

AWC

Using the PAK-III I/O Coprocessor

© 1998 by AWC, All Rights Reserved
AWC
310 Ivy Glen Ct.
League City, TX 77573
stamp@al-williams.com
<http://www.al-williams.com/awc/awce.htm>
v1.1 25 Jan 99

Overview

The Stamp Pak III is an 8-bit I/O coprocessor with a versatile interface that makes it easy to interface it with a variety of microcontrollers including the Basic Stamp from Parallax. However, you can easily interface the Pak III with practically any modern microprocessor. Here are some of the Pak III's major features:

- Robust, speed independent serial interface
- Uses as few as 2 pins to connect to the host
- Can connect multiple devices to the same 2 pins with one additional pin per device
- Works with Stamp II's ShiftOut and ShiftIn commands
- Synchronous operation – read results when you are ready for them
- Adds 8-bits of general-purpose digital I/O pins
- 64 bytes of EEPROM storage
- Advanced I/O features (PWM, Count, Pulse I/O, etc.)
- Easy to use

The Pak III is a standard 18-pin IC. In order to operate it must have a regulated supply of 5V and connection to a clock element. The Stamp Pak III includes a 10Mhz ceramic resonator that you can use to clock the chip.

Other than the power and clock connections, the Pak III requires at least two wires to connect between your microcontroller (the host) and itself. Depending on the host's capabilities and your application, you may want to use more pins for I/O (up to 5 pins).



If You Need Help

If you require assistance with your PAK-III, 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 (281) 538-2147. Be sure to check out our Web page for updates at www.al-williams.com/awc/awce.htm.

Registering Your PAK-III

Please take a moment to register your e-mail address with AWC. Simply send an e-mail message to aspreg@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.

Pin Connections

Pin	Name	Type	Description
18	Busy/Mode	Input/Output	This pin has two functions. At start up, if the pin is pulled up by a resistor, the PAK3 uses this pin as a busy indicator. The pin will read 0 when the PAK3 is not busy. However, if at startup the pin is pulled to ground, the PAK3 will indicate its busy status on the Enable/Busy pin.
17	Enable/Busy	Input or Open Collector Output	If Busy/Mode is pulled high at start up, the Enable/Busy pin is only an input. Bring it low to disable the PAK3, or high to enable it. However, if Busy/Mode is pulled low at start up, this pin serves as an open-collector I/O pin. If the host pulls the pin low, it disables the PAK3. If the PAK3 is busy, it pulls the pin low.
3	CLK	Input	The host pulses the CLK pin to shift data in or out of the PAK3.
1	SIN	Input	The PAK3 reads data from this pin. You may short SIN and SOUT and use the same pin for input and output if the host is capable of treating a single pin as an I/O pin.
2	SOUT	Open Collector Output	The PAK3 writes data to the host on this pin. Since the pin is open collector, you'll need a resistor to 5V on this pin. You may short SIN and SOUT together to use one pin for I/O
4	RESET	Input	Hardware resets the PAK3 when low.
5	VSS	Power	Ground
14	VDD	Power	+5V
15	RES1	Clock	Connects to resonator
16	RES2	Clock	Connects to resonator
6-13	B0-B7	I/O	General purpose I/O

