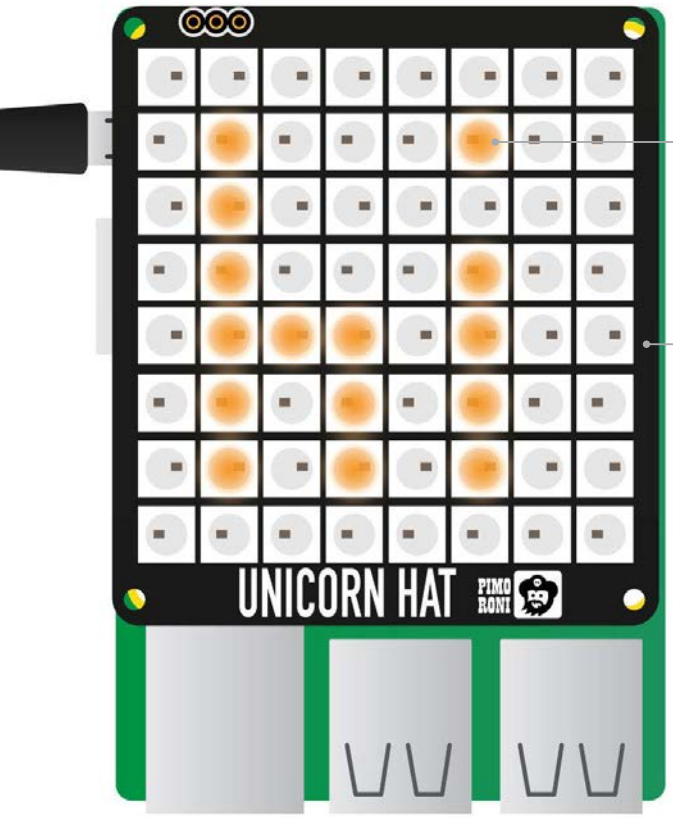
SEAN MCMANUS

Sean McManus is the author of *Raspberry Pi For Dummies* (with Mike Cook), and *Scratch Programming in Easy Steps*.
sean.co.uk
twitter.com/musicandwords



The code displays multicoloured red, purple and blue text

A diffuser layer makes it easier to read, and protects your eyes from the bright LEDs!

MAKE TEXT SCROLL ON THE UNICORN HAT

You'll Need

- Unicorn Hat
- Frosted or Ninja diffuser
- Pibow case (optional)
- Internet access

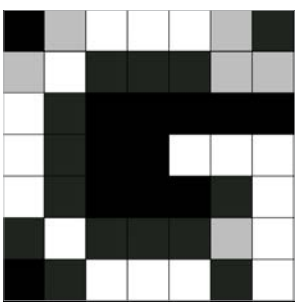
The Unicorn HAT provides a compact and colourful way to display scrolling messages on your Raspberry Pi. Here's how it's done...

It's easy to get colourful special effects from the Unicorn HAT, an 8×8 matrix of RGB LEDs. But did you know you can also use it to output text? This project shows you how to scroll messages across it, and could form the basis of any project that needs to display information, such as a robot or Twitter display. Nobody wants to hand-design (or even hand-code) a whole new font for this, so this project uses Pygame to scan a font on screen instead, capturing that font data in a Python dictionary.

>STEP-01

Create your font

The key to this project is to use one of the fonts already on your Pi, **FreeSans.ttf**, and convert it to a useful form for the Unicorn HAT scroller program. Open Python 2 and enter **Listing 1**. It first displays each character in turn and scans it to check the brightness of each pixel. Next, it builds a list of values for each pixel in the letter, and compiles a dictionary of those list values. Finally, it trims the character data back to the minimum width required.



Right The font creator program uses three different colour shades. On the Unicorn HAT, these shades are multiplied by the colour numbers to create the letters

>STEP-02

Manually modify selected characters

This step is optional, depending on your application. Some of the punctuation symbols (including @) don't render clearly at the size we need to display them for scanning. If you want, you can define them manually. Each character has a list that contains another list of data for each row. Use 1 to plot a point and 0 for an empty point, as shown for @ in Listing 1. Here's a shortcut to save hand-designing the characters first: the Amstrad CPC6128 manual (available at **bit.ly/1AgpY7J**) shows the design for that classic computer's font, which also uses an 8×8 grid (see Chapter 7).

>STEP-03

Enter the scroller code

Listing 2 is your scroller code. Enter it into a new window in Python 2 and save it as `scroller.py`. The code assumes you have your Raspberry Pi with the USB ports on the left, so you can stand it up on the side that has no cables going into it. Change the orientation at the start if your Pi is a different way around, to 0, 90 or 270.

