RAMPack B

Serial RAM Module
*Byte/Block Addressing
*Sequential Buffer Mode
*Multiple Baud Rates
*8K to 32K Bytes of RAM

SOLUTIONS CUBED
256 East 1st Street
Chico, CA 95928
phone (530) 891-8045
fax (530) 891-1643
www.solutions-cubed.com

MINIATURE ENGINEERING MODULES

RAMPack B
Serial RAM Module

FEATURES

- Addressable writing
- Byte or block reading
- FIFO (first-in-first-out) buffer mode
- Easy to use 1 or 2 wire serial interface
- 1.2K, 2.4K, 4.8K, 9.6K, or 19.2K baud rates supported
- Expandable to handle 32Kx8 bytes of RAM
- RAM socketed for easy replacement
- No external components
- Easy to use SIP package

DESCRIPTION

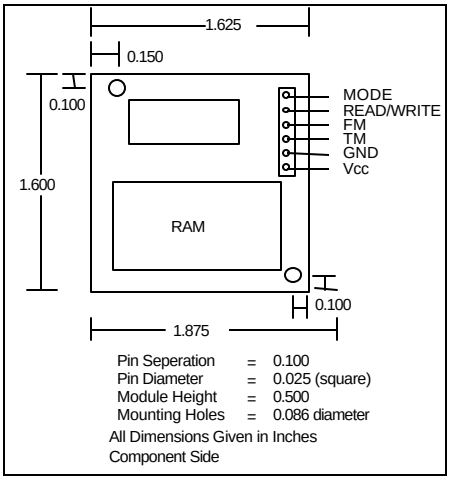
The RAMPack B module maintains 8Kx8 bytes of RAM. The module makes use of buffers and an on board microcontroller to allow addressable access to the RAM using only one or two i/o lines.

An automatic baud rate detection scheme allows the user to use a variety of baud rates without setting hardware jumpers. The RAMPack B supports 1200, 2400, 4800, 9600, and 19200 baud rates.

The 8Kx8 bytes of low cost static RAM comes socketed. If the user requires non-volatile RAM(RAM whose data stays valid after power is removed), a NVRAM can be purchased and inserted into the socket. A software command allows the RAMPack B to be expanded to handle 32Kx8 RAM.

An additional buffer mode has been implemented in the RAMPack B. It is a FIFO(first-in-first-out) buffer. Bytes of data may be sent to the device and each byte will be stored sequentially in RAM. Initiating a read in FIFO mode will cause the RAMPack B to send all of the stored data to the master device. No commands or addressing are required to store data with FIFO mode. The FIFO mode does require a 4 pin interface to control data flow.

PIN CONFIGURATION AND MECHANICAL SPECIFICATIONS



MODE	Has weak pull up resistor(200k ohm). Set or unconnected for normal operation. Pulled low for FIFO mode.
READ/WRITE	Has weak pull up resistor(200k ohm). Only used in FIFO mode. Set for a buffer READ. Clear for buffer WRITE.
FM	Data received from master on this pin.
TM	Data sent to master on this pin.
GND	Ground potential, all grounds are common.
Vcc	Supply for control circuitry.

CUBED SOLUTIONS
530-891-8045

SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

note: These are stress ratings only. Stresses above those listed below may cause permanent damage and/or affect device reliability. The operational ratings should be used to determine applicable ranges of operation.

Storage Temperature	-45°C to +85°C
Operating Temperature	0°C to +70°C
Supply Voltage(Vcc)	0 to 7.0V
Voltage on TM, FM, GND, MODE, READ/WRITE pins	-0.6V to (Vcc+0.6V)

DC ELECTRICAL CHARACTERISTICS

At T_A = 25°C and Vcc = 5.0V unless otherwise noted.

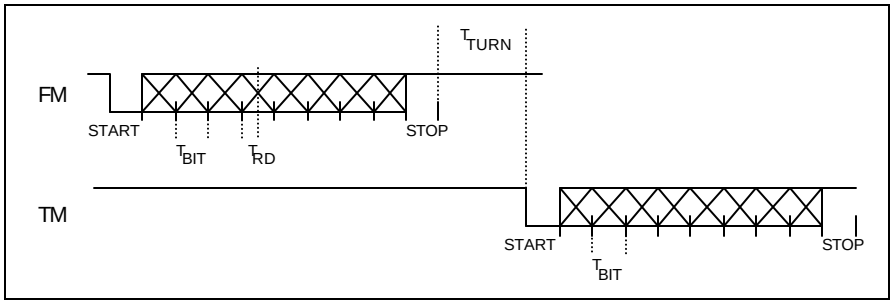
Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Supply Voltage	Vcc	4.5		5.5	V	
Vcc rise time to ensure good reset	SVdd	0.05			V/ms	If this is not met, the RAMPack B may start up in an unknown state and may not communicate correctly.
Supply Current	Icc		1	80	mA	Maximum can occur during a write/read. It will last only 20us per byte.
FM Input Low Voltage	V_{IL}	GND		0.2Vcc	V	
FM Input High Voltage	V_{IH}	2.0 0.2Vcc+1V		Vcc	V	4.0<Vcc<5.0 Full Vcc range User may use better of two specs.
TM Output Low Voltage	V_{OLTM}			0.6	V	
TM Output High Voltage	V_{IHTM}	Vcc			V	TM is open collector
TM Output Pull Up current	I_{TMPU}	2.5	5.0	5.5	mA	TM open collector is tied to Vcc with a 5% 1kΩ resistor.
Vcc for RAM data retention	V_{RET}	2			V	Microcontroller drops out at 3V

Note: “Typ” values are for design guidance only and are not guaranteed.

