programming system. Nick Baldwin from TEP tours you through using and understanding this amazing new resource.



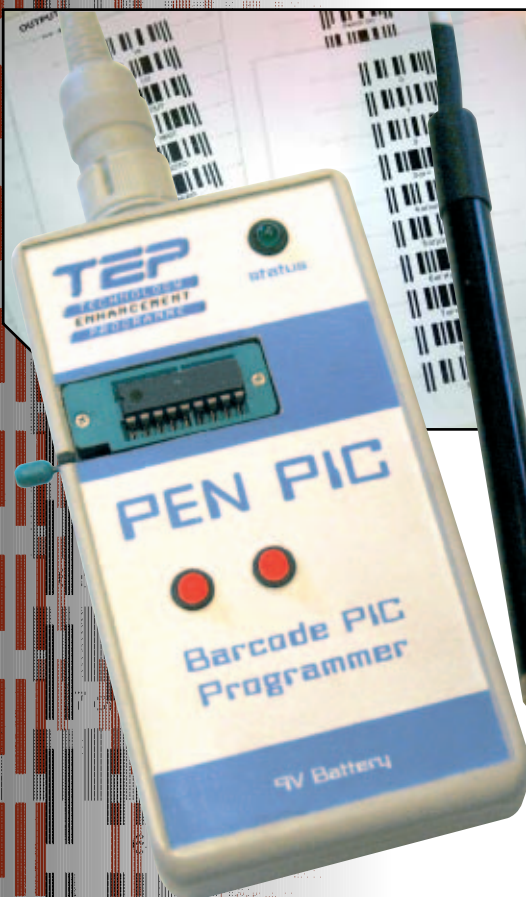
Introduction

PENPIC is set to become the first and last word for ease of use across all the Key Stages. This is a remarkable alternative way of inputting programs into a PIC and relies exclusively on a series of printed bar codes. Each unique code represents a value or command in basic that the PIC interprets and stores.

The pupil or student selects and/or assembles their sequence of program lines as bar codes and successively swipes each code into the PIC.

What is really different about this system is the almost failure proof nature of the programming method. It requires no IT infrastructure to run in class and provides audible prompts if a program line is unsuitable or has not correctly scanned.


Normally PIC microcontrollers require a programming language and or flow chart from a computer program. PENPIC has been designed so that students simply scan the codes directly into the PIC chip without having to type in commands or draw diagrams on screen. It is still a good idea however to plan program lines and outcomes in advance.



The PENPIC starter kit is supplied with a handheld/desktop programmer and wand and a laminated set of bar codes. The unique set of codes can also be downloaded from the TEP site and reprinted in class and laminated to prolong life or cut and pasted from the site into student portfolios to provide an instant record and aide memoire of program ideas and development.

Initially PENPIC utilises a **16F628 PIC** not familiar to most colleagues who more usually will have used 16F627 and 16F84 PICs. The Pin-Outs are the same as the 16F627 and 16F84 and in fact the operating and programming strategy is the same but the 628 simply has more room! Because the PENPIC programmer relies on the PIC having a Bootstrap or header program already on the PIC to interpret and sort the incoming barcode data it simply needed more memory. The programmer is powered by a 9 volt battery and does need turning off after use!

The 16F628 PIC works in all the project boards and demo boards already available and will work with all published PIC 18 pin PCB designs in use.

 **PENPIC Programming Unit**
The final production version will vary from this image

Continued Overleaf →

PENPIC

Continued



Getting started with PENPIC

The programmer unit features a ZIF socket to hold PICs while programming and the plug in wand has a reflective optical infra red head.

