ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #36019

# The Morse Code Game: Morse in a Minute

**Heather Morrell**
**Aaron Muldrew, Northeastern University**

I am a second year student at Northeastern University pursuing a Bachelor's of Science in Mechanical Engineering and Physics. I have work experience in a local machine shop and have developed many basic machining skills. From my education at Northeastern, I have developed CAD skills as well as many design skills and research techniques. I am interested in working in the aerospace industry after completing my degree.

**Prof. Nathan E Israeloff, Northeastern University**

Dept of Physics

**Prof. Don Heiman**

Don Heiman, PhD, Department of Physics, Northeastern University, Boston, MA 02115 email: heiman@neu.edu; http://northeastern.edu/heiman/research/index.html

**Prof. Haridas Kumarakuru, Northeastern University**

Haridas Kumarakuru, PhD, MInstP. Assistant Teaching Professor, Department of Physics, College of Science, Northeastern University, Boston, MA 02115 E.Mail: h.kumarakuru@northeastern.edu

# The Morse Code Game: Morse in a Minute

**Heather Morrell[1], Aaron Muldrew[1,2], Nathan Israeloff[1], Don Heiman[1] and Haridas Kumarakuru[1*]**

[1]Department of Physics, Northeastern University, Boston, MA 02115
[2]Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115

[*]*Corresponding author*: h.kumarakuru@northeastern.edu

## Abstract

"Morse in a Minute" is a simple Morse Code translation game for players of all levels. The two major components of the game, the decoder and the 4-bit binary counter, work to give the contestants a fun challenge. The decoder translates the morse code written by the contestants, with the dot, dash, and slash buttons, and displays the translated alphabetic code on an LCD screen. The counter displays the time passed to ensure that all contestants have the same amount of time to write their selected word or phrase. This device was built to fulfill an Electronics Course requirement assigned as "project-based learning."

## 1. Introduction

During our undergraduate lab course, "Electronics for Scientists," in fall 2021, we were asked to design and build a prototype device as a part of "project-based learning" that accomplished a specific purpose. During the semester, we were inspired with a variety of electronic lab-based experiments, trouble shooting and data analysis. We always enjoyed playing games in our childhood and now that electronics is such an important part of our lives, we decided to design a simple electronic game to entertain people in small gatherings.

We designed a game based on "Morse Code" [1], which is named to honor one of the inventors of telegraph, Samuel Morse, born in 1791 in Charlestown Massachusetts, a few miles from the lab in which this game was designed. Morse Code is a code for translating letters to dots, "•", which represent short signal duration, and dashes, "—", which represent long signal duration. One objective behind making this game was that it would be a game for contestants with varying experience with morse code. Thus, a time-independent game was created such that beginners would not need to learn how to make the relative lengths of dots and dashes. Instead, the circuit has 3 buttons, for dot, dash, and slash, where a slash represents a separation between letters and two slashes represents a separation between words.

## 2. Theory and Experimental Design

There are two main components to the electronics: the decoder and the counter. The decoder translates the morse code "written" by the contestants using the three buttons and displays the translated alphabetic code on an LCD screen. Meanwhile, the counter will count and display the time passed to ensure that all contestants have the same amount of time to write their word or

phrase. As the contestants use the dot, dash, and slash buttons to write their selected word or phrase, the decoder uses an Arduino connected to a computer to perform the morse code to alphabetic translation. There is a computer code to dictate how the Arduino interacts with the circuit. This code is responsible for translating the dots and dashes to letters and outputting these letters on the LCD screen. The code uses "if-else" statements to associate each letter with an ASCII number and displays each letter after the buttons are pressed. A variable called "letter" is used to store the values of the buttons pressed and the appropriate letter is printed onto the LCD screen when the user is finished typing each letter.

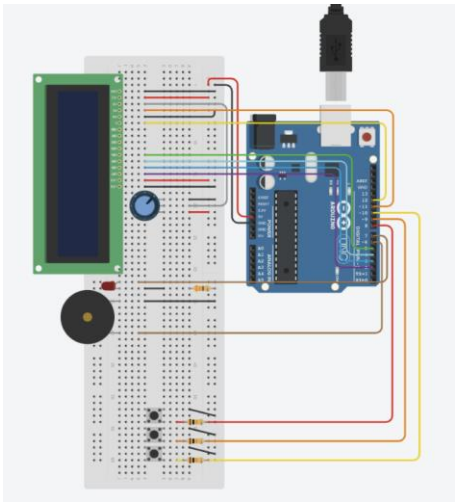

**Fig. 1:** *Diagram of decoder, showing the 3 buttons and associated LEDs, piezo, LED alpha-numeric display and Arduino.*

