AWC

PAK-XII Data Sheet

AWC
1279 FM 518 Rd #2
Kemah, TX 77564
(281) 334-4341
http://www.al-williams.com/awce.htm
V1.2 12 August 2004

Table of Contents

Overview

The PAK XII is a 32-bit floating-point math coprocessor with an easy-to-use interface that makes it simple to use with practically any microcontroller including the Basic Stamp from Parallax. However, you can easily interface the PAK XII with practically any modern microprocessor. The PAK can add, subtract, multiply, and divide numbers between +/- 6.80564693 x 10^{38}. The smallest number the PAK I can represent is +/- 1.17549435 x 10^{-38}.

- Here are some of the PAK's major features:
- Simplified RS232 9600 baud ASCII text interface
- Uses as few as 2 pins to connect to the host
- Analog comparator
- 6 channel A/D converter
- Easy to use

The PAK is a standard 28-pin IC. In order to operate it must have a regulated supply of 5V and connection to a clock element. The PAK includes a 10MHz ceramic resonator that you can use to clock the chip. Note that while the ASCII RS232 interface is easy to use, it is not as efficient as the protocol used by our other math PAKs (such as the PAK-IX). If you want higher speeds (and the possibility of concurrent operation) you should consider a PAK-IX (or the similar PAK-I and II coprocessors). However, if you want the easiest-to-use math coprocessor, you want the PAK-XII.

Other than the power and clock connections, the PAK requires at least two wires to connect between your microcontroller (the host) and itself. In addition, the analog section may be connected to your power bus or an external reference.

If You Need Help

If you require assistance with your PAK, please feel free to contact us. The best way to get support is via e-mail (stamp@al-williams.com). However, you may also call between 9AM - 4PM Central Time at (281) 334-4341. You can also fax to (314) 596-9925. Be sure to check out our Web page for updates at www.awce.com.

Registering Your PAK

Please take a moment to register your e-mail address with AWC. Simply send an e-mail message to pakreg@al-williams.com. AWC will not make your address available to other companies, but we may periodically send you updated technical notes. You'll also receive information about new microcontroller products and specials.

Update: PAK-XII "A" Version

PAK-XII chips manufactured after 12 August 2004 have the ability to store up to 10 macros. The total storage available for macros is 511 characters (and each macro takes one extra character). At start up, there are no macros. You define each macro in sequence by using the { command. You can use the macro by using the }n command where n is a digit from 0 to 9. You can clear all macros by issuing a {{ command. In either case, the { command ends with a carriage return. For example, here a program defines a macro that doubles the number on the stack:

```
{2*
OK
5}0=
10.0000000
```

You can define macros at any time although once defined, they can't be changed or deleted (except by deleting all macros or restarting the PAK). A macro can invoke another macro, but any commands in the first macro that follow the call to the second macro will be ignored.

When the PAK successfully records a macro, it prints OK and a carriage return. If there was an error, the PAK prints ## and a carriage return.

Pin Connections

Pin	Name	Type	Description
1	Reset	Input	Hardware resets the PAK when low. Connect to 5V for normal operation
2	RX	Input	Connection to RS232 sent from host computer. This is a TTL-level signal. Communications are at 9600 baud, 8 bits, 1 stop bit, no parity.
3	TX	Output	TTL-level RS232 sent to host computer. Communications are at 9600 baud, 8 bits, 1 stop bit, no parity.
4-6	N/C		Not connected
7	Vdd	Power	+5V
8,22	Vss	Power	Ground
9, 10	CLK1, CLK2	Clock	Connect pin 9 to one end pin of supplied 10MHz resonator and pin 10 to the other end pin (it does not matter which end connects to which pin). The resonator's center pin is ground.
11	N/C		Not connected
12	COMP+	Analog	+ input for analog comparator
13	COMP-	Analog	- input for analog comparator
14	HS-	Input	Must be grounded to enable the PAK to send data to the host (used for optional handshaking).
15	HS+	Input	Must be disconnected or high to enable the PAK to send data to the host (used for optional handshaking)
16-19	N/C		Not connected
20	AVdd	Power	+5V for analog circuitry (must be within 0.3V of Vdd)
21	ARef	Analog	Analog reference voltage. To use internal references, you can leave this pin disconnected or to improve performance, connect a .1uF capacitor between the pin and ground. If you prefer, you can connect an external reference voltage to this pin, although this will preclude using the internal references. The external reference can range from 2V to AVdd.
23-28	A0-A5	Analog	Analog input channels. These all read as 10-bit channels, although the accuracy of channels 4 and 5 is limited to 8 significant bits

Command Summary

The PAK uses reverse polish notation (RPN) to enter numbers and operators. RPN has you enter numbers into a stack, and then perform operations on them. So if you send the PAK the string "1.1,5.4" you've just loaded the stack. The current number at the top of the stack is 5.4 and "below" that is the number 1.1.