>STEP-04

Paste in your font dictionary

Run the font creator program (`python fontmaker.py`), highlight the font dictionary when it's shown

Language
>PYTHON

fontmaker.py (Listing 1)

```
# -*- coding: utf-8 -*-
import pygame
pygame.init()
canvas=pygame.display.set_mode((100,100))
pygame.mouse.set_visible(0)
char_set = "QWERTYUIOP ASDFGHJKL ZXCVBNM \
1234567890-= !$%^&*()_+"
char_set += "[]{} ;'#:@~ ,./<>?\"\\"
font_dictionary = dict()

for letter in char_set: #main dictionary creation loop
    canvas.fill((0,0,0)) #clear the canvas
    fontObj = pygame.font.Font(\
'/usr/share/fonts/truetype/freefont/FreeSans.ttf',9)
    textSurface = fontObj.render(
    letter,True,(255,255,255),(0,0,0))
    textRectObj = textSurface.get_rect()
    canvas.blit(textSurface, textRectObj)
    pygame.display.update() #display the letter

    letter_data = []
    for y in range(8): #check each row of
    # the letter on canvas
        letter_row=[]
        for x in range(8):
            #each x position of letter on canvas
            colour = canvas.get_at((x,y+2))
            if colour[0]>200:
                letter_row.append(1)
            elif colour[0]>100:
                letter_row.append(0.75)
            elif colour[0]:
                letter_row.append(0.15)
            else:
                letter_row.append(0)
        letter_data.append(letter_row)

    for x in range(7,-1,-1): # Trim excess space on right of letter
        column=[letter_data[y][x] for y in range(8)]
        if max(column)==0:
            for i in range(8):
                del letter_data[i][x]

    font_dictionary[letter]=letter_data
font_dictionary[' ']=[[0]*4]*8 #space gets trimmed to empty otherwise
font_dictionary['@']=[[0,1,1,1,1,1,0],[1,1,0,0,0,1,1],[1,1,0,1,1,1,1],\
[1,1,0,1,0,0,1],[1,1,0,1,1,1,1],[1,1,0,0,0,0,0],[0,1,1,1,1,1,0],[0]*7]

pygame.quit()
print font_dictionary
```

in the shell, and use SHIFT+**CTRL+C** to copy it all. It starts and ends with a curly bracket and might span more than one screen. Paste it in place of the curly brackets, where **font_dictionary** is defined near the top of Listing 2. Now your scroller program has the font data it needs. Save your program.

>STEP-05

Restart and install software

There appears to be a conflict between Pygame and the Unicorn HAT, so you can't use them both in the same session. If you see random flashing on the Unicorn HAT when you run the scroller program, this is probably the reason why. Restart your Raspberry Pi now. If you haven't already installed the Unicorn HAT drivers, go into the command line and issue the command:

\curl -sS get.pimoroni.com/unicornhat | bash

>STEP-06

Run your scroller

You should view your Unicorn HAT through a diffuser layer. You can buy one from Pimoroni, designed for use as a lid on the full-size Pibow case. To run your scroller, open the command-line and go to the folder containing your code in. Enter **sudo python scroller.py**. The program will ask you for text to scroll and then scroll it across the display. When you build this code into other applications, put the message you want to display into the variable **string_to_show**.

scroller.py (Listing 2)

```
import unicornhat as unicorn
import time
unicorn.rotation(180) #adjust for your Pi's orientation
unicorn.brightness(0.4)
#warning: Altering this value can make LED VERY bright!
font_dictionary={} # paste in your font dictionary here
string_to_show=raw_input("Enter the text to scroll: ")
scroll_rows=[[0]*8]*8 #blank space at start of message

for character in string_to_show:
   if character.upper() in font_dictionary:
       character_rows = font_dictionary[character.upper()]
   else:
       character_rows = font_dictionary['-']
   for i in range(8):
       scroll_rows[i] = scroll_rows[i]+character_rows[i]
       scroll_rows[i] += [0] #gap between letters

for i in range(8):
   scroll_rows[i]+=[0]*8 #blank space at end of message

for scroll_position in range(len(scroll_rows[0])-8):
   for y in range(8):
      thisrow = scroll_rows[y]
      for x in range(8):
          pixel_shade=thisrow[x+scroll_position]
          unicorn.set_pixel(x,y,int((95+x*20)*pixel_shade),\
              int(100*pixel_shade),int((95+y*20)*pixel_shade))
   unicorn.show()
   time.sleep(0.04)
```