AC ELECTRICAL CHARACTERISTICS

At T_A = 25°C and Vcc = 5.0V unless otherwise noted.

Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Communication bit period	TBIT				µS	The bit period is determined by an on-board oscillator, and is temperature sensitive
1200 baud		826	832	838		
2400 baud		413	416	419		
4800 baud		206	208	211		
9600 baud		103	104	105		
19200 baud		51	52	53		
Offset when a bit is read	TRD				µS	This is used to ensure a bit is valid when read. A bit must be valid for at least this long in order for the communication to not be erroneous
1200 baud		360	400	440		
2400 baud		180	200	220		
4800 baud		90	100	110		
9600 baud		45	50	60		
19200 baud		22	25	27		
Time for a command from master to be responded to	TTURN	450	500	550	µS	This time is used to allow for a master to change from transmission mode to reception mode
Time for module to detect a start bit after entering FIFO write mode	TVAL	12	16	20	µS	
Baud generator error	BGE		2	6	%	
Time FM needs to be high to end a Write command	TWREND				mS	To end a Write command the FM pin must be stable high for a period greater than T_{BIT} * 20
1200 baud			16.64			
2400 baud			8.32			
4800 baud			4.17			
9600 baud			2.08			
19200 baud			1.04			



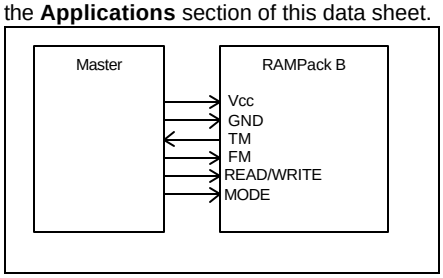
Communication Timing

OPERATION

The RAMPack B provides simple means of interfacing to RAM. Because there are no external components necessary for use, the RAMPack B is an extremely easy device to use.

Hardware Hook up

The connection diagram below shows the basic setup for using the RAMPack B. Information about using the module with only one serial line is given in the **Applications** section of this data sheet.



Basic connection diagram

Power(Vcc) must be supplied to the RAMPack B from either a master processor or an external supply. Both the master's ground and the RAMPack B's GND pin must be at the same potential. As the diagram shows, the TM pin on the RAMPack B provides the communication path to the master from the RAMPack B; while the FM pin on the RAMPack B provides the communication path from the master to the RAMPack B. The MODE and READ/WRITE pins can be left unconnected when the FIFO buffer is not in use.

FIFO BUFFER MODE

The RAMPack B has a FIFO buffer mode built into it. FIFO stands for first-in-first-out. This buffer allows the user to store bytes of data seqentially in RAM. Written data is not held to the timing requirements of a write command. In FIFO mode the MODE and READ/WRITE pins take the place of sync bytes and commands. As long as the RAMPack B is in FIFO buffer write mode it is looking for data to store. You can send a string of data, and then wait for a period of time, and send a new string. Data received in FIFO buffer mode is written first at address '00 00'h. Each additional byte of data is written to the next address location. If the RAM is filled during a FIFO write then the additional data is rolled over and starts again at '00 00'h. If the RAMPack B is in expanded RAM mode then this rollover occurs after 32,768 bytes of data are written to RAM. After a write operation is completed the data stored in RAM can be accessed with a FIFO read operation. The FIFO read will send all data that was stored during a previous write operation to the Master unit. Data is sent as a stream at the baud rate that the RAMPack is set for. There is a 500us delay between the MODE pin going low and the data being sent on the TM pin. The default baud rate for FIFO mode is 2400 baud. The FIFO Baud Set command can be used to change the default baud rate. The MODE and READ/WRITE pins are used to activate the FIFO buffer mode. See AN-052 for detailed source code for utilizing the FIFO buffer.

The FIFO buffer mode makes use of an address pointer that is cleared after a FIFO read operation. If the RAMPack B is going to make use of both the FIFO buffer mode and the regular addressing operations, such as Write, Byte Read, or Block Read, then the user will have to be aware of this pointer. See AN-054 for some insight into resetting the address pointer when using a FIFO write operation.

EXPANDING MEMORY

The RAMPack B can be used with 32Kx8 SRAM and NVRAM. To expand the RAM use the Expand RAM software command. On power up the RAMPack B is set to interface to 8Kx8 RAM. If the larger 32Kx8 RAM is used simply send the Expand RAM command after power up. Normal RAM allows access to addresses from '00 00'h -'1F FF'h, expanded RAM allows access to addresses ranging from '00 00'h to '7F FF'h. To use the Expand RAM command you must replace the 8Kx8 RAM with a 32Kx8 RAM chip. See AN-053 for an example of code implementing software expanded RAM . A 32Kx8 SRAM chip is the Mosel Vitelic V62C518256LL-70P. This part is available from Arrow Electronics.

