# thingSoC

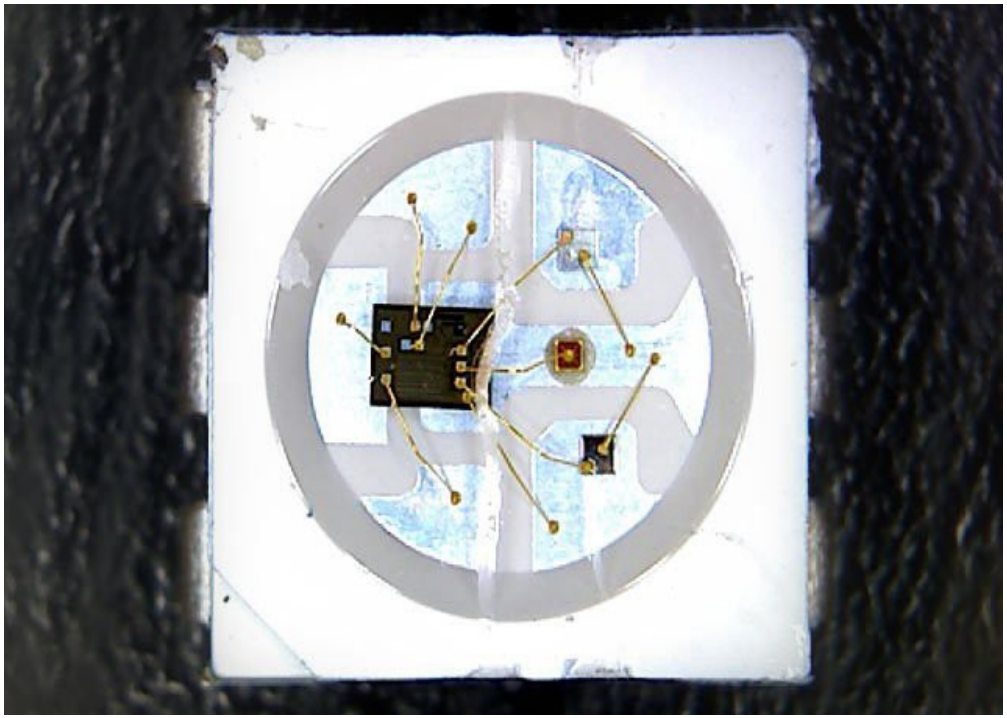
## NEOLED Overview:

## An Introduction to WS2812 LEDs


By
@PatternAgents

# Addressable LEDs

* WS2810/WS2812 (a.k.a. NEOPixels by Adafruit)
  Integrates an LED driver chip with Red, Green, Blue LED's

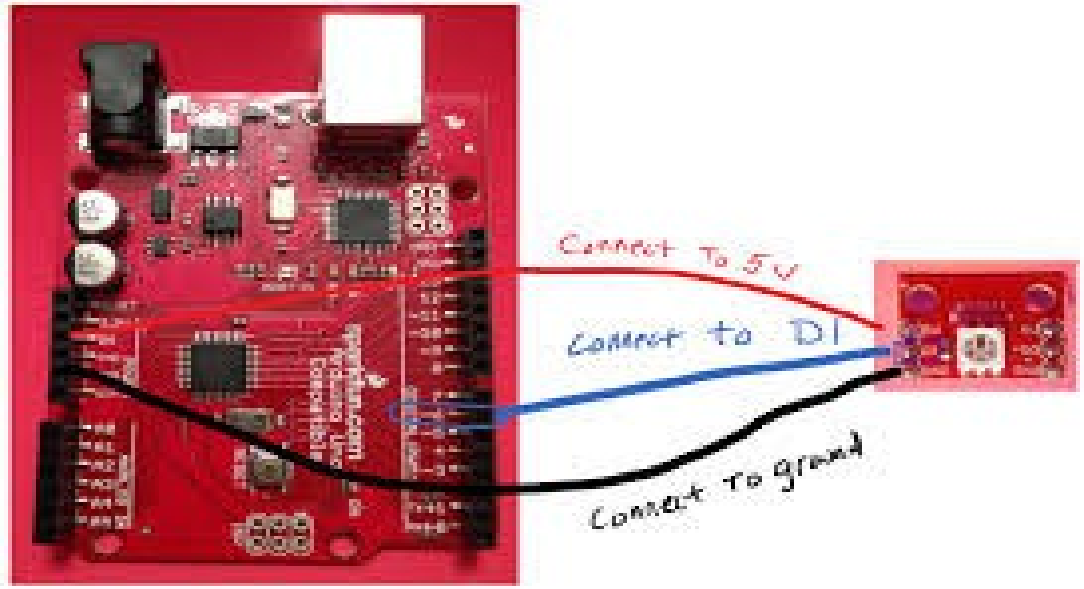


# WS2812 (SK6812) Color LEDs

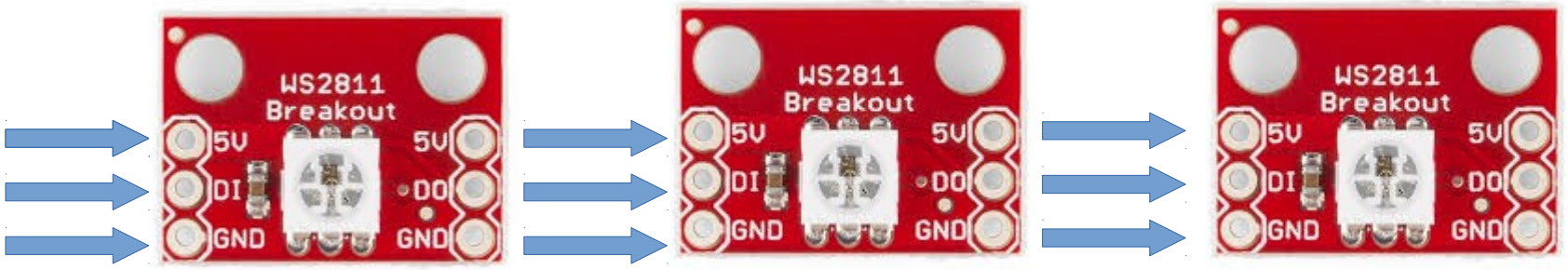
* Likely the most common addressable LED you'll find (called NEOPixels when you buy them at Adafruit!)
* They use a serial bit stream (UART like) protocol that can be difficult for some microcontrollers or Linux single board computers (SBC) to drive directly (tight timing...)
* It can also be difficult to drive and synchronize hundreds of WS2812 LEDs in multiple strings without hardware acceleration of some kind (special circuit, DMA, etc.)

