SIK Circuit 4C: DIY Who Am I – Lab Report

# Introduction

This report describes building the second circuit in the SparkFun Inventors Kit (SIK) [1]. The circuit is ….

# Materials

1 x Sparkfun Inventors Kit (SIK) [1]

1 x Windows/Mac/Linux computer with Arduino IDE installed [2]

1 x USB cable

1 x …

# Design

The aim of this circuit is ….

The circuit diagram, is show below:

…

# Procedure

The step-by-step procedure for building the circuit is described in the SIK guide booklet and online [3]. For this build, I used the online guide.

1. **Gather the required components**

I collected the Arduino board, the …

1. **Assemble the circuit**

I assembled the circuit according to the diagram using the Hookup Table provided..

1. **Enter the source code & upload**

I typed in the source code for the sketch. I ….

1. **Observe**

At this point the ….

# Source Code

All source code and images for this project are hosted on GitHub [4].

**Sketch-Circuit-4C-DIY-Who-am-i.ino** - online [4] and full source listed in Appendix A.

# Discussion

* I did …:
* I haven’t e….

# Conclusion

The …. I’m ready to move on to the next circuit.

# References

[1] <https://www.sparkfun.com/products/15267>

[2] <https://www.arduino.cc/en/software>

[3] [https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v41/](https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v41)

[4] <https://github.com/shanecastle/comp444>

# Appendix A – Circuit-4C-DIY-Who-am-I.ino

/\* COMP444 - Circuit 4C - Heads Up Game

This is a DIY version of the popular Heads Up party game. To play, one person resets the RedBoard and holds the LCD

facing away from them so that they cannot see it (usually on their forehead). The display will show a short countdown

then display random words. The other player(s) who can see the screen must yell out clues until time runs out or the player

guesses what word is on the screen. If they guess correctly, they can press the button on the breadboard and another word

will be displayed.

This sketch was written by SparkFun Electronics, with lots of help from the Arduino community.

This code is completely free for any use.

\*/

#include <LiquidCrystal.h> //the liquid crystal library contains commands for printing to the display

LiquidCrystal lcd(13, 12, 11, 10, 9, 8); // tell the RedBoard what pins are connected to the display

int buttonPin = 2; // pin that the button is connected to

int buzzerPin = 6; // pin for driving the buzzer

int buttonPressTime = 0; // variable to show how much time the player has left to guess the word (and press the button)

long timeLimit = 15000; // time limit for the player to guess each word

long startTime = 0; // used to measure time that has passed for each word

int roundNumber = 0; // keeps track of the roundNumber so that it can be displayed at the end of the game

const int arraySize = 25;

const char \*words[arraySize] = {"moose", "beaver", "bear", "goose", "dog", "cat", "squirrel", "bird", "elephant", "horse",

"bull", "giraffe", "seal", "bat", "skunk", "turtle", "whale", "rhino", "lion", "monkey",

"frog", "alligator", "kangaroo", "hippo", "rabbit"};

// the start value in the sequence array must have a value that could never be an index of an array

// or at least a value outside the range of 0 to the size of the words array - 1; in this case, it can't be between 0 to 24

int sequence[] = {-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}; // start with an array full of -1's

void setup()

{

pinMode(buttonPin, INPUT\_PULLUP); // set the button pin as an input

lcd.begin(16, 2); // tell the LCD library the size of the screen

generateRandomOrder(); // generate an array of random numbers from 0 to 24 that will determine which order the words are shown in

showStartSequence(); // print the start sequence text

}

void loop()

{

for (int i = 0; i < arraySize; i++)

{ // for each of the 25 words in the sequence

lcd.clear(); // clear off the array

roundNumber = i + 1; // the array starts at 0, but the roundNumber will start counting from 1

lcd.print(roundNumber); // print the roundNumber (this is the current round number)

lcd.print(": "); // spacer between the number and the word

lcd.print(words[sequence[i]]); // print a random word from the word array

startTime = millis(); // record the time that this round started

while (digitalRead(buttonPin) == HIGH)

{ // do this until the button is pressed...