COMMANDS

Write Command ('00'h) In order to use the Write command you must send the RAMPack B the sync byte, Write command byte, and the two byte address to start writing at. The four bytes required by this command, as well as some data bytes, are displayed as a timing diagram on page 6 of this document. Data is sent to the RAMPack B on the FM pin(From Master). The RAMPack B will determine the baud rate to use from the sync byte. The first byte sent is the sync byte('55'h) which is followed by the Write command('00'h). The next two bytes are the high and low address bytes. The address bytes should be between '00 00'h and '1F FF'h for normal operation. When in expanded RAM mode, using 32Kx8 RAM, the address range is extended from '00 00'h to '7F FF'h. Writing beyond the values detailed above will cause the RAMPack B to roll the address over and overwrite the contents starting a '00 00'h. Data bytes to be written follow the address bytes.

A Write command ends when the FM line has been high for a time period greater than 2 byte lengths, calculated from the baud rate currently in use. This can be easily done with a delay loop or pause routine. To calculate this delay use the following equation.

$$\textbf{Delay} = T_{BIT} * 20$$

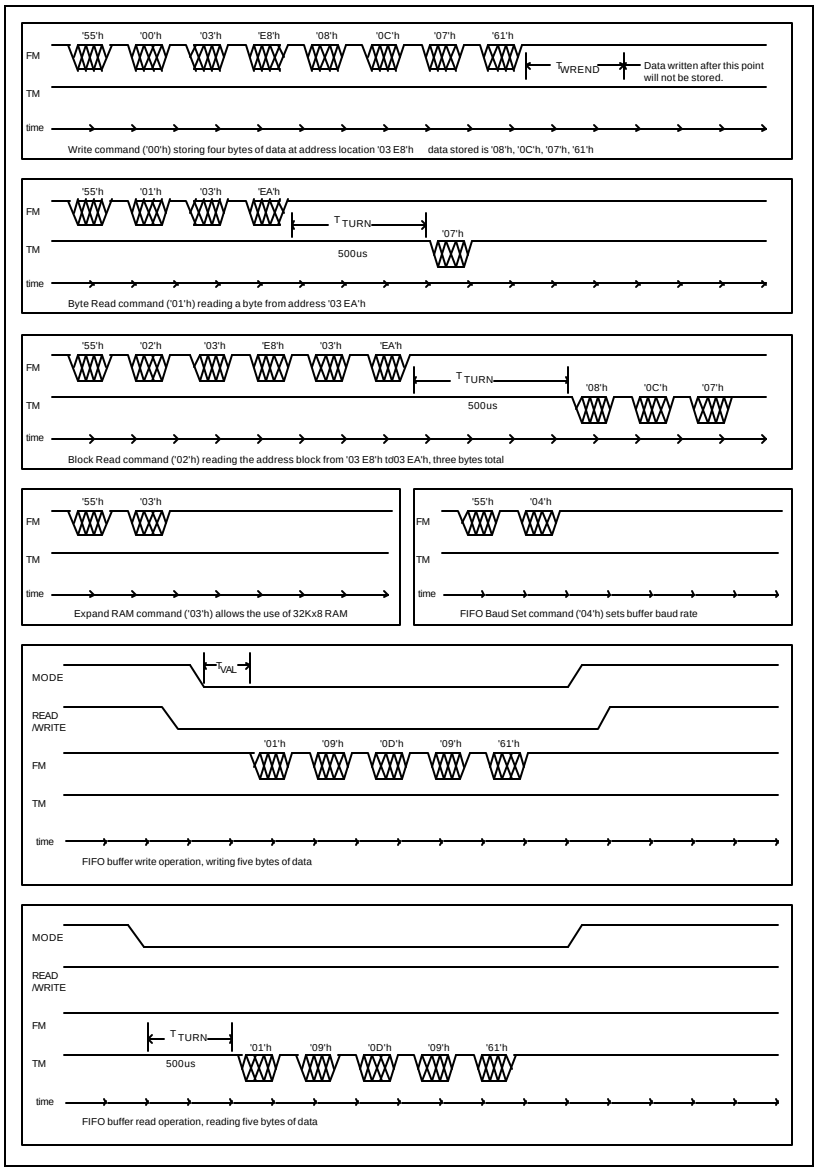
Byte Read Command ('01'h) To initiate a Byte Read command the sync byte through address bytes are sent on the FM pin. The first byte sent is the sync byte('55'h) which is followed by the Byte Read command('01'h). The next two bytes are the high and low address bytes. After the address bytes are received, the RAMPack waits for 500us. When this period elapses the data byte will be sent LSB first on the TM(To Master)pin. The byte will be read from the location pointed to by the address bytes that were previously sent.

Block Read Command ('02'h) To initiate a Block Read command the sync byte through address bytes are sent on the FM pin. The first byte sent is the sync byte('55'h) which is followed by the Block Read command('02'h). The next two bytes are the high and low address bytes that the block read will start at. Following the start address bytes the user must send two bytes that point to the end of the block read. After the end of read address bytes are received, the RAMPack waits for 500us. When this period elapses the data bytes will be sent LSB first on the TM(To Master)pin. The data block read will begin at the start address and end at the end address.

