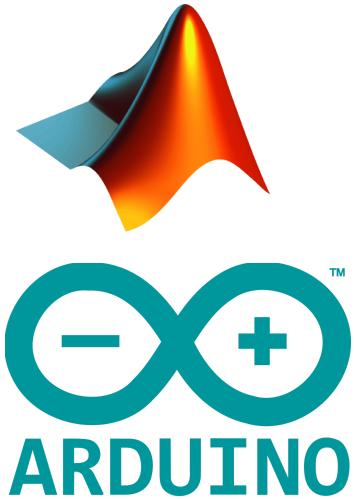




# MATLAB GUI & Arduino Programming | 5



Electrical and Computer Engineering 5  
**Incorporating Arduino into MATLAB**  
Developed by Henry Liu, Alexis Yu, and Nate Seekins

# Overview

In Lab 5, we will be creating a whack-a-mole game that ties in your understanding and experiences with both Arduino and the GUI function on MATLAB. Technical skills that will be built upon include coding with MATLAB, reading fritzing and circuit diagrams, and building their respective circuits. Throughout the challenges, we will guide you in the building of your Arduino board and the whack-a-mole game that we will eventually play via the GUI in MATLAB. By the end of the lab, you will be able to play this virtual game by pressing on the pushbuttons on your Arduino board. We hope that you will have fun learning to make and playing with this arcade styled project!

Some of the technical foundations that will be reviewed in Lab 5 include embedded systems and control and programming in C/C++. Through the use of these topics, you will gain a deeper understanding of the process in which a signal from a push button on a breadboard can be registered on and connected to the MATLAB GUI. You will also be revisiting the user interface capability on the MATLAB GUI and utilizing its features to create this game and its inputs. Lastly, we will be using OnShape to create a 3D prototype of what this game may look like.

# What You Will Need

## Materials:

- (1) Arduino Mega (or Uno)
- (1) USB Cable A-B
- (1) Breadboard
- (4) Pushbuttons
- (5) IR LED
- (1) Piezo Buzzer
- Jumper Cables
- (5) 330 Ohm Resistors
- (4) 10k Ohm Resistors

## Machinery:

- Desktop Computer/Laptop

## Software:

- MATLAB
  - GUI
  - Add Ons: Arduino Hardware
- Arduino IDE

# Challenge #1: Pushbuttons and LEDs

## Objective

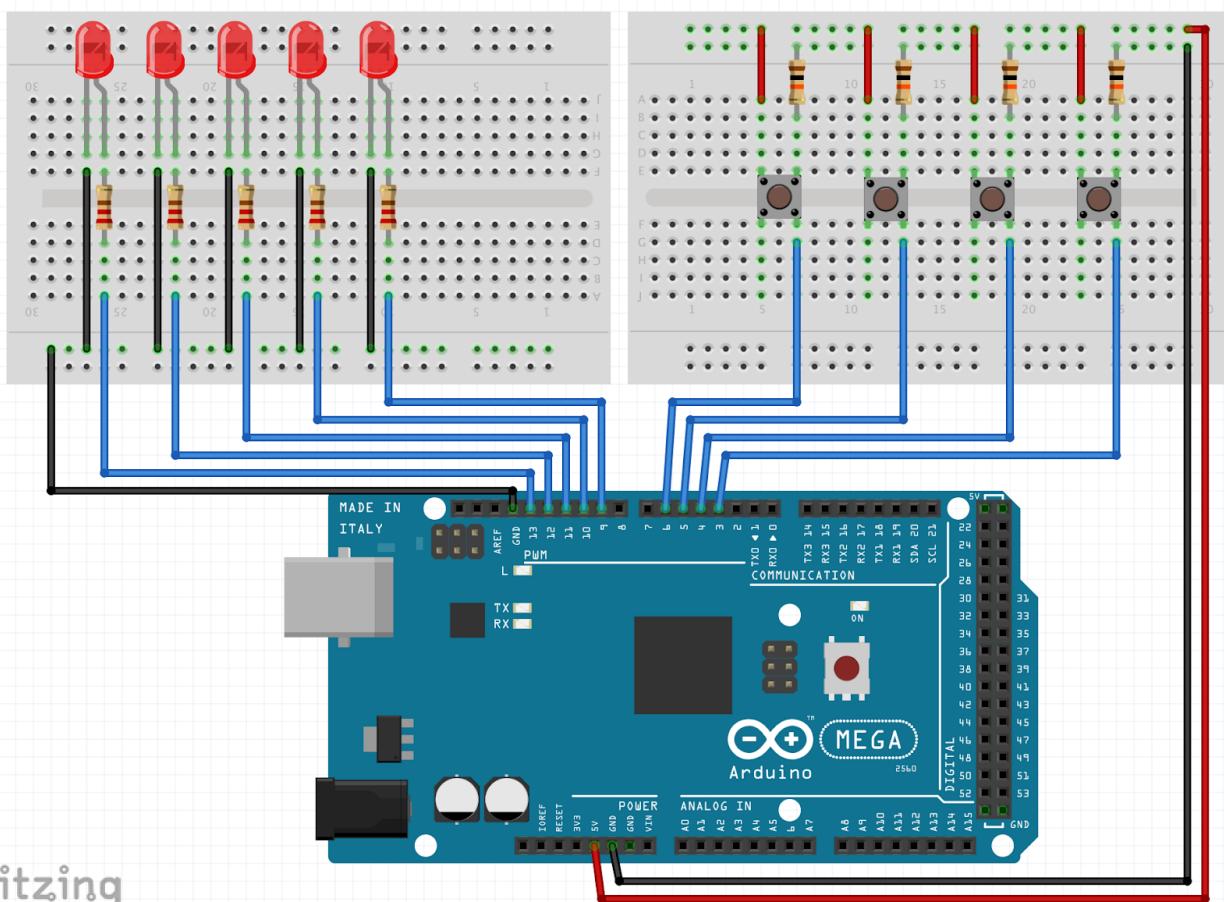
In challenge #1, we will set up a push button and LED lights on the Arduino for the whac-a-mole game. Every two times you push the button, an extra LED should illuminate. After the 10th button press, all 5 LEDs should be on. Do not tear down this circuit as it will be used in a later challenge.

This will test the user's ability to build a circuit based on the provided fritzing and circuit diagrams.

## Components

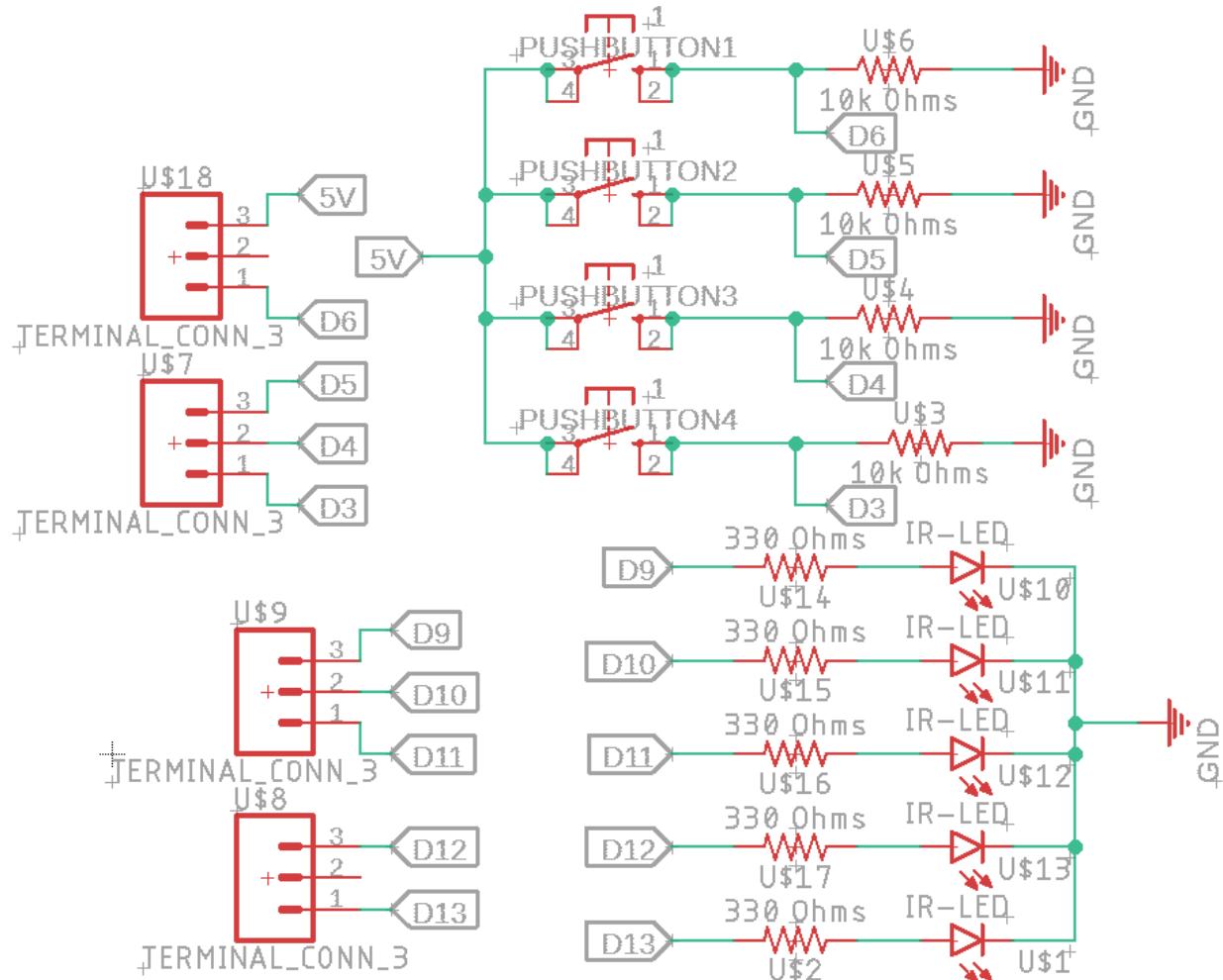
- (5) LEDs
- Arduino Mega 2560
- (2) Breadboards
- Jumper wires
- (4) Push Button
- (5) 330 Ohm Resistors
- (4) 10k Ohm Resistor

