

Innovative Tech Events for Libraries: Engage, Educate, and Inspire

Chess

- **Fast Chess:** https://en.wikipedia.org/wiki/Fast_chess
- **To play chess in the browser locally:** <http://plainchess.timwoelfle.de/>
- **A free chess clock:** <https://chessclock.org/>
- **A demonstration of speed chess using a giant chess set:** <https://youtu.be/MunplSRRZIM?si=vce1ITvG90HuxqZI>

Weather Sensor

The weather sensor we built ran on an Arduino Mega 2560, an open-source microcontroller with a massive community of creators. You can learn more about Arduino here: <https://www.arduino.cc/>

Parts List

- 1 Arduino Mega 2560 (or compatible microcontroller)
- 1 USB A to B cable (for plugging the microcontroller into a computer for programming)
- 1 breadboard
- 1 5461AS or YY3651AH 4-digit 7 segment displays (compatible models)
- 1 red LED
- 1 blue LED
- 1 DHT11 Module (it should have 3 pins)
- 1 button
- 6 1K Ω resistors
- 2 short red wires
- 1 short black wire
- 1 long red wire
- 1 long black wire
- 4 short matching color wires
- 16 long wires

Code

Written in C++ in the Arduino IDE. You can find guides to help with coding: <https://www.arduino.cc/en/Guide/>

```

/**
 * DHT11 Weather Info Readout
 * This sketch reads humidity and temperature from a DHT11 sensor and outputs it
 * on a seven segment display, using two LEDs to indicate the current display mode
 * and a button to toggle between modes. In the spirit of the open-source community
 * and making electronics projects accessible to everyone, this code is entirely
 * free to use.
 *
 * Author: Alex Griffin
 * License: MIT
 */

#include <DHT11.h>
#include <SevSeg.h>

// Declare what pins the components connect to
const int SENSOR_PIN = 13;
const int BUTTON_PIN = 12;

const int TEMPERATURE_LIGHT_PIN = 11;
const int HUMIDITY_LIGHT_PIN = 10;

const int DIGIT_ONE = 7;
const int DIGIT_TWO = 6;
const int DIGIT_THREE = 5;
const int DIGIT_FOUR = 4;
const int SEGMENT_A = 53;
const int SEGMENT_B = 52;
const int SEGMENT_C = 50;
const int SEGMENT_D = 48;
const int SEGMENT_E = 46;
const int SEGMENT_F = 44;
const int SEGMENT_G = 42;
const int SEGMENT_DOT = 40;

// Logic variables
int timer = 0; //How many repeats since last read the sensor
const int READ_SENSOR_WHEN = 10000;
int temperature;
int humidity;
int output; // What will be displayed on the screen
bool mode = 0;
bool pressingButton = 0; // Makes switching modes easier for the user
/*
 * temperature mode = 1
 * humidity mode = 0
 */

// Name and start libraries.
DHT11 sensor(SENSOR_PIN);
SevSeg screen;
void setupScreen() {
    byte numDigits = 4;
    byte digitPins[] = {
        DIGIT_ONE,

```

```

    DIGIT_TWO,
    DIGIT_THREE,
    DIGIT_FOUR
};
byte segmentPins[] = {
    SEGMENT_A,
    SEGMENT_B,
    SEGMENT_C,
    SEGMENT_D,
    SEGMENT_E,
    SEGMENT_F,
    SEGMENT_G,
    SEGMENT_DOT
};
screen.begin(COMMON_CATHODE, numDigits, digitPins, segmentPins);
}

// Define important functions
void updateScreen(int number) {
    screen.setNumber(number, 0);
    screen.refreshDisplay();
}

void readSensor() {
    /* Reads information from the sensor and stores it in these variables:
     * results: stores the error code, if any (otherwise it stores "0" to say there's
no errors)
     * temperature: stores temperature
     * humidity: stores humidity
     */
    int results = sensor.readTemperatureHumidity(temperature, humidity);
    if (results != 0) {
        // Prints the error code to the serial monitor
        Serial.println(DHT11::getErrorString(results));
    }
}

void pressButton() {
    if (digitalRead(BUTTON_PIN) & !pressingButton) {
        mode = !mode;
        pressingButton = 1;
    }
    if (!digitalRead(BUTTON_PIN)) {
        pressingButton = 0;
    }
}

void setup() {
    Serial.begin(9600); //Optional, adds shell to print debug messages in
    Serial.println("Hello world"); //Optional, example message
    setupScreen();

    // Setup lights
    pinMode(TEMPERATURE_LIGHT_PIN, OUTPUT);
    pinMode(HUMIDITY_LIGHT_PIN, OUTPUT);
}

```

```

// Setup button
pinMode(BUTTON_PIN, INPUT);

// The sensor has already been set up on line 38

// Initial temperature read
readSensor();
}

void loop() {

  pressButton(); // Changes mode

  // Determine output based on mode
  if (mode) {
    output = temperature;
  } else {
    output = humidity;
  }

  /* Determine whether to show information on
  the display or read the sensor */
  timer++;
  if (timer < READ_SENSOR_WHEN) {
    // Show information on the display
    updateScreen(output);
    digitalWrite(TEMPERATURE_LIGHT_PIN, mode);
    digitalWrite(HUMIDITY_LIGHT_PIN, !mode); // Opposite of temp light
  }
  else {
    timer = 0; // Resets the timer, to loop again
    readSensor();
  }
}

```

Sphero

Sphero is a robotics company that makes a wide line of programmable robots designed for complete beginners to the world of robotics and programming. We use the Sphero BOLT at our library and learning commons. You can find out more about their company and available robots here:

<https://sphero.com/>

Code



References

- **3D Printed Cardboard Maze Connectors:** <https://www.thingiverse.com/thing:6991196>
- **Sphero STEAM Activities: Red Light, Green Light:** https://youtu.be/Whgmil_UkLY?si=7y-94aHckx0JLyQY&t=70%E2%80%8B
- **Red Light, Green Light Program (and the source of inspiration for our code):** <https://edu.sphero.com/remixes/1666750>

3D Printed Mascot Technologies

Our mascot coasters were designed in Blender, a free and open-source 3D modelling platform. You can find more information about Blender here: <https://www.blender.org/about/>

Our marketing department uses Pantone colors for our mascot, so we used the Pantone color system to find the right color plastic for the job. This process involved market research into what companies would

make custom-colored filaments, their unit costs, and shipping times. You can learn more about Pantone here: <https://www.pantone.com/about-pantone>

We used a Dremel Digilab 3D45 3D printer for our project and two different colored plastic filaments to produce these coasters. You can use any 3D printer that prints in plastic filament (as opposed to resin, or other technologies).