Source : @PatternAgents

# WS2812 Wiring



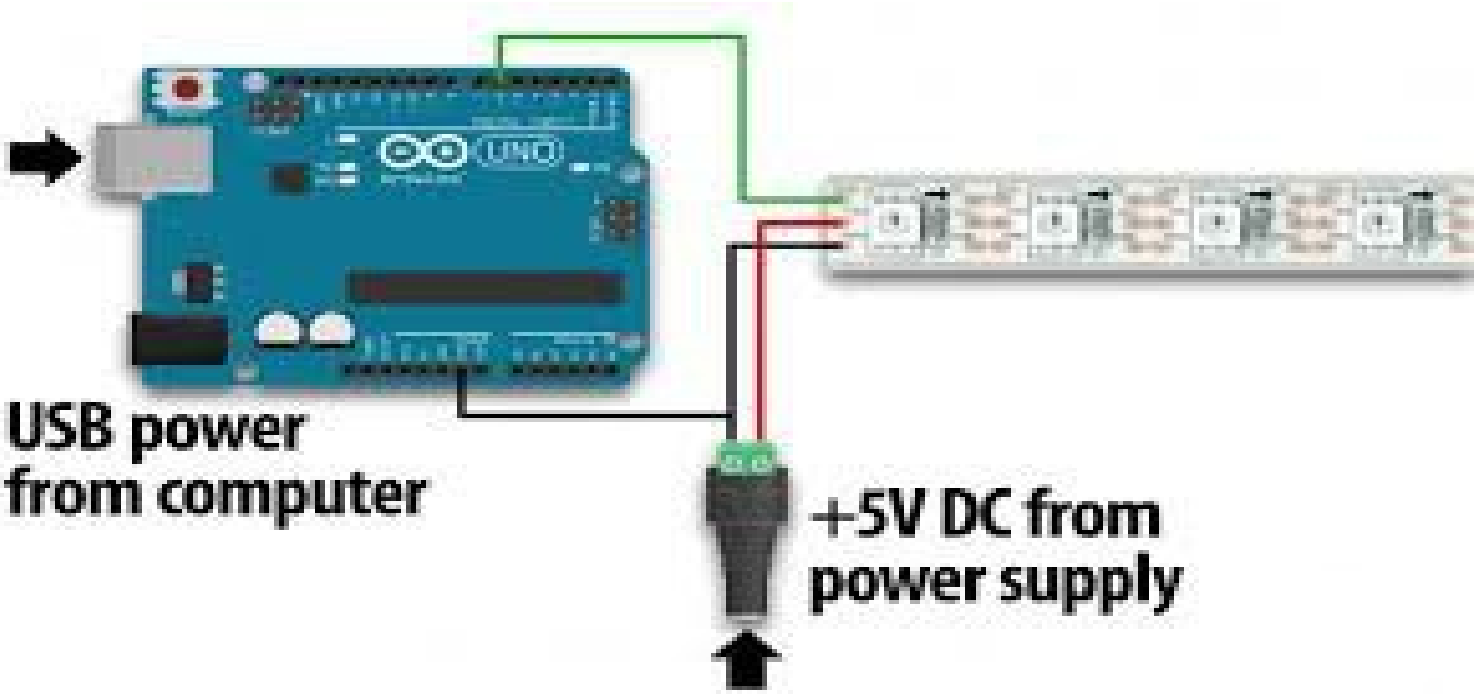

* Needs 5.0 Volt Power
* Needs 5.0 Volt Signal
* Can draw AMPS of Power with many LEDs
* Do NOT do this with more than a few LEDs, it can fry your Arduino boards...




Source : Sparkfun

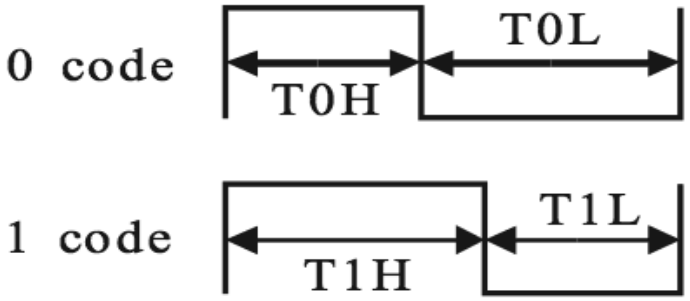
# WS2812 Wiring

* Best Practice is to power the LED strings separately:



# WS2812 Timing

* The older WS2810 was a 400KHz serial data signal
* The newer WS2811/WS2812 are now 800KHz data signals




**Data transfer time** (TH+TL=1.25μs±300ns)

| | | | |
|---|---|---|---|
| T0H | 0 code ,high voltage time | 0.4us | ±150ns |
| T1H | 1 code ,high voltage time | 0.8us | ±150ns |
| T0L | 0 code , low voltage time | 0.85us | ±150ns |
| T1L | 1 code ,low voltage time | 0.45us | ±150ns |
| RES | low voltage time | Above 50μs | |

Source : Adafruit

# WS2812 LED Color Order

**Composition of 24bit data:**

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G7 | G6 | G5 | G4 | G3 | G2 | G1 | G0 | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |

* Eight (8) bits each for Green, Red, Blue
* Different chips can use different color orders (beware!)
* Some add White or Yellow LEDs for a 32 bit color pixel

Source : Adafruit

# Arduino Libraries

* Adafruit NeoPixel Library :
  Well written, supports AVR, ESP8266, SAMD, etc.