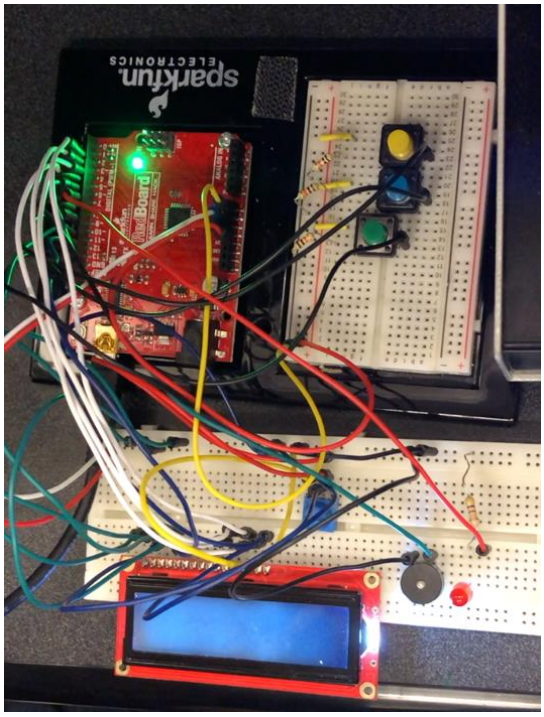

**Fig. 2:** *Photo of the decoder installed in the device.*

When the slash button is pressed, the code will recognize this is the end of a letter and the letter will show up on the LCD screen, the brightness of which will be controlled by the potentiometer. For additional amusement, an LED and piezo were added into the circuit. When the "dot" button is pushed, a short buzz will sound on the piezo and the LED will light up for a short amount of time, inspired by the typical time-dependent morse code. When a "dash" button is pushed, a long buzz will sound on the piezo and the LED will light up for a longer amount of time. A circuit diagram of the decoder can be seen in Figure 1 and a photo of the decoder prototype is shown in Figure 2. The code is written in C++ and uses standard Arduino commands. The code uses if-else statements to perform the translation and uses variables to store the inputted morse code messages. The Arduino allows the electronics to interact with the code, creating a physical user interface typical of Arduino designs [2]. A sample of the code, including the input storage system and the translation process is shown Figure 3.

```
if (digitalRead(dot)==HIGH)
{
  letter=10*letter;
  letter=(letter+1);
  tone(6,200,200);
  digitalWrite(7,HIGH);
  delay(200);
  digitalWrite(7,LOW);
}
if (digitalRead(dash)==HIGH)
{
  letter=10*letter;
  letter=(letter+2);
  tone(6,200,500);
  digitalWrite(7,HIGH);
  delay(500);
  digitalWrite(7,LOW);
}
if (digitalRead(slash)==HIGH)
{
  if (letter==12)
  {lcd.print("A");}

  if (letter==2111)
  {lcd.print("B");}

  if (letter==2121)
  {lcd.print("C");}
```

**Fig. 3:** *A sample of the code, including the input storage system and the translation process.*

| Decimal | Binary |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| 10 | 1010 |
| 11 | 1011 |
| 12 | 1100 |
| 13 | 1101 |
| 14 | 1110 |
| 15 | 1111 |

**Fig. 4**. *Four-bit decimal binary conversion.*

| Input | Input | Output |
|---|---|---|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Fig 5.** *Dual input NAND gate's Truth Table.*

While the decoder is working to translate the morse code written by the contestants, there is a 4-bit binary counter cycling through the binary numbers $0000_2$ to $1001_2$, or decimal numbers 0 to 9, to keep track of time passed, like a stopwatch. (See Figure 4 for the decimal -binary conversion.) The counter uses two dual input JK flip-flops (SN74HC112) where each flip-flop's output corresponds to one of the 4 bits. The JK flip-flops were connected in series and the inputs were set to "1", or high, in this case, 5 volts. A common function generator, with a TTL output was used to deliver clock pulses [3].

To display the time passed to the contestants, a 7 segment LED display was connected to the circuit. However, a decoder (CD4511) was needed to translate the binary digits from the counter to the decimal numbers displayed. The decoder was added to the circuit, connected to power, and the 4 outputs of each of the JK flip-flops of the binary counter circuit were connected to the inputs

of the decoder. Then, the outputs of the decoder were connected to the corresponding inputs of the seven-segment LED display (LTS-2801AB) using 220 Ω resistors.

At this point, the circuit described above would attempt to cycle through all 16 digits of the 4-bit binary number range. However, it was desired that the 7-segment display only displayed numbers 0 to 9 and then cycle back through the digits. In order to tackle this issue, all that was needed was simple binary logic to truncate the cycle such that the digit after 9 would not display but rather the cycle would reset to 0. Next, the binary representation of 10, that is, $1010_2$, had to be configured. It was noticed that the binary number $1010_2$ represents the first time that the leftmost digit and the digit second from the right are both 1 in the binary numbers 0 through 10. It is important to note that from right to left, the binary digits represent the JK flip-flop outputs Q1, Q2, Q3, and Q4, respectively. What this means is that when the counter cycle reaches 10, or $1010_2$, Q4 (the leftmost digit) and Q2 (the second to the right) are both 1, that is, the Q4 and Q2 JK flip-flop outputs are high. In order to reset the cycle to zero, an addition to the circuit had to be made such that when Q4 and Q2 are high, the circuit resets the JK flip-flop inputs to 0.

