Transmitter Code

```
#define LED_PIN 1  // TX0 pin
char message[] = "EV BEHIND";  // Message to transmit
void setup() {
  Serial.begin(38400);  // Initialize serial communication for debug output
  pinMode(LED_PIN, OUTPUT);
  Serial.println("Setup complete");  // Debug output
}

void loop() {
  Serial.println("ev behind ");  // Debug output
  for (int i = 0; message[i] != '\0'; i++) {
    // Convert character to binary (assuming ASCII)
    char byte = message[i];
    for (int j = 7; j >= 0; j--) {
      int bit = (byte >> j) & 1;
      // Simulate light pulse using LED
      digitalWrite(LED_PIN, bit);
      delayMicroseconds(1000); // Adjust pulse width based on desired baud rate
      digitalWrite(LED_PIN, LOW);
      delayMicroseconds(1000); // Adjust inter-symbol time
    }
  }
  delay(1000); // Delay between transmissions (optional)
  Serial.println("alert ");  // Debug output
}
```

Receiver Code

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Pins for the LCD: RS, EN, D4, D5, D6, D7

void setup() {
  Serial.begin(38400); // Start serial communication at 9600 baud rate
  lcd.begin(16, 2);  // Initialize the LCD with 16 columns and 2 rows
}

void loop() {
  String receivedString = "";
  if (Serial.available()) {
    while (Serial.available()) {
      char c = Serial.read();
      if (isPrintableAscii(c)) { // Check if character is printable ASCII
        receivedString += c;
      }
    }
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Received:");
    lcd.setCursor(0, 1);
    lcd.print(receivedString);

    if (receivedString.length() > 0) { // Check if there's valid data to print
      Serial.print("Data received and displayed on LCD: ");
      Serial.println(receivedString);
      delay(10000);

      // Display received data on Serial Monitor
```

```
      Serial.print("Received: ");
      Serial.println(receivedString);
    }
  } else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("No data received");

    Serial.println("No data received from Serial");
  }
}

bool isPrintableAscii(char c) {
  return (c >= 32 && c <= 126); // Printable ASCII characters range from 32 to 126
}
```