

To add a new Breadboard:

1. Create an image of the desired breadboard.
 - a. Sometimes it will be easier to start with an existing PEBBLE breadboard image or draw a new image from scratch.

On other occasions an image may be available from a website though these often need some tidy-up to remove unwanted lines, etc., and even rotate the entire or parts of the image.
So be aware it is unlikely one will find an image that does not require some work.
 - b. Scale the image so that the holes are on a 28 x 28 pixel grid.
 - c. Save the image in .GIF format in the folder PEBBLE/Images with a name in the format:
BBrrpcv.GIF
where rr = number of vertical rows of holes
p = number of power bus/rails (typically 0,1 or 2)
c = power rail colours (typically 1= red top most, 2 = black topmost)
v = variant on a scheme – eg the 4th 30 row board type

At least keeping with the “BBrr” part of the file name will make it easier to see what an image represents.

2. In the directory c:/PEBBLE/ for both the files P_for_FF.html and P_for_IE.html insert an additional line of code in the breadboard pulldown menu structure.

This is under the section commencing:

```
<form name="menubuttonForm">  
<select id="breadboardselector"  
class="myselect"
```

The line will look like:

```
<option value="BB62">50R 0Pwr Datak 12-600B Altronics H0719
```

The “BB62” part is the identifier that the program uses elsewhere to know which is the current/active breadboard to display. Please note that the “62” is not the number of rows but the next sequential number in the PEBBLE register/database.

The part after the > character is the text that will appear in the breadboard selection dropdown in the PEBBLE floating menu.

To ascertain the next number to us as the breadboard identifier, look in the PEBBLE/Javascript folder in the file named **imagedata.js**

Breadboard and protoboards are entered at the top of the file and the last entry will be at the bottom of the BBxx section with a line entry such as:

```
var BB62 = new Image(); // Datak 12-600B/Altronics H0719 50x40 Protoboard
```

note that the text after the // is a remark/comment only for our information when returning to the program.

As the breadboard images are typically larger files and only 1 is required at a time, to save memory and time to open all of the images, initially only the first file (i.e. BB1) is registered within the PEBBLE program and loaded immediately from within the javascript file **imagedata.js** for display on the screen.

The other breadboards are opened as required in the javascript file **breadboards.js** as and when they are selected from the PEBBLE floating menu dropdown list.

3. In the PEBBLE/javascript folder, open the file **breadboards.js** and add new entries as follows:
 - a. In the function **setBreadboardType(val)**
Scroll to the bottom of the function case statements and add a new entry for the new breadboard. This will be in form:

```

case "BB62" :
    globalBreadBoardSize = "Datak12-600B";
    globalBBPower = "Datak12-600B";
    o.style.width = "1471px";
    o.style.height = "1167px";
    if(globalTopParkVisible == "false" || globalTopParkVisible == false || globalTopParkVisible == null)
    {
        q.style.top = "1237px";
    }
    else
    {
        q.style.top = "1532px";
    }
    break;

```

The key element here are

o.style.width = the width of the breadboard image in pixels

o.style.height = the height of the breadboard image in pixels

for q.style.top,

the first entry will be the image height + 70 pixels

the second entry will be the image height + 365 pixels

these two entries determine the top edge of the “Bottom parking area” for placing off-board components

The first two entries can be matched to existing size and power rail configuration if you identify an identical existing format otherwise provide a unique name.

b. In the function **generatePointArrays()**

(i) Under the section commencing with

```
dropPointsX = new Array();
```

At the bottom of the if(globalBreadBoardSize series of statements we need to add a new entry to define the board and image details. The entry will be in the format:

```

if(globalBreadBoardSize == "Datak12-600B") // Datak Prototyping board == Altronics H0719
{
    var bbrows = 50; // actually this board has 51 rows (not 50 per silk mask)
    var bbxr = 65;
    var lsmarg = 60;
}

```

Where: bbrows = the number of rows of holes – 1 (note that 0 is also a row number here)

bbxr = the right side distance in pixels from edge of image to centre of right holes

Lsmarg = the left side margin/distance in pixels from edge of image to centre of left holes

(ii) Under the section commencing with

```

// - - - - -
dropPointsY = new Array();

```

at the bottom of the if (globalBBPower == ... series of statements we need to add a new entry to define the board and image details. The entry will be in the format:

```

if (globalBBPower == "Datak12-600B")
{
    bbcols = 50 ; // Datak 12-600B / Altronics H0719 proto board
}

```

This represents the total number of rows of horizontal holes including 1 as a space to the off-board parking area and 9 for the offboard area. Thus our Datak 12-600B has 40 actual holes and row 0 is the first row so we have $(40 - 1) + 2$ (board margin) + 9 (in off-board parking area).

- c. Next an entry must be added to define the distance from the top of the breadboard to the top row of holes. This will be in the format:

```

if(globalBreadBoardSize == "Datak12-600B")
{
    var topadj = topadj + 29 // var topadj = 47; // note initial/default value is 18
    yquant = 1070 / 39;
}

```

The value yquant above is determined to scale the holes vertically. In the example there are 40 rows or 39 spaces and top to bottom is approx. 1170 pixels.

- d. Next it is necessary to consider any rows/columns of holes where there are absolutely no holes and thus are better excluded from the defined matrix of holes.
All of the 51 vertical lines are complete (except the 4 corner most holes) so we allow all 51 rows of holes to be used.

In our example protoboard, at the bottom of the board the margin equates to 2 rows without any holes. To exclude these two rows of holes before the “off board component parking area” were include a piece of test code at the bottom of the section commencing:

```

// - - - - -
for(var m = 0; m <= bbcols; ++ m)

```

The added code in our example is:

```

if(globalBBPower == "Datak12-600B")
    if(m == 40 || m == 41 )
    {
        continue;
    }

```

Remember that in terms of row and column numbering, that the first is row/column 0 thus the above lines exclude the 41st and the 42nd horizontal lines of holes from being used to anchor the top left of a component.

- e. The final addition to the code is required to tell the PEBBLE program the file name for the .gif image file to be used for the new breadboard. In the case of the Datak 12-600B use in the above

example, the lines of code which are to be placed in the function **loadreqdBBimage(val)** within the breadboard.js files are as follows:

```
case "BB62" :  
    BB62.src = "images/BB50004.gif";  
    break;
```

by placing each section of code for a new breadboard/protoboard at the bottom of the function it will help to keep track of the BBxx numbers allocated for each breadboard.