

Useless Joke Bot

Authors: Jennifer Biette Kowalski, Lucia Cipolina Kun

What is the Useless Joke Bot?

The Useless Joke Bot is a machine that tells you a joke with the wrong punchline, every time. For instance, “How do you cut the ocean in half?” “The library.” Built with Arduino, experience the complete and utter lack of a payoff as you wait for a joke to scroll across the LCD display with a nonsensical response.

Required hardware for the display

- Arduino Uno
- LCD screen
- Breadboard(s)
- Jumper wires
- Push button
- Alligator clips (if using an arcade push button)
- 10k ohm Potentiometer
- 220 ohm Resistor
- battery/powersource for Arduino

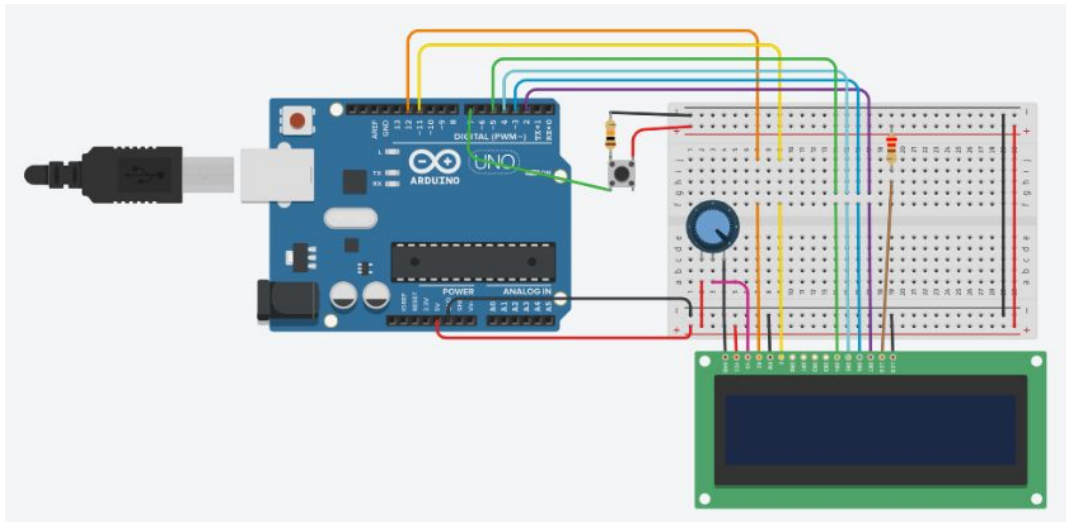
Required materials for the box

- Balsa wood (or other wood)
- Box Cutter (or other way to cut wood)
- Wood glue/gorilla glue
- Tape or clamps

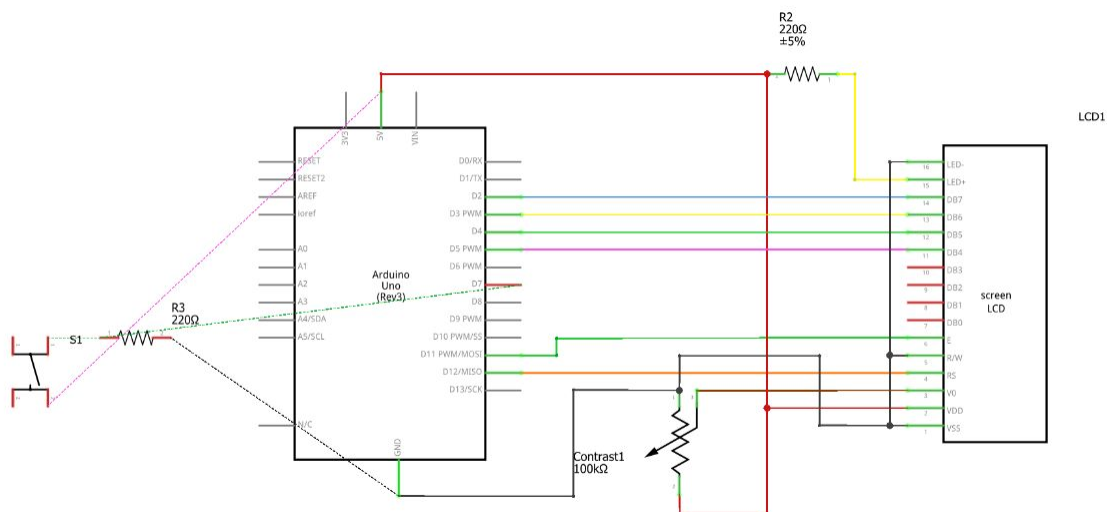
Circuit

The breadboard circuit and schematics to replicate the project are below. Here's how to build your own Useless JokeBot:

1. Connect the LCD to the Arduino board following the schematics below. This is the same connection settings as explained in the official ["HelloWorld" Arduino tutorial](#) [1].
2. Add a push-button connected to Pin 7 in one leg and to a pull-up resistor in the other leg. The schematic shows a standard 4-pin button. To use an [arcade button](#) like the one we used in our example, connect one pin of the push-button to Pin 7 and the other to power in the breadboard.