## Fritzing Diagram



**Note:** Remember to use black wires for GND and any other color for the analog ports.

### Circuit Diagram



### Resources

For more information regarding how the code works and runs, please visit:

<https://medium.com/arduino-playground/checking-for-a-button-press-in-arduino-7681cbb7bde7>

### Task

- Construct the circuit consisting of 5 LEDs and 4 push buttons
- Read through the code below and fill in the blanks. Pay attention to the comments, and make sure you understand what is going on.

- Upload the code on Arduino and test to see if the buttons work. After pressing any push button twice an LED should illuminate. Repeat this step until all 5 LEDs are turned on

## {=}{=} Code

```
//Initialize all of the pins
int buttonPin1 = _____; //Number of push button pin
int buttonPin2 = _____;
int buttonPin3 = _____;
int buttonPin4 = _____;
int led1 = _____; //Number of LED pin
int led2 = _____;
int led3 = _____;
int led4 = _____;
int led5 = _____;
//Keep track of LED state
int ledState1 = _____;
int ledState2 = _____;
int ledState3 = _____;
int ledState4 = _____;
int ledState5 = _____;
//Keep Track of how many times the buttons have been pushed
int count = 0;

void setup() {
    pinMode(_____, INPUT);
    pinMode(_____, OUTPUT);
    Serial.begin(9600);

}

void loop() {
    if(buttonPressed(buttonPin1)){
        count++;
    //We want to see if a button has been pressed
        Serial.println("Button 1 pressed!");
    //Turn on the LED
        turnOnLight();
    }
    if(buttonPressed(buttonPin2)){
        count++;
        Serial.println("Button 2 pressed!");
        turnOnLight();
    }
    if(buttonPressed(buttonPin3)){
        count++;
        Serial.println("Button 3 pressed!");
    }
}
```

```

        turnOnLight();
    }
    if(buttonPressed(buttonPin4)){
        count++;
        Serial.println("Button 4 pressed!");
        turnOnLight();
    }
    delay(100);

}

int buttonPressed(int button) {
    static uint16_t lastStates = 0;
    uint8_t state = digitalRead(____);
    if (state != ((lastStates >> button) & 1)) {
        lastStates ^= 1 << button;
        return state == ____;
    }
    return false;
}

//Check the count value so that the LEDs will light up every other button
press
bool checkCount(int c){
    if(c>0 && (c%2 == 0)){
        return ____;
    }else{
        return ____;
    }
}

//Check to see if an LED has already been turned on
bool checkLight(int light){
    int state = light;
    if(state == ____){
        return ____;
    }else{
        return ____;
    }
}

//Function to turn an LED on once any of the buttons have been pushed twice
void turnOnLight(){
    if(checkCount(count) && checkLight(ledState1)){
        digitalWrite(led1 , ____ ); //turn on the LED
        ledState1 = ____;
        Serial.println("Light 1 On!");
    }else if(checkCount(count) && checkLight(ledState2)){
        digitalWrite(led2 , ____ );
    }
}

```

```
ledState2 = ____;
Serial.println("Light 2 On!");
}else if(checkCount(count) && checkLight(ledState3)){
    digitalWrite(led3 , ____ );
    ledState3 = ____;
    Serial.println("Light 3 On!");
}else if(checkCount(count) && checkLight(ledState4)){
    digitalWrite(led4 , ____ );
    ledState4 = ____;
    Serial.println("Light 4 On!");
}else if(checkCount(count) && checkLight(ledState5)){
    digitalWrite(led5 , ____ );
    ledState5 = ____;
    Serial.println("Light 5 On!");
}
}
```

**NOTE:** WAIT! Do not take down the circuit! You will need it for the next challenge.

# Challenge #2: Piezo Buzzer

## Objective

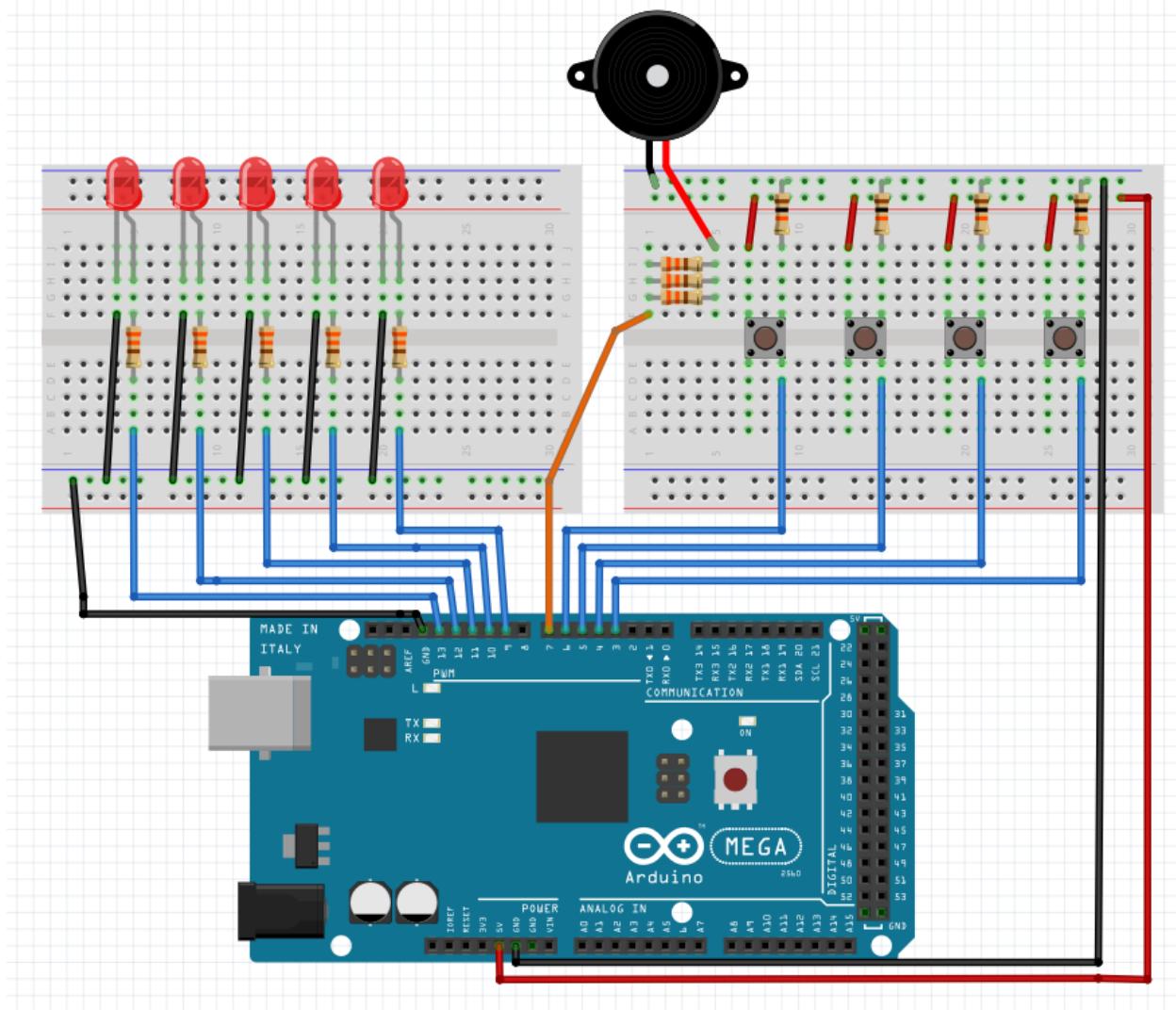
In challenge #2, we will build on the circuit built in challenge #1 to incorporate a sound capability when a button is pressed. Each button press will make a sound and when all 5 LEDs are illuminated, there will be a special tune played.

