

Sensible Cartesian Co-ordinate graphics pack

The standard Cartesian co-ordinate system places 0,0 in the bottom left corner of the view (ignoring offsets) e.g. like a piece of graph paper.

Modern windowing and LCD panel co-ordinate systems place 0,0 in the top left position.

The following is a series of graphics primitive subroutines that make it easy to pass conventional x,y co-ordinates and plot them correctly using the native windowing co-ordinates, using old the school drawing commands PLOT, DRAW and MOVE. This makes it a cinch to produce nice looking graphs without tying your brain in a knot with windowing co-ordinates. It also introduces the concept of a graphics cursor which leaves the "pointer" at the last referenced point on the display.

I have more to add (including a scaling ORIGIN command to place the co-ordinate system anywhere you choose) but they need some honing. **Note:** Always uses the current graphics pen (colour)

Example

```
cls
For n =0 To 239 Step 8 '239 To 0 Step -4
    move 239-n,n
    draw n,239
Next
For x=0 To Gxm Step 10
    For y=0 To Gym Step 10
        Move x,y
        circle Gx,Gy,4
        DrawR rnd*20-10,rnd*20-10
    Next y,x
```

Preamble

```
'screen dimensions follow LCDPANEL settings
Const Gxm=MM.HRes-1,Gym=MM.VRes-1 ' 0 to zzz

'The global variables Gx and Gy always contain the current graphic cursor in
native co-ordinates
Dim Integer Gx, Gy ' current native x,y (i.e. not cartesian) coord
system
```

The Code

```
'plot a point
Sub Plot(x As Integer,y As Integer)
    Gx=x:Gy=Gym-y
    Pixel Gx,Gy
End Sub
```

```
'plot a point relative to the cursor e.g. PlotR -1,-1 plots a point one
pixel left and down from the current graphics cursor
  Sub PlotR(x As Integer,y As Integer)
    Gx=Gx+x:Gy=Gy-y
    Pixel Gx,Gy
  End Sub

'Draw a line from the current graphics cursor to the x,y given
  Sub Draw(x As Integer,y As Integer) 'draw a line from the current
graphics cursor to the specified point
    Line Gx,Gy,x,Gy-y
    Gx=x:Gy=Gy-y
  End Sub

'Draw a line from the current graphics cursor to the relative point x,y from
the cursor
  Sub DrawR(x As Integer,y As Integer)
    Line Gx,Gy,Gx+x,Gy-y
    Gx=Gx+x:Gy=Gy-y
  End Sub

'move the graphics cursor to the given x,y but doesn't draw anything
  Sub Move(x As Integer,y As Integer)
    Gx=x:Gy=Gy-y
  End Sub

'move the graphics cursor to the point x,y relative to the current cursor
but doesn't draw anything
  Sub MoveR(x As Integer,y As Integer)
    Gx=Gx+x:Gy=Gy-y
  End Sub

'sets the global variables Gx and Gy to native LCD panel co-ordinates from
Cartesian so
'other drawing commands e.g. Text, Circle, Box etc. can use the Cartesian
system. Same as Move but more intuitive
  Sub Native (x,y)
    Gx=x:Gy=Gy-y
  End Sub
```

From:
<http://fruitoftheshed.com/wiki/> - FotS

Permanent link:
http://fruitoftheshed.com/wiki/doku.php?id=mmbasic:sensible_cartesian_co_ordinate_graphics_pack

Last update: **2024/01/19 09:30**