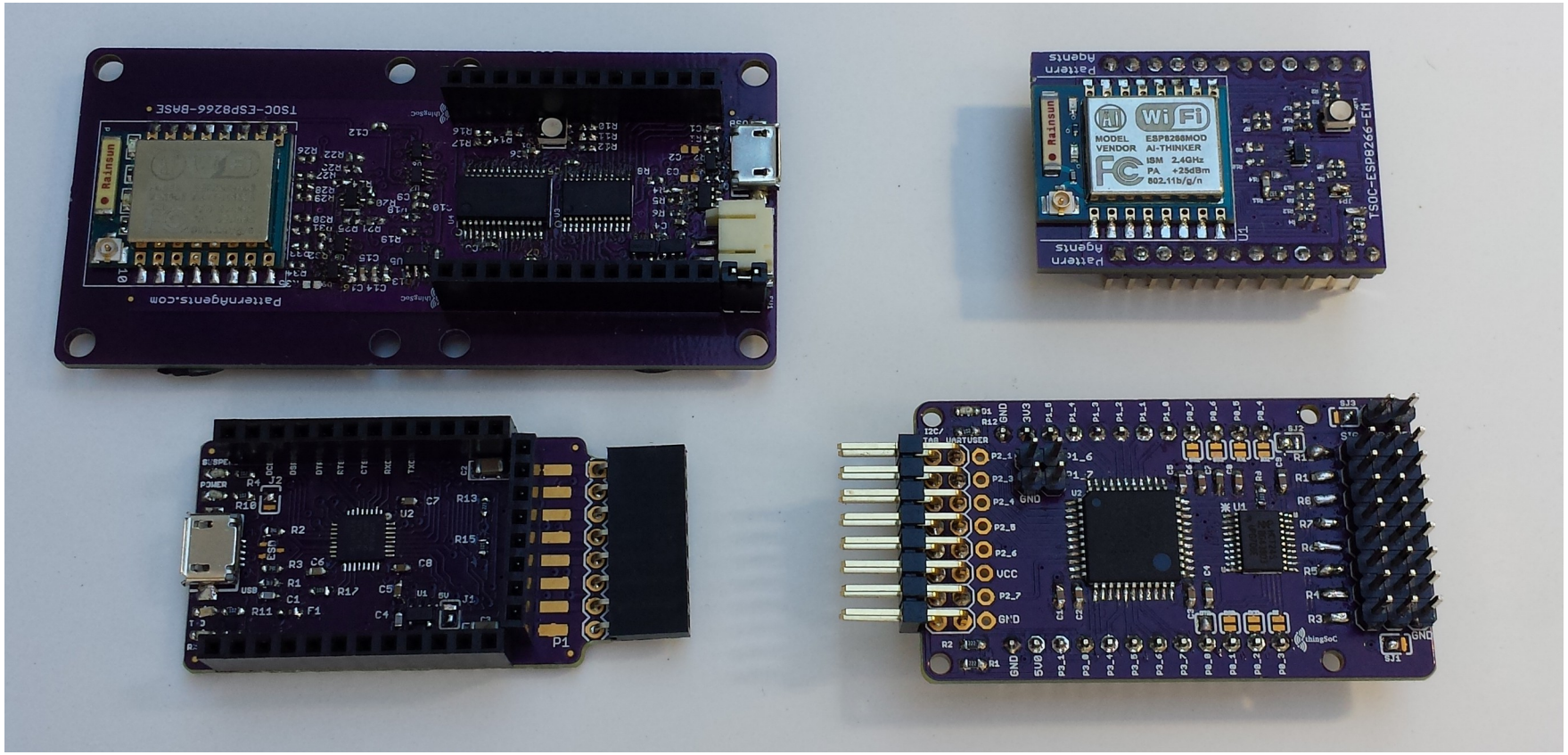
* FastLeds Library :
  Optimized, with higher performance and update rates

* Problems :
  1) Can interfere with other functions, like audio, servos, etc.
  2) Can use up a lot of the “scarce” memory resource
  3) Doesn't work well with the Edison, Currie, Galileo, etc.