This will test the user's ability to build a circuit based on the provided fritzing and circuit diagrams.

## Components

- Piezo buzzer
- (3) 300 Ohm Resistors
- Jumper Wire

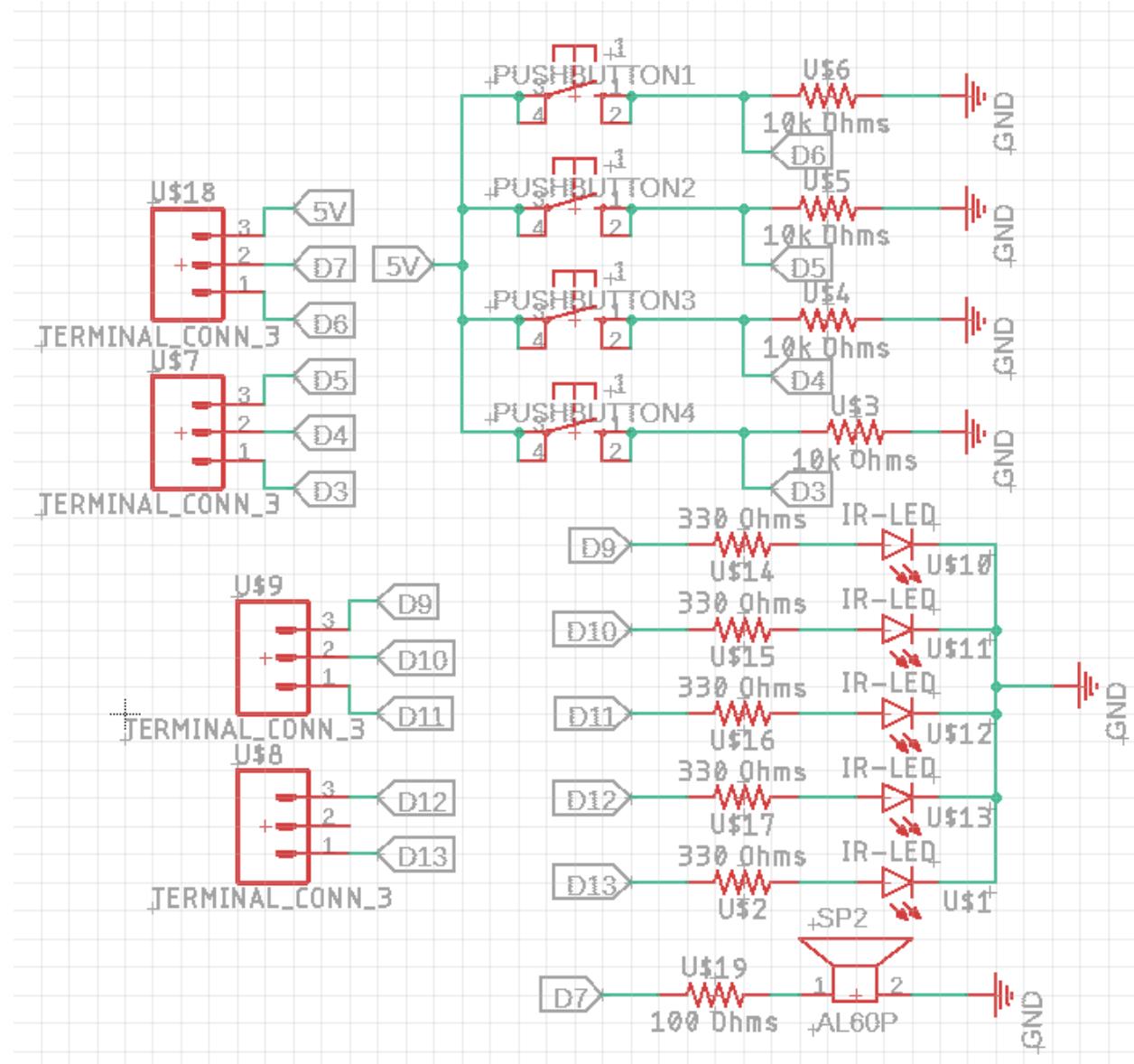
## Fritzing Diagram



**Note:** When viewing the provided links for wiring a piezo buzzer, we notice that it asks for a 100 ohm resistor which we do not have. To create a resistance close to 100 ohms with our kit's provided 330 ohm resistors we can make use of parallel circuits and its total resistance as calculated by the formula:

$$\frac{1}{R_{\text{total}}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots$$

### Circuit Diagram



## ✓ Task

- Incorporate a piezo buzzer to the circuit constructed from Challenge #1
- Read through the code below and fill in the blanks. Pay attention to the comments, and make sure you understand what is going on.
- For an extra challenge, try recreating “Mary Had a Little Lamb” from the provided sheet music and with the piezo buzzer.
- Upload the code on Arduino and test to see if the piezo buzzer produces a sound each time a button is pressed and when the LEDs are all illuminated.

### Coding Challenge

Try creating “Mary Had a Little Lamb” with the last function of the provided code! For reference, here is the sheet music:

### Mary Had a Little Lamb

Folk song

The sheet music consists of two staves of musical notation in G major (one sharp) and 4/4 time. The first staff contains three measures of the melody, each corresponding to the lyrics "Mar - y had a lit - tle lamb," repeated three times. The second staff continues the melody with the lyrics "Mar - y had a lit - tle lamb, its fleece was white as snow." The source for the sheet music is bethsnotes.com.

Feel free to experiment with other songs you would like to use and try to create special sound effects for each time you press a push button! For more information on how to code your piezo buzzer, please visit:

<https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45>

## {=} Code

```
//Initialize all of the pins
int piezoPin = ____; //Number of piezo buzzer pin
int buttonPin1 = ____; //Number of push button pin
int buttonPin2 = ____;
```

```
int buttonPin3 = [REDACTED];
int buttonPin4 = [REDACTED];
int led1 = [REDACTED]; //Number of LED pin
int led2 = [REDACTED];
int led3 = [REDACTED];
int led4 = [REDACTED];
int led5 = [REDACTED];
//Keep track of LED state
int ledState1 = [REDACTED];
int ledState2 = [REDACTED];
int ledState3 = [REDACTED];
int ledState4 = [REDACTED];
int ledState5 = [REDACTED];
//Keep Track of how many times the buttons have been pushed
int count = 0;

void setup() {
    pinMode([REDACTED], INPUT);
    pinMode([REDACTED], OUTPUT);
    Serial.begin(9600);

}

void loop() {
    if(buttonPressed(buttonPin1)){
        count++;
        //We want to see if a button has been pressed
        Serial.println("Button 1 pressed!");
        //we want to use one of the new functions at the end of our code to produce
        a sound effect each time we press the button
        [REDACTED];
    }
    //Turn on the LED
    turnOnLight();
    //we want to use the other new function that we created at the end of our
    code to play a tune once all of the LEDs are illuminated
    [REDACTED];

}
if(buttonPressed(buttonPin2)){
    count++;
    Serial.println("Button 2 pressed!");
    [REDACTED];
    turnOnLight();
    [REDACTED];

}
if(buttonPressed(buttonPin3)){
    count++;

}
```

```

Serial.println("Button 3 pressed!");
████████;
turnOnLight();
████████;

}

if(buttonPressed(buttonPin4)){
    count++;
    Serial.println("Button 4 pressed!");
    ██████████;
    turnOnLight();
    ██████████;

}

delay(100);

}

int buttonPressed(int button) {
    static uint16_t lastStates = 0;
    uint8_t state = digitalRead(████████);
    if (state != ((lastStates >> button) & 1)) {
        lastStates ^= 1 << button;
        return state == ██████████;
    }
    return false;
}

//Check the count value so that the LEDs will light up every other button
press
bool checkCount(int c){
    if(c>0 && (c%2 == 0)){
        return ██████████;
    }else{
        return ██████████;
    }
}

//Check to see if an LED has already been turned on
bool checkLight(int light){
    int state = light;
    if(state == ██████████){
        return ██████████;
    }else{
        return ██████████;
    }
}

//Function to turn an LED on once any of the buttons have been pushed twice

