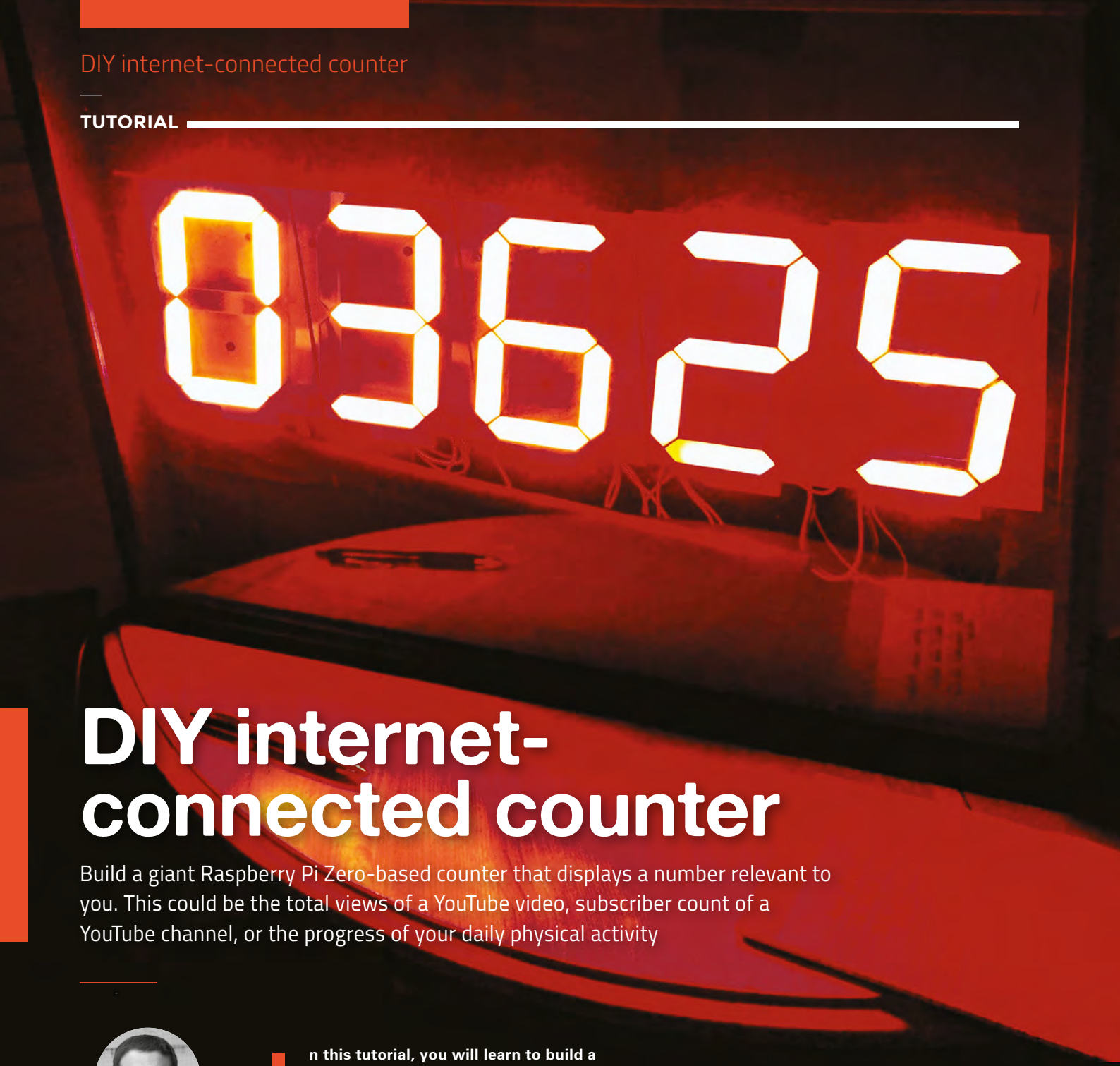
TUTORIAL



DIY internet-connected counter

Build a giant Raspberry Pi Zero-based counter that displays a number relevant to you. This could be the total views of a YouTube video, subscriber count of a YouTube channel, or the progress of your daily physical activity

Sai Yamanoor

yamanoor.com

Sai Yamanoor is a Mechatronics Engineer and a DIY enthusiast. In his free time he likes to build things using the Raspberry Pi and the Arduino. You can check out Sai's projects at **saiyamanoor.com**

In this tutorial, you will learn to build a gigantic counter that could be used as a visual aid to motivate yourself. This visual aid could be a countdown towards your charity fundraiser goal or your daily step goal on your fitness tracker. At the end of this tutorial, we have included some ideas on how to use the counter to motivate yourself.