Expand RAM Command ('03'h) The expand RAM command is a software command used to configure the RAMPack B to handle 32Kx8 byte RAM chips. This allows the end user to store four times more information than with an 8Kx8 RAM chip. In order to utiliize this function the 8Kx8 RAM that is sold with the RAMPack B will have to be replaced with a 32Kx8 RAM chip. Mosel Vitelic part number V62C518256LL-70P is one example of a 32Kx8 RAM chip. If the larger RAM chip is used in this design then the user should configure the RAMPack B for expanded RAM mode as soon as the device is powered up.

FIFO Baud Set Command ('04'h) The FIFO baud set command is used to change the default baud rate for the FIFO buffer mode. When the RAMPack B is powered up the default baud rate for the FIFO buffer is 2400. To change this baud rate just send the FIFO Baud Set command preceded by a sync byte to the RAMPack B. The baud rate that you send this command at is the baud rate that the FIFO buffer will operate in. Each time the device is powered up the baud rate for the FIFO buffer mode will revert to 2400. The FIFO Baud Set command must be sent to the RAMPack B while it is not in FIFO buffer mode.

Any valid command received by the RAMPack B will effectively modify the baud rate for the FIFO buffer. So you may set the baud rate with a Byte Read, Read Pointer, Write Pointer, etc.



Command Set Examples

Write Pointer Command ('05'h) The Write Pointer command is sent to the RAMPack B sync byte first, followed by the command byte('05'h), followed by the high and then low address bytes destined for the address pointer. The RAMPack B makes use of address pointers to determine where write or read operations should occur in RAM. For instance, in the Write command, bytes three and four represent the address location that the command will start the Write operation at. These address values are loaded into an address pointer inside of the RAMPack B. Any time that you write or read from RAM address information is required for the command to function.

The only time that you will likely need to use the Write Pointer command is when you are using the FIFO buffer as well as addressed Write or Read commands. When the FIFO buffer is used to store data the address pointer is reset to '00 00'h after a FIFO read occurs. In this way FIFO write and read operations work in conjunction to allow the user to make use of their functions with no addressing operations. But if you wanted to store data with the FIFO buffer, and then use a Write or Read command you would corrupt the addressing scheme for the FIFO buffer. The next FIFO write would occur at the last address that the Write or Read command accessed.

The Write Pointer command can be used to set the FIFO buffer to start writing at any RAM location. It may also be used to reset the starting address to '00 00'h after another command has modified the internal address pointers. See AN-055 for applied use of the Write Pointer command.

You should not send the RAMPack B address pointer locations outside the range of addresses available within the RAM being used.

Read Pointer Command ('06') The Read Pointer command is sent starting with a sync byte, followed by the command byte('06'h). After a 500us delay the RAMPack B will return the contents of this address pointer to the Master unit. Data is returned as two bytes with the high byte returned first. The Read Pointer command can be used to read an address pointer in the RAMPack B. The address pointer read is related to the last memory location accessed in RAM. If used after the Write or Block Read commands the address pointer will be incremented by one from the last address accessed.

The Read Pointer command can be used to determine how many bytes were written to RAM during a Write command. In most instances the user will know how many bytes have been written. But in those cases where the information is not available to the Master unit it can be read using the Read Pointer Command.

Read End of FIFO Command ('07') The Read End of FIFO command is sent starting with a sync byte, followed by the command byte('07'h). After a 500us delay the RAMPack B will return the contents of this address pointer to the Master unit. Data is returned as two bytes with the high byte returned first.

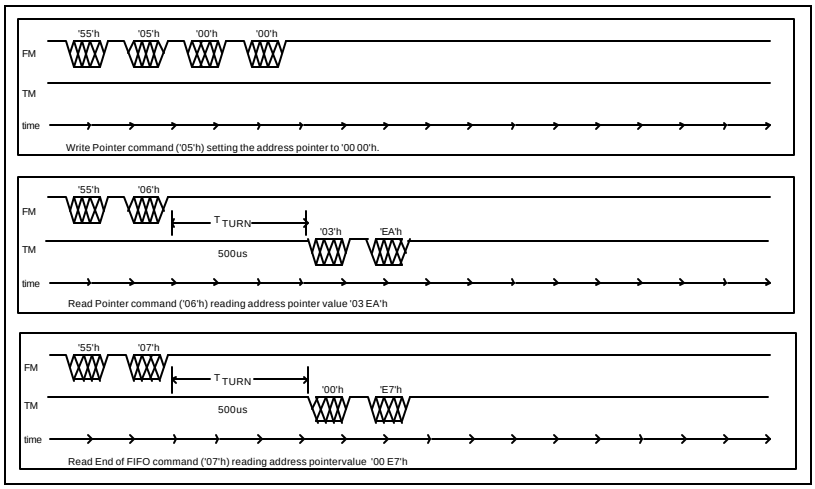
The Read End of FIFO can be used to set up the Block Read of data written to RAM using the FIFO buffer. For example, suppose your Master unit is only capable of reading and storing five bytes of data. You could not use a FIFO read operation if 1000 bytes of data were written to RAM using the FIFO buffer(A FIFO read operation would return a string of 1000 bytes of data, and then reset the address pointers). But with a Read End of FIFO command you may determine the last address location accessed by the FIFO write operation and read data in blocks of five bytes, until all data has been loaded.

Command Structure

note: all values are given in decimal unless otherwise noted.

