**Morse Code Reader with OLED Display**

**✅ Project Title:**

**Morse Code Reader with Star Animation, Word Detection & Screensaver using ESP32 and OLED**

**📌 Objective:**

To build a Morse Code reader using an ESP32 board that detects button input as Morse code (dot/dash), decodes it to English letters and words, displays output on an OLED screen, and includes a stylish UI with a star animation, centred welcome screen, and idle screensaver.

**🧰 Hardware Requirements:**

| **Component** | **Quantity** | **Description** |
| --- | --- | --- |
| ESP32 Dev Board | 1 | Main controller |
| Push Button | 1 | To input Morse code (dots and dashes) |
| OLED Display (SSD1306) | 1 | 128x64 I2C OLED for text and animation display |
| Buzzer | 1 | For audio feedback when button is pressed |
| LED | 1 | For visual feedback during button press |
| Resistors (10k) | 1 | For pull-down or pull-up with button (if needed) |
| Breadboard + Wires | - | For circuit assembly |

**🔌 Pin Configuration:**

**(**[**URL:-https://app.cirkitdesigner.com/project/4701fc8e-6df0-4dbc-aaeb-de212902bd25**](URL:-https://app.cirkitdesigner.com/project/4701fc8e-6df0-4dbc-aaeb-de212902bd25)**)**

| **Component** | **Pin Name** | **ESP32 Pin** |
| --- | --- | --- |
| Button | Digital | D14 |
| Buzzer | Digital | D25 |
| LED | Digital | D26 |
| OLED | SDA | D21 |
| OLED | SCL | D22 |

**🔁 Working Principle:**

1. **Button Press Logic:**
   * **Short Press < 300 ms:** Treated as a **dot (.)**
   * **Long Press ≥ 300 ms:** Treated as a **dash (-)**
2. **Letter Pause:**
   * If no button is pressed for 500 ms after a dot/dash sequence, it's decoded into a character.
3. **Word Pause:**
   * If there is a ≥ 3 sec pause between inputs, a space is added to form words.
4. **Idle Screensaver:**
   * After 15 seconds of inactivity, the system displays a **moving star screensaver**.
   * Pressing the button exits the screensaver.

**🔁 CODE**

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

#define OLED\_RESET    -1

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

// Pin definitions

#define BUTTON\_PIN 14

#define BUZZER\_PIN 25

#define LED\_PIN    26

// Timing variables

unsigned long pressStart = 0;

bool buttonPressed = false;

String morseBuffer = "";

String decodedText = "";

unsigned long lastInputTime = 0;

bool inScreensaver = false;

// Thresholds (ms)

const int dotThreshold = 300;

const int letterPause = 500;

const int wordPause = 3000;

const int idleTimeout = 15000;  // 15 seconds of inactivity → go to screensaver

// 8x8 star bitmap

const unsigned char starBitmap[] PROGMEM = {

  0b00011000,

  0b00111100,

  0b01111110,

  0b11111111,

  0b01111110,

  0b00111100,

  0b00011000,

  0b00000000

};

// Morse code mapping

struct MorseMap {

  const char\* code;

  char letter;

};

MorseMap morseTable[] = {

  {".-", 'A'}, {"-...", 'B'}, {"-.-.", 'C'}, {"-..", 'D'},

  {".", 'E'}, {"..-.", 'F'}, {"--.", 'G'}, {"....", 'H'},

  {"..", 'I'}, {".---", 'J'}, {"-.-", 'K'}, {".-..", 'L'},

  {"--", 'M'}, {"-.", 'N'}, {"---", 'O'}, {".--.", 'P'},

  {"--.-", 'Q'}, {".-.", 'R'}, {"...", 'S'}, {"-", 'T'},

  {"..-", 'U'}, {"...-", 'V'}, {".--", 'W'}, {"-..-", 'X'},

  {"-.--", 'Y'}, {"--..", 'Z'},

  {"-----", '0'}, {".----", '1'}, {"..---", '2'}, {"...--", '3'},

  {"....-", '4'}, {".....", '5'}, {"-....", '6'}, {"--...", '7'},

  {"---..", '8'}, {"----.", '9'}

};

// Decode Morse

char decodeMorse(String code) {

  for (MorseMap m : morseTable) {

    if (code == m.code) return m.letter;

  }

  return '?';

}

// Centered text helper

void drawCenteredText(int y, String text) {

  int16\_t x1, y1;

  uint16\_t w, h;

  display.getTextBounds(text, 0, y, &x1, &y1, &w, &h);

  display.setCursor((SCREEN\_WIDTH - w) / 2, y);

  display.print(text);

}

// Show welcome screen

void showWelcomeScreen() {

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(WHITE);

  for (int y = -8; y < SCREEN\_HEIGHT + 8; y += 2) {

    display.clearDisplay();

    for (int i = 0; i < 6; i++) {

      int x = 10 + i \* 20;

      int yOffset = (y + i \* 7) % SCREEN\_HEIGHT;

      display.drawBitmap(x, yOffset, starBitmap, 8, 8, WHITE);

    }

    display.display();

    delay(50);