If you then sent a +, the PAK would remove both numbers and put the result (6.5) on the top of the stack. You can push multiple numbers on the stack. For example, consider:

```
1 2 3 * +
```

This string would multiply 2 by 3 and then add 1 to the result (leaving 7 on the top of the stack).

The PAK uses a 15-element stack and you must enter numbers without using scientific notation. So if you wanted to compute 5.1 divided by 1.15 you would send this string to the PAK:

```
5.1, 1.15 /=
```

In general, a space, carriage return, or comma will separate numbers and the subtraction operator. Otherwise spaces are ignored. So you could also send:

```
5.1,1.15/=
```

In the chart below, X is the top of the stack and Y is the number below the top of the stack.

The = sign produces output. The first character will be a space for positive numbers, a – for negative numbers, or an error flag. The remaining characters will be the result followed by a carriage return.

Code	Name	Description
|	ABS	X=|X|
%	SQRT	X=Square_Root(X)
^	POW	X=X to the Y power
+	ADD	X=X+Y
-	SUB	X=Y-X
*	MULT	X=X*Y
/	DIV	X=Y/X
@	DUP	Y=X (copies top of stack)
=	DISP	Print result using decimal notation (always shows 7 digits after decimal point)
E	DISPE	Print result in scientific notation
e	e	X=e (2.718281828)
#	EQ	If X=Y then output ASCII “1” else output ASCII “0” – does not disturb stack
<	LT	If Y<X then output ASCII “1” else output ASCII “0” – does not disturb stack
>	GT	If Y>X then output ASCII “1” else output ASCII “0” – does not disturb stack
?	ISNAN	If X is not a number then output ASCII “1” else output ASCII “0” – does not disturb stack
_	ISINF	If X is infinity then output ASCII “1” else output ASCII “0” – does not disturb stack (this is the underscore character, not a dash)
;	DROP	Discard X
!	CLR	Clear entire stack (two clears recommended at program start)
:	DUMP	Dumps all numbers in stack (prints backslash + 0 as last number)
~	SWAP	Swap X and Y
l	LOG	X=ln(X)
L	LOG10	X=log10(X)
c	COS	X=cos(X)
C	ACOS	X=acos(X)
s	SIN	X=sin(X)
S	ASIN	X=asin(X)
t	TAN	X=tan(X)
T	ATAN	X=atan(X)
p	PI	X=3.14159265
Pn	PACE	Set serial pacing in ms (n=0 to 9)
r	SQRT2	X=Square Root(2)
R0	AREF	Set analog reference to AREF pin
R1	AVDD	Set analog reference to AVDD pin (Must not connect AREF to voltage)
R3	A256	Set analog reference to 2.56V (Must not connect AREF to voltage)
A	ACOMP	Set X to 1.0 or 0.0 depending on analog comparator output
an	ADC0	X=analog channel n (see text; n can range from 0 to 5)
[n	STO	Store X in register n (n can range from 0 to 9); does not disturb stack
]n	RCL	X = contents of register n (n can range from 0 to 9)

Notes: All angles are in radians

About Floating Point Numbers

The 32-bit floating point representation used by the PAK internally is the IEEE754 format. This means that there is a 24 bit numeric part and an 8 bit exponent. So although the PAK can represent numbers that are very large and very small, there are only 24 bits of significance to the numbers. This is not peculiar to the PAK – it affects any program (including PC software) that uses IEEE754 numbers.

Because 2 to the 24th power is 16,777,216 numbers that require a representation larger than this will have built in inaccuracies. For example, consider this output from the PAK:

17000000 1 + = 17000000.0000000

This is obviously wrong, but it exceeds the ability of the PAK to store numbers that are both large and small. On the other hand, consider this:

17000000 2 *= 34000000.0000000

That's the correct answer, but there are only 2 significant digits in the answer (in other words, the PAK internally represents this number as 3.4E07). The distinction between numeric range and significant digits is very important and you should understand the ramifications of this before using any floating point math on any computer, not just a PAK.