Command	Byte Sent
WRITE	'00'h
BYTE READ	'01'h
BLOCK READ	'02'h
EXPAND RAM	'03'h
FIFO BAUD SET	'04'h
WRITE POINTER	'05'h
READ POINTER	'06'h
READ END OF FIFO	'07'h

Table of commands



Command Set Examples

EXAMPLES OF POINTER OPERATIONS

The main Address pointer and the End of FIFO pointer are shown. The tables depict both new values loaded into RAM and the memory location that the two pointers point to after a particular operation.

RAMPack B power up:

hex address	**hex value at address**	**address pointer**
00 00	xx unknown value	← End of FIFO, Address

After a FIFO write operation writing ‘EF’h, and ‘02’h

hex address	**hex value at address**	**address pointer**
00 00	EF	
00 01	02	
00 02	xx	← End of FIFO, Address

After a Byte Write to address ’00 03’h of byte ‘DD’h

hex address	**hex value at address**	**address pointer**
00 00	EF	
00 01	02	
00 02	xx	← End of FIFO
00 03	DD	
00 04	xx	← Address

After a FIFO read

hex address	**hex value at address**	**address pointer**
00 00	EF	← Address
00 01	02	
00 02	xx	←End of FIFO

COMMUNICATION PROTOCOL

Communication with the RAMPack B is accomplished with a two-wire (labeled TM and FM), asynchronous, serial communication channel. The FM pin carries data and commands from the master device to the RAMPack B. The TM pin carries data and commands to the master device from the RAMPack B. All communication is 8N1, least significant bit first, 1 start bit, and 1 stop bit. The RAMPack B accommodates 1200, 2400, 4800, 9600, and 19200 baud. All communication must be initiated by the master processor. Every communication must be started with a '55'h sync byte. This allows the RAMPack B to automatically sync on the baud rate. After this sync byte, the command may be sent along with any additional information, if necessary. The RAMPack B will ignore all incoming data, until it sees a '55'h sync byte. Responses from the RAMPack B to the master do not use the sync byte. Any response that the RAMPack B sends to the master will be at the most recent baud rate used by the master.

OPERATING AT 19,200 BAUD

The RAMPack B makes use of a low cost ceramic resonator. The resonator operates at a relatively low frequency of 4Mhz. All serial timing is based on this 4Mhz "clock". The accuracy of these low cost resonators can vary from part to part. They are also susceptible to frequency shifts over temperature. Operating the RAMPack B at 19,200 baud provides little room for error. If you seem to be having trouble running the RAMPack B at 19200 baud try shifting the master's baud rate slightly

All other baud rates operate without this concern.

ABOUT RAM

There are lots of types of RAM(random access memory). Most RAM is volatile RAM. This means that when power is removed from the circuit the data is lost. NVRAM, or non-volatile RAM, is battery backed up RAM. When power is removed NVRAM does not lose it's data. Dallas Semiconductor makes both 8Kx8(DS1225) and 32Kx8(DS1230) NVRAM. Both of these parts can be purchased from Jameco, and are compatible with the RAMPackB. RAM does not have limitations on the amount of times that a particular address can be written to, unlike other types of memory storage technology. One of the primary disadvantages of RAM is the number of pins required to correctly address a RAM chip(usually more than 20). With the RAMPack B access to RAM is reduced to 1 or 2 pins(or 4 for FIFO buffer mode).

DIFFERENCES BETWEEN RAMPACK AND RAMPACK B

The command structures and methods of communication between the old RAMPack and the RAMPack B are similar. The new FIFO mode was not available in the previous version of the RAMPack. Also new to the RAMPack B is the Expand RAM command that allows the user to communicate with 32Kx8 bytes of RAM.

RAMPack B Upgrades

- Greater baud rate range
- First-in-first-out buffer mode
- Smaller size
- PCB/breadboard suitable module

Converting Master Software

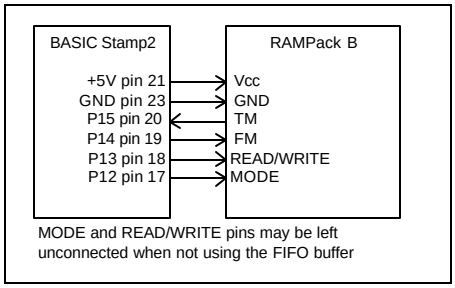
When using the Write command the user no longer needs to send the number of bytes that are being written. A new method of determining when a write operation has been completed is now being used in the RAMPack B. See the Write command for details on this.

A new Byte Read command allows the user to read just one byte of data from a specific address. The Block read can be used to read multiple, or blocks, of bytes. The user is no longer restricted to reading just 255 bytes of data, as was the case with the RAMPack.

APPLICATIONS

The following examples show how to interface the RAMPack B to various master processors in various configurations. These examples should make it fairly easy for an end user to custom design their own programs and uses for the RAMPack B.

AN-051 'Interfacing to a Parallax BASIC Stamp II', AN-052 'Using the FIFO buffer', AN-053 'Single Wire Interface with the BASIC Stamp 1/Expanded RAM', and AN-054 'Using the FIFO Buffer and Addressed Read and Write' are included here.



