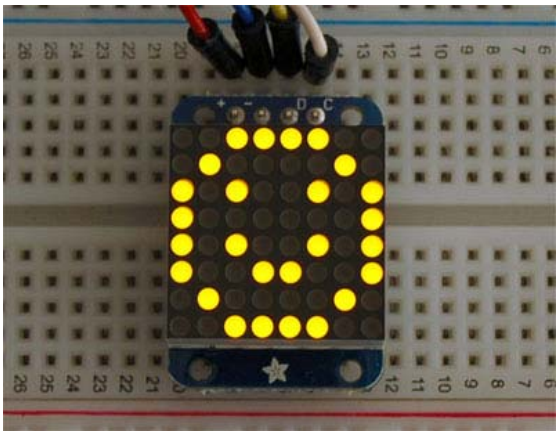
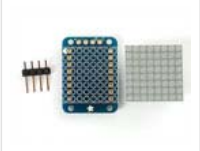
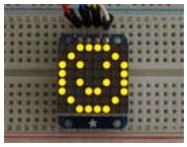
Adafruit Mini 8x8 LED Matrix w/I2C Backpack - Yellow -

ID: 871







Description

What's better than a single LED? Lots of LEDs! A fun way to make a small display is to use an **8x8 matrix** or a **4-digit 7-segment display**. Matrices like these are 'multiplexed' - so to control 64 LEDs you need 16 pins. That's a lot of pins, and there are **driver chips like the MAX7219** that can control a matrix for you but there's a lot of wiring to set up and they take up a ton of space. Here at Adafruit we feel your pain! After all, wouldn't it be awesome if you could control a matrix without tons of wiring? That's where these adorable LED matrix backpacks come in. We have them in two flavors - **a mini 8x8** and a **4-digit 0.56" 7-segment**. They work perfectly with the matrices we stock in the Adafruit shop and make adding a bright little display trivial.

The matrices use a driver chip that does all the heavy lifting for you: They have a built in clock so they multiplex the display. They use constant-current drivers for ultra-bright, consistent color (**the images above are photographed at the dimmest setting to avoid overloading our camera!**), 1/16 step display dimming, all via a simple I2C interface. The backpacks come with address-selection jumpers so you can connect up to four mini 8x8's or eight 7-segments (or a combination, such as four mini 8x8's and four 7 -segments, etc) on a single I2C bus.

The product kit comes with:

- A fully tested and assembled LED backpack
- **Ultra-bright 8x8 yellow matrix**
- 4-pin header

A bit of soldering is required to attach the matrix onto the backpack but its very easy to do and only takes about 5 minutes.

Of course, in classic Adafruit fashion, **we also have a detailed tutorial showing you how to solder, wire and control the display**. We even wrote **a very nice library for the backpacks so you can get running in under half an hour, displaying images on the matrix or numbers on the 7-segment**. If you've been eyeing matrix displays but hesitated because of the complexity, his is the solution you've been looking for!