

TUGAS BESAR MIKROPROSESOR DAN ANTARMUKA

Reaction Game Berbasis STM32F103

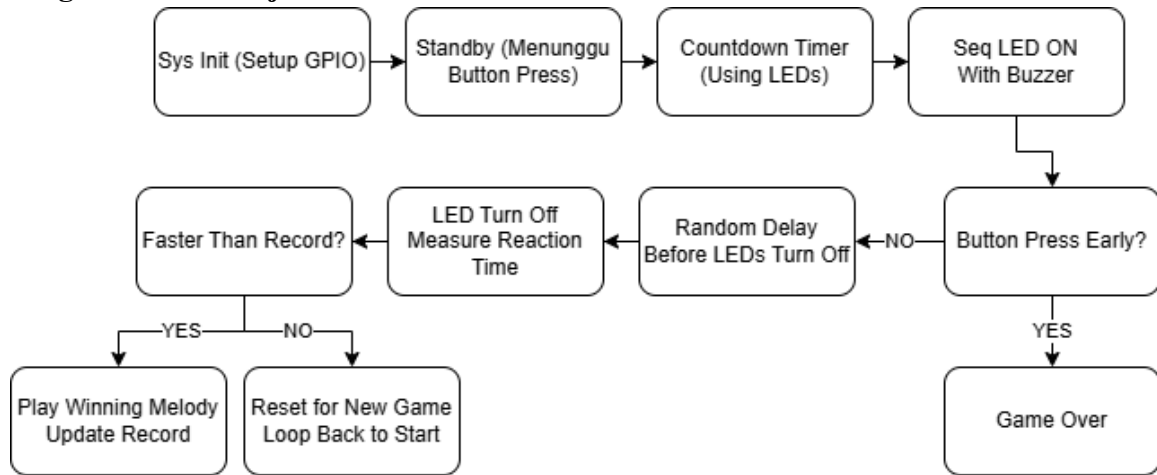
Sub-Module Main Game Loop dan LED Control



Disusun Oleh
Dhanendra Nivadirokhman 1103220113

Teknik Komputer
Fakultas Teknik Elektro
Universitas Telkom
2024

A. Diagram Cara Kerja



B. Penjelasan

1. Main Game Loop

Main game loop adalah bagian utama dari program yang menjalankan alur permainan reaction game. Loop ini bertanggung jawab untuk menunggu input dari pengguna, menjalankan logika permainan, dan mengatur pengendalian LED, buzzer, serta menghitung waktu reaksi. Berikut ini adalah langkah-langkah utamanya:

2. LED Control

Sub-modul ini menangani pengaturan LED, termasuk penyalan secara berurutan selama permainan dan mematikan LED setelah permainan dimulai. LED digunakan sebagai indikator visual untuk memberikan informasi kepada pemain tentang waktu yang tepat untuk menekan tombol.

C. Cara Kerja

1. Main Game loop

a. Menunggu Penekanan Tombol

- Sistem menunggu hingga tombol ditekan untuk memulai permainan.
- Jika tombol tidak ditekan, loop akan terus berjalan dalam kondisi standby.

b. Countdown Sebelum Permainan

- Countdown dilakukan selama 3 detik sebelum permainan dimulai untuk memberi waktu persiapan kepada pemain.

c. Pengaturan LED

- LED menyala secara berurutan, diiringi dengan bunyi beep dari buzzer.

d. Pengecekan Reaksi Pemain

- Jika tombol ditekan terlalu dini saat LED masih menyala, permainan dianggap gagal, dan buzzer memainkan melodi kekalahan.

- Jika pemain menunggu hingga LED mati dan menekan tombol, waktu reaksi dihitung.
- e. **Penghitungan Rekor**
- Jika waktu reaksi pemain lebih cepat dari rekor sebelumnya, rekor diperbarui, dan melodi kemenangan dimainkan.
- 2. LED Control**
- a. **Penyalan LED**
- LED dinyalakan secara berurutan dengan durasi tertentu (LED_DELAY).
 - Ketika tombol ditekan terlalu cepat, seluruh LED akan dimatikan sebagai tanda kegagalan.
- b. **Pemadaman LED**
- Setelah waktu acak (waitDelay), semua LED dimatikan, dan pemain harus menekan tombol secepat mungkin.

C. Hasil Implementasi

Source Code Main Loop :

```
while (1) {
    if (!btnPressed) {
        continue; // Standby mode
    }

    btnPressed = 0; // Reset button flag
    HAL_Delay(3000); // Countdown

    for (int i = 0; i < NUM_LEDS; i++) {
        if (btnPressed) {
            // Too early, play losing melody
            playMelody(loseMelody, loseDurations, 5);
            break;
        }
        HAL_GPIO_WritePin(GPIOA, ledPins[i], GPIO_PIN_SET); // Turn on LED
        HAL_Delay(LED_DELAY); // Wait
        HAL_GPIO_WritePin(GPIOA, ledPins[i], GPIO_PIN_RESET); // Turn off LED
    }

    uint32_t waitDelay = MIN_WAIT_DELAY + (rand() % (MAX_WAIT_DELAY -
MIN_WAIT_DELAY + 1));
    HAL_Delay(waitDelay);

    ledsOffTime = HAL_GetTick();
    for (int i = 0; i < NUM_LEDS; i++) {
        HAL_GPIO_WritePin(GPIOA, ledPins[i], GPIO_PIN_RESET); // Turn off all LEDs
    }
}
```

```

}

while (!btnPressed); // Wait for button press
reactionTime = HAL_GetTick() - ledsOffTime;

if (reactionTime < record) {
    record = reactionTime; // Update record
    playMelody(winMelody, winDurations, 6); // Play winning melody
}
}

```

Source Code LED Control :

```

for (int i = 0; i < NUM_LEDS; i++) {
    HAL_GPIO_WritePin(GPIOA, ledPins[i], GPIO_PIN_SET); // Turn on LED
    HAL_Delay(LED_DELAY); // Delay between transitions
    HAL_GPIO_WritePin(GPIOA, ledPins[i], GPIO_PIN_RESET); // Turn off LED
}

for (int i = 0; i < NUM_LEDS; i++) {
    HAL_GPIO_WritePin(GPIOA, ledPins[i], GPIO_PIN_RESET); // Ensure all LEDs are off
}

```

D. Hubungan dengan Laporan Proyek

Berdasarkan laporan proyek di dokumen "Reaction game berbasis Bluepill," main game loop dan LED control sepenuhnya mendukung alur permainan sebagaimana dijelaskan dalam bagian Cara Kerja:

1. Mode standby, countdown, penyalan LED, dan deteksi reaksi telah terintegrasi dalam main game loop.
2. LED control digunakan untuk memberi indikator visual selama permainan berlangsung, sesuai fungsi yang dirancang.