Alexander Shah
Project 1 – Morse Code LED
2024-02-04
EN.605.715.81.SP24

Table of Contents

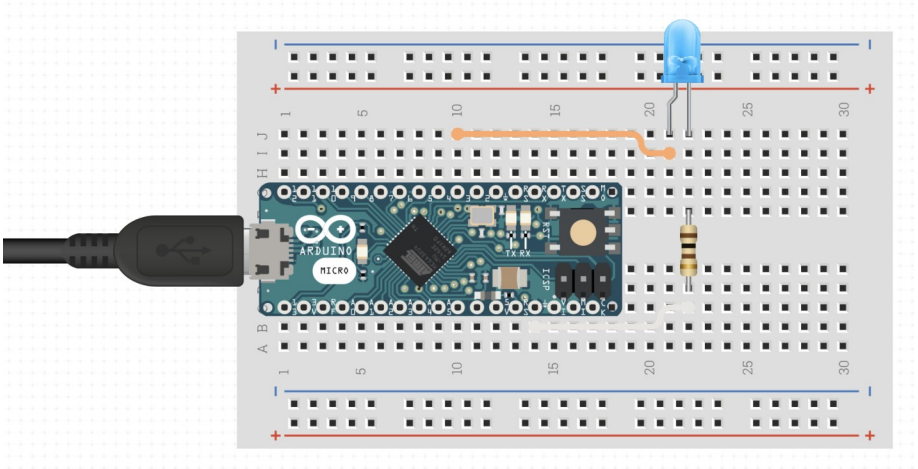
NOTE to the Grader: My Arduino kit arrives the day after the deadline for this project, so the demonstration uses the integrated LEDs on an Arduino Micro to flash the message, but the code and functionality is the same.

Requirements

This projects uses an Arduino Micro, an LED, a resistor, and wiring, to create a blinking morse code beacon using a connected device for text input. The program uses a loop to capture input until a quit command is given, either control + Z or the string “QQ”. The project was written in Arduino IDE in a C like language and deployed on a knock off Arduino Micro. These designs requirements are from the Project 1 prompt.

Design

The Arduino Micro, LED, resistor, and wiring are assembled simply on a breadboard, as seen in Figure 1 below. The Arduino pin defined in the code uses pin 9 at the corner of the board, compared to the image below which shows another pin. The long arm of the LED should be connected to the output pin, and the resistor should be connected between the short leg and ground, shown in barely visible white below the board in the image below. The demonstrated implementation does not contain the breadboarding, but instead uses the integrated LEDs of an arduino micro to flash the message. The code and functionality remains the same.



