Battery Powered Raspberry Pi Displays w/ RaspiRobot Shield

Created by Simon Monk

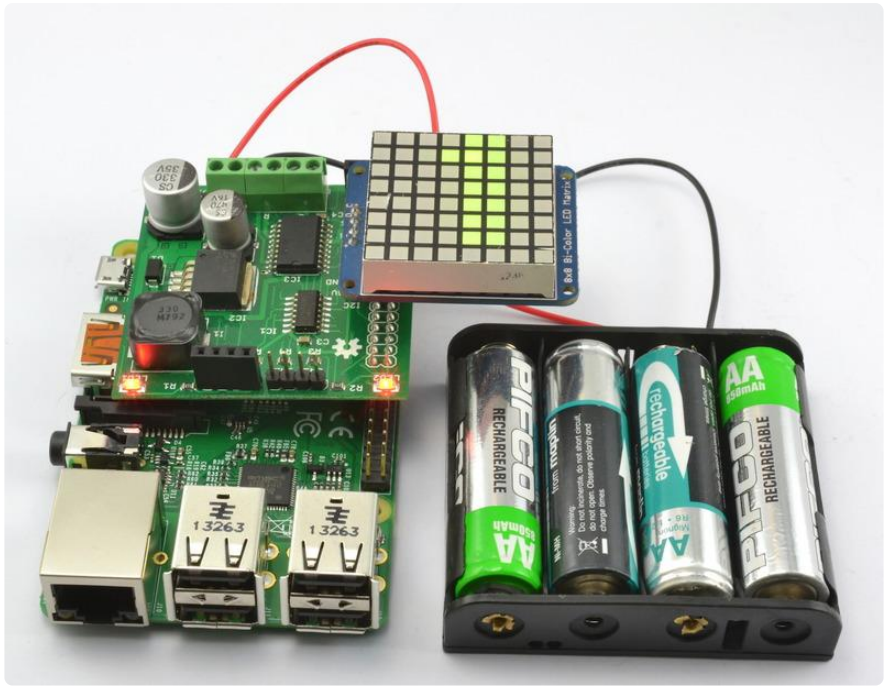
https://learn.adafruit.com/raspirobot-battery-powered-raspberry-pi-displays

Last updated on 2023-08-29 02:35:25 PM EDT

Table of Contents

Overview

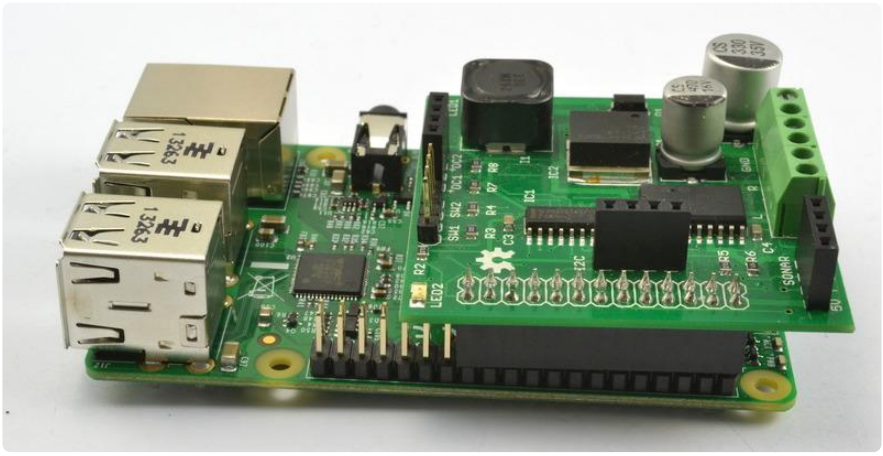
Although the RasPiRobot Board is primarily intended as a motor controller, it also includes a high current switch mode voltage regulator that allows you to power a Raspberry Pi from a wide range of battery types.



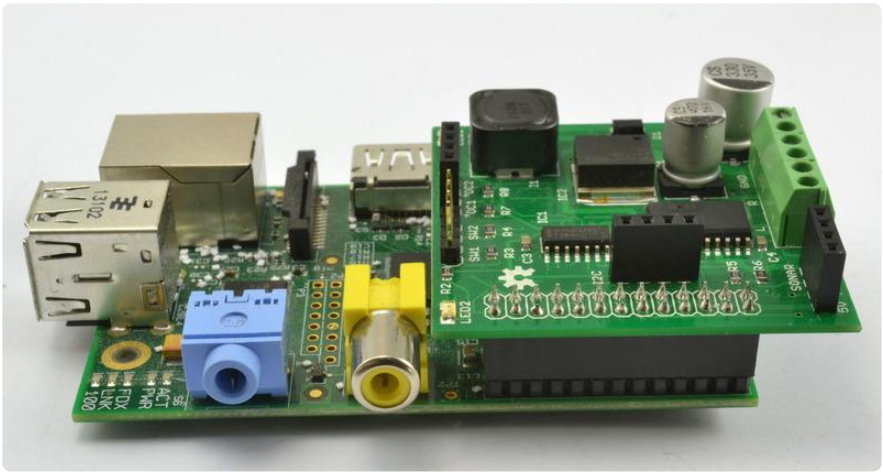
In this tutorial you will learn how to use the RasPiRobot Shield and an AA battery pack to make a portable Raspberry Pi powered display.
The display will show the current time as scrolling text, but the project can easily be modified to display other scrolling text messages.
The parts used in this project are listed in the Featured Products bar on the right of your screen.

Wiring

Wiring is really too strong a word for it. Just plug the RasPiRobot Board V2 onto the Raspberry Pi as shown below. Note that if you are using a Raspberry Pi B+ with extra pins, then make sure that the RasPiRobot Board V2 fits over the GPIO pins at the right hand side of the Raspberry Pi.