int roundedTime = round((timeLimit - (millis() - startTime)) / 1000); // calculate the time left in the round (dividing by 1000 converts the number to seconds

lcd.setCursor(14, 1); // set the cursor in the lower right corner of the screen

lcd.print(" ");

lcd.setCursor(14, 1); // set the cursor in the lower right corner of the screen

lcd.print(roundedTime); // print the time left in the time limit

delay(15);

if (millis() - startTime > timeLimit)

{ // if the time limit is up before the button is pressed

gameOver(); // end the game

}

if (digitalRead(buttonPin) == LOW)

{

tone(buzzerPin, 272, 10); // emit a short beep when the button is pressed

}

} // exit this loop when the button is pressed

delay(500); // delay before going to next round, so the button press doesn't get registered twice

}

// if you finish all 25 words

winner(); // show the you win message

}

//--------------FUNCTIONS------------------------------

// DISPLAYS A COUNTDOWN TO START THE GAME

void showStartSequence()

{

lcd.clear(); // clear the screen

lcd.setCursor(0, 0); // move the cursor to the top left corner

lcd.print("Category:"); // print "Category:"

lcd.setCursor(0, 1); // move the cursor to the bottom left corner

lcd.print("Animals"); // print "Animals:"

delay(2000); // Wait 2 seconds

lcd.clear(); // clear the screen

lcd.print("Get ready!"); // print "Get ready!"

delay(1000); // wait 1 second

lcd.clear(); // clear the screen

lcd.print("3"); // print "3"

delay(1000); // wait 1 second

lcd.clear(); // clear the screen

lcd.print("2"); // print "3"

delay(1000); // wait 1 second

lcd.clear(); // clear the screen

lcd.print("1"); // print "3"

delay(1000); // wait 1 second

}

// GENERATES A RANDOM ORDER FOR THE WORDS TO BE DISPLAYED

void generateRandomOrder()

{

randomSeed(analogRead(0)); // reset the random seed (Arduino needs this to generate truly random numbers

for (int i = 0; i < arraySize; i++)

{ // do this until all 25 positions are filled

int currentNumber = 0; // variable to hold the current number

boolean match = false; // does the currentNumber match any of the previous numbers?

// generate random numbers until you've generated one that doesn't match any of the other numbers in the array

do

{

currentNumber = random(0, arraySize); // generate a random number from 0 to 24

match = false; // we haven't checked for matches yet, so start by assuming that it doesn't match

for (int i = 0; i < arraySize; i++)

{ // for all 25 numbers in the array

if (currentNumber == sequence[i])

{ // does the currentNumber match any of the numbers?

match = true; // if so, set the match variable to true

}

}

} while (match == true); // if the match variable is true, generate another random number and try again

sequence[i] = currentNumber; // if the match variable is false (the new number is unique) then add it to the sequence

}

}

// GAME OVER

void gameOver()

{

lcd.clear(); // clear the screen

lcd.setCursor(0, 0); // move the cursor the top left corner

lcd.print("Game Over"); // print "Game Over"

lcd.setCursor(0, 1); // move to the bottom row

lcd.print("Score: "); // print a label for the score

lcd.print(roundNumber - 1); // print the score (the score is equal to the previous level/round number)

// play the losing fog horn

tone(buzzerPin, 130, 250); // E6

delay(275);

tone(buzzerPin, 73, 250); // G6

delay(275);

tone(buzzerPin, 65, 150); // E7

delay(175);

tone(buzzerPin, 98, 500); // C7

delay(500);

while (true)

{

} // get stuck in this loop forever

}

// WINNER

void winner()

{

lcd.clear(); // clear the screen

lcd.setCursor(7, 0); // move the cursor to the top center of the screen

lcd.print("YOU"); // print "You"

lcd.setCursor(7, 1); // move the cursor to the bottom center of the screen

lcd.print("WIN!"); // print "WIN!"

// play the 1Up noise

tone(buzzerPin, 1318, 150); // E6

delay(175);

tone(buzzerPin, 1567, 150); // G6

delay(175);

tone(buzzerPin, 2637, 150); // E7

delay(175);

tone(buzzerPin, 2093, 150); // C7

delay(175);

tone(buzzerPin, 2349, 150); // D7

delay(175);

tone(buzzerPin, 3135, 500); // G7

delay(500);

while (true)

{

} // get stuck in this loop forever

}