2.6.4 Using Array Variables

A single Array Variable can be visualized as a list of values. The variable name is how you access the list, and a number –or index– is used to point to any single value within the list.

The index value is enclosed in brackets and appended after the array variable's name.

```
myarray[index] = 0
```

The index can be a literal number, a variable, or an expression.

For the following examples, an array named "stored" is used:

```
stored  VAR  BYTE[8]       'Create an array named "stored"
                           with 8 elements.
```

Note that the 8 elements in our array are numbered 0 through 7. There is no element-8. This is very important, because PBP doesn't place a limit at the end of the array. If you write "stored[8]", PBP won't generate an error and you will be accessing memory outside of the array. (The end of the array is stored[7] and there is no stored[8].) This could have disastrous results and be very difficult to debug.

The elements of "stored" can be written individually:

```
stored[0] = 12
stored[1] = 25
```

Or, you might want to write values with a loop. You could read PORTB once per second and save 8 readings:

```
FOR index = 0 TO 7          'Loop 8 times
    stored[index] = PORTB   'Save value of PORTB
    PAUSE 1000              'Wait a second
NEXT index                  'Loop again
```

PBP offers commands ARRAYREAD and ARRAYWRITE that make it very easy to store multiple values in a BYTE array:

```
ARRAYWRITE stored, [12,25,100,0,0,100,200,40]
```

A multi-dimensional table of values can be emulated using multiple index variables to construct an index expression. To treat our array as a table with 2 columns (x) and 4 rows (y), we can write: