a.

Morse code transmission of "SOS" includes three short blinks (S) followed by three longer blinks (O) and ending with another three short blinks (S). The LED blinks with brief illumination for dots during the Morse code while dashes are marked by extended LED illumination.

For hardware installation a 5mm external LED received power through pin 9 of the Arduino Uno with a 100-ohm resistor functioning as a limit for electrical current. One endpoint of the LED was linked to the ground reference point.

I began to create the Arduino code following the completion of hardware setup. The program contained an outline that established the LED pins as an output in addition to scope lodging functions with precise time intervals for Morse code behavior. The programming was transferred to the Arduino board through the Arduino Integrated Development Environment (IDE). The upload operation was checked for any coding issues while successful upload allowed me to inspect the LED functions. The observations became the basis for altering the timing parameters until the dots and dashes timing reached accuracy standards.

b.

Morse code transmissions highly depends on proper timing execution. The duration for dots was set to 200 milliseconds in the program and dashes were programmed for 500 milliseconds during this project. Within a Morse letter the duration between dot and dash elements was set to 200 milliseconds. The program used 1000 milliseconds as the gap duration to separate Morse code letters from each other. The system along with its specified timing made it possible for the blinking LED to easily transmit distinct characters.

c.

// Define the pin where the LED is connected

int ledPin = 9; // Pin 9 is used for the external LED

// Setup function

void setup() {

pinMode(ledPin, OUTPUT); // Set ledPin as an output

}

// Function to flash the LED

void flash(int duration) {

digitalWrite(ledPin, HIGH); // Turn the LED on

delay(duration); // Wait for 'duration' milliseconds

digitalWrite(ledPin, LOW); // Turn the LED off

delay(duration);

}

// Loop function

void loop() {

// S: dot dot dot

flash(200);

flash(200);

flash(200);

delay(300);

// O: dash dash dash

flash(500);

flash(500);

flash(500);

delay(300);

// S: dot dot dot

flash(200);

flash(200);

flash(200);

delay(1000); // Wait before repeating the SOS signal

}

d. Provide screenshot. A screenshot of a computer

Description automatically generatede. Take picture and addA circuit board with wires connected to it

Description automatically generated

f.

I examined the memory consumption after sending the program to Arduino Uno. The downloaded code allocation requires 1-2 KB of SRAM and 5-10% Flash memory on an Arduino Uno board. Memory usage efficiency enables additional modifications and expansions of the system.

g.

I would create a mapping of characters to their corresponding Morse representations. A mapping system of characters to Morse codes will enable this code to transmit any desired Morse message. The application includes a functionality that accepts string input parameters to convert individual characters into Morse code patterns while operating the blinking LED.