Source : Adafruit

# thingSoC is a OSHW Standard

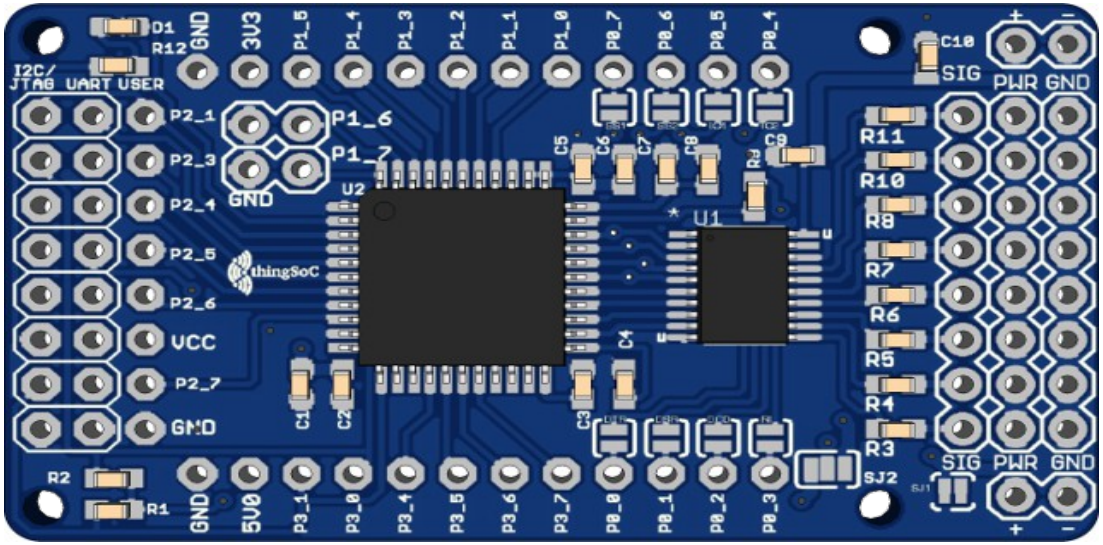
Growing into a family of inter-operable products:




source: @PatternAgents

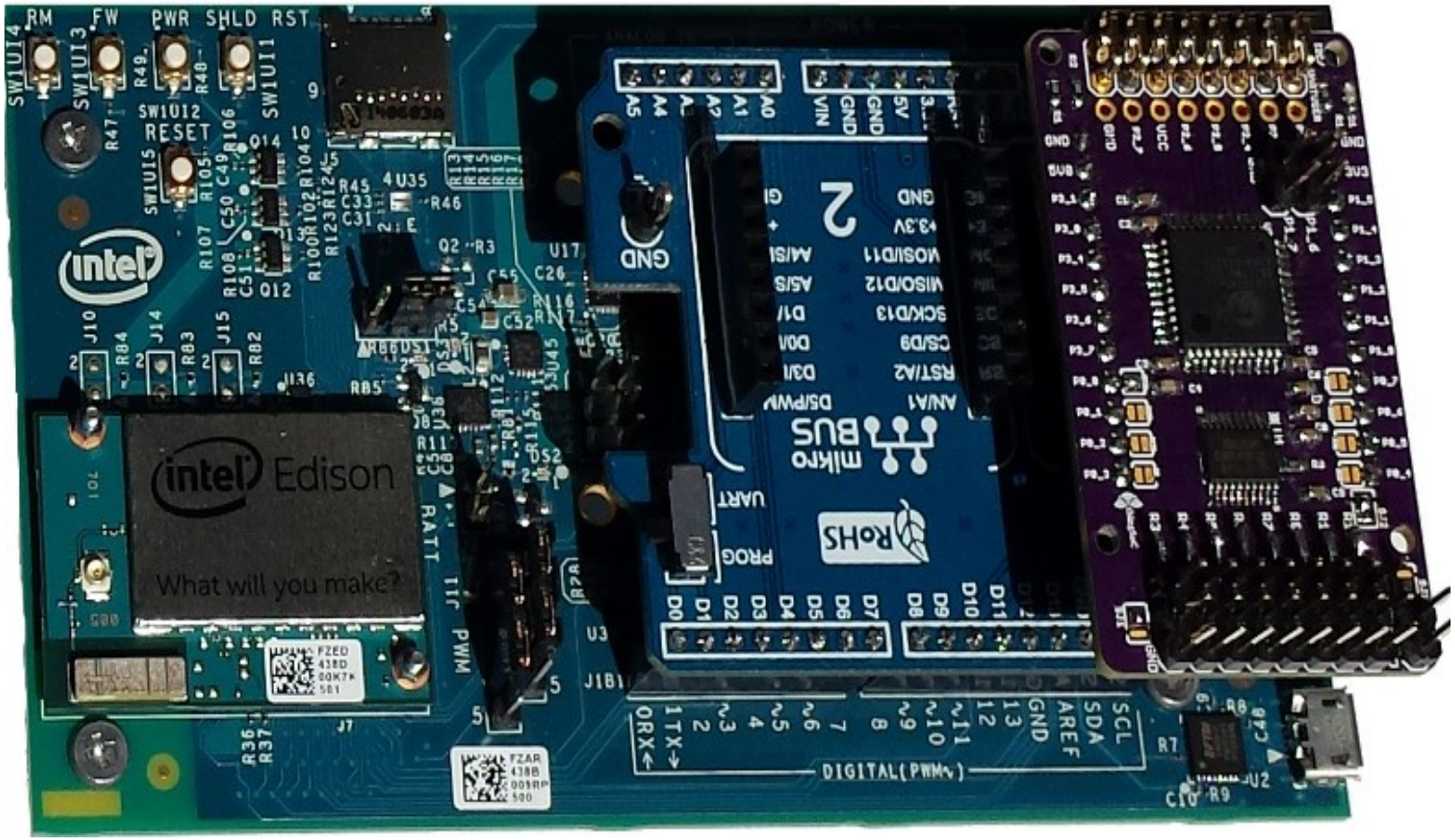
# TSOC_NEOLED_EM

* Eight (8) Channel RBG(Y) LED Driver Board (Embeddable Module)
  ***Makes driving LED arrays as easy as writing to an I2C memory***
* Reprogrammable for other functions (servo motors, etc.)
* Works with other thingSoC products
* Coming soon to Crowd Supply here in Portland!




Source : @PatternAgents

# Driving WS2812 with Edison




source: @PatternAgents

# Driving WS2812 with Edison

* While there are ways to drive WS2812 LEDs directly from the Edison (MRAA), they are not easy and have other problems.
* Using the TSOC_NEOLED board eliminates the timing and voltage level translation issues for you.
* The TSOC_NEOLED board looks like a simple I2C memory device to the Edison, simplifying your code.

source: @PattemAgents

## TSOC_NEOLED Arduino Sketch Examples

Under the "examples" directory are several Arduino IDE examples for using the TSOC_NEOLED board with Arduino IDE. Using the Arduino "Wire" library for I2C communications, it is simple to control the TSOC_NEOLED