Schematic



Source Code

Here's the programming rationale behind the code, which you can download from the [hackaday site](#). The main idea is to generate a punchline that does never matches the joke. This makes the whole sentence not particularly funny, and probably confusing.

To achieve this, we placed each joke and its corresponding punchline on two different arrays and picked the unmatching elements of both. The code is written so that they never appear together in the same sentence. This is achieved by generating two different random numbers, which are used as the location index on each array. The rest of the code takes care of the display and the responses to the button state.

```
/* Useless Joke Bot
   Creates messed up jokes that are not funny.*/

/* The circuit:
 * LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 * LCD R/W pin to ground
 * 10K resistor:
 * ends to +5V and ground */

// include the library code:
#include <LiquidCrystal.h>

// === ARDUINO CONNECTIONS ===
// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to

// initialize the library with the numbers of the interface pins
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// === CONSTANTS AND GLOBALS ===
//set button variables and pin numbers:
const int buttonPin = 7;    // constant number of the pushbutton pin
int currState = LOW;
int prevState = LOW;

//set variables to hold a random number generated by the random() function
int randomNumberOne;
int randomNumberTwo;
int arr[2];    //initializes an array of 2 integers//writing coordinates: it will
write to the upper most line
```

```

//Define the Jokes arrays. The first is the joke and the second the punchline
const char *array1[] = {
  "How do you cut the ocean in half?",
  "What has no legs but can do a split?",
  "What kind of shoes do frogs wear?",
  "What do you get when you cross a pig and a Christmas tree?",
  "What do you call a lazy kangaroo?",
  "What foods are good for young people?",
  "What is a mummy's favorite food?",
  "What building has the most stories?",
  "What kind of tea cannot be taken into space?",
  "What did one triangle say to the other triangle?",
  "What happens when you cross a singer and a rocking chair?",
  "What do you call a pig that does karate?",
  "What goes tick-tock, woof-woof?",
  "Why is a bad joke like a bad pencil?",
  "Why do fish always sing off key?",
  "What do you call a pile of cats?",
  "What kind of tea is the hardest?",
  "What did one campfire say to the other?",
  "Why is a pancake like the sun?",
  "What room can no one enter?"
};
const char *array2[] = {
  " With a sea saw.",
  " A banana.",
  " Open toed.",
  " A porcupine.",
  " A pouch potato.",
  " The pro-teens!",
  " Wraps.",
  " The library.",
  " Gravity.",
  " Let's get together and square dance.",
  " You rock to the beat.",
  " A pork chop.",
  " A watch dog.",
  " Because it has no point.",
  " Because you can't tuna fish.",
  " A meowtain.",
  " Reality.",
  " Let's go out one of these days!",
  " Because it rises in the yeast.",
  " A mushroom.",
};

void setup() {
  //Set up the LCD's number of columns and rows:
  lcd.begin(16, 1); //We define the LCD as having only one row as we will only
  display on the top row
  //Serial monitor
  Serial.begin(9600);
  //Initialize the pushbutton pin as an input:

```

```

        pinMode(buttonPin, INPUT);
    }

    void loop() {
        currState = digitalRead(buttonPin);
        delay(5); //delay a little bit to avoid debouncing (i.e. noise reading from a
        button)

        //Serial.println(currState); //To DEBUG

        //Display "Press button" text the first time
        lcd.print( "Press button  ");

        // -- Read the state of the pushbutton value and displays text --
        if (currState != prevState && currState) // rising edge
        {
            //Changes the values inside the array or random ints
            rndNbr(arr);
            //Generates and displays rnd text
            rndText(arr); //the 'arr' has been modified by the previous function
            //Displays "Press button" text
            reset();
        }

        //Save the current state as the last state, for the next time through the loop
        prevState = currState;
    }

    void rndNbr(int arr[]){
        // -- generate a random number and changes the value of the array passed--

        //Set rnd seed
        randomSeed(millis() + analogRead(A0) + analogRead(A1) + analogRead(A2) +
            analogRead(A3) + analogRead(A4) + analogRead(A5));

        //Size of arrays
        //The first call will tell you how much memory the whole array is using. The
        tells you how much memory one element is using. The division then tells you how many
        elements in the array.
        const int arrLen1 = sizeof(array1) / sizeof(array1[0]);
        const int arrLen2 = sizeof(array2) / sizeof(array2[0]);

        //Generate random numbers
        randomNumberOne = random(0, arrLen1);
        randomNumberTwo = random(0, arrLen2);

        //Check whether they are different
        do {
            randomNumberTwo = random(0, arrLen2);
        } while (randomNumberOne == randomNumberTwo);

        arr[0] = randomNumberOne;
        arr[1] = randomNumberTwo;
    }