Software Commands

Bit Pattern	Name	Argument	Returns	Description
0001BBB0	INPUT	None	0	Sets pin to input
0001BBB1	OUTPUT	None	0	Sets pin to output
0010BBB0	LOW	None	0	Sets pin to output and set state low
0010BBB1	HIGH	None	0	Sets pin to output and set state high
0011BBB0	BITREAD	None	0 or 1	Reads input pin
0100BBB0	REVERSE	None	0	Sets input to output and vice versa
0100BBB1	TOGGLE	None	0	Toggles output pin
0101BBB0	PULSOUT	PLEN	0	Pulses out (duration is a 16-bit word specifying 4uS units)
0110BBB0	PULSIN0	None	0,PLEN	Measures low pulse in 4uS units (PLEN is 16-bits)
0110BBB1	PULSIN1	None	0,PLEN	Same as PULSIN0 but measures a high pulse
0111BBB0	SHIFTIN	MBITS	0,BYT	Shift data in using pin as data and the next pin as clock; MBITS specifies a mode and bit length.
0111BBB1	SHIFTOUT	MBITS, BYT	0	Shift data out using pin as data and the next pin as clock; MBITS specifies a mode and bit length
1000BBB0	COUNT	PER	0,CT	Counts transitions on pin for specified period (16-bit word in 8uS units). Returns 16-bit count
1001BBB0	PWM	DUTY, PER	0	PWM on pin with specified duty cycle (byte) and period (2ms per unit; 16-bits)
1010BBB0	RCTIME0	None	0,PLEN	Measure RC charge in 4uS units (16-bits)
1010BBB1	RCTIME1	None	0,PLEN	Measure RC discharge in 4uS units (16-bits)
10110000	READ	ADDR	0,BYT	Read EEPROM (address is a byte from 0-63)
10110010	WRITE	ADDR, BYT	0	Writes EEPROM (address is a byte from 0-63)
10110011	GETRAND	None	0,BYT	Gets quasi-random byte
11000000	PULLOFF	None	0	Turns off pull up resistors
11000001	PULLON	None	0	Turns on pull up resistors
1101BBB0	FREQOUT	FREQ, PERIOD	0	Makes single tone. Frequency is 16-bit word (Hertz), Period is 16-bit word (milliseconds)
1101BBB1	FREQOUT2	FREQ, FREQ, PERIOD	0	Like FREQOUT, but makes two tones
10110110	CHECK	None	\$31	Communications/version check

Notes and Key:

- BBB = Bit number (0-7) corresponding to B0 to B7
- PLEN = 16-bit word specifying pulse length in 4uS units
- MBITS = 8-bit value. Top 4 bits specify mode (see reference); bottom 4 bits indicate bit length (1-8)
- BYT = Byte value
- PER = 16-bit period in 8uS units
- DUTY = Byte indicating duty cycle in 1/255 units
- ADDR = EEPROM address (byte) from 0 to 63
- FREQ = Frequency (16-bits) in Hertz
- PERIOD = Duration (16-bits) in milliseconds

I/O Considerations

Each I/O pin on the Pak III can sink 25mA or source 20mA subject to a total limit of 150mA total sink and 100mA total source. Inputs consider voltages below .8V as a 0 and voltages greater than 2.4V as a 1. Outputs will be less than .6V or greater than 4.3V (for a 1 or a 0, respectively).

Resetting

There are several ways you can reset the Pak III. It is a good idea to reset the unit before using it, or any time that you want to make sure it is in a known state. This is especially true when using the Stamp. Each time the Stamp resets or wakes up from sleep, the I/O pins briefly become inputs. This can fool the Pak III into starting a data transfer. Always reset the Pak III first.

The best way to reset the Pak III (when using only one unit) is to send a special reset sequence over the clock and data lines. This has the advantage that it doesn't require any extra I/O from the host. To send a reset sequence, bring the data pin to 0 and raise the clock to a 1. While the clock remains in the 1 state, bring the data pin high. This will indicate to the Pak III that you wish to reset. When the clock returns to a zero state, the Pak III will reset. The reset doesn't change any register values or port pins, but it does reset communications to a known state.

You can get the same result by bringing the enable pin low and then returning it to high to enable the Pak III. This is useful if you are connecting more than one Pak to the same data and clock lines. You'll need to use the enable pin then anyway, and it makes sure that the selected Pak is always in a known state. Don't use the data line reset when using multiple Paks on the same clock and data lines.

Finally, you can force a hardware reset by bringing the reset pin low. This might be useful if your circuit generates a hardware reset signal based on a brown-out detector or other master reset circuit. Normally, you'll just connect the reset pin to the +5V supply and allow the Pak III to reset itself on power up. If you do want to drive this pin, make sure that it is at 5V for normal operation. You can use a reset switch or other device if you pull up the reset pin with a 10K-22K resistor. This is the only reset that will interrupt an operation in progress.

Communications

There are several schemes you can use to communicate with the Pak III. All of them revolve around a synchronous protocol involving a clock pin and 1 or 2 data pins. Data is shifted most significant bit first, and samples at the rising edge of the clock. The Pak III exposes a separate input (SIN) and output (SOUT) pins for hosts that can't easily handle bi-directional I/O lines. However, for hosts like the Stamp or PIC, it is a simple matter to tie these lines together since SOUT is open collector.