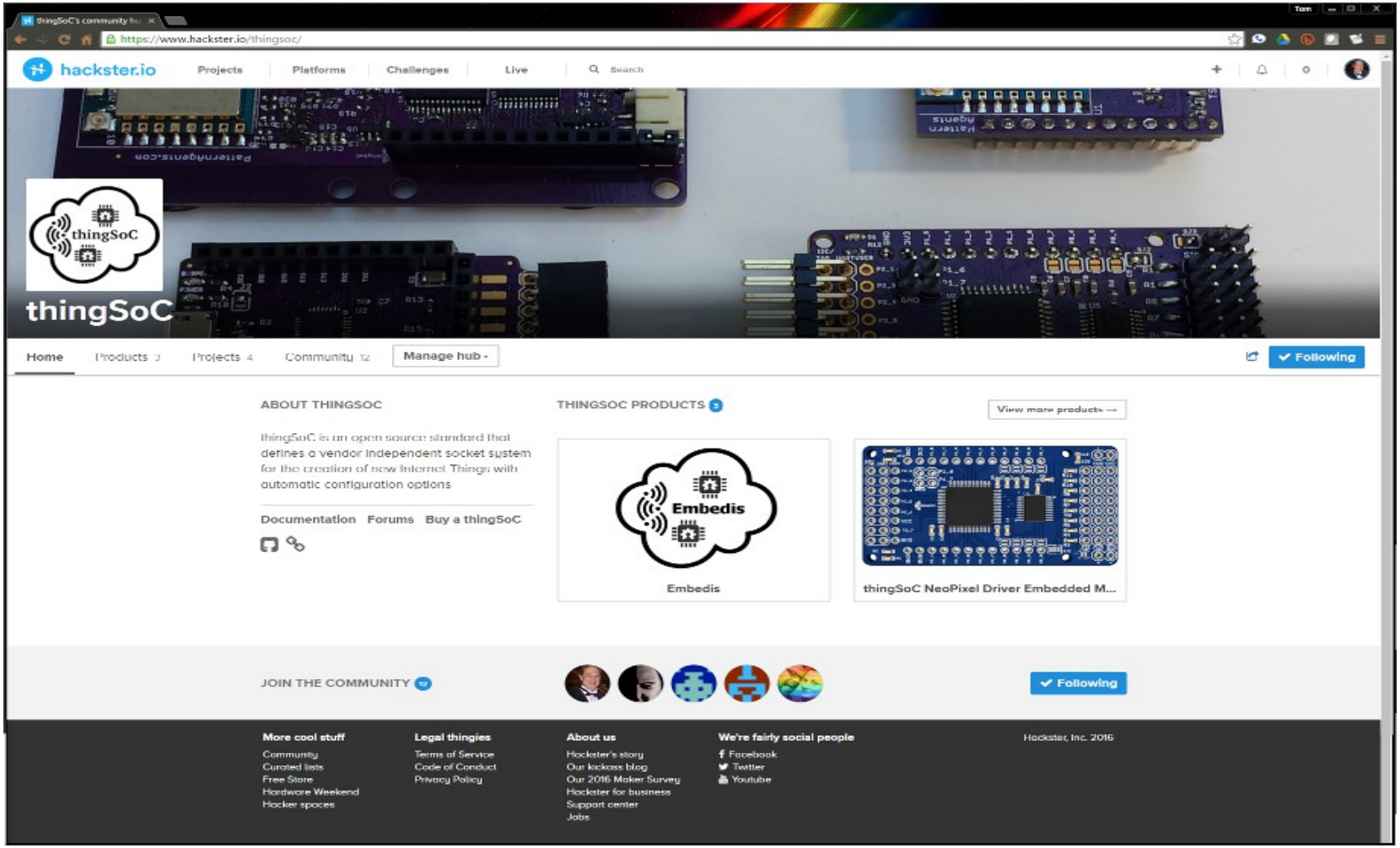
```
// Start the Arduino I2C Interface
wire.begin();


// Turn the first LED On (i.e. green LED #1)
// the LED's are (GRBY) color order in memory
byte row, column = 0;
Wire.beginTransmission(TSOC_NEOLED_I2CADDRESS);  // Start the I2C transaction
Wire.write(row);                                 // Send the high byte of 16 bit memory address
Wire.write(column);                              // Send the low byte of 16 bit memory address
Wire.write(TSOC_NEOLED_LED_ON);                  // Send a single data byte  (we could send more...)
Wire.endTransmission();                          // End the I2C transaction

// Turn the first LED Off (i.e. green LED #1)
// the LED's are (GRBY) color order in memory
Wire.beginTransmission(TSOC_NEOLED_I2CADDRESS);  // Start the I2C transaction
Wire.write(row);                                 // Send the high byte of 16 bit memory address
Wire.write(column);                              // Send the low byte of 16 bit memory address
Wire.write(TSOC_NEOLED_LED_OFF);                  // Send a single data byte  (we could send more...)
Wire.endTransmission();                          // End the I2C transaction
```