```

```

void turnOnLight(){
    if(checkCount(count) && checkLight(ledState1)){
        digitalWrite(led1 , _____ ); //turn on the LED
        ledState1 = _____ ;
        Serial.println("Light 1 On!");

    }else if(checkCount(count) && checkLight(ledState2)){
        digitalWrite(led2 , _____ );
        ledState2 = _____ ;
        Serial.println("Light 2 On!");

    }else if(checkCount(count) && checkLight(ledState3)){
        digitalWrite(led3 , _____ );
        ledState3 = _____ ;
        Serial.println("Light 3 On!");

    }else if(checkCount(count) && checkLight(ledState4)){
        digitalWrite(led4 , _____ );
        ledState4 = _____ ;
        Serial.println("Light 4 On!");

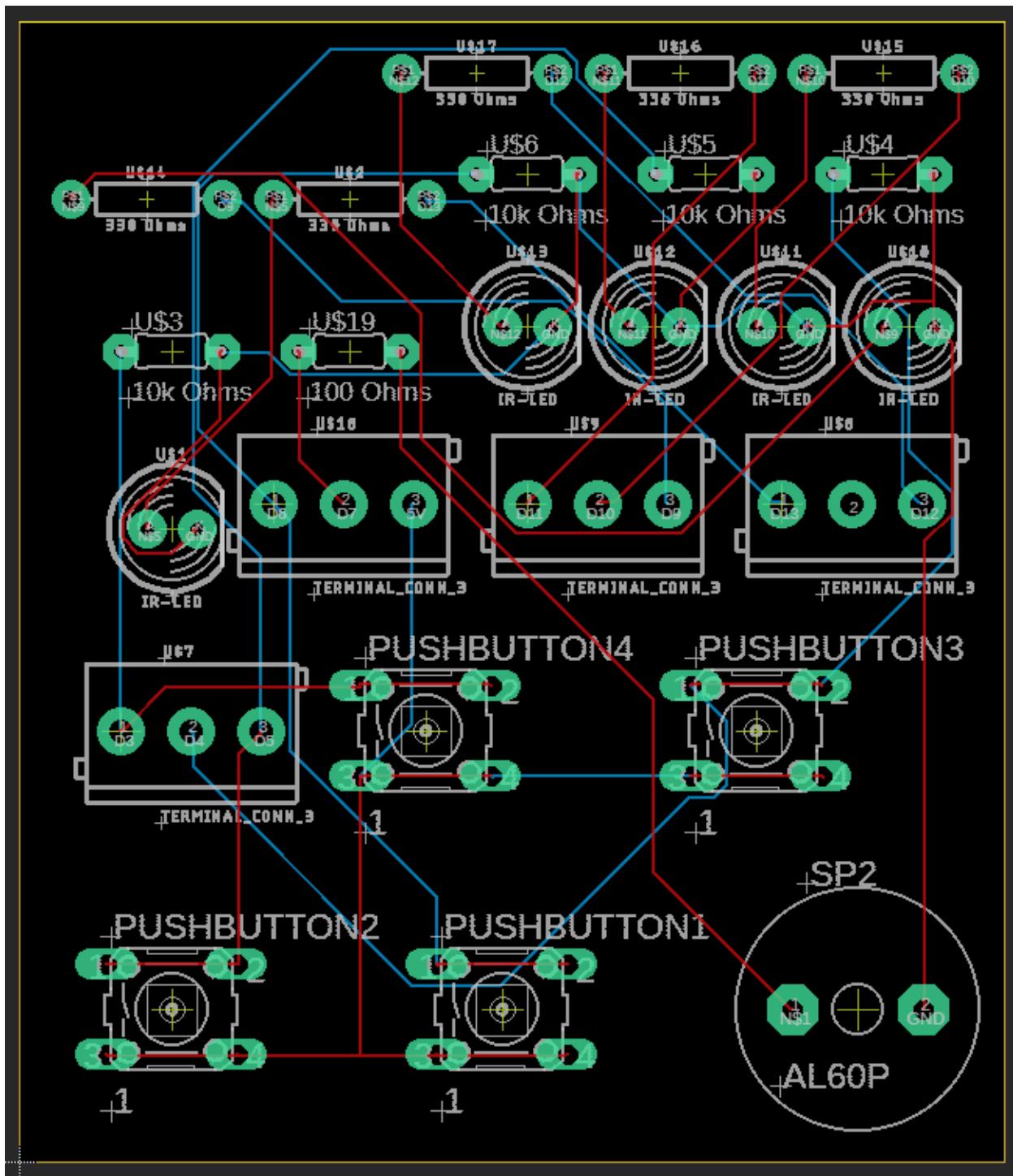
    }else if(checkCount(count) && checkLight(ledState5)){
        digitalWrite(led5 , _____ );
        ledState5 = _____ ;
        Serial.println("Light 5 On!");
    }
}

//function to play sound for when a button is pressed
void soundEffect(){
//Use the provided resources that are linked below to create a sound effect
for each button press in your game
    _____ ;
}

//this function checks that all of the LEDs are illuminated and when they
are, it will play a tune
void playtune(){
    if (count == _____ ){
        Serial.println("Congratulations you won!!!!");
    //Use the provided resources that are linked below to create a tune for
    your game.
    //For an extra challenge, try recreating the song "Mary Had a Little Lamb."
        _____ ;
    }
}

```

## Printed Board Design



The above PCB diagram is an example of what our circuit may look like if we were to print one out from a manufacturer.

## Resources

For what a piezo buzzer and how it works, please visit:

<https://www.programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/>

<https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45>

For more on calculating the total resistance in a parallel circuit, please visit:  
<https://www.goodscience.com.au/year-9-physics/electrical-resistance-and-ohms-law/5-calculating-total-resistance-in-a-circuit/>

For an explanation of how a parallel circuit works, please visit:  
<https://learn.sparkfun.com/tutorials/series-and-parallel-circuits/all>

**NOTE:** WAIT! Do not take down the circuit! You will need it for a later challenge.

# Challenge #3: MATLAB GUI

## Objective

In challenge #3, we will set up the frontend of the game on MATLAB GUI. This will consist of making the layout of the game and setting up graphics on the GUI. There will be a start button, a score counter, and graphics of the moles.

This will build user's skills in matlab specifically GUI which helps the user build their front and back end skills.

## Components

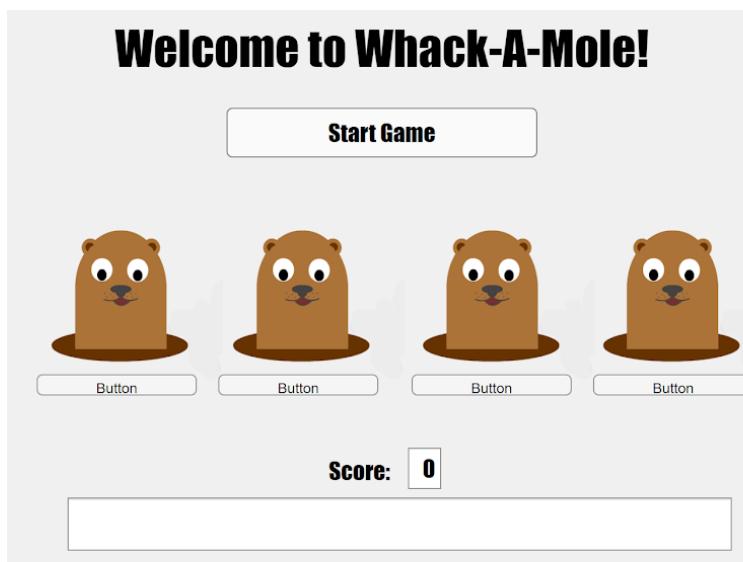
- Matlab

## Resources

Use this link to understand how to set up an app in GUI:

[https://www.mathworks.com/help/matlab/creating\\_guis/about-the-simple-guide-gui-example.html](https://www.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html)

Your job is to create the layout of the whack-a-mole app like the example shown below. It will need a title, a start button, a score counter, four graphics for the moles, and a transparent button on top of each of the graphics (The button is visible on one of the graphics to show what it's supposed to look like when seen)



## Task

- Use MATLAB GUI to create the layout of Whack-A-Mole

# Challenge #4: MATLAB GUI Whack-A-Mole Game

## Objective

In Challenge #4, we will set up the GUI user interface. We will be coding the whack-a-mole game so that it will be playable with a mouse. In this app, the user will play the reactionary game so that each time the user clicks within a reasonable time frame the score will increase by one. The objective is to get to a score of 10.