Hook-Up diagram for AN-051 and AN-052

Software Listing AN-051 ' Interfacing to a BASIC Stamp II'

This routine writes a byte of data to a random address and then reads the data back. This Write - Byte Read operation occurs at 1200 baud. After this occurs the data byte is incremented and written to RAM at the next three address locations. This data is then read back with a Block Read. This Write - Block Read operation takes place at 2400 baud. Tests for the accuracy of the data are included in the code. The randomizing of the address, which occurs in the MIX_IT routine, is simplified by ensuring that the address counter won't roll-over in the RAMPack B.

```
            INPUT     15
            OUTPUT    14
            VALUE     VAR       BYTE
            VALUE               = $00                 'Initial values of variable registers
            b4                  = $00
            b5                  = $00
START:
            HIGH      14        'Set FM high
            PAUSE     3000
MORE_STUFF:
            SEROUT    14,813,[$55,$00,b5,b4,VALUE]    'Write a byte then read it back at 1200 baud
            PAUSE     20
            SEROUT    14,813,[$55,$01,b5,b4]
            SERIN     15,813,100,NOISE2,[b15]
DISPLAY:
            DEBUG     ISHEX2 b5,TAB
            DEBUG     ISHEX2 b4,TAB
            DEBUG     ISHEX2 VALUE,CR,CR
            if  VALUE <> b15 then NOISE
            VALUE     = VALUE +1
            b4        = b4 + 1
            b6        = b4 + 2                        'Write three bytes then read them back at 2400 baud
            SEROUT    14,396,[$55,$00,b5,b4,VALUE,VALUE,VALUE]
            PAUSE     10
            SEROUT    14,396,[$55,$02,b5,b4,b5,b6]
            SERIN     15,396,100,NOISE2,[b15,b14,b13]
            if VALUE  <> b15 then NOISE
            if VALUE  <> b14 then NOISE
            if VALUE  <> b13 then NOISE
            DEBUG     "BLOCK READ OK",CR,CR
MIX_IT:
            RANDOM    W2
            IF b4     > $FC THEN MIX_IT
            IF b5     > $1F THEN MIX_IT
            GOTO      MORE_STUFF
NOISE:
            DEBUG     BELL
```

```
DEBUG       "DATA RETURNED BAD",CR
            GOTO        DONE
NOISE2:
            DEBUG       BELL
            DEBUG       "NO DATA RETURNED",CR
DONE:
END:
```

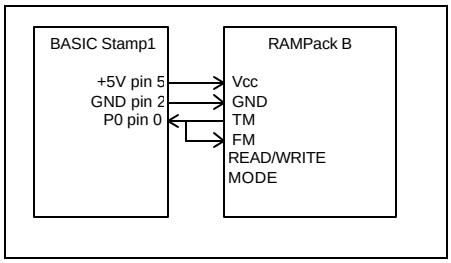
Software Listing AN-052 'Using the FIFO Buffer'

This routine writes 7 bytes of data to the FIFO buffer at 9600 baud. 1 byte is sent every 50ms. After the data is sent the routine performs a FIFO read and checks the data read versus the expected data. If everything checks out, the routine displays the variable VALUE, increments the variable, and returns to loop through the program again. A BASIC Stamp II is the master.

```
            INPUT                   15
            OUTPUT                  14
            OUTPUT                  13
            OUTPUT                  12
            VALUE                   VAR         BYTE
'Initial values of variable registers
            VALUE                   = $00
            b4                      = $00
            b5                      = $00
START:
            HIGH        14          'FM pin
            HIGH        13          ' read/write pin(low for write)
            HIGH        12          'mode pin(low for buffer mode)
            PAUSE       3000
                                    'FIFO Baud Set for 9.6k baud
            SEROUT      14,84,[$55,$04]
            PAUSE       100
MORE_STUFF:
            LOW         13          'enter "write" mode
            LOW         12          'enter "buffer" mode
                                    'send 7 bytes;  1 every 50ms
            SEROUT      14,84,50,[VALUE,$55,$99,$88,$32,$57,$69]
            HIGH        12          'exit "buffer" mode
            PAUSE       2
            HIGH        13          'enter "read" mode
            LOW         12          'enter "FIFO" mode
                                    'read in 7 bytes of data
            SERIN       15,84,100,NOISE,[b15,b14,b13,b12,b11,b10,b9]
            HIGH        12          'exit "FIFO" mode
DISPLAY:
            DEBUG       ISHEX2  VALUE,CR
            if VALUE    <> b15 then NOISE
            if b14      <> $55 then NOISE
            if b13      <> $99 then NOISE
            if b12      <> $88 then NOISE
            if b11      <> $32 then NOISE
            if b10      <> $57 then NOISE
            if b9       <> $69 then NOISE
            VALUE       = VALUE +1
            GOTO        MORE_STUFF
NOISE:
            DEBUG       BELL
            DEBUG       "READ FAILED"
END:
```

AN-053 'Single Wire Interface with the BASIC Stamp 1/Expanded RAM'

