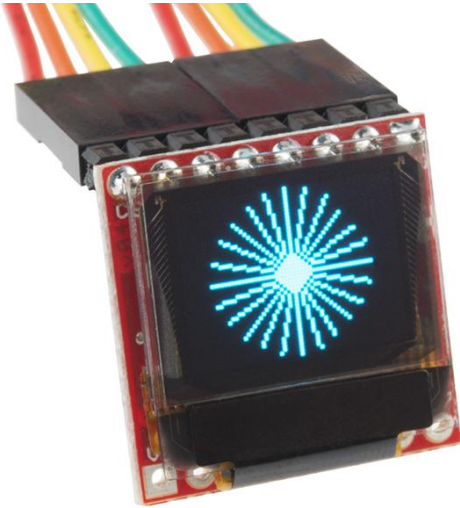


Photon OLED fun 265xp possible!

<https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-for-photon-experiment-guide/experiment-11-oled-apps---weather--clock>



Part 1. A basic counter 15xp

Step 1 Hook up the OLED

Follow the hardware instructions [here](#). You only need to hook up the oLED display. You don't need the potentiometer or the light sensor.

Step 2 The basic code

```
// This #include statement was automatically added by the Particle IDE.
#include "SparkFunMicroOLED/SparkFunMicroOLED.h"

#include "math.h"

////////////////////////////////////////
// MicroOLED Definition //
////////////////////////////////////////
#define PIN_OLED_RST D6 // Connect RST to pin 6
#define PIN_OLED_DC D5 // Connect DC to pin 5 (required for SPI)
#define PIN_OLED_CS A2 // Connect CS to pin A2 (required for SPI)
MicroOLED oled(MODE_SPI, PIN_OLED_RST, PIN_OLED_DC, PIN_OLED_CS);
int n = 0;

void setup()
{
    oled.begin(); // Initialize the OLED
}

// loop() looks at the clockState variable to decided which digital watch
// function to draw on the screen.
void loop()
{
    oled.clear(PAGE);
    oled.setFontType(3); // set font to large
    oled.setCursor(0,0); // the top left of the screen
    oled.print(n); // print the number
    oled.display();
    delay(1000); // delay one second
    n = n + 1;
}
```

Step 3 Save the code

Step 4 Adding the required library

See the instructions [here](#) right under the picture of the redboard

Step 5 Flash the code

You should see the display showing incrementing numbers.

Part 2. Countdown timer 15xp

Create a timer that counts down from 100 (100, 99, 98, 97 ...)

Part 3. Temperature and Humidity - 20xp

Step 1. Hook up the hardware for the temp/humidity sensor

As shown [here](#).]

Step 2. The code (make sure to save it)

```
/// This #include statement was automatically added by the Particle IDE.
#include "SparkFunRHT03/SparkFunRHT03.h"

////////////////////////////////////
// Pin Definitions //
////////////////////////////////////
const int RHT03_DATA_PIN = D3; // RHT03 data pin
const int LED_PIN = D7; // LED to show when the sensor's are being read
double temp = 0;
double humidity = 0;
////////////////////////////////////
// RHT03 Object Creation //
////////////////////////////////////
RHT03 rht; // This creates a RTH03 object, which we'll use to interact with the
sensor
```

```

void setup()
{
    rht.begin(RHT03_DATA_PIN); // Initialize the RHT03 sensor

    // Don't forget to set the pin modes of our analog sensor (INPUT) and the
    LED (OUTPUT):
    pinMode(LED_PIN, OUTPUT); // Set the LED pin as an OUTPUT
    digitalWrite(LED_PIN, LOW); // Initially set the LED pin low -- turn the
    LED off.
    Particle.variable("humidity", &humidity, DOUBLE);
    Particle.variable("temp", &temp, DOUBLE);
}

void loop()
{
    digitalWrite(LED_PIN, HIGH); // Turn the LED on -- it'll blink whenever the
    sensor is being read.

    // Use the RHT03 member function `update()` to read new humidity and
    temperature values from the sensor.
    // There's a chance the reading might fail, so `update()` returns a success
    indicator. It'll return 1
    // if the update is successful, or a negative number if it fails.
    int update = rht.update();

    if (update == 1) // If the update succeeded, print out the new readings:
    {
        humidity = rht.humidity();
        temp = rht.tempF();
    }
    delay(RHT_READ_INTERVAL_MS); // The RHT03 needs about 1s between read
    attempts

    digitalWrite(LED_PIN, LOW); // Turn the LED off

    delay(2000); // delay 2 seconds
}

```

Step 3. Add the library

Add the **SparkFunRHT03** library as you did the OLED one above.