## Components

- Matlab

## Resources

Use this link to better understand how to make callbacks in MATLAB GUI:

[https://www.mathworks.com/help/matlab/creating\\_guis/write-callbacks-using-the-guide-workflow.html](https://www.mathworks.com/help/matlab/creating_guis/write-callbacks-using-the-guide-workflow.html)

For more on setting the visibility of the buttons:

<https://www.mathworks.com/matlabcentral/answers/428524-how-do-you-set-the-visibility-of-a-gui-window-to-off-through-code>

## Task

- Read through the code below and fill in the blanks. Pay attention to the comments, and make sure you understand what is going on.

## {-=} Code

```
methods (Access = private)

    function load = molesAppear(app)
        r = randi([_____],1,100); %Load random number between 1-4
        count = 1;
        while app.ScoreEditField.Value < _____ %Run until score is 10
            if r(count) == _____
                set(app.button1, 'visible', 'off');
                pause(1);
                set(app.button1, 'visible', 'on');
                drawnow;

            elseif r(count)==_____
                set(app.button2, 'visible', 'off');
                pause(1);
                set(app.button2, 'visible', 'on');
                drawnow;
```

```

        elseif r(count) == _____
            set(app.button3, 'visible', 'off');
            pause(1);
            set(app.button3, 'visible', 'on');
            drawnow;

        else
            set(app.button4, 'visible', 'off');
            pause(1);
            set(app.button4, 'visible', 'on');
            drawnow;

        end
        count = count +1;
    end
    app.EditField.Value = '_____'; %Text When Game is Won
    drawnow;

end
end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: StartGameButton
    function StartGame(app, event)
        set(app.button1, 'visible', 'on');
        set(app.button2, 'visible', 'on');
        set(app.button3, 'visible', 'on');
        set(app.button4, 'visible', 'on');
        app.ScoreEditField.Value = _____; %Set score to 0
        drawnow;
        molesAppear(app);
    end

    % Button pushed function: _____ %Put the name of your first button
    function ButtonPushed(app, event)
        if strcmp(app.button1.Visible, 'off')
            app.ScoreEditField.Value = _____; %Add to score
            drawnow;
            pause(2);
            set(app.button1, 'visible', 'on');
            drawnow;
        else
            app.ScoreEditField.Value = _____; %Subtract score
            drawnow;
        end
    end

```

```

        end
    end

    % Button pushed function: _____ %Second Button Name
    function Button_2Pushed(app, event)
        if strcmp(app.button2.Visible, 'off')
            app.ScoreEditField.Value = _____; %Add to score
            drawnow;
            pause(2);
            set(app.button2, 'visible', 'on');
            drawnow;
        else
            app.ScoreEditField.Value = _____; %Subtract score
            drawnow;
        end
    end

    % Button pushed function: _____ %Third Button Name
    function Button_3Pushed(app, event)
        if strcmp(app.button3.Visible, 'off')
            app.ScoreEditField.Value = _____; %Add to score
            drawnow;
            pause(2);
            set(app.button3, 'visible', 'on');
            drawnow;
        else
            app.ScoreEditField.Value = _____; %Subtract score
            drawnow;
        end
    end

    % Button pushed function: _____ %Fourth Button Name
    function Button_4Pushed(app, event)
        if strcmp(app.button4.Visible, 'off')
            app.ScoreEditField.Value = _____; %Add to score
            drawnow;
            pause(2);
            set(app.button4, 'visible', 'on');
            drawnow;
        else
            app.ScoreEditField.Value = _____; %Subtract score
            drawnow;
        end
    end
end

```

# Challenge #5: MATLAB GUI Whack-A-Mole Game

## Objective

Challenge #5 is where everything will be integrated together to create a working whack-a-mole game. You will need to connect the circuit you created in Challenge #1 to your MATLAB GUI created in the previous challenge. We will be using the buttons on the Arduino board to “whack” the moles with each press.

## Components

- Matlab
- Arduino Mega 2560
- Circuit Made in Challenge #2

## Resources

Links for connecting Arduino to MATLAB GUI

[https://www.mathworks.com/help/matlab/import\\_export/read-streaming-data-from-arduino.html](https://www.mathworks.com/help/matlab/import_export/read-streaming-data-from-arduino.html)

<https://www.mathworks.com/hardware-support/arduino-matlab.html>

<https://www.allaboutcircuits.com/projects/arduino-interface-with-matlab/>

<https://www.mathworks.com/videos/arduino-and-matlab-reading-inputs-and-writing-outputs-106502.html>

In this task, we will only be editing the molesAppear() function from the last challenge as well as adding a lights() function. We will be using new functions to control the arduino through matlab which include:

- readDigitalPin (this is to read the button)
- playTone (this is to play the piezo buzzer)
- writePWMSignal (this is to turn on the led)

## Task

- Read through the code below and fill in the blanks. Pay attention to the comments, and make sure you understand what is going on.
- Play the game!! And Debug any errors

## {=} Code

```
function load = molesAppear(app)
    r = randi([1 4],1,100);
    count = 1;
    a = arduino('COM3', 'Mega2560'); %Connecting MATLAB to arduino
    while app.ScoreEditField.Value < 10
        if r(count) == 1
            set(app.button1, 'visible', 'off');
            pause(1);
            num = readDigitalPin(a, "____"); %Put pin number in form
D__ for the button
            if num==1
                app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
                playTone(a, '____', 440, .5) %Pin number of piezo
                lights(app,a);
            end
            set(app.button1, 'visible', 'on');
            drawnow;

        elseif r(count)==2
            set(app.button2, 'visible', 'off');
            pause(1);
            num = readDigitalPin(a, "____"); %Pin Number of button
            if num==1
                app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
                playTone(a, '____', 440, .5) %Pin number of piezo
                lights(app,a);
            end
            set(app.button2, 'visible', 'on');
            drawnow;

        elseif r(count) == 3
            set(app.button3, 'visible', 'off');
            pause(1);
            num = readDigitalPin(a, "____"); %Pin Number of button
            if num==1
                app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
                playTone(a, '____', 440, .5) %Pin number of piezo
                lights(app,a);
            end
            set(app.button3, 'visible', 'on');
            drawnow;

        else
```

```

        set(app.button4, 'visible', 'off');
        pause(1);
        num = readDigitalPin(a, "___"); %Pin Number of button
        if num==1
            app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
            playTone(a, '___', 440, .5) %Pin number of piezo
            lights(app,a);
        end
        set(app.button4, 'visible', 'on');
        drawnow;
    end
    count = count +1;
end
app.EditField.Value = 'CONGRATS YOU WON!';
for c = 1:20
    %Celebratory Light Show
    writePWMMVoltage(a,'___',5); %Pin Number for led
    writePWMMVoltage(a,'___',0); %Pin Number for led
    end
    drawnow;

end
function results = lights(app, a)
    c = app.ScoreEditField.Value;
    if c==2
        writePWMMVoltage(a,'___',5); %Pin Number for led
    elseif c==4
        writePWMMVoltage(a,'___',5); %Pin Number for led
    elseif c==6
        writePWMMVoltage(a,'___',5); %Pin Number for led
    elseif c==8
        writePWMMVoltage(a,'___',5); %Pin Number for led
    elseif c==10
        writePWMMVoltage(a,'___',5); %Pin Number for led
    end
end
end