```

```

}

long rndText(int arr[]) {
    //:inputs: arr[] is a Ref to an array of integers (arrays are always passed by
    Ref)

    // -- generate random sentence --
    String TEXT = String(array1[randomNumberOne]) + String(array2[randomNumberTwo]);

    // -- print sentence on the LCD --
    lcd.clear();          //Clear screen first time
    lcd.setCursor(16, 0); //Right most space as we will scroll to the left
    lcd.autoscroll();    //Scrolls the chars from right to left

    //Need to print one char at a time
    for (int iChar = 0; iChar < TEXT.length(); iChar++) {
        lcd.print(TEXT[iChar]);
        delay(400);
    }

    delay(400);          //give time to read
    lcd.noAutoscroll(); // turn off automatic scrolling
    lcd.clear();         // clear screen for the next loop
}

void reset(){
    //Displays text on screen

    String TEXT = String("Press button ");

    // -- print sentence on the LCD --
    lcd.setCursor(16, 0); //Right most space as we will scroll to the left
    lcd.autoscroll();    //Scrolls the chars from right to left

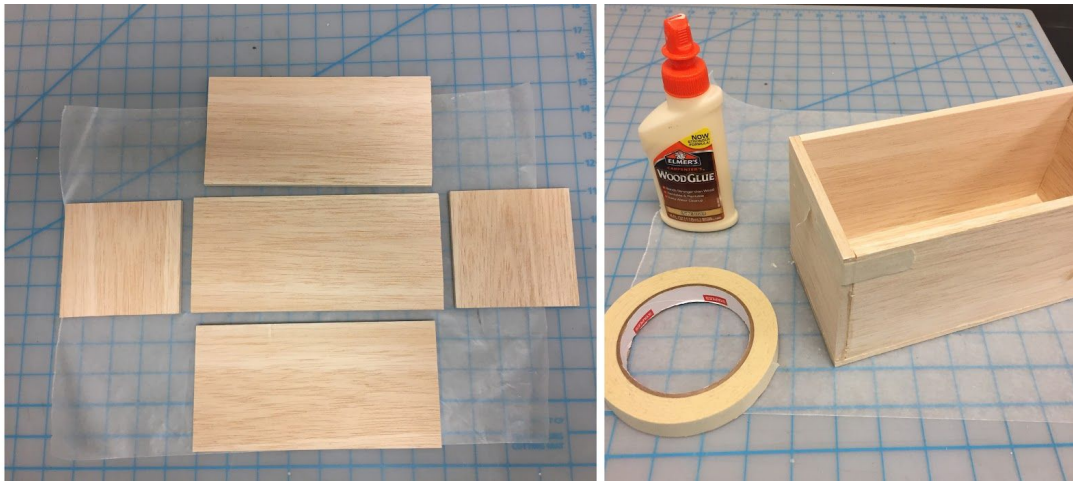
    //Need to print one char at a time
    for (int iChar = 0; iChar < TEXT.length(); iChar++) {
        lcd.print(TEXT[iChar]);
        delay(300);
    }

    lcd.noAutoscroll(); // turn off automatic scrolling
}

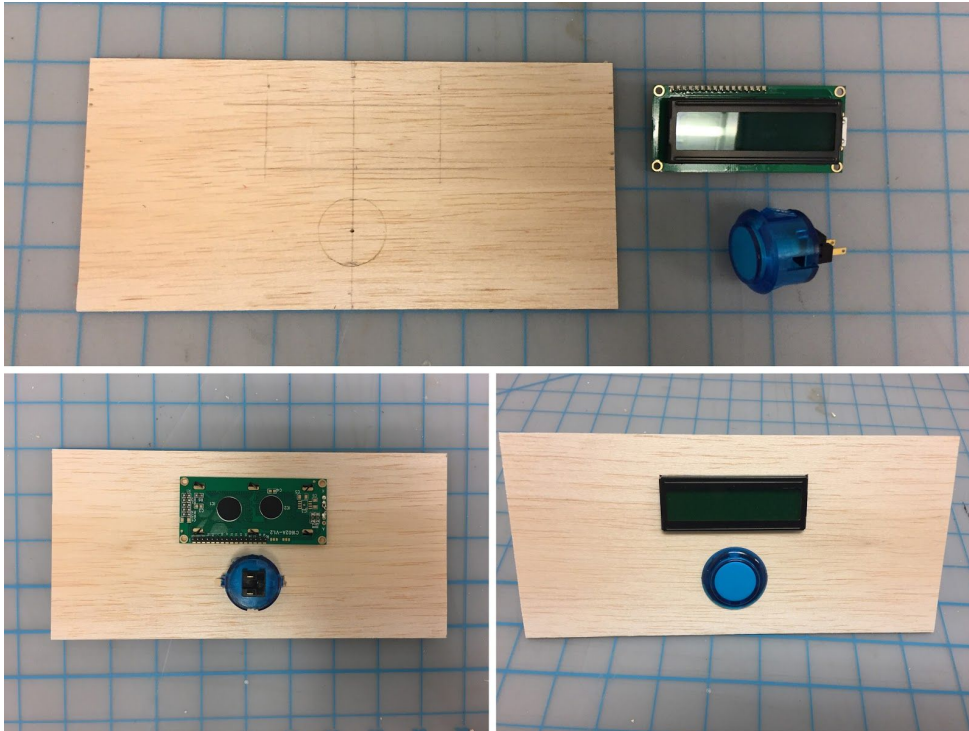
```