Once a PIC is inserted in the unit it automatically erases the previous data entries allowing the user to scan each new line in, which is auto saved to the PIC in the order in which they are scanned. This means being very clear about the intended sequence of commands. This is not at all difficult with the bar codes and commands in front of the user. The PENPIC board uses a 9 volt supply that will seem at odds with the concept of 'flashing' the program onto a PIC that nationally requires a higher voltage. Of course what users are really doing is loading data from the bar codes into the PICs EEPROM using the interpreter software already on the PIC in the header program. The 16F628 is available from a number of sources but users will need to ensure they 'burn' the PENPIC header program onto the PIC first using a conventional programmer or alternatively purchase the 16F628 with the header already on them. This is the same cost as a blank PIC from Teaching Resources and will minimise teacher's hassles. You can download the hex file direct from the TEP site:

www.tep.org.uk/Frames/_f_PenPIC.html

The 16F628 pre-headed PIC is inserted in the ZIF socket and the unit gives an audible double 'beep' indicating that the chip is inserted and cleared of its previous program. This is accompanied by an LED flash that stays on. The wand is swiped from left to right in a natural fairly brisk movement right across the bar code. Incorrect swipes that are too slow or too fast (less likely) are prompted by a failed to input tone that is two tones high followed by low.

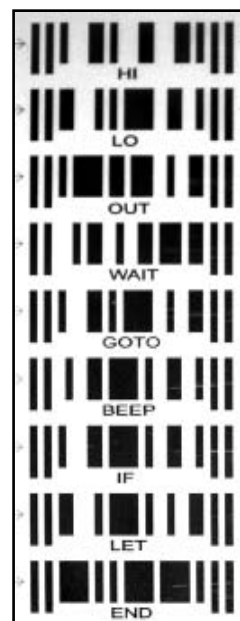
The individual elements of code as they are input are monitored by one of the LED's. There are 10 black bands that constitute each command or numeric value. At each end of the code are two very similar bands that set the scan speed reference and end of coding data for the chip for each scan operation.

You can photocopy or print off bar code sheets but will need to ensure they do not get too scruffy or dirty or lose resolution at the copy stage as the light and dark bands of coding are critical and could easily be degraded without a little care.

Programming commands

For each command and numerical value there is an associated bar code. Pupils can print out a sequence of codes or assemble a program in advance or can selectively swipe the codes from the printed sheets of codes direct.

Command	Statement
High	The command to switch an output high
Low	The command to switch an output low
OUT	The binary command to turn on a combination of outputs 0-255
WAIT	The command to create a delay or pause in 0.1 increments (WAIT 001= 0.1second WAIT 100= 10 seconds)
GOTO	To make the program loop to a different line (Lines are numbered 00-99 as two digits)
BEEP	Sets the Beep sound for piezo output
IF	The conditional statement for an input or variable
LET	The variable maths statement
END	The last line of program.



Each line of code is 'swiped' by the reflective opto pen and each successful swipe of a command is greeted with a single high tone beep or numerical value with a single or double beep. Each sequence of commands or program needs to finish with an END command.

Editing

Once the programming sequence has started you cannot edit a line out as the sequence is already 'stacked' in the PIC so the only option at that point is to re-enter the commands and data by resetting the PIC in the socket which erases the old or incorrect sequence.

Programming a simple routine

The simplest program to input is a flashing LED program that repeats or loops to line 00 to create an LED flashing with 1 second intervals.

Scan	High	followed by	Scan	1	means output 1 high
Scan	Wait		Scan	010	means WAIT 1 second
Scan	Low		Scan	1	means output 1 low
Scan	Wait		Scan	010	means WAIT 1 second
Scan	GOTO		Scan	00	means GOTO line 00 (0-99)
Scan	END				means End of program

This program takes about 15 or so seconds to perform. Pupils can actively modify commands from the start and record their ideas as a series of codes cut and pasted into their folder or folio or simply write down sequentially the instruction list prior to scanning the codes direct from the sheet

IF statements

IF statements, involves a condition to test the program or inputs. This determines what loop or line to GOTO in a program. The inputs tested are 0, 1, 2 or 3 in the case of the PIC inputs and up to two variables within the program, which we will call x and y.

IF statement is scanned in followed by input 0 or 1,2,3 followed by the digital input condition 0 or 1. This is then followed by the routine or command of GOTO or High or Low or OUT.