There are cases where the PAK may not be able to find an answer for you. For example, if you try to divide by 0, there is no correct answer for the result. When you use the DISP or DISPE functions, the first character of the result will always be a space for a nonnegative number or a "-" for a negative number. However, the first character will be "#" if the result is "not a number" (an undefined result) or "^" if the result is infinity.

The result in these cases is always 0.0000000. So, for example, consider this exchange with the PAK:

1 0 /=^0.0000000

You can also extract numbers in scientific notation by using the DISPE command ("E"). This format is often useful when dealing with very large or very small numbers. The numeric part follows the usual format (including the leading tag character). After the numeric part, however, there is an E and an exponent (two digits and sign). This exponent is the power of 10, so 1.1000000E+01 is actually the number 11 (1.1 time 10 to the first power). For example, consider this dialog:

1 10000000000/= 0.0000000000

1 10000000000/E 9.9999990E-11

The first line shows the answer as zero because with only 7 digits after the decimal point, it is zero! But the real answer is actually a very small number. Keep in mind that like nearly every computer program, the PAK stores numbers in binary. That means there are often small rounding errors that occur because binary numbers can't precisely store certain decimal fractions. So in the above example, it is clear that the answer should be 1E-10, but there is a very small rounding error.

You can't directly enter scientific notation, although it is easy enough to use the MULT and POW operators. So to enter 2.2E-5 you could send:

2.2 10 -5 ^ *

For debugging purposes, you can dump the entire stack using the DUMP command (":"). This outputs each number on the stack starting with the top of the stack and working backwards. The end of the list contains a backslash (instead of a blank or minus sign) and a 0.0000000 to mark the end.

Analog Capabilities

The PAK-XII has several exciting analog capabilities. In particular, there is an analog comparator and four 10-bit analog inputs. There are also two analog inputs that have 8 bit accuracy (although they report 10 bits of result).

The analog comparator is connected to pins 12 (+) and 13 (-). If the voltage on the + pin is higher than the voltage on the – pin, the comparator read command (a) will put 1.0 on the top of the stack. Otherwise, the comparator will put 0.0 on the top of the stack. The comparator has a maximum offset voltage of 20mV and leakage current less than +/- 50nA. The typical propagation delay of the comparator is under 500nS (at 5V).

The six analog inputs have a minimum 55 mega ohm input impedance (typical values are 100 mega ohms) and a typical bandwidth of 38.5kHz. You can measure voltages based on the AVdd pin, the ARef pin, or an internal 2.56V reference.

The ARef pin can accept an external reference (it has a typical input impedance of 32K). However, if you do feed an external reference, you can't use the AVdd (typically 5V) or internal 2.56V reference. If you plan on using either AVdd or the 2.56V reference, either leave ARef disconnected, or place a .1uF capacitor between ARef and ground to improve the reference accuracy.

You can use the Rn command to choose one of the available references. After you change the reference you should discard one analog reading to allow the converter to stabilize. The default reference is ARef.

Assume you have ARef connected to 5V (or that you are using the AVdd reference). Then a 2.5V input on A0 would cause the "a0" command to place the value 512.0 on the top of the stack. The maximum count (10-bit) is 1023.0.

Of course, if you use a different reference, 1023.0 counts will be the value of the reference voltage. So a 2.5V input when using the 2.56V reference would return 1022 or 1023. The “a” command always returns a whole number (you can’t read 500.25 counts, for example). However, it is possible to average multiple samples directly (which can improve performance). For example:

a0 a0 a0 a0 + + + 4/=

This reads 4 samples, adds them together and divides by 4. The spaces are not necessary, but are included to make the commands more readable. You can also convert the counts to volts in the same step. For example, suppose the reference is 5V. You might send this command sequence:

a0 a0 a0 a0 + + + 4/ 5* 1024/=

This reads 4 samples, adds them together and divides by four to get the average reading. It then multiplies the result by 5 and divides by 1024. The resulting number is the voltage in volts.

Communications

Some host processor can’t keep up with the PAK’s serial output, even with hardware handshaking (the Basic Stamp, in particular may have this problem). By default, the PAK does not pause between characters, but by using the “P” command you can set a pacing amount ranging from 0 (the default) to 9 mS. The delay must be a single digit.