```

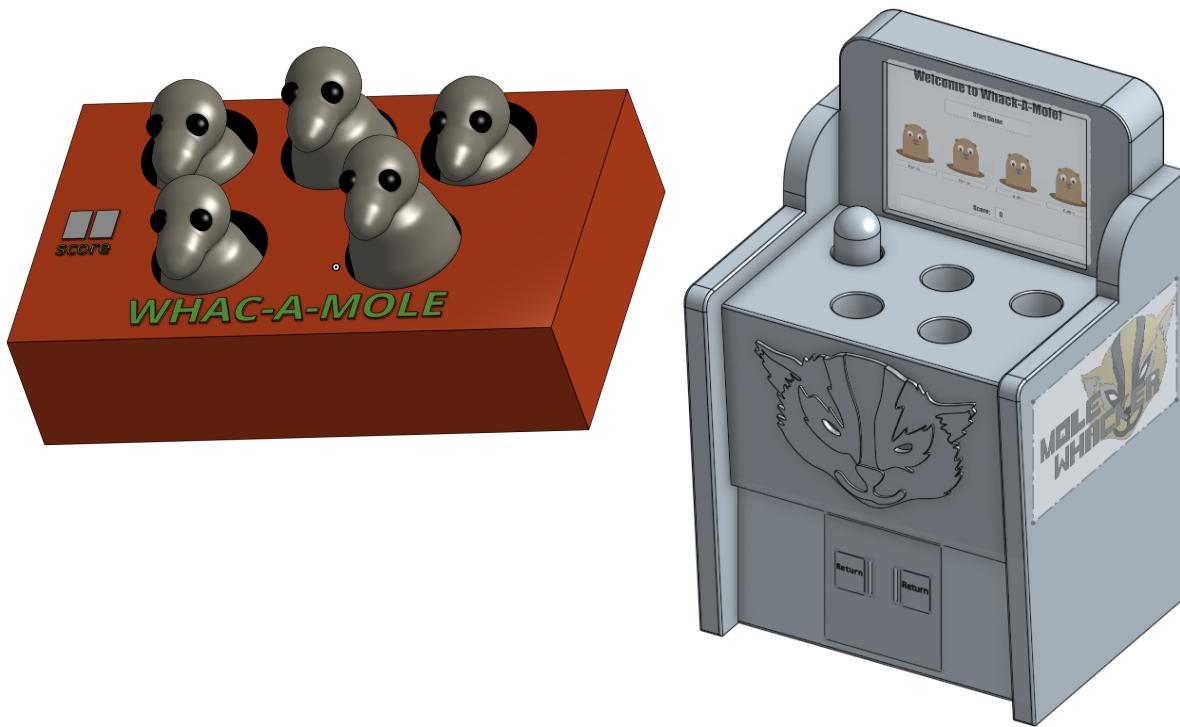
# Challenge #6: 3D Prototyping with OnShape

## Objective

If you were to play this created game at an actual arcade, how would this game look? Using 3D prototyping via OnShape, create a 3D representation of what this game would look like for you.

This challenge will test your command of OnShape and familiarize you with its assembly ability.

The provided images below serve as an example of what you will be doing and should be creating.



## Resources

For more information regarding how to use OnShape's assembly feature, please visit

<https://www.youtube.com/watch?v=YgxjGmrjDRU>

# Appendix - Resources

## Challenge #1

For more information regarding how the code works and runs, please visit:

<https://medium.com/arduino-playground/checking-for-a-button-press-in-arduino-7681cbb7bde7>

## Challenge#2

For more information regarding how a piezo buzzer makes noise, wired, and coded, please visit the following links:

<https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45>

Watching the following video may be especially helpful

<https://www.programmingelectronics.com/an-easy-way-to-make-noise-with-arduino-using-tone/>

For more on calculating the total resistance in a parallel circuit, please visit:

<https://www.goodscience.com.au/year-9-physics/electrical-resistance-and-ohms-law/5-calculating-total-resistance-in-a-circuit/>

For an explanation of how a parallel circuit works, please visit:

<https://learn.sparkfun.com/tutorials/series-and-parallel-circuits/all>

## Challenge#3

Use this link to understand how to set up an app in GUI:

[https://www.mathworks.com/help/matlab/creating\\_guis/about-the-simple-guide-gui-example.html](https://www.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html)

[https://www.mathworks.com/help/matlab/creating\\_guis/lay-out-apps-in-app-designer.html](https://www.mathworks.com/help/matlab/creating_guis/lay-out-apps-in-app-designer.html)

## Challenge#4

MATLAB useful links:

[https://www.mathworks.com/help/matlab/creating\\_guis/app-designer-startup-function.html](https://www.mathworks.com/help/matlab/creating_guis/app-designer-startup-function.html)

[https://www.mathworks.com/help/matlab/creating\\_guis/graphics-support-in-app-designer.html](https://www.mathworks.com/help/matlab/creating_guis/graphics-support-in-app-designer.html)

## Challenge#5

Links for connecting Arduino to MATLAB GUI

[https://www.mathworks.com/help/matlab/import\\_export/read-streaming-data-from-arduino.html](https://www.mathworks.com/help/matlab/import_export/read-streaming-data-from-arduino.html)

<https://www.mathworks.com/hardware-support/arduino-matlab.html>

<https://www.allaboutcircuits.com/projects/arduino-interface-with-matlab/>

<https://www.mathworks.com/videos/arduino-and-matlab-reading-inputs-and-writing-outputs-106502.html>

### **Challenge#6**

For additional help/information on how to use OnShape and its assembly feature, watch these videos:

This first video is the basics of getting started on OnShape and creating parts

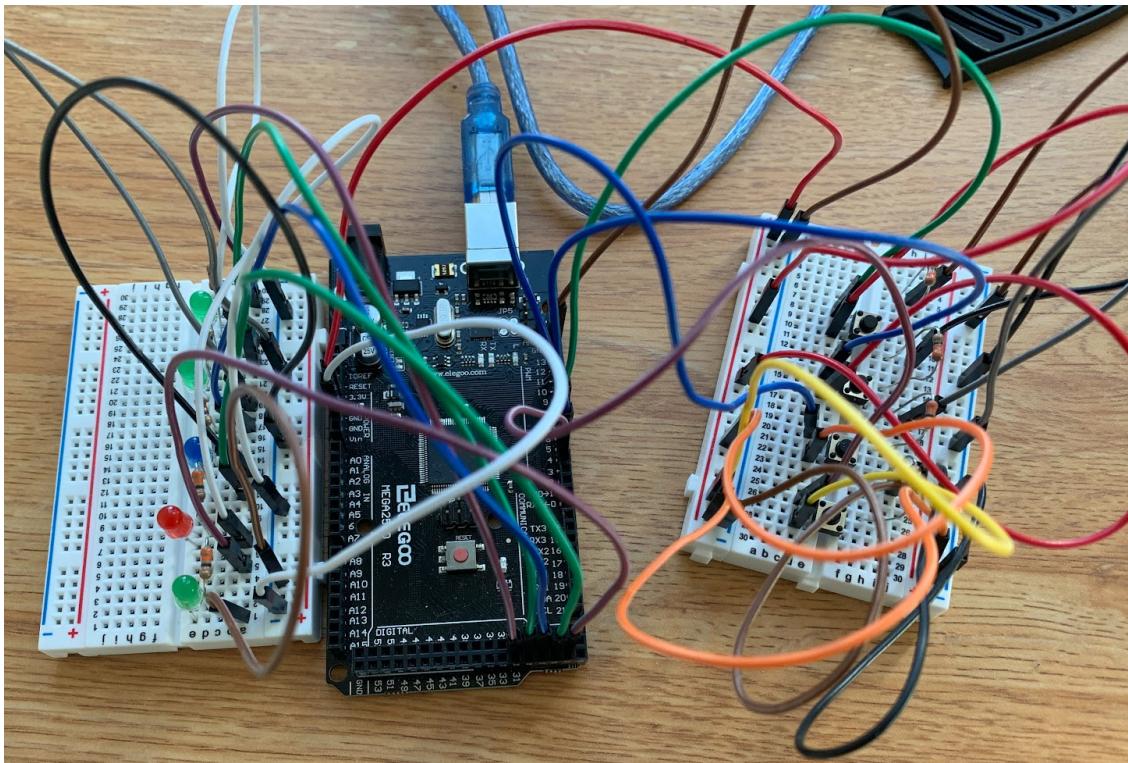
<https://www.youtube.com/watch?v=kGLZcrLkmHM>

This second video teaches how to put together the different parts created into an assembly