If you are using a Raspberry Pi model B, then the RasPiRobot Board V2 will fit over all the GPIO pins, as shown below.

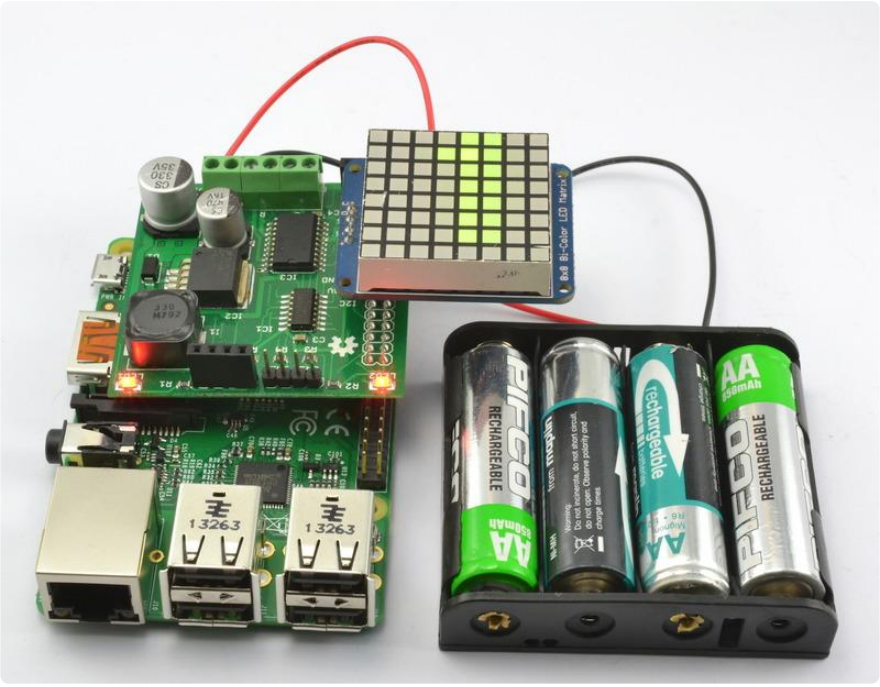


You will need to solder together the LED Matrix display as it comes in kit form. You can find full instructions for this on the display's product page (http://adafru.it/902).

Then fit the LED Matrix display into the I2C socket o the RasPiRobot Board V2. Make sure that you get it the right way around, and if you are using the latest version of the RasPiRobot Board V2 with extended headers make sure that the header pins cannot reach up as far as the bare connections on the underside of the LED matrix module. If

you think they might, then just fold a bit of electrical tape or even Scotch tape over the extended GPIO pins. Attach the flying leads of the battery box to the screw terminals maked Vin and GND on the RasPiRobot Board V2. Fit some AA batteries (rechargeable or single-use). Turn on the switch to the battery box and you will see the power light on the Raspberry Pi light up, as well as the two LEDs on the RasPiRobot Board V2.

For the battery pack - the red wire goes to VIN and the black wire goes to Ground



Software

To setup your Raspberry Pi to use I2C (needed by the display module) follow the instructions here ().
Please follow the steps here () to install the Adafruit I2C library and PIL (Python Imaging Library).
You may also wish to try out some of the examples in the latter guide to check that I2C and the LED Matrix display is working correctly.
The following example takes the display examples a bit further, allowing you to display scrolling text on them.
Open an editor window using:

```
nano scrolling_clock.py
```

Then paste in the following text and save the file using CTRL-X and then Y and then ENTER.

```
import time
from datetime import datetime
from PIL import ImageFont
from PIL import Image
from PIL import ImageDraw
from Adafruit_LED_Backpack import Matrix8x8

display = Matrix8x8.Matrix8x8()
display.begin()

font = ImageFont.truetype("/usr/share/fonts/truetype/freefont/FreeSansBold.ttf", 9)

im = Image.new("1", (8, 8), "black")
draw = ImageDraw.Draw(im)
width, ignore = font.getsize("88 : 88 : 88")

def format_time():
    d = datetime.now()
    return "{:%H : %M : %S}".format(d)

message = format_time()
x = 8
while True:
    x = x - 1
    if x &lt; -(width + 20):
        x = 8
        message = format_time()
    draw.rectangle((0, 0, 7, 7), outline=0, fill=0)
    draw.text((x, -1), message, 1, font=font)
    display.set_image(im)
    display.write_display()
    time.sleep(0.1)
```

You can now run the program using the command:

```
sudo python scrolling_clock.py
```

The time should now slowly scroll across the display.

The code uses a number of libraries to do its job. The Adafruit_LED_Backpack library handles the low level interface to the matrix display. To be able to write text onto the display, we need to use the Python Imaging Library (PIL) and write the text to an image that can then be written to the display.
After initializing the display, the Truetype font FreeSansBold size 9 point is loaded. This seemed to work the best from the pre-installed fonts. This can be changed to one of the other fonts in the directory /usr/share/fonts/truetype/freefont/ if you want to experiment. The fonts are really not intended to work in 8x8 pixels, so the results are legible but not great.
An 8x8 image is created with a bit depth of 1. We are just going to write text in green for now. The following command works out how many pixels wide the text for the time is going to be:

```
width, ignore = font.getsize("88 : 88 : 88")
```

The formatTime function just returns the current time formatted for the display.
The main loop of the program produces the scrolling text effect. It does this by writing the text of the time onto the display with a decreasing x offset, so that the message appears to move. Most of the time, the value of x will be negative (off the left of the display), but the draw.text function automatically crops to the visible part of the display.

Next Steps

This arrangement can also be used to drive other Adafruit Backpack displays, such as the 4 digit 7 segment display (http://adafru.it/878)and other matrix backpack displays ().