Since the Basic Stamp has to fetch instruction from a slow EEPROM, this pacing is often required if you are trying to read data and process it at the same time.

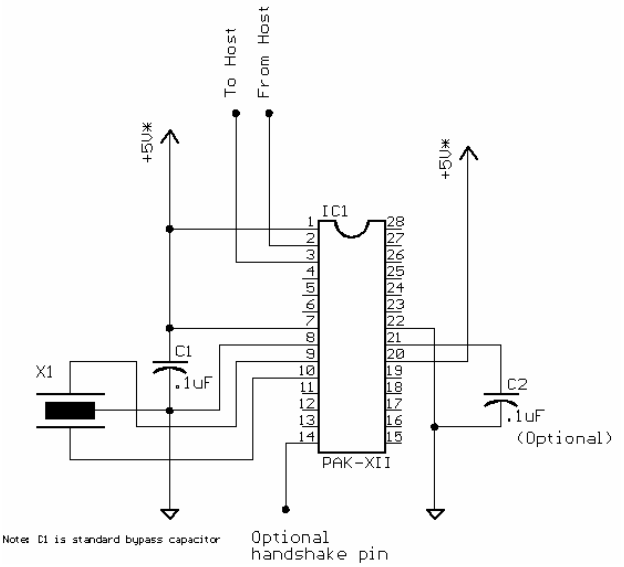
Notice that if the host computer is expecting data from the PAK, it must not send any data until the PAK sends the final carriage return. Any data sent while the PAK is transmitting will be

interpreted as an interrupt, which will terminate the current output processing and cause the PAK to process the new command.

Typical Circuits and Software

Connecting the PAK to its external components is simple. Supply 5V to Vdd, AVdd and ground the Vss pins. If you don't plan to use handshaking, you can ground HS- and leave HS+ disconnected. Connect the two outer pins of the ceramic resonator to the RES1 and RES2 pins (the order does not matter). Ground the center pin of the resonator. Connect the RESET pin to 5V.

The host computer sends TTL RS232 data to pin 2 and receives data with the same levels from pin 3 of the PAK. If you want to prevent the PAK from sending you can hook up pin 14 (or pin 15 if the handshaking is inverted). In many cases, you won't need to handshake. The PAK has a serial input buffer, so there is no need for it to stop you from sending data.



Example 1. Connections to a host computer (note all I/O levels are TTL levels, not RS232). If not using pin 14/15, ground pin 15.

Consider Example 1 connected to a Basic Stamp BS2P. Pin 2 of the PAK-XII connects to P2 of the Basic Stamp and Pin 3 connects to P3. If you want to use hardware handshaking, you can connect P4 to pin 14. Of course, you can use any Stamp pins you like as long as you change the software to match.

Here's a simple program for the BS2P and the circuit in Example 1:

```
'{$STAMP BS2p}
'{$PBASIC 2.5}
TX PIN 2
RX PIN 3
' Optional hardware handshake line
'HS PIN 4
Baud CON 240  ' Use 84 for regular BS2

c VAR byte(16)
w VAR Word
x VAR Word

top:
HIGH HS
' good idea to reset the chip first
SEROUT  TX,Baud,["!!"]
' Compute square root of 31
X=31
SEROUT TX,Baud,[DEC x,"%="]
GOSUB GetResult
END

' This just reads a string
' You can find a version that
' reads numeric variables online
' Note the Stamp is not fast enough
' to read a character at a time,
' but it can read the whole string
' at once
GetResult:
 SERIN RX\HS,Baud,[STR c\16\13]
 DEBUG STR c
 RETURN
```

Specifications

Absolute Maximum Ratings

Parameter	Minimum	Typical	Maximum
Supply voltage	4.5V	5V	6V
Supply current	-	15mA	20mA
Comparator offset voltage (Vin=Vdd/2)			20mV
Comparator leakage current (Vin=Vdd/2)	-50nA		50nA
Comparator propagation delay		500nS	
A/D absolute accuracy		1.75LSB	
VRef Input	2V		AVdd
A/D Bandwidth		38.5kHz	
A/D input resistance	55Mohm	100Mohm	
VRef input resistance		32Kohm	

Temperatures

Ambient temperature under bias	-55°C to +125°C
Storage temperature	-65°C to +150°C