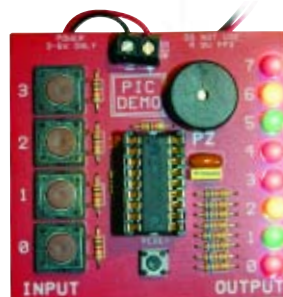
IF statements include:

Is equal to	=
Is greater than	>
Is less than	<
Is not equal to	==

Programming using IF statements

A simple routine scanning an input and turning all the outputs on for a fixed period

Scan	IF 0	followed by	Scan	1	means if input 0 high
Scan	GOTO	followed by	Scan	00	means GOTO line 00 (loop)
Scan	OUT	followed by	Scan	255	means OUTPUT 255 (all high)
Scan	WAIT	followed by	Scan	030	means WAIT 3 seconds
Scan	OUT	followed by	Scan	000	means OUTPUT 000 (all low)
Scan	GOTO	followed by	Scan	00	means GOTO line 00 (0-99)
Scan	END				means END of program



Continued Overleaf →

PENPIC

Continued



Using LET statements

Using LET statements we can create Variables x and y with values that can be used to create useful integer maths routines which can be incremented or decremented using input keys or embedded into the program maths.

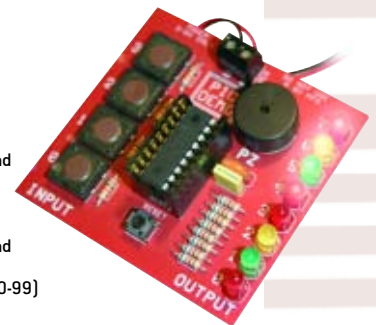
If a variable x or y is scanned the program requires a condition for that variable to be scanned:

LET statements include:

x or y variable is equal to	=
x or y variable is itself plus a value	+=
x or y variable is itself minus a value	-=

So an incrementing program to sequentially increase the binary output values to create a simple moving LED display might be:

Scan	LET	followed by	Scan	4 or x	means LET x variable		
Scan	3 or =	followed by	Scan	000	means x=000		
Scan	LET	followed by	Scan	7 or +	followed by	001	means x=x+1
Scan	WAIT	followed by	Scan	001	means WAIT 0.1 second		
Scan	OUT	followed by	Scan	4 or x	means output value x		
Scan	WAIT	followed by	Scan	001	means WAIT 0.1 second		
Scan	GOTO	followed by	Scan	00	means GOTO line 00 (0-99)		
Scan	END				means end of program		



More about Bar Codes

Essentially a bar code is a series of light and dark bars representing 1's and 0's. The bar code has a unique ability to instruct the header program on the PIC and also keeps check of the user's own hand scan speed! Scanning is from left to right of course they could easily be turned upside for left hand scanning. Sharing programs between user groups, classes or even through this journal couldn't be easier. The lamination on the printed bars is required or the dirt and colour fade on paper would rapidly set in. A future alternative method of scanning is of course 'swiping' each individual code on a separate card past the reflective IR opto head.

Included with the PENPIC resource is a comprehensive leaflet that takes the user through aspects of using IF statements and Variables and it is worth taking a look at them here as it conveys just how comprehensive the programmer is. The current PIC 'header' available does not include analogue inputs however these are planned as a future release and once again will be free downloads off the TEP website or at no extra cost pre-programmed from Teaching Resources.

Future Proof Policy

What we have found difficult to appreciate in testing and trying out this resource is that it is another TEP product totally future proofed in that, future PIC releases that emerge on the education market will be catered for by this programmer and users will simply download the latest 'headers' as they become available.

Providing some exemplar programs on paper and getting students to edit and re-edit them and try 'hands-on' program changes is a useful starting point.

Alongside the PhonePIC programmer, PICtalk and Chip Factory Logic System, PENPIC completes a unique suite of PIC programming resources and amply demonstrates a wide range of electronic communication methods. As such they complement each other and are a great family of resources for any teacher of Systems and Control.

PENPIC is available from Teaching Resources

▶ PENPIC Starter Kit – Stock code BAR 001