Therefore, the minimum number of lines you need are two. An output for the clock and an I/O line to connect to SIN and SOUT. You'll connect the Enable/Busy pin to +5V and use a pull up resistor to 5V on the Busy/Mode pin. This allows you to reset the device

and communicate with it. The only problem is, you will have to make sure not to ask the device for data while it is processing. Most operations are quite fast, but some operations (generating a tone, or writing to EEPROM, for example) can take a long time depending on what parameters you specify.

If you need to disable and enable the device, you'll want to drive the Enable/Busy pin. Using this pin allows you to share the SIN, SOUT, and CLK lines with other devices that use a similar protocol. So to connect 5 Paks you'd only need 7 I/O lines (5 enables, 1 SIN/SOUT, and 1 CLK line). You drive the Enable/Busy pin low to disable the device. If you don't plan to use this capability, just pull the pin high with a pull up resistor. If you are using the Busy/Mode input (see below) you can still use a pull up resistor, or just tie the Enable/Busy pin directly to +5V.

If you ground the Busy/Mode input at start up, the device will also use the Enable/Busy pin to tell you when it is unavailable. It does this by driving the line low when it is not ready. That means the host must drive the pin with an open collector output if it wants to disable the device. If you don't need to disable the device, you can just connect the pin to a pull up resistor and connect the Enable/Busy pin to an input on the host.

If you want to use a normal output to drive the Enable/Busy pin, but you still want a busy indication, you can connect the Busy/Mode pin to 5V through a pull up resistor. Then the Pak III will use the Busy/Mode pin to signify it is busy (the pin is high when busy). In this case, the Enable/Busy pin is always an input and you can drive it normally.

One thing to note about the status returns: All of the commands will return zeros in the top bit of the response. You can use this as a form of busy indication if you are unwilling to use any other pins. Simply read the data pin until it is zero. This indicates that the device is sending the response. The READX and RIO commands do not follow this convention, but they take very little time to execute anyway (less than 4 uS).

To summarize, here are the ways you can synchronize with the Pak III:

1. Monitor the Enable/Busy line while Busy/Mode is grounded. If you bring Enable/Busy low it indicates that the Pak III is disabled. If the Pak III brings the pin low, it means it is engaged in operations.
2. Place a pull up resistor on the Busy/Mode pin. This will cause the Busy/Mode pin to be high when the Pak III is busy.
3. Wait until the data pin goes low before reading responses.

Command Reference

BITREAD

Reads the logical state of the specified pin (1 or 0). This command does **not** automatically force the pin into the input state. Returns a 0 or 1 byte.

CHECK

Communications check and version identify. Always returns \$31 (\$3 for this PAK, 1 for version number).

COUNT

Count returns the number of transitions on the specified pin during the interval you ask for. Send the period (16-bits) in 8uS units. The PAK-III will return a zero and the 16-bit count.

FREQOUT

This command emits a tone for a specified period on a particular pin. You'll want to filter the output with an RC network to filter the high-frequency noise generated. Send the frequency in hertz (two bytes) and the duration in milliseconds (also two bytes). The PAK-III returns 0.

FREQOUT2

This command emits two tones for a specified period on a particular pin. You'll want to filter the output with an RC network to filter the high-frequency noise generated. Send the first frequency in Hertz (two bytes), the second frequency in Hertz (two bytes) and the duration in milliseconds (also two bytes). The PAK-III returns 0.

GETRAND

You can use this command to read a number that is suitable for seeding a random number generator. This number is actually a free-running clock inside the PAK-III. The PAK-III will return a zero and the quasi-random byte.

HIGH

Forces the specified pin to be an output and outputs a 1 on that pin. Always returns 0.

INPUT

Converts the pin number to an input. Always returns zero.

LOW

Forces the specified pin to be an output and outputs a 0 on that pin. Always returns 0.

OUTPUT

Changes the specified pin to an output. Always returns 0.

PULLON/PULLOFF

These commands enable and disable the weak pull up resistors (about 20K) on the 8 I/O ports. The default is disabled. The return value is always 0.

PULSIN0, PULSIN1

Measures low and high pulses. Returns the length, 16-bits, in 4uS units.

PULSOUT

Pulses the output for the specified duration (a 16-bit word specifying 4uS units).

PWM

PWM generates a pulse width train of the specified duty cycle (a byte; 0 is 0%, 255 is 100%) for the allotted time (a 16-bit number in 2ms units).