Therefore, Q4 and Q2 were input into a NAND gate (74HC00) so that when both are 1, as is the case when the cycle reaches $1010_2$, the output of the NAND gate is 0 as shown in the NAND gate truth table in Figure 5. The output of the NAND gate was then put into the reset pin of all 4 inputs of the flip flops so that when Q4 and Q2 are both 1, the flip-flops will be in reset mode, bringing them back to 0. Then, they cycle up to 9 and reset again. A circuit diagram and image of the binary counter assembled in the lab and are displayed in Figures 6 and 7, respectively.

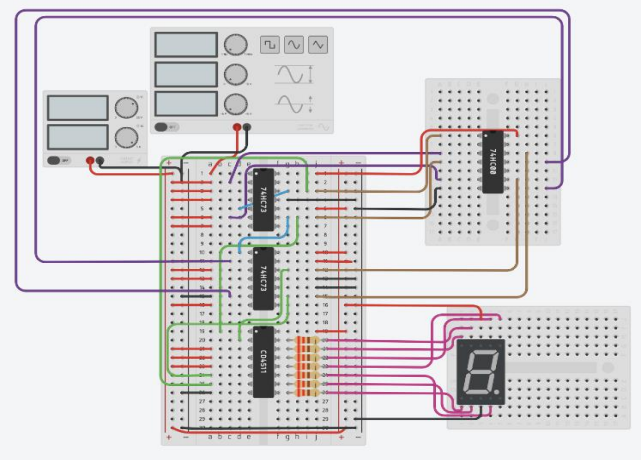

**Fig. 6:** *Illustration of binary counter circuit, showing the JK and NAND ICs and 7-segment digital display.*

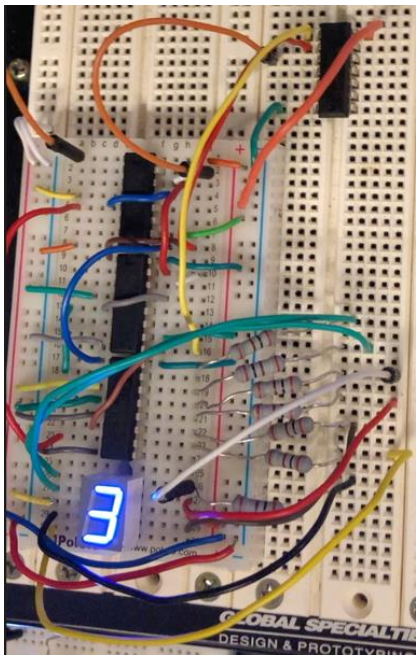

**Fig. 7:** *Photo of binary counter built in the device.*

## 3. Playing the game

To play "Morse in a Minute," the players will be given the morse code alphabet alongside the corresponding letters and a key showing how much each letter is worth. Like in Scrabble®, more uncommon letters will be worth more points, shown in Table 1. Once the game begins, each player will have one minute, timed by the counter, to write the highest scoring word they can. When time is up, if the letters displayed on the LCD screen make up a word, the player will be awarded a score equal to the sum of all the letters' points. However, if the letters displayed on the LCD screen do not make up a word, the player will receive no points. The player with the highest score wins!

| Points Earned | Letter(s) |
|---|---|
| 1 | A, E, I, O, U, L, N, S, T, R |
| 2 | D, G |
| 3 | B, C, M, P |
| 4 | F, H, V, W, Y |
| 5 | K |
| 8 | J, X |
| 10 | Q, Z |

**Table 1.** *Points assigned per letter characters. [5]*

## 4. Future Improvements

The preliminary design was successful and was demonstrated in the lab. Currently we are developing the design with some additional features and looking forward to presenting the project at ASEE-NE 2022. In the future, this project could be improved by turning the 4-bit up-counter into a timer, with two 7-segment LED displays that counts down from 60 and plays a buzzer when 0 is reached, so that the time passed on the counter does not need to be monitored. Additionally, an RGB LED could be added to differentiate buttons with unique colors. The code will incorporate the scoring system for the words spelled and output a final score when the time runs out. The code will also have to use specific frequencies to output the right colors on the RGB LED.


## Acknowledgements

Thank you to Bin Zhu for providing hands-on assistance during the assembly of our prototype. We would also like to thank the Northeastern University Department of Physics for financially supporting our experience at the ASEE-NE 2022 conference.


## References


1. "History of Morse," NRICH, https://nrich.maths.org/2198.
2. T. A. Team, "What is Arduino?," Arduino. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction.
3. "JK Flip-Flop Explained in Detail - Dcaclab Blog," https://dcaclab.com/blog/j-k-flip-flop-explained-in-detail/.
4. "Boolean Algebra Truth Tables for Logic Gate Functions," Basic Electronics Tutorials, March 24, 2020, https://www.electronics-tutorials.ws/boolean/bool_7.html.
5. Hasbro, "Scrabble FAQ: Answers to Your Scrabble Questions: Hasbro," scrabble, https://scrabble.hasbro.com/en-us/faq