; Scrolling Text App -5x5 LED Dot Matrix-V4

`Written by Christian McMahon 14/5/2018

`Modified by Pat McMahon 30/11/2017, 150 to 300 Advance,

`added digits 0-9 etc.

; Constants

symbol displayWidth = **5**

symbol characterSpacing = **0**

symbol bptrStart = **28** ; Skip variables on m2 parts (w0, w1 etc)

symbol advanceRateMS = **1000** ; Advance every this many milliseconds

symbol advanceStep = displayWidth ; Advance by this many columns

symbol advanceRateCount = advanceRateMS / displayWidth

symbol tableStart = **200**

; Constants for scrolling text

;symbol characterSpacing = 1

;symbol advanceRateMS = 300

;symbol advanceStep = 1

; Constants for static text

;symbol characterSpacing = 0

;symbol advanceRateMS = 1000

;symbol advanceStep = displayWidth

; Variables

symbol letter = b0

symbol column = b1

symbol varB = b2

symbol varC = b3

symbol value = b4

symbol columnCount = w2

symbol currentColumn = w3

symbol counter = w4

symbol counterB = w5

; Convenience Macro

#MACRO LoadColumn(\_column)

@bptrinc = \_column

#ENDMACRO

; Convenience Macro

#MACRO LoadBlankColumns(\_blankColumnCount)

value = \_blankColumnCount

if value != **0** then

for counterB = **1** to value

@bptrinc = **0**

next counterB

endif

#ENDMACRO

; Convenience Macro

#MACRO AppendLetter(\_letter)

letter = \_letter

gosub appendLetterSub

#ENDMACRO

main:

; Initialise

let dirsB = **%11111111**

let dirsC = **%00010111**

let columnCount = **0**

let currentColumn = **0**

let bptr = bptrStart

setfreq m32

LoadBlankColumns(displayWidth)

TABLE tableStart,("PERSONAL LOCAL GLOBAL ") ; save text in table

counter = tableStart;

letterLoop:

readtable counter, value ; read character from table

if value != **0** then ; stop loading characters at the end of the text

AppendLetter(value)

LoadBlankColumns(characterSpacing)

inc counter

goto letterLoop

endif

columnCount = bptr - bptrStart

drawDisplay:

; Draw columns for a set period of time before advancing

for counter = **1** to advanceRateCount

; Draw all columns on display

bptr = bptrStart + currentColumn

for column = **1** to displayWidth

; portC

select case column

case **5**

varC = **%00010111**

case **4**

varC = **%00010110**

case **3**

varC = **%00010101**

case **2**

varC = **%00010011**

case **1**

varC = **%00000111**

endselect

; portB

if column != **5** then

varB = @bptr | **%00100000**

else

varB = @bptr

endif

; Clear all, setting pinsB first to ensure clean transition

pinsB = **%00100000**

pinsC = **%00010111**

pinsC = varC

pinsB = varB

bptr = bptr + **1**

next column

next counter

; Advance

currentColumn = currentColumn + advanceStep

counter = currentColumn + displayWidth

if counter > columnCount then

currentColumn = **0**

endif

goto drawDisplay

appendLetterSub:

select case letter

case "A"

LoadColumn(**%11110**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

LoadColumn(**%11110**)

case "B"

LoadColumn(**%11111**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%01010**)

case "C"

LoadColumn(**%01110**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

case "D"

LoadColumn(**%11111**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%01110**)

case "E"

LoadColumn(**%11111**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

case "F"

LoadColumn(**%11111**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

case "G"

LoadColumn(**%01110**)

LoadColumn(**%10001**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%01101**)

case "H"

LoadColumn(**%11111**)

LoadColumn(**%00100**)

LoadColumn(**%00100**)

LoadColumn(**%00100**)

LoadColumn(**%11111**)

case "I"

LoadColumn(**%00000**)

LoadColumn(**%00000**)

LoadColumn(**%11111**)

LoadColumn(**%00000**)

LoadColumn(**%00000**)

case "J"

LoadColumn(**%00000**)

LoadColumn(**%01000**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%01111**)

case "K"

LoadColumn(**%11111**)

LoadColumn(**%00100**)

LoadColumn(**%01010**)

LoadColumn(**%10001**)

LoadColumn(**%00000**)

case "L"

LoadColumn(**%11111**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

case "M"

LoadColumn(**%11111**)

LoadColumn(**%00010**)

LoadColumn(**%00100**)

LoadColumn(**%00010**)

LoadColumn(**%11111**)

case "N"

LoadColumn(**%11111**)

LoadColumn(**%00010**)

LoadColumn(**%00100**)

LoadColumn(**%01000**)

LoadColumn(**%11111**)

case "O"

LoadColumn(**%01110**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%01110**)

case "P"

LoadColumn(**%11111**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

LoadColumn(**%00010**)

case "Q"

LoadColumn(**%00110**)

LoadColumn(**%01001**)

LoadColumn(**%01001**)

LoadColumn(**%01001**)

LoadColumn(**%10110**)

case "R"

LoadColumn(**%11111**)

LoadColumn(**%00101**)

LoadColumn(**%00101**)

LoadColumn(**%01101**)

LoadColumn(**%10010**)

case "S"

LoadColumn(**%10010**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%01001**)

case "T"

LoadColumn(**%00001**)

LoadColumn(**%00001**)

LoadColumn(**%11111**)

LoadColumn(**%00001**)

LoadColumn(**%00001**)

case "U"

LoadColumn(**%01111**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%01111**)

case "V"

LoadColumn(**%00111**)

LoadColumn(**%01000**)

LoadColumn(**%10000**)

LoadColumn(**%01000**)

LoadColumn(**%00111**)

case "W"

LoadColumn(**%11111**)

LoadColumn(**%01000**)

LoadColumn(**%00110**)

LoadColumn(**%01000**)

LoadColumn(**%11111**)

case "X"

LoadColumn(**%10001**)

LoadColumn(**%01010**)

LoadColumn(**%00100**)

LoadColumn(**%01010**)

LoadColumn(**%10001**)

case "Y"

LoadColumn(**%00001**)

LoadColumn(**%00010**)

LoadColumn(**%11100**)

LoadColumn(**%00010**)

LoadColumn(**%00001**)

case "Z"

LoadColumn(**%10001**)

LoadColumn(**%11001**)

LoadColumn(**%10101**)

LoadColumn(**%10011**)

LoadColumn(**%10001**)

case "1"

LoadColumn(**%00000**)

LoadColumn(**%00000**)

LoadColumn(**%11111**)

LoadColumn(**%00000**)

LoadColumn(**%00000**)

case "2"

LoadColumn(**%11101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10111**)

case "3"

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%11111**)

case "4"

LoadColumn(**%00111**)

LoadColumn(**%00100**)

LoadColumn(**%00100**)

LoadColumn(**%11111**)

LoadColumn(**%00100**)

case "5"

LoadColumn(**%10111**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%11101**)

case "6"

LoadColumn(**%11111**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%11101**)

case "7"

LoadColumn(**%10001**)

LoadColumn(**%01001**)

LoadColumn(**%00101**)

LoadColumn(**%00011**)

LoadColumn(**%00001**)

case "8"

LoadColumn(**%11111**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%11111**)

case "9"

LoadColumn(**%10111**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%10101**)

LoadColumn(**%11111**)

case "0"

LoadColumn(**%11111**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%10001**)

LoadColumn(**%11111**)

case " "

LoadBlankColumns(displayWidth);

else

; draw underscore

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

LoadColumn(**%10000**)

endselect

return