<https://www.youtube.com/watch?v=YgxjGmrjDRU>

# Challenge #1: Completed Circuit Diagram & Code

## Challenge #1 Completed Circuit Diagram



## {-=}{-} Challenge #1 Completed Code

```
//Initialize all of the pins
int buttonPin1 = 4; //Number of push button pin
int buttonPin2 = 5;
int buttonPin3 = 6;
int buttonPin4 = 7;
int led1 = 22; //Number of LED pin
int led2 = 24;
int led3 = 26;
int led4 = 28;
int led5 = 30;
//Keep track of LED state
int ledState1 = LOW;
int ledState2 = LOW;
int ledState3 = LOW;
int ledState4 = LOW;
int ledState5 = LOW;
//Keep Track of how many times the buttons have been pushed
int count = 0;
```

```
void setup() {
    pinMode(buttonPin1, INPUT);
    pinMode(led1, OUTPUT);
    Serial.begin(9600);

}

void loop() {
    if(buttonPressed(buttonPin1)){
        count++;
        //We want to see if a button has been pressed
        Serial.println("Button 1 pressed!");
        //Turn on the LED
        turnOnLight();
    }
    if(buttonPressed(buttonPin2)){
        count++;
        Serial.println("Button 2 pressed!");
        turnOnLight();
    }
    if(buttonPressed(buttonPin3)){
        count++;
        Serial.println("Button 3 pressed!");
        turnOnLight();
    }
    if(buttonPressed(buttonPin4)){
        count++;
        Serial.println("Button 4 pressed!");
        turnOnLight();
    }
    delay(100);

}
int buttonPressed(int button) {
    static uint16_t lastStates = 0;
    uint8_t state = digitalRead(button);
    if (state != ((lastStates >> button) & 1)) {
        lastStates ^= 1 << button;
        return state == HIGH;
    }
    return false;
}

//Check the count value so that the LEDs will light up every other button
press
bool checkCount(int c){
    if(c>0 && (c%2 == 0)){
        return true;
```

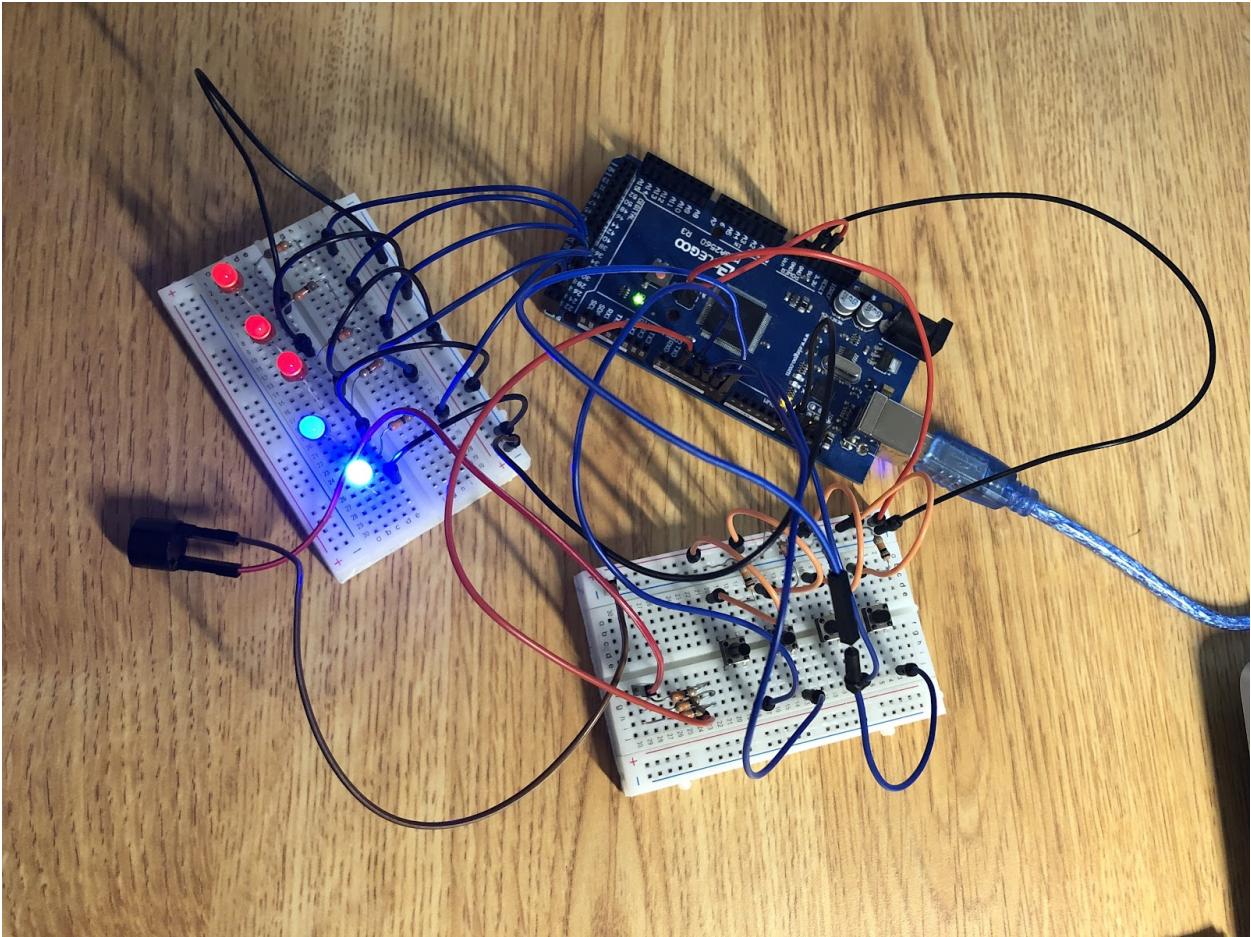
```
        }else{
            return false;
        }
    }

//Check to see if an LED has already been turned on
bool checkLight(int light){
    int state = light;
    if(state == HIGH){
        return false;
    }else{
        return true;
    }
}

//Function to turn an LED on once any of the buttons have been pushed twice
void turnOnLight(){
    if(checkCount(count) && checkLight(ledState1)){
        digitalWrite(led1 , HIGH ); //turn on the LED
        ledState1 = HIGH;
        Serial.println("Light 1 On!");
    }else if(checkCount(count) && checkLight(ledState2)){
        digitalWrite(led2 , HIGH );
        ledState2 = HIGH;
        Serial.println("Light 2 On!");
    }else if(checkCount(count) && checkLight(ledState3)){
        digitalWrite(led3 , HIGH );
        ledState3 = HIGH;
        Serial.println("Light 3 On!");
    }else if(checkCount(count) && checkLight(ledState4)){
        digitalWrite(led4 , HIGH );
        ledState4 = HIGH;
        Serial.println("Light 4 On!");
    }else if(checkCount(count) && checkLight(ledState5)){
        digitalWrite(led5 , HIGH );
        ledState5 = HIGH;
        Serial.println("Light 5 On!");
    }
}
```

## Challenge #2: Completed Circuit Diagram & Code

### Challenge #2 Completed Circuit Diagram



### {=}{=} Challenge #2 Completed Code

```
//Initialize all of the pins
int piezoPin = 2; //Number of piezo buzzer pin
int buttonPin1 = 4; //Number of push button pin
int buttonPin2 = 5;
int buttonPin3 = 6;
int buttonPin4 = 7;
int led1 = 22; //Number of LED pin
int led2 = 24;
int led3 = 26;
int led4 = 28;
int led5 = 30;
//Keep track of LED state
int ledState1 = LOW;
int ledState2 = LOW;
```

```
int ledState3 = LOW;
int ledState4 = LOW;
int ledState5 = LOW;
//Keep Track of how many times the buttons have been pushed
int count = 0;

void setup() {
    pinMode(buttonPin1, INPUT);
    pinMode(led1, OUTPUT);
    Serial.begin(9600);

}

void loop() {
    if(buttonPressed(buttonPin1)){
        count++;
        //We want to see if a button has been pressed
        Serial.println("Button 1 pressed!");
        //we want to use one of the new functions at the end of our code to produce
        a sound effect each time we press the button
        soundEffect();
        //Turn on the LED
        turnOnLight();
        //we want to use the other new function that we created at the end of our
        code to play a tune once all of the LEDs are illuminated
        playtune();

    }
    if(buttonPressed(buttonPin2)){
        count++;
        Serial.println("Button 2 pressed!");
        soundEffect();
        turnOnLight();
        playtune();

    }
    if(buttonPressed(buttonPin3)){
        count++;
        Serial.println("Button 3 pressed!");
        soundEffect();
        turnOnLight();
        playtune();

    }
    if(buttonPressed(buttonPin4)){
        count++;

    }
}
```

```

    Serial.println("Button 4 pressed!");
    soundEffect();
    turnOnLight();
    playtune();

}

delay(100);

}

int buttonPressed(int button) {
    static uint16_t lastStates = 0;
    uint8_t state = digitalRead(button);
    if (state != ((lastStates >> button) & 1)) {
        lastStates ^= 1 << button;
        return state == HIGH;
    }
    return false;
}

//Check the count value so that the LEDs will light up every other button
press
bool checkCount(int c){
    if(c>0 && (c%2 == 0)){
        return true;
    }else{
        return false;
    }
}

//Check to see if an LED has already been turned on
bool checkLight(int light){
    int state = light;
    if(state == HIGH){
        return false;
    }else{
        return true;
    }
}

//Function to turn an LED on once any of the buttons have been pushed twice
void turnOnLight(){
    if(checkCount(count) && checkLight(ledState1)){
        digitalWrite(led1 , HIGH ); //turn on the LED
        ledState1 = HIGH;
        Serial.println("Light 1 On!");

    }else if(checkCount(count) && checkLight(ledState2)){