(Figure 1) Assembled project layout

Implementation

The code for the Project, written in Arduino IDE, can be compiled and run as a single file, the contents for the morse.ino file are:

```
 /* Project 1 - Arduino Morse Code
 Alex Shah
 1/28/24
  */

 //using a knock off arduino micro, assign pin 9 for led output, or use rx/tx led
to blink
 int iLEDPIN = 9;
 int iRXLED = 17;
 int iTXLED = 30;

 //1 unit for dot
 //3 units for dash
 //1 unit between dot/dash in same letter
 //3 units space between letters
 //7 units between words
 int iUnit = 300;

 //use control + Z to terminate loop, ascii character is '\x1A'
 bool fInterrupt = false;

 void setup() {
   pinMode(iLEDPIN, OUTPUT);
   pinMode(iRXLED, OUTPUT);
   pinMode(iTXLED, OUTPUT);

   Serial.begin(9600);
   while (!Serial) {
     //wait for serial connection
     ;
   }
   Serial.println("Enter some text (A-Z,a-z), then press enter and watch the led
flash your message! Use control + z or 'QQ' to quit.");
 }

 void loop() {
   while (!fInterrupt) {
     if (Serial.available()) {
       // convert input to morse code until interrupted with control + Z, or QQ
because control + z not sent through Arduino IDE
       String input = Serial.readStringUntil('\n');
       input.trim();
       if (input == '\x1A' || input.equalsIgnoreCase("QQ")) {
         fInterrupt = true;
       } else {
         //sleep to prevent first input getting eaten
         stop_leds();
         delay(2000);

         show_morse(input);
         stop_leds();
       }
     }
     // sleep on loop break
```

```
    delay(1000);
  }
  Serial.println("Program exiting...");
  stop_leds();
  delay(1000);
  exit(0);
}

void show_morse(String input) {
  for (int i = 0; i < input.length(); i++) {
    char character = toUpperCase(input[i]);
    switch (character) {
      case ' ':
        word_space();
        break;
      case 'A':
        dot();
        dash();
        letter_space();
        break;
      case 'B':
        dash();
        dot();
        dot();
        dot();
        letter_space();
        break;
      case 'C':
        dash();
        dot();
        dash();
        dot();
        letter_space();
        break;
      case 'D':
        dash();
        dot();
        dot();
        letter_space();
        break;
      case 'E':
        dot();
        letter_space();
        break;
      case 'F':
        dot();
        dot();
        dash();
        dot();
        letter_space();
        break;
      case 'G':
        dash();
        dash();
        dot();
        letter_space();
        break;
      case 'H':
        dot();
```

```
  dot();
  dot();
  dot();
  letter_space();
  break;
case 'I':
  dot();
  dot();
  letter_space();
  break;
case 'J':
  dot();
  dash();
  dash();
  dash();
  letter_space();
  break;
case 'K':
  dash();
  dot();
  dash();
  letter_space();
  break;
case 'L':
  dot();
  dash();
  dot();
  dot();
  letter_space();
  break;
case 'M':
  dash();
  dash();
  letter_space();
  break;
case 'N':
  dash();
  dot();
  letter_space();
  break;
case 'O':
  dash();
  dash();
  dash();
  letter_space();
  break;
case 'P':
  dot();
  dash();
  dash();
  dot();
  letter_space();
  break;
case 'Q':
  dash();
  dash();
  dot();
  dash();
  letter_space();
```

```
  break;
case 'R':
  dot();
  dash();
  dot();
  letter_space();
  break;
case 'S':
  dot();
  dot();
  dot();
  letter_space();
  break;
case 'T':
  dash();
  letter_space();
  break;
case 'U':
  dot();
  dot();
  dash();
  letter_space();
  break;
case 'V':
  dot();
  dot();
  dot();
  dash();
  letter_space();
  break;
case 'W':
  dot();
  dash();
  dash();
  letter_space();
  break;
case 'X':
  dash();
  dot();
  dot();
  dash();
  letter_space();
  break;
case 'Y':
  dash();
  dot();
  dash();
  dash();
  letter_space();
  break;
case 'Z':
  dash();
  dash();
  dot();
  dot();
  letter_space();
  break;
default:
  Serial.write("Invalid letter or missed something...");
```

```
        break;
    }
  }
  stop_leds();
}

void letter_space() {
  delay(3 * iUnit);
}

void word_space() {
  delay(7 * iUnit);
}

void dot() {
  //dot on for 1 unit
  digitalWrite(iLEDPIN, HIGH);
  digitalWrite(iRXLED, LOW);
  digitalWrite(iTXLED, LOW);
  delay(iUnit);
  //delay 1 unit between dot/dash
  digitalWrite(iLEDPIN, LOW);
  digitalWrite(iRXLED, HIGH);
  digitalWrite(iTXLED, HIGH);
  delay(iUnit);
}

void dash() {
  //dash on for 3 units
  digitalWrite(iLEDPIN, HIGH);
  digitalWrite(iRXLED, LOW);
  digitalWrite(iTXLED, LOW);
  delay(3 * iUnit);
  //delay 1 unit between dot/dash
  digitalWrite(iLEDPIN, LOW);
  digitalWrite(iRXLED, HIGH);
  digitalWrite(iTXLED, HIGH);
  delay(iUnit);
}

void stop_leds() {
  digitalWrite(iLEDPIN, LOW);
  digitalWrite(iRXLED, HIGH);
  digitalWrite(iTXLED, HIGH);
}
```

Demo

Please see the video at the link below for a code walkthrough, and demonstration of functionality:
https://drive.google.com/file/d/1ABUOe7u5lMC8gpXaE0C6s5jQ8Dmne5eB/view?usp=sharing

References

https://jhu.instructure.com/courses/64026/files/9535686?wrap=1
https://docs.arduino.cc/built-in-examples/basics/Blink/