RCTIME0, RCTIME1

Measures the time for a capacitor to charge or discharge. Returns a zero and the time in 4uS units (16-bits).

READ

Use this command to read a byte from the PAK-III's EEPROM. Write the address (from 0 to 63) and the return value will be a zero byte, followed by the data byte. Please note that accessing EEPROM is relatively slow, so be sure to wait for the PAK-III's acknowledgement before sending more commands.

REVERSE

Changes an input pin to an output pin and vice versa. Always returns 0.

SHIFTIN

ShiftIn requires a single byte argument. The top four bits of this argument specify the mode. The bottom four bits is the number of bits to shift (1 to 8). The mode may be any of the following:

- 0 – MSBPRES (MSB first, sample before clock)
- 1 – LSBPRE (LSB first, sample before clock)
- 2 – MSBPOST (MSB first, sample after clock)
- 3 – LSBPOST (LSB first, sample after clock)

The PAK-III returns a 0 and the byte shifted in.

SHIFTOUT

ShiftOut requires a byte argument followed by a data byte. The top four bits of this argument specify the mode. The bottom four bits is the number of bits to shift (1 to 8). The mode may be any of the following:

- 0 – LSBFIRST (LSB shifted out first)
- 1 – MSBFIRST (MSB shifted out first)

The PAK-III returns a 0.

TOGGLE

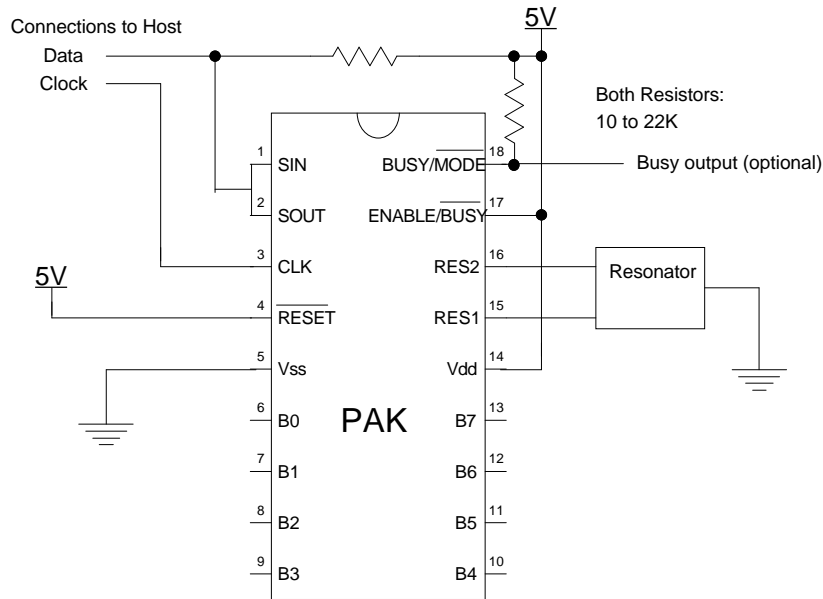
Forces the specified pin to be an output and then reverses its state (that is makes a 1 a 0 and a 0 a 1). Always returns 0.

WRITE

This command writes a byte to the PAK-III's EEPROM. Data stored here will remain indefinitely, even surviving power loss. Send an address byte (from 0 to 63) and the data byte. The PAK-III will return a 0. Please note that accessing EEPROM is relatively slow, so be sure to wait for the PAK-III's acknowledgement before sending more commands. Also keep in mind that there is a maximum number of write cycles the EEPROM can handle. Once you write to a cell a certain number of times (between 1 million and 10 million writes) it will fail. This is typical of all EEPROM.

Typical Circuits and Software

Connecting the Pak-III to its external components is simple. Supply 5V to Vdd and ground the Vss pin. 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. For the simplest mode of operation connect the ENABLE/BUSY pin to 5V and the BUSY/MODE pin to 5V through a pull up resistor (10K-22K). Of course, you'll also need to make connections to the host microprocessor. You may also want to make different connections to ENABLE/BUSY and BUSY/MODE if you want to use different methods of synchronizing.