Putting it in a box

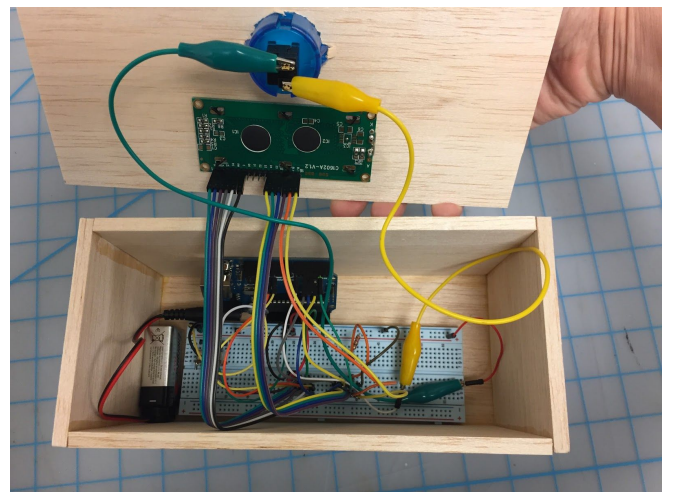
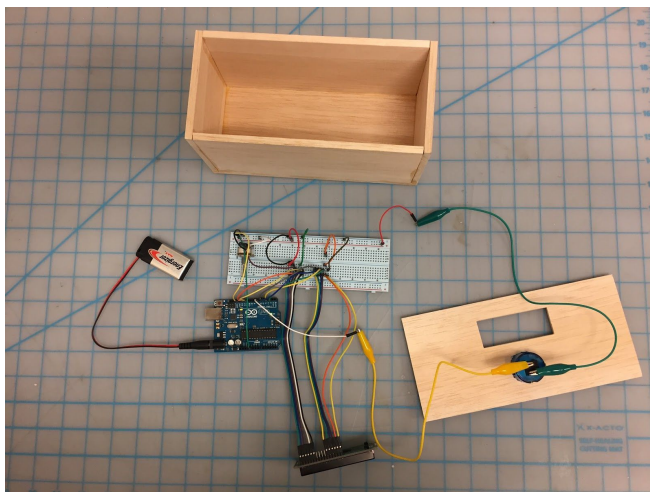
1. Cut the wood to size. The box in this example is made from Balsa Wood, the lid from $\frac{1}{8}$ inch balsa and the sides and base from $\frac{1}{4}$ inch balsa, so that the wood can be cut with a box cutter. Cut the following wood pieces:
 - Long sides: 2 pieces of 8 in x 4 in x $\frac{1}{4}$ in wood
 - Short sides: 2 pieces of 4 in x 4 in x $\frac{1}{4}$ in wood
 - Base: 1 piece of 8 $\frac{1}{2}$ inch x 4in x $\frac{1}{4}$ in wood
 - Lid: 1 piece of 8 $\frac{1}{2}$ inch x 4in x $\frac{1}{8}$ in wood



2. Glue the walls of the box so that the edges of the box sit on top of the base. Clamp with tape.
3. While the glue dries, measure LCD screen and button, cut out hole for each in the lid of box. This particular LCD has a face that is 2 $\frac{3}{4}$ inches x 1 inch and this button needs an approximately 1 $\frac{1}{8}$ inch hole.



4. Once the glue on the box is dry, remove any tape or clamps. Attach the Arduino and other components to a power source like a battery. CAREFULLY place the circuit components in the box. Set the lid on top, taping into place if necessary. Make sure you can still access inner components to reconnect any loose wires or unplug the battery.



References

1. Arduino LCD tutorial
<https://www.arduino.cc/en/Tutorial/HelloWorld>