We got the idea of building a gigantic counter from the book *Deep Work* by Cal Newport. The author states that in order to overcome procrastination, you need to make a big gesture for yourself. Hence, we built this gigantic counter to motivate ourselves to stay physically active. In this tutorial, we would like to share how to build one for your own goals.

LET'S GET STARTED!

Before you start building your own internet-connected counter, you need to determine its feasibility; i.e. you need to find out if it is indeed possible to fetch the data in question in a known format (such as JSON).

GUTS OF THE SYSTEM

The display consists of five seven-segment displays that are interfaced to an Arduino board. The Arduino drives the seven-segment displays. The Arduino is interfaced to the Raspberry Pi Zero via USB. The Raspberry Pi Zero connects to the internet and fetches the requested information.

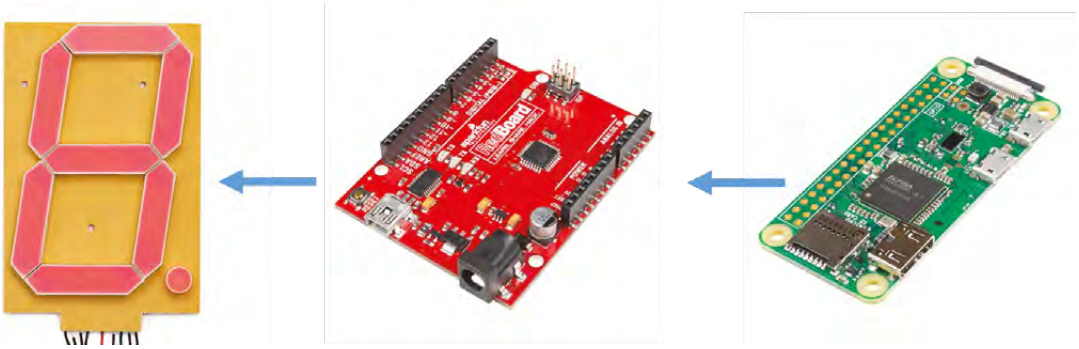


Figure 1
Guts of the display

credit: SparkFun
CC BY-SA 4.0

SETTING UP SEVEN-SEGMENT DISPLAYS

We chose to use five seven-segment displays in our project. This enabled us to display any number up to 99 999.

The first step is to solder the display drivers on to the seven-segment display (as shown in **Figure 1**). Ensure that the ten castellated pins are soldered facing the bottom side of the seven-segment display. The ten pins of the driver need to be aligned with the traces of the display (as shown in **Figure 2**).

The seven-segment display driver comes with two sets of six header pins named IN and OUT. Connect the five seven-segment displays in series, such that the OUT pins of one display are connected to the IN pins of the next display (as shown in **Figure 3**).

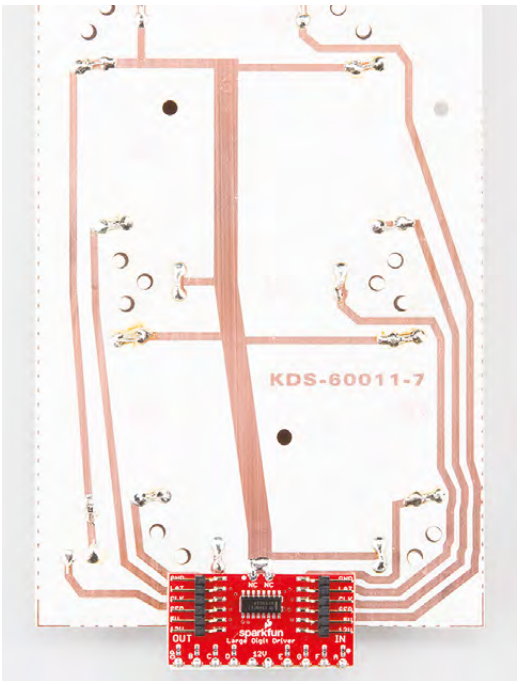


Figure 2
Solder the drivers onto the seven-segment display

Source: sparkfun.com

Once this is done, connect the IN pins of the first display to the Arduino (**Figure 4**) in the following order:

> GND to Arduino GND
> LAT to Arduino 5
> CLK to Arduino 6
> SER to Arduino 7
> 5v to Arduino 5v
>12v to Arduino Vin

The seven-segment display requires a 12 V power supply, and the power supply pin of the isplay driver is connected to the 'VIN' pin of the Arduino. Therefore, connect a 12 V, 1 A power supply to the Arduino. The seven-segment display and the Arduino interface needs to be tested. The test code sample (**seven_segment_interface_test.ino**) is available from **hsmag.cc/ZYrmkT**. Compile and flash the Arduino with the code sample. The seven-segment display should start displaying numbers. It should incrementally display numbers from '00000' to '99 999'. If the numbers aren't displayed, check the connections between the Arduino and the Raspberry Pi Zero. If the segments of the display don't light up properly, check the solder connections for lack of continuity or whether the displays are heating up due to short circuit etc.

After verification of the seven-segment display setup, it is time to flash the device firmware that drives the display. The firmware file (**sketch_sep01a.ino**) is available in the repository provided earlier in this article.

The Arduino listens for any incoming messages from the Raspberry Pi via the serial port. When the display needs to be updated, the Raspberry Pi transmits a number (between 0 and 99 999) as a string with a preceding 'S'. →

YOU'LL NEED

- **1 × Raspberry Pi Zero W**
- **5 × 6.5" Seven-segment display**
  sparkfun.com/products/8530
- **5 × Sparkfun large digit driver**
  sparkfun.com/products/13279
- **1 × Sparkfun redboard/ Arduino Uno**
- **1 × 12 V, 1 A power supply**
- **Set of jumper cables**
- **1 × 18" by 24" shadow box**
  (available at arts & crafts stores)
- **2 × micro USB cables**
- **1 × USB A to C adapter**
- **Set of M3 screws and nuts**

FIXING FLICKERS

If you notice the digits of your display flickering or the brightness of the display is not the same across all digits, check your connections and ensure that you are using a power supply of appropriate current rating. Connect the 12 V power supply to both the ends of the seven-segment display. This would ensure that the brightness is even across all the digits of the display.

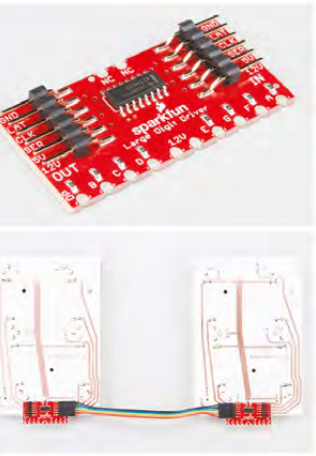


Figure 3
Connect the seven-segment displays in series

Source: sparkfun.com

```
        if(c == 'S'){
              while(Serial.available()){
                c = Serial.read();
                steps = (steps * 10) + (c -
'0');
              }
              Serial.println(steps);
              showNumber(steps);
            }
```

The Arduino converts the string into a number and updates the display.

```
        void showNumber(long value)
        {
          long number = abs(value); //Remove
negative signs and any decimals
          for (byte x = 0 ; x < 5 ; x++)
          {
            int remainder = number % 10;
           postNumber(remainder, false);
           number /= 10;
          }
         //Latch the current segment data
          digitalWrite(segmentLatch, LOW);
          digitalWrite(segmentLatch, HIGH);
        }
```

ASSEMBLY OF THE DISPLAY

Now that the display and the Arduino are operational, it is time to assemble the seven-segment display and the Arduino into a shadow box. Shadow boxes are a type of a picture frame available from craft stores.

Line up the seven-segment displays in a straight line on the shadow box frame. Then, mark the drill points to mount the display. Before mounting the display, wire them up (as shown earlier) and cover the back frame using a liner material (preferably black). Assemble the back frame into the shadow box and verify that the displays are in a straight line and also verify the overall operation of the display. When the Arduino turns on, it displays zero by default.

LET'S FETCH SOME NUMBERS!

We are assuming that you are familiar with the Raspberry Pi Zero and that you have already completed setup of your Raspberry Pi Zero (including network setup etc.). We are going to discuss some potential uses for this giant counter. For example: let's track the total number of subscribers on the Raspberry Pi Foundation's YouTube channel. We are going to write some Python code to track the subscriber count. When we started writing this article, the total subscriber count was at 25 970 and it ticked up by two in 60 minutes.