Step 4. Flash and check the results

https://api.particle.io/v1/devices/DEVICE_ID/temp?access_token=ACCESS_TOKEN

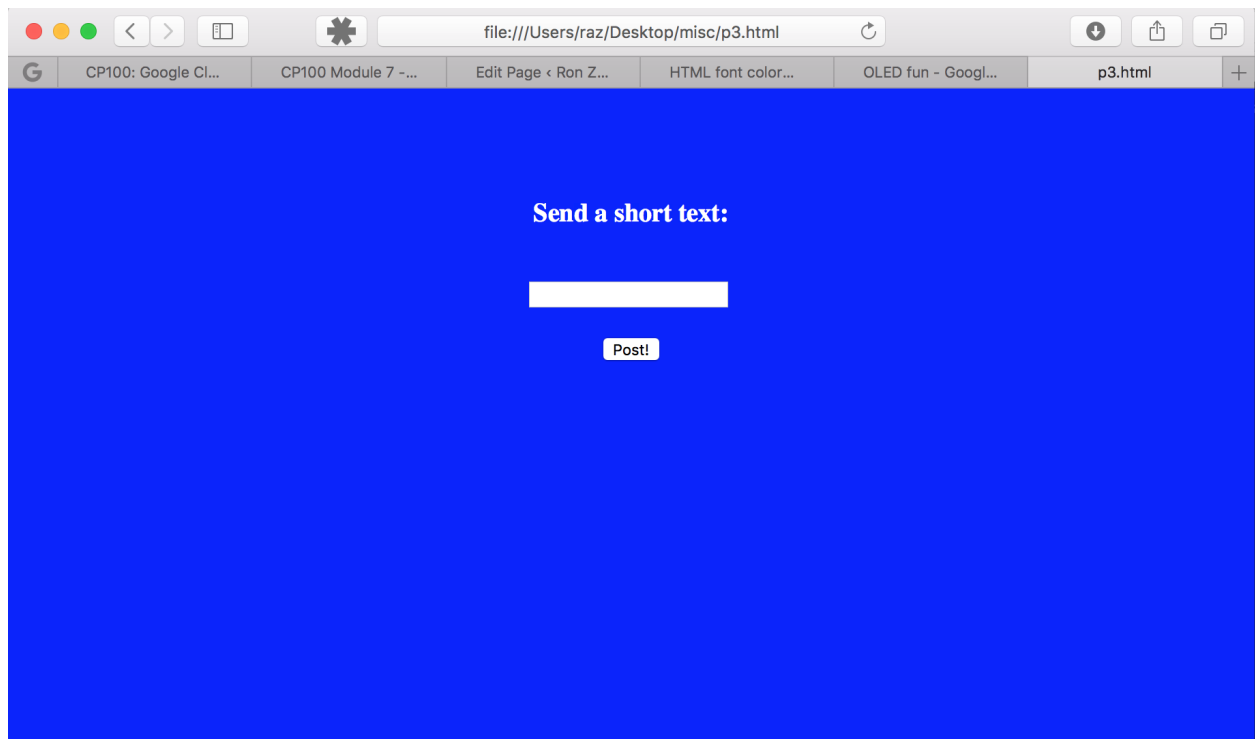
b

Part 4. weather 45xp

I would like the display to indicate the current temperature and humidity level. Here are [pictures using some previous hardware](#).

Part 5. mini-Twitter 45xp

I would like a web interface like this:



(to get that text input box I used:

```
<input type="text" name="args"></input>
```

I would like the text the person types to appear on the OLED display.

Add the **HTTPCLIENT** library as you did the OLED one above.