The RAMPack B uses a two wire serial interface for communication. However, its TM pin is pseudo open-collector. This coupled with the fact that the FM pin and the TM pin never communicate simultaneously allows the RAMPack B to communicate via 1 wire if the master is able to support this mode. In order for 1 wire communication to work, the master processor must be able to both send and receive with one I/O line. Because of this, a standard PC serial port is not able to work with just 1 wire. However, most microcontrollers can support this mode. This example shows how to interface the RAMPack B to a Parallax BASIC Stamp I with only one wire. The diagram on the next page shows the hardware connections necessary. Connections to the mode control pins, if FIFO mode is used, are identical to those of other application notes. In addition to a single line interface AN-053 implements the Expand RAM command that allows the RAMPack B to be used with 32Kx8 byte RAM chips. From this application note you can derive three things, how to use a 1 wire interface, use of the Expand RAM command, and the command structure for use with the BASIC Stamp I.




```
START:
SYMBOL    NEWDAT1                 =         B0          '
SYMBOL    DAT1                    =         B1          '
SYMBOL    ADDR_HI                 =         B2          '
SYMBOL    ADDR_LO                 =         B3          '
          HIGH      0                                   'Make sure FM pin is high
          PAUSE     500                                 'Pause 500ms
          SEROUT    0,T2400,($55,$03)                   'Set up for 32Kx8 RAM
          PAUSE     1000
AGAIN:
          RANDOM                  W1                    'Randomize address
          IF ADDR_HI > $7F THEN AGAIN                   'Max address is 7Fxx
SEND_BYTE:
          SEROUT    0,T2400,($55,$00,ADDR_HI,ADDR_LO,DAT1)
          PAUSE     10                                  'Delay for Write to end
RECEIVE_BYTE:
          SEROUT    0,T2400,($55,$01,ADDR_HI,ADDR_LO)
          SERIN     0,T2400,NEWDAT1                     'Read in data
          DEBUG     $ADDR_HI,CR                         'Display address
          DEBUG     $ADDR_LO,CR,CR
          IF NEWDAT1 <> DAT1 THEN PROBLEM               'Make sure write matches read
          DAT1      =         DAT1 +1                   'Change write data
          GOTO      AGAIN
PROBLEM:
          DEBUG     "BAD DATA"                          'DAT1 did not match NEWDAT1
END:
```

AN-054 'Using the FIFO Buffer and Addressed Read and Write'

The FIFO buffer mode is designed in such a way as to reset the address pointer after every FIFO read operation. The addressable Write, Byte Read, and Block Read commands do not do this. Because of this if a Write, Byte Read, or Block Read is used prior to a FIFO write operation the buffer will begin writing at one address location above the last address accessed. For instance, if three bytes were written starting at address '10 00'h, and then a FIFO write was implemented, it would begin at address location '10 04'h. A subsequent FIFO read operation would return all data from '00 00'h to the end of the previous FIFO write.

Some users may want to use upper memory addresses to store various bytes of data, and at the same time use the lower address location for FIFO write and FIFO read operations. In order to do this you just need to reset the address pointer to '00 00'h prior to any FIFO write operation. A Byte Read command accessing address location '00 00'h, followed by a FIFO write operation is the fastest way to reset the address pointer to '00 00'h without using the Set Pointer command. The following BASIC Stamp II code snippet can be used to do this without corrupting any data that is already stored. This piece of code should be located prior to any code implementing a FIFO write operation.

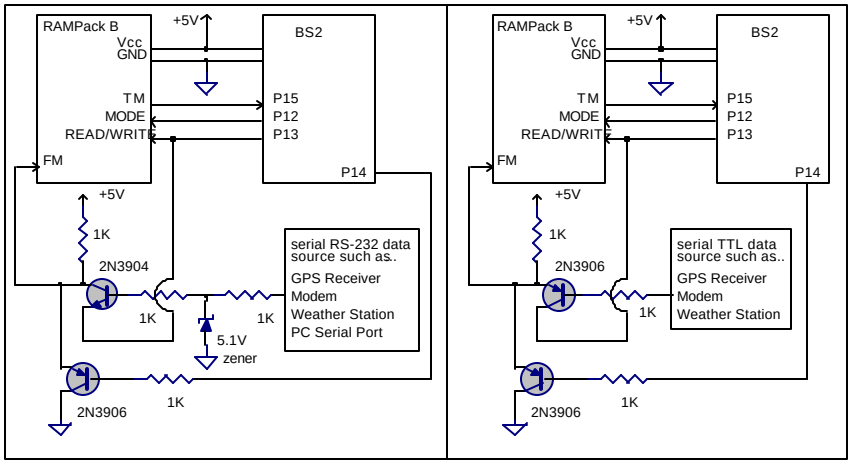
```
SEROUT      14,84,[$55,$01,$00,$00]     'Read a byte at '00 00'h, 9600 baud
LOW  13                                 'Pull READ/WRITE pin low
LOW  12                                 'Pull MODE pin low
PAUSE       20                          'Wait for single byte to transfer
HIGH 12                                 'Let READ/WRITE pin return high
HIGH 13                                 'Let MODE pin return high
```

After this code is implemented the address pointer in the RAMPack B is reset to '00 00'h. Using the Byte Read command to point to addresses, it is possible to store multiple FIFO write operations at different address locations. If this were done, and the length of a FIFO write were known, then the Block Read command could be used to retrieve the data.

AN-055'Interfacing to the BS2 and a serial data device'

This application makes use of the FIFO buffer mode, a simple interface circuit, and some various commands. The circuit and software can allow the BS2(Basic Stamp II)to control a serial bus. This circuit actually serves to automate serial data collection.