The first step is to install some requisite Python libraries:

```
sudo pip install --upgrade configparser
request google-api-python-client
```

In order to get started, you need a Google developer account. The instructions to sign up are available from here: **hsmag.cc/UDbAZZ**. Sign up for a developer account and save the API key to a file called **youtube_config.ini** in the following format:

```
[CREDENTIALS]
API_KEY = key
```

The Python code sample for this example is available along with this project's repository. By making use of the YouTube API, we can obtain the subscriber count as follows:

```
URL=(
"https://www.googleapis.com/youtube/v3/
channels?part=statistics&id=" +
 CHANNEL +
 "&key=" +
 API_KEY
)

def get_subscriber_count():
    data = None
    try:
        response = requests.get(URL)
    except:
        logging.info("Requests error")
    data = response.json()
    return data["items"][0]["statistics"]
["subscriberCount"]
```

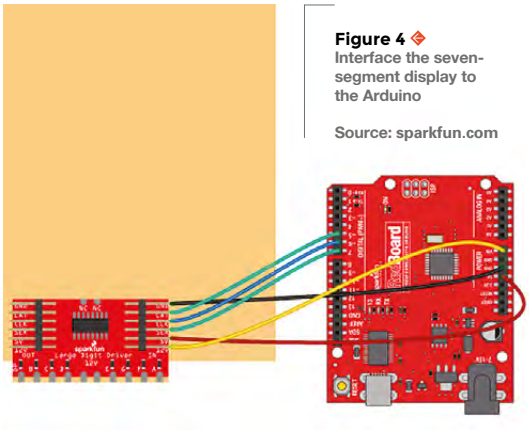


Figure 4
Interface the seven-segment display to the Arduino

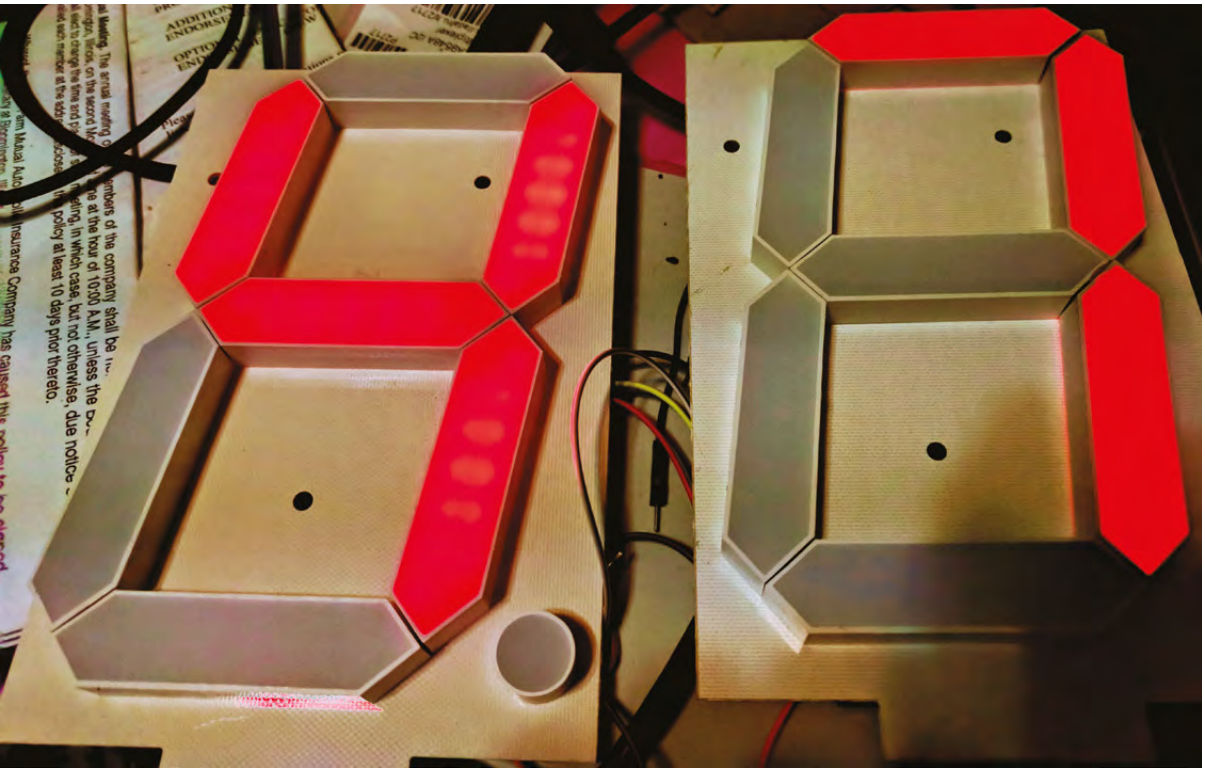
Source: sparkfun.com

QUICK TIP

It is possible to drive the seven-segment display directly using the Raspberry Pi Zero. You need to write some drivers (clocking, latching etc.) for the serial bus of the seven-segment driver.

STAY FIT

We originally built this dashboard to keep track of our physical activity. We made use of the Fitbit API to count down from our daily step goal. Check out the Python client for the Fitbit API at: **hsmag.cc/FvrDcZ**



Left
Interface the seven-segment display to the Arduino

BRIGHT LIGHTS

