

## AN\_002

### Using the RAMPack WRITE & READ commands with a Parallax Basic Stamp1

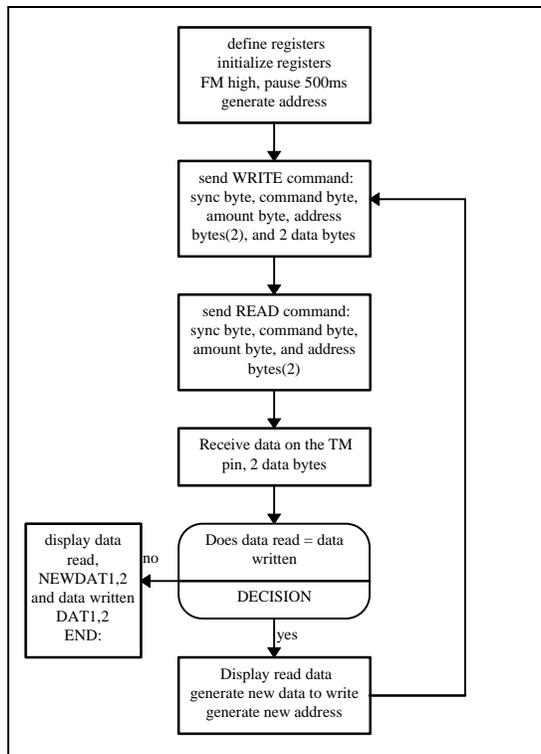
The following software listing depicts one way to utilize the RAMPack with a Basic Stamp1. This software generates a pseudo-random address for the RAMPack. At each address generated the Basic Stamp writes two data bytes at 2400 baud. The Basic Stamp then reads back the two data bytes at 2400 baud with the READ command.

Data sent to the Basic Stamp is generated by the Basic Stamp's RANDOM command(as is the address location). The two data bytes sent to the Basic Stamp are then compared to the two data bytes produced by the READ command. If these values are not equal the software jumps to a DEBUG routine that displays the written data along with the data that was read. At this point the program ends. If the data written is the same as the data read from the RAMPack then the Basic Stamp displays the data value to the PC by means of the DEBUG command. After this occurs the cycle is repeated.

This application note describes the methods to use the WRITE and READ commands. It also details a method for generating random address bytes and keeping them within the RAMPacks memory map. This software listing includes a method of generating and storing random data bytes, as well as insuring that data written to the RAMPack is the same as data read from the RAMPack.

When using the SERIN command with a Basic Stamp1, if serial data is not received the Basic Stamp will appear to lock up. Essentially the Basic Stamp1 continues to wait for data that never comes. This problem should not occur with the RAMPack. All communication has been fully tested. But if there is significant noise in your system this could occur. The Basic Stamp2 incorporates a "timeout" period, so that if data is not received the Basic Stamp2 software can resend the command.

#### Software Flow Chart



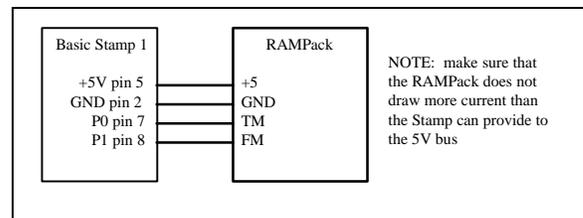
#### Software Listing

START:

```

SYMBOL NEWDAT1      =B0      'data received register
SYMBOL NEWDAT2      =B1      'data received register
SYMBOL ADDR_HI      =B2      'high address byte register
SYMBOL ADDR_LO      =B3      'low address byte register
SYMBOL DAT1         =B4      'data to write register
SYMBOL DAT2         =B5      'data to write register
HIGH 1
PAUSE 500           'pause 500ms
ADDR_HI             = $00     'initialize address bytes
ADDR_LO             = $00     'initialize address bytes
DAT1                = $00     'initialize data bytes
DAT2                = $00     'initialize data bytes
.
AGAIN:
RANDOM W1            'W1 = B3,B2 address bytes
IF ADDR_HI > $1F THEN AGAIN 'if too high try again
.
'send sync, command, amount(2), address, and data bytes data bytes
.
SEND:
SEROUT 1,T2400,($55,$00,$02,ADDR_HI,ADDR_LO,DAT1,DAT2)
.
'send READ command, receive data and store in NEWDAT1,2
.
RECEIVE:
SEROUT 1,T2400,($55,$01,$02,ADDR_HI,ADDR_LO)
SERIN 0,T2400,NEWDAT1,NEWDAT2
.
DEBUG $NEWDAT1,$NEWDAT2 'display data received
IF NEWDAT1 <> DAT1 THEN PROBLEM 'does data read = written
IF NEWDAT2 <> DAT2 THEN PROBLEM 'does data read = written
RANDOM W2              'OK, generate new data
GOTO AGAIN           'repeat cycle
.
'data not the same, display read, written data end program
.
PROBLEM:
DEBUG CR,$DAT1,$NEWDAT1,CR,$DAT2,$NEWDAT2,CR,CR
END:
  
```

#### Hook-up Diagram



**SOLUTIONS CUBED**

3029-F Esplanade  
Chico, CA 95973  
phone (916) 891-8045  
fax (916) 891-1643  
www.solutions-cubed.com