In both circuits below the serial data generator has access to the RAMPack B only when the Master(BS2) places the RAMPack B in FIFO write mode. The Master device always has access to the bus. Both of these circuits are low cost and easy to build. The circuit on the left provides an interface to RS-232 level signals. This circuit was tested for functionality on a PC port. We do recommend that any users research the effectiveness of these circuits on their particular serial data generator prior to use. The circuit depicted in the left schematic could also be used to invert data formats. The circuit on the right provides the same bus control capabilities as that on the left, but can be used with TTL level signals. In both circuits the READ/WRITE pin is connected to the leg of the serial data transistor at the point which would normally be grounded. When the BS2 places the RAMPack B in FIFO write mode, READ/WRITE is pulled to ground, then the serial data device has access to the FM pin. At any time when the READ/WRITE pin is not at ground potential the serial data device does not have access to the RAMPack B. This provides a simple method for the BS2 to switch back and forth from the buffer mode and check for new data.



Hardware Connection for AN-055

'AN-055 Software Listing

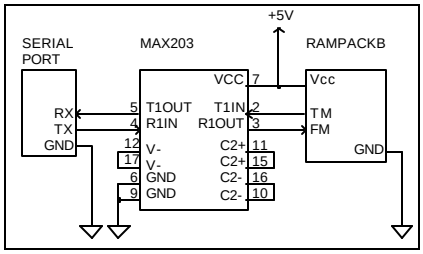
'This routine monitors the RAMPack B to see if serial data has been loaded into it. The serial data is tested for every 15 'seconds. If serial data has been written to the RAMPack B then the data is read out of the RAMPack B and displayed to the 'DEBUG screen until a carriage return code($0D) is read from RAM. The BS2 uses the Read End of FIFO command to see if 'data has been loaded into RAM. If the FIFO address pointer is at '00 00'h then no data was written to the buffer during the 'previous 15 second interval. When data has been written to RAM the BS2 uses the Byte Read command, instead of a FIFO 'read, to read one byte at a time. Because a FIFO read is not used the address pointer must be reset before the FIFO write 'mode can be used again. This is done simply with the Write Pointer command. The address pointer is set to '00 00'h before a 'FIFO write is implemented.

```
START:
        ADDR_HI         VAR         BYTE
        ADDR_LO         VAR         BYTE
        HIGH            14                              'Set FM pin
        HIGH            13                              'Set read/write pin
        HIGH            12                              'Set mode pin
        PAUSE           2000
FIFO_RESET:
        ADDR_LO         =           $00                 'Default address for read
        ADDR_HI         =           $00                 '           "
        SEROUT          14,84,[$55,$05,$00,$00]         'Reset FIFO pointer
MORE_STUFF:
        LOW             13                              'Enter FIFO write mode
        LOW             12
PAUSE                   15000                           'Load data for 15 seconds
        HIGH            12
        HIGH            13                              'Exit FIFO write mode
        SEROUT          14,84,[$55,$07]                 'Read end of FIFO pointer
        SERIN           15,84,100,NO_DATA,[b15,b14]     'Receive FIFO pointer
        IF b15 <> $00 THEN READ_MORE                    'Test for new data
        IF b14 =  $00 THEN NO_NEW_DATA
```

```
READ_MORE:
        SEROUT      14,84,[$55,$01,ADDR_HI,ADDR_LO]     'Send byte read command
        SERIN       15,84,100,NO_DATA,[b15]             'Receive byte
        DEBUG       b15                                 'Display byte
        IF b15 = $0D THEN FIFO_RESET            'Test for end of string
        ADDR_LO     = ADDR_LO+1                 'Increment address
        IF ADDR_LO = $00 THEN HIGH_ADDR
        GOTO        READ_MORE                   'Read in more data
HIGH_ADDR:
        ADDR_HI = ADDR_HI +1                    'Increment upper address
        GOTO        READ_MORE
NO_DATA:
        DEBUG       "NO DATA",CR                'Read did not occur
        GOTO        FIFO_RESET
NO_NEW_DATA:
        DEBUG       "NO NEW DATA",CR            'No new serial data in RAM
        GOTO        MORE_STUFF
END:
```

AN-056 Interfacing to a PC Serial Port

The RAMPack B can be easily interfaced to a PC serial port, thereby allowing any PC program which can control the serial port to interface to the RAMPack B. The two wire interface of the RAMPack B allows for two way communication with no special hardware configuration.



The schematic shows a Maxim 203 RS-232 level translator chip for changing the TTL levels of the RAMPack B to RS-232 levels. While there are methods available of interfacing to RS-232 levels without a level translation chip, Solutions Cubed can make no claims as to their reliability.

For a standard PC serial port use the following pin out for the serial connector. These connections should work with most computers, but you should check to make sure.

Signal Name	DB-25	DB-9
TX	2	3
RX	3	2
GND	7	5

Distributors

Digi Key
701 Brooks Ave. South
Thief River Falls, MN 56701-0677
800-344-4539
www.digikey.com

Jameco
1355 Shoreway Road
Belmont, CA 94002-4100
800-831-4242
www.jameco.com

JDR Microdevices
1850 South 10th Street
San Jose, CA 95112-4108
800-538-5000
www.jdr.com

Mondo-tronics
4286 Redwood Hwy #226
San Rafael, CA 94960
415-491-4600
www.mondo.com

Parallax
599 Menlo Drive Suite 100
Rocklin, CA 95765
888-512-1024
www.parallaxinc.com