  }

  display.clearDisplay();

  drawCenteredText(20, "Welcome to");

  drawCenteredText(35, "Morse Code Reader");

  display.display();

  delay(3000);

  display.clearDisplay();

  display.display();

}

// Show Morse & Message

void showOLED() {

  display.clearDisplay();

  display.setTextSize(1);

  display.setTextColor(WHITE);

  display.setCursor(0, 0);

  display.print("Morse: ");

  display.println(morseBuffer);

  display.setCursor(0, 20);

  display.print("Message:");

  display.setCursor(0, 32);

  display.println(decodedText);

  display.display();

}

// Screensaver animation

void showScreensaver() {

  static int yOffset = -8;

  display.clearDisplay();

  for (int i = 0; i < 6; i++) {

    int x = 10 + i \* 20;

    int y = (yOffset + i \* 7) % SCREEN\_HEIGHT;

    display.drawBitmap(x, y, starBitmap, 8, 8, WHITE);

  }

  display.display();

  yOffset += 2;

  if (yOffset > SCREEN\_HEIGHT) yOffset = -8;

  delay(50);

}

void setup() {

  pinMode(BUTTON\_PIN, INPUT);

  pinMode(BUZZER\_PIN, OUTPUT);

  pinMode(LED\_PIN, OUTPUT);

  Serial.begin(115200);

  if (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) {

    Serial.println("OLED not found");

    while (1);

  }

  randomSeed(analogRead(0));

  showWelcomeScreen();

  showOLED();

  lastInputTime = millis();

}

void loop() {

  bool state = digitalRead(BUTTON\_PIN);

  // Wake up from screensaver

  if (inScreensaver && state == HIGH) {

    inScreensaver = false;

    showOLED();

    delay(300);  // debounce

  }

  // If in screensaver, animate stars

  if (inScreensaver) {

    showScreensaver();

    return;

  }

  // Button pressed

  if (state == HIGH && !buttonPressed) {

    pressStart = millis();

    buttonPressed = true;

    digitalWrite(BUZZER\_PIN, HIGH);

    digitalWrite(LED\_PIN, HIGH);

  }

  // Button released

  if (state == LOW && buttonPressed) {

    unsigned long pressDuration = millis() - pressStart;

    buttonPressed = false;

    digitalWrite(BUZZER\_PIN, LOW);

    digitalWrite(LED\_PIN, LOW);

    morseBuffer += (pressDuration < dotThreshold) ? "." : "-";

    lastInputTime = millis();

    showOLED();

  }

  // Decode letter

  if (!buttonPressed && morseBuffer.length() > 0 && millis() - lastInputTime > letterPause) {

    char decodedChar = decodeMorse(morseBuffer);

    decodedText += decodedChar;

    morseBuffer = "";

    lastInputTime = millis();  // reset last activity

    showOLED();

  }

  // Detect word pause

  if (!buttonPressed && morseBuffer.length() == 0 && decodedText.length() > 0 &&

      millis() - lastInputTime > wordPause) {

    if (decodedText[decodedText.length() - 1] != ' ') {

      decodedText += " ";

      showOLED();

    }

  }

  // Enter screensaver after long idle time

  if (!buttonPressed && millis() - lastInputTime > idleTimeout) {

    inScreensaver = true;

  }

}

**🎨 User Interface:**

1. **Startup Animation:**
   * Falling/moving 8x8 star-bitmaps appear in a dynamic scrolling animation.
2. **Welcome Message:**
   * Centered text:
   * Welcome to
   * Morse Code Reader
3. **Main Display:**
   * Displays live Morse input and decoded message:
   * Morse: ..-. .- -...
   * Message:
   * FAB
4. **Screensaver Mode:**
   * Shows floating stars in vertical motion during inactivity.

**🧠 Key Code Logic:**

**1. Morse Buffering:**

morseBuffer += (pressDuration < dotThreshold) ? "." : "-";

**2. Letter Decoding:**

char decodedChar = decodeMorse(morseBuffer);

decodedText += decodedChar;

**3. Word Detection:**

if (millis() - lastInputTime > wordPause)

decodedText += " ";

**4. Screensaver Timer:**

if (millis() - lastInputTime > idleTimeout)

inScreensaver = true;

**5. Star Bitmap:**

const unsigned char starBitmap[] PROGMEM = {

0b00011000,

0b00111100,

0b01111110,

0b11111111,

0b01111110,

0b00111100,

0b00011000,

0b00000000

};

**🧪 Testing Instructions:**

1. Upload code using Arduino IDE with **ESP32 board selected**.
2. Power on the device – the OLED should show **welcome animation**.
3. Use the **button**:
   * Short press for **dot**
   * Long press for **dash**
4. Wait 0.5 sec after a letter → character gets decoded.
5. Pause for 3 sec → **space** added to separate words.
6. Inactivity of 15 sec → shows **screensaver**, which resumes on button press.

**📈 Enhancements Suggested:**

* Add support for **backspace/clear** via another button.
* Show **decoded Morse letter** alongside each code.
* Use a **piezo speaker tone** for better Morse feedback.
* Store full message and allow **scrolling on OLED**.

**📂 Libraries Used:**

* Adafruit\_GFX.h
* Adafruit\_SSD1306.h
* Wire.h