```

```
digitalWrite(led2 , HIGH );
ledState2 = HIGH;
Serial.println("Light 2 On!");

}else if(checkCount(count) && checkLight(ledState3)){
  digitalWrite(led3 , HIGH );
  ledState3 = HIGH;
  Serial.println("Light 3 On!");

}else if(checkCount(count) && checkLight(ledState4)){
  digitalWrite(led4 , HIGH );
  ledState4 = HIGH;
  Serial.println("Light 4 On!");

}else if(checkCount(count) && checkLight(ledState5)){
  digitalWrite(led5 , HIGH );
  ledState5 = HIGH;
  Serial.println("Light 5 On!");
}

//function to play sound for when a button is pressed
void soundEffect(){
//Use the provided resources that are linked below to create a sound effect
for each button press in your game
  tone(piezoPin, 90,800);
  delay(20);
  tone(piezoPin, 300,500);
  delay(20);
  tone(piezoPin, 800,400);
}

//this function checks that all of the LEDs are illuminated and when they
are, it will play a tune
void playtune(){
  if (count == 10){
    Serial.println("Congratulations you won!!!");
//Use the provided resources that are linked below to create a tune for
your game.
//For an extra challenge, try recreating the song "Mary Had a Little Lamb."
    tone(piezoPin, 900,500);
    delay(600);
    tone(piezoPin, 800,500);
    delay(600);
    tone(piezoPin, 700, 500);
    delay(600);
    tone(piezoPin, 800,500);
    delay(600);
```

```
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 900,700);
delay(600);
tone(piezoPin, 800,500);
delay(600);
tone(piezoPin, 800,500);
delay(600);
tone(piezoPin, 800,700);
delay(600);
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 1000,500);
delay(600);
tone(piezoPin, 1000,700);
delay(600);
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 800,500);
delay(600);
tone(piezoPin, 700, 500);
delay(600);
tone(piezoPin, 800,500);
delay(600);
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 800,500);
delay(600);
tone(piezoPin, 800,500);
delay(600);
tone(piezoPin, 900,500);
delay(600);
tone(piezoPin, 800,500);
delay(600);
tone(piezoPin, 700, 900);
delay(600);
}
```

## Challenge #4: Completed Code

```
methods (Access = private)

    function load = molesAppear(app)
        r = randi([1 4],1,100);
        count = 1;
        while app.ScoreEditField.Value < 10
            if r(count) == 1
                set(app.button1, 'visible', 'off');
                pause(1);
                set(app.button1, 'visible', 'on');
                drawnow;

            elseif r(count)==2
                set(app.button2, 'visible', 'off');
                pause(1);
                set(app.button2, 'visible', 'on');
                drawnow;

            elseif r(count) == 3
                set(app.button3, 'visible', 'off');
                pause(1);
                set(app.button3, 'visible', 'on');
                drawnow;

            else
                set(app.button4, 'visible', 'off');
                pause(1);
                set(app.button4, 'visible', 'on');
                drawnow;

            end
            count = count +1;
        end
        app.EditField.Value = 'CONGRATS YOU WON!';
        drawnow;

    end
end

% Callbacks that handle component events
methods (Access = private)
```

```

% Button pushed function: StartGameButton
function StartGame(app, event)
    set(app.button1, 'visible', 'on');
    set(app.button2, 'visible', 'on');
    set(app.button3, 'visible', 'on');
    set(app.button4, 'visible', 'on');
    app.ScoreEditField.Value = 0;
    drawnow;
    molesAppear(app);
end

% Button pushed function: Button
function ButtonPushed(app, event)
    if strcmp(app.button1.Visible, 'off')
        app.ScoreEditField.Value = app.ScoreEditField.Value + 1;
        drawnow;
        pause(2);
        set(app.button1, 'visible', 'on');
        drawnow;
    else
        app.ScoreEditField.Value = app.ScoreEditField.Value - 1;
        drawnow;
    end
end

% Button pushed function: Button_2
function Button_2Pushed(app, event)
    if strcmp(app.button2.Visible, 'off')
        app.ScoreEditField.Value = app.ScoreEditField.Value + 1;
        drawnow;
        pause(2);
        set(app.button2, 'visible', 'on');
        drawnow;
    else
        app.ScoreEditField.Value = app.ScoreEditField.Value - 1;
        drawnow;
    end
end

% Button pushed function: Button_3
function Button_3Pushed(app, event)
    if strcmp(app.button3.Visible, 'off')
        app.ScoreEditField.Value = app.ScoreEditField.Value + 1;
        drawnow;
        pause(2);
        set(app.button3, 'visible', 'on');
        drawnow;
    else

```

```
    app.ScoreEditField.Value = app.ScoreEditField.Value - 1;
    drawnow;
end

% Button pushed function: Button_4
function Button_4Pushed(app, event)
    if strcmp(app.button4.Visible, 'off')
        app.ScoreEditField.Value = app.ScoreEditField.Value + 1;
        drawnow;
        pause(2);
        set(app.button4, 'visible', 'on');
        drawnow;
    else
        app.ScoreEditField.Value = app.ScoreEditField.Value - 1;
        drawnow;
    end
end
end
```

## Challenge #5: Completed Code

```
function load = molesAppear(app)
    r = randi([1 4],1,100);
    count = 1;
    a = arduino('COM3', 'Mega2560');
    while app.ScoreEditField.Value < 10
        if r(count) == 1
            set(app.button1, 'visible', 'off');
            pause(1);
            num = readDigitalPin(a, "D4");
            if num==1
                app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
                playTone(a, 'D53', 440, .5)
                lights(app,a);
            end
            set(app.button1, 'visible', 'on');
            drawnow;

        elseif r(count)==2
            set(app.button2, 'visible', 'off');
            pause(1);
            num = readDigitalPin(a, "D5");
            if num==1
                app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
                playTone(a, 'D53', 440, .5)
                lights(app,a);
            end
            set(app.button2, 'visible', 'on');
            drawnow;

        elseif r(count) == 3
            set(app.button3, 'visible', 'off');
            pause(1);
            num = readDigitalPin(a, "D6");
            if num==1
                app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
                playTone(a, 'D53', 440, .5)
                lights(app,a);
            end
            set(app.button3, 'visible', 'on');
            drawnow;
```

```

        else
            set(app.button4, 'visible', 'off');
            pause(1);
            num = readDigitalPin(a, "D7");
            if num==1
                app.ScoreEditField.Value =
app.ScoreEditField.Value + 1;
                playTone(a, 'D53', 440, .5)
                lights(app,a);
            end
            set(app.button4, 'visible', 'on');
            drawnow;

        end
        count = count +1;
    end
    app.EditField.Value = 'CONGRATS YOU WON!';
    for c = 1:20
        writePWMMVoltage(a,'D9',5);
        writePWMMVoltage(a,'D10',5);
        writePWMMVoltage(a,'D11',5);
        writePWMMVoltage(a,'D12',5);
        writePWMMVoltage(a,'D13',5);
        writePWMMVoltage(a,'D9',0);
        writePWMMVoltage(a,'D10',0);
        writePWMMVoltage(a,'D11',0);
        writePWMMVoltage(a,'D12',0);
        writePWMMVoltage(a,'D13',0);
    end

    drawnow;

end

function results = lights(app, a)
    c = app.ScoreEditField.Value;
    if c==2
        writePWMMVoltage(a,'D9',5);
    elseif c==4
        writePWMMVoltage(a,'D10',5);
    elseif c==6
        writePWMMVoltage(a,'D11',5);
    elseif c==8
        writePWMMVoltage(a,'D12',5);
    elseif c==10
        writePWMMVoltage(a,'D13',5);
    end
end

```

end