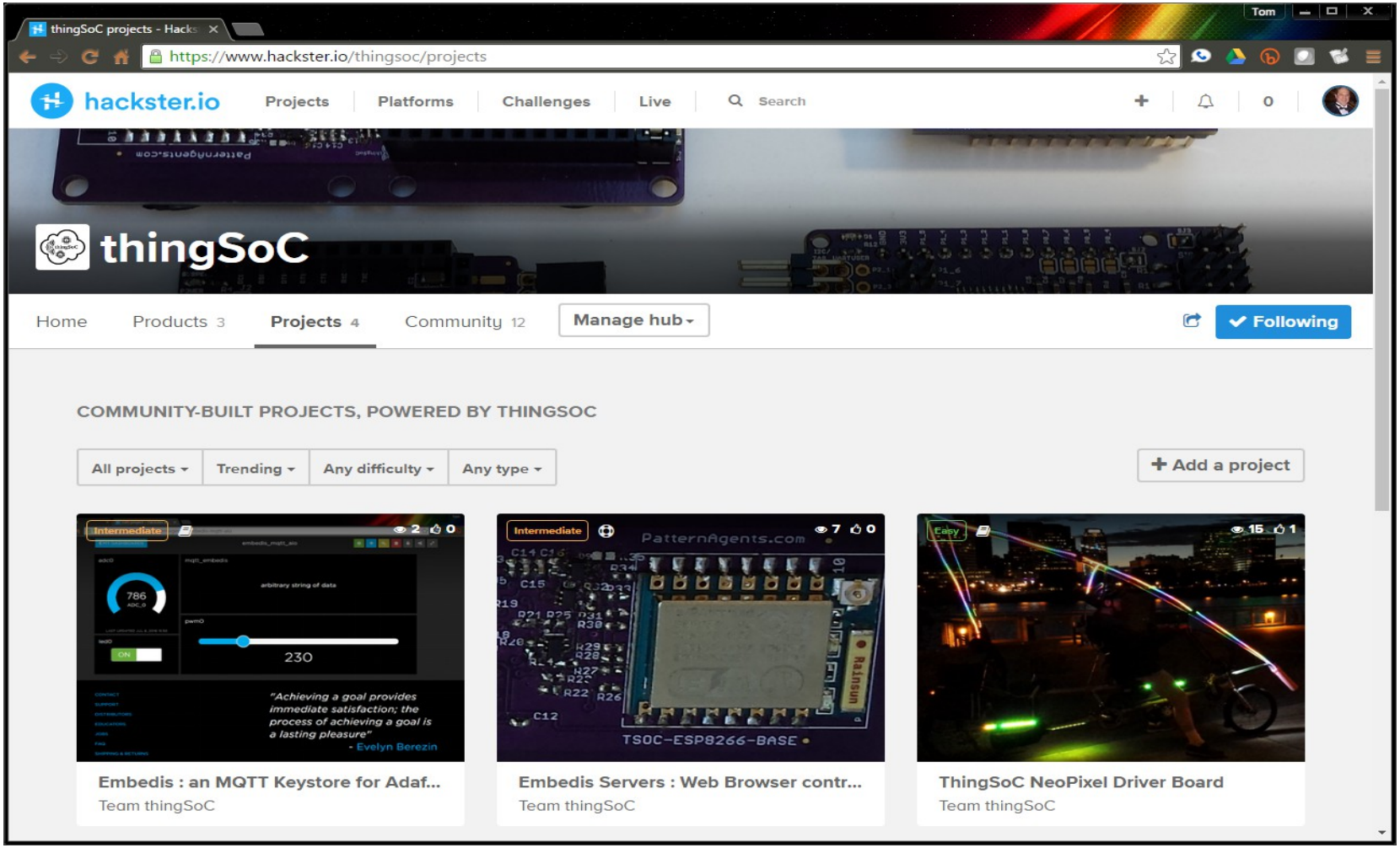
source: @PatternAgents

# Hackster.io Platform




source: PatternAgents

# Hackster.io Projects




source: PatternAgents

# If you would like to help…

* Sign Up for Hackster.IO
* Please go to :
  https://www.hackster.io/thingsoc
  and “Click” on the blue “Following” button
* Look for our Crowd Supply campaign
  coming soon!

# Thank You !

Thank You!

Source : @PatternAgents

# @Pattern Agents

* PatternAgents supplies Open Source Design Patterns for Hardware, Firmware, Software, and Applications.

* A tailor or dressmaker is very familiar with the concept of a "pattern"; they select a “pattern” for the style of garment that they want to make, and then size that “pattern” for the customers individual measurements.

* PatternAgents is doing the same thing for  electronics, in embedded systems for communications, robotics, metering and automation.