Part 6. A clock 15xp

```
#include "math.h"

////////////////////
// Button Pins //
////////////////////
#define MODE_PIN    D2

////////////////////
// MicroOLED Definition //
////////////////////
#define PIN_OLED_RST D6 // Connect RST to pin 6
#define PIN_OLED_DC  D5 // Connect DC to pin 5 (required for SPI)
#define PIN_OLED_CS  A2 // Connect CS to pin A2 (required for SPI)
MicroOLED oled(MODE_SPI, PIN_OLED_RST, PIN_OLED_DC, PIN_OLED_CS);

// Set your Time zone below, adjust from UTC.
// E.g. -6.0 is mountain time in the United States,
// -4.0 would be eastern, -7.0 pacific
#define TIME_ZONE -5.0 // Mountain time (Niwot, CO; SparkFun HQ!)

////////////////////
// Button Debounce Variables //
////////////////////
// debounceTime defines the number of milliseconds that must pass between
// button presses.
#define debounceTime 100 // 100ms debounce time
unsigned int lastPresTime = 0; // keeps track of the last time a button was pressed

void setup()
{
    pinMode(MODE_PIN, INPUT_PULLUP);
    oled.begin(); // Initialize the OLED
    Time.zone(TIME_ZONE); // Set up the timezone
}

void loop()
{
    displayClock(Time.hour(), Time.minute(), Time.second());
}

// displayClock draws the current hour and minute in a HH:MM format.
// Our large font doesn't leave room for seconds, so we'll use seconds to
// blink the colon.
```

```

void displayClock(unsigned int hours, unsigned int minutes, unsigned int seconds)
{
    oled.clear(PAGE); // Clear the display
    oled.setFontType(3); // Switch to the large-number font
    oled.setCursor(0, 0); // Set the cursor to top-left

    // printWithLeadZero adds as many '0''s as required to fill out a number's
    // digit count. E.g., if hours = 7, this will print "07"
    // The second parameter defines the number of digits a number should fill.
    printWithLeadZero(hours, 2); // Print two-characters for hours

    // Next the colon (:), or not.
    if ((seconds % 2) == 0) // If seconds is even
        oled.print(":"); // Print a colon
    else // Otherwise
        oled.print(" "); // Print a spaces

    // Another printWithLeadZero, this time for minutes.
    printWithLeadZero(minutes, 2);

    oled.display(); // Update the display
}

// Prints as many lead zero's as necessary given a number 'n', and a
// number of 'digits' it should fill out.
// E.g. printWithLeadZero(42, 3) will print "042"
// printWithLeadZero(7, 2); will print "07"
void printWithLeadZero(unsigned int n, int digits)
{
    for (int i=1; i<digits; i++) // Cycle through digit-1 times
    {
        if (n < (pow(10, i))) // If a number is less than 10^i
            oled.print("0"); // Print a leading zero
    }
    oled.print(n); // print the rest of the number
}

// softwareDebounce keeps our button from "bouncing" around. When a button is
// pressed, the signal tends to fluctuate rapidly between high and low.
// This function filters out high-frequency button presses, limiting
// them to debounceTime ms.

```

Part 7. Alarm Clock 45xp

Can you alter the part 6 code to make an alarm clock. You can hard code the alarm time for, for example, 2:30. Normally the OLED displays the current time. You can decide what happens when the alarm goes off. Possibilities include:

- The current time flashes on the screen

- Some buzzer sounds (or short tune) +10xp

When a user presses a button the alarm goes off.

Snooze button +25xp

Can you add a snooze button? When the user presses that button the alarm stops for 10 minutes (well, maybe for debugging, 1 minute is a better snooze period).

Part 8 - a remix of either the basic clock or the alarm clock +15

Can you add a temp reading to the clock? So it displays the time for 5 seconds say, then switches to the temp for 5, and then back again to the clock?

Part 9 - the magic 8 ball 25- 50 xp

Remember those magic 8 balls?



If not check out [this wikipedia page on Magic Eight Balls](#)

I would like you to implement a Magic Eight Ball using the LCD and any other components you want. The more creative your implementation the more XP.

You may want to generate a random number. To do so you can use the function `random`. For example,

```
random(10);
```

Will generate a random number from 0 to (and including) 9.

Some thoughts off the top of my not very creative head:

- press a button and display a random answer
 - simply display the answer
 - answers are wizzing by and when you press the button it stops on one answer.
 - the button press starts the wizzing (something like a roulette wheel) which gradually slows and displays the answer

+20 xp: how about you press a button and the photon sends a random positive text message to your phone?