If you install this gigantic display in a darker area of your home, it would light up the room and act as a night light.

The subscriber count obtained using the YouTube API is written to the serial port as follows:

```
def serial_write(serial_client,count):
    message = 'S' + count
    serial_client.write(bytes(message,
encoding='utf-8'))
```

The Arduino will parse the string and update the display.

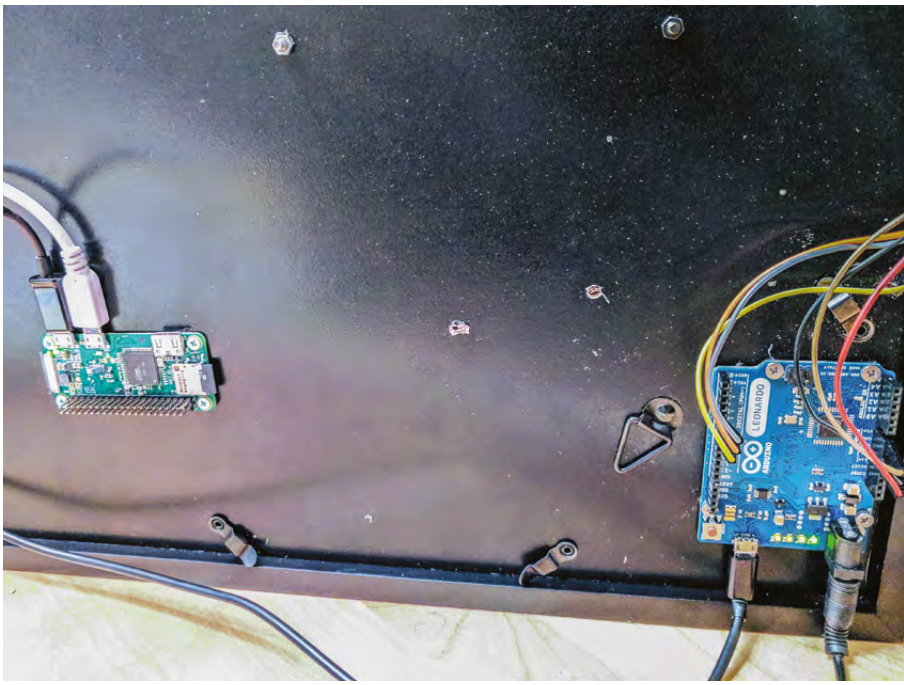
Once the Raspberry Pi's function is tested, you can mount the Arduino and the Raspberry Pi Zero on the backside of the panel.

Install this display somewhere and watch it perform a live update of the total subscriber count. We use this giant counter to keep track of our daily physical activity goals. It counts down from our daily step goal. The code sample for the fitness goal tracker is also available in the repository.

Here are some recommendations for potential uses of this giant display:

- **Tracking your savings (check out ifttt.com)**
- **Tracking your social media statistics (YouTube views, Twitter followers)**
- **Displaying the total number of people in a given space**
- **Tracking your sales/marketing target etc.**

Below
Raspberry Pi Zero and the Arduino mounted on the back panel



MINDFUL MOUNTING

In this tutorial, we mounted the Raspberry Pi Zero and the Arduino on the back of the shadow box for easy trouble-shooting. This made things difficult when we tried to mount the shadow box on a wall. Consider mounting the electronics on the front side of the frame and concealing it using a cloth matching the same colour as the background of the frame.