Example 1. Connections to Stamp or similar host

If you are using the Basic Stamp II or Basic Stamp IISX, you'll have no problems using the ShiftIn and ShiftOut commands. Many Basic Stamp-compatible compilers have these commands too. If you don't have access to these commands, don't worry. Here is some simple code (written in simple PBasic) that shows the actions you have to take:

```
' B1 is the byte to shift
shiftoutput:
output datapin
for b7=0 to 7
' Set data pin to 0 or 1
  low datapin
  if (b1 & $80) <> $80 then so0
  high datapin
so0:
  b1=b1*2          ' shift byte left
  high clkpin
  low clkpin       ' could use pulsout
next
return
```

```
' B1 is byte to input byte
shiftinput:
input datapin
b1=0
for b7=0 to 7
  b1=b1*2
  b1=b1 | in15
  high clkpin
  low clkpin      ' could use pulsout
next
return
```

You'll find a library of Basic Stamp II routines that allow you to access the PakIII on the companion disk. Here is a simple program that generates random numbers while blinking an LED connected to the PakIII's B4 pin and scanning a button on B3:

```
agn: xpin=4      ' toggle pin 4
gosub xtoggle
gosub xgetnumber ' get random number & print it
debug hex2 xoutvalue,cr
xpin=3          ' check switch on pin 3
gosub xread
if xoutvalue=1 then nobz ' if switch up, no buzz
xpin=5          ' switch down: buzz on B5 speaker
xA=1000
xB=500
gosub XFreqout
nobz:
goto agn       ' do it all again
```

Frequently Asked Questions

Q: Can the PAK-III work while the host does other operations?

A: Of course! Just send your commands and proceed without waiting for the response code. Be sure to read the response before sending another command (unless you reset the PAK-III).

Q: Can I run the Pak III from a different clock source?

A: Yes. But it isn't a good idea because many of the commands are clock-sensitive.

Q: Is there a second source for the PakIII?

A: No. However, if you have a high-volume application and you are concerned about availability, contact AWC about obtaining a license to produce your own Pak III's or obtain them from third parties. We can also make SOIC and high temperature versions available as a special order item.

Q: How can I get more I/O pins?

A: Look for our Pak IV, coming soon.

Basic Stamp II Library

On the enclosed disk you'll find FUTIL.BS2 which contains a simple library for the PAK-I. Here are the calls in that file:

Call	Function	Notes
XCount	Count pulses	Time in xA, return in Xoutvalue
XEERead	Read EEPROM	Address in xA, return in Xoutvalue
XEEWrite	Write EEPROM	Address in xA, byte in xB
XFreqOut	Generate tone	Freq=xA, Duration=xB
XGetNumber	Get time-sensitive random number	Result in Xoutvalue
XHigh	Set pin high	Pin in Xpin
XInput	Set pin to input	Pin in Xpin
XLow	Set pin low	Pin in Xpin
XOutput	Set pin to output	Pin in Xpin
XPulseIn0	Read low pulse	Pin in Xpin, result in Xoutvalue
XPulseIn1	Read high pulse	Pin in Xpin, result in Xoutvalue
XPulseOut	Generate pulse	Pin in Xpin, duration in xA
XPWM	Generate PWM	Pin in Xpin, duty cycle in xA, duration in xB
XRCTime0	RCTime low	Pin in Xpin, return in Xoutvalue
XRCTime1	RCTime high	Pin in Xpin, return in Xoutvalue
XRead	Read input	Pin in Xpin, result in Xoutvalue
XReset	Reset the coprocessor	Always call first
XShiftIn	Shift data in (MSBFIRST)	Pin in Xpin, 8 bit result in Xoutvalue
XShiftOut	Shift data out (MSBFIRST)	Pin in Xpin, 8 bit data in xA
XToggle	Toggle pin	Pin in Xpin

Specifications

Absolute Maximum Ratings

Ambient temperature under bias	-55°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on VDD with respect to VSS	-0.3 to +7.5V
Total power dissipation (1)	800 mW
Maximum current out of VSS pin	150 mA
Maximum current into VDD pin	100 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	20 mA
Maximum current sunk by B0-B7	150 mA
Maximum current sourced by B0-B7	100 mA

DC Characteristics

Parameter	Minimum	Typical	Maximum
Supply voltage	4.5V	5V	5.5V
Vdd rise time on power up	.05V/ms	-	-
Supply current @ 10Mhz	-	5mA	10mA