LAPORAN MODUL BUZZER
TUGAS BESAR KELOMPOK SURJONO

Reaction Game Berbasis STM32F103

Disusun Oleh:

Maulana Hafiz 1103223105

Mata Kuliah:

MIKROPROSESOR DAN ANTARMUKA

Teknik Komputer

Fakultas Teknik Elektro

Universitas Telkom

2024

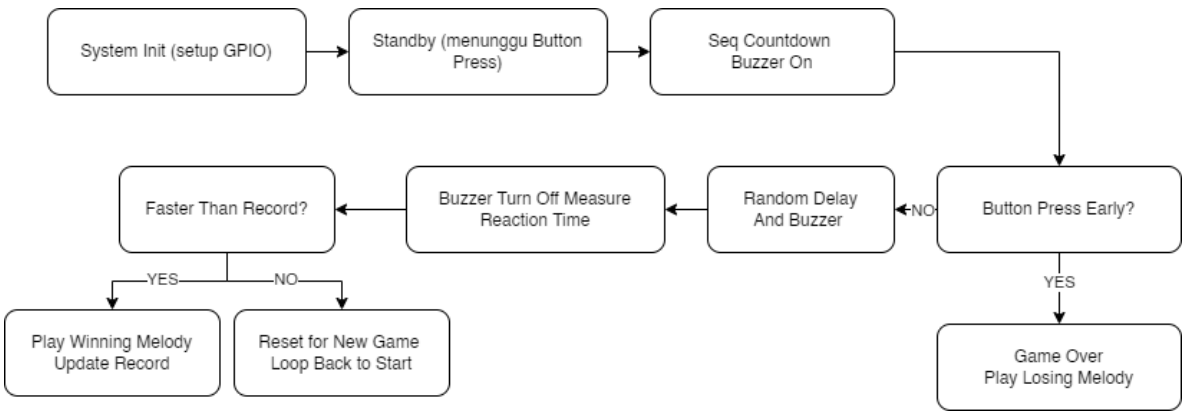
A. Latar Belakang

Suara atau melodi sering digunakan dalam berbagai aplikasi atau permainan untuk memberikan *feedback* kepada seseorang. Pada proyek ini, buzzer digunakan untuk memberikan suara sebagai *feedback* dalam permainan *Reaction Time* berbasis mikrokontroler STM32F103. Ketika seseorang menekan tombol terlalu cepat atau berhasil menekan tombol tepat waktu, sistem akan memutar sebuah suara atau memberikan *feedback* sebagai tanda kemenangan atau kekalahan. *Feedback* suara ini bertujuan untuk meningkatkan sebuah permainan agar lebih interaktif dan lebih terkesan hidup.

B. Alat dan Bahan

1. STM32F103 Blue Pill
2. Buzzer Passive
3. Button
4. LED
5. Kabel Jumper
6. Breadboard
7. FTDI
8. STM32CubeMX

C. Cara kerja



Pada sistem ini, buzzer digunakan untuk memberikan *feedback* suara berupa melodi atau suara beep. Ada dua jenis *feedback* yang diberikan oleh sistem:

1. Melodi Kekalahan:
   Ketika pemain menekan tombol terlalu cepat, buzzer akan memutar sebuah melodi yang menunjukkan kekalahan.
2. Melodi Kemenangan:
   Jika pemain berhasil menekan tombol setelah LED dimatikan dengan waktu reaksi tercepat, buzzer akan memutar sebuah melodi kemenangan.

Selain itu, buzzer juga memberikan *feedback* suara berupa beep singkat setiap kali LED menyala, yang memberi tahu pemain bahwa mereka harus siap menekan tombol setelah LED dimatikan. Proses pemutaran melodi dan *feedback* suara dilakukan dengan menggunakan fungsi playMelody() yang mengatur frekuensi dan durasi setiap nada. Berikut adalah cara kerjanya:

- Melodi kemenangan atau kekalahan, setiap melodi terdiri dari serangkaian nada yang dimainkan satu per satu dengan durasi tertentu.
- Feedback beep, ketika LED menyala, buzzer menghasilkan suara beep singkat yang memberi sinyal kepada pemain.

D. Algoritma

Melodi dimainkan dengan menggunakan array frekuensi yang berisi nilai-nilai frekuensi untuk setiap nada yang ingin diputar. Setiap nada diputar dalam durasi tertentu. Fungsi playMelody() bertanggung jawab untuk memutar melodi yang terdiri dari beberapa nada berturut-turut. Berikut adalah langkah-langkah dalam implementasi melodi:

- Array Frekuensi: Setiap melodi terdiri dari array frekuensi (dalam Hertz) untuk setiap nada yang ingin diputar.
- Array Durasi: Durasi dari setiap nada didefinisikan dalam array durasi (dalam milidetik).
- Looping Melodi: Fungsi playMelody() memutar setiap nada dari array frekuensi dengan durasi yang sesuai menggunakan fungsi HAL_Delay() untuk mengatur jeda antara nada.

Secara keseluruhan, buzzer digunakan untuk memberikan *feedback* suara yang penting selama permainan, baik untuk mengindikasikan kekalahan, kemenangan, atau sebagai penanda waktu dan reaksi. Berikut adalah source code nya:

- Inisialisasi Pin Buzzer :

```
GPIO_InitStruct.Pin = BUZZER_PIN;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);
HAL_GPIO_WritePin(GPIOC, BUZZER_PIN, GPIO_PIN_RESET);
```

Pada fungsi GPIO_Init(), pin buzzer (PC14) diinisialisasi sebagai output untuk mengontrol buzzer. Inisialisasi ini memastikan buzzer siap digunakan.

- Pemutaran melodi

```
void playMelody(uint16_t *melody, uint16_t *durations, uint8_t length) {
  for (int i = 0; i < length; i++) {
    HAL_GPIO_WritePin(GPIOC, BUZZER_PIN, GPIO_PIN_SET);
    HAL_Delay(durations[i]);
    HAL_GPIO_WritePin(GPIOC, BUZZER_PIN, GPIO_PIN_RESET);
    HAL_Delay(50);
  }
}
```

Fungsi playMelody() digunakan untuk memutar melodi ketika game dimulai atau ketika ada *feedback* suara untuk pemain. Di sini buzzer dinyalakan (GPIO_PIN_SET) dan dimatikan (GPIO_PIN_RESET) sesuai dengan durasi nada yang dimainkan.

- *Feedback* kekalahan

```
uint16_t loseMelody[] = {220, 196, 174, 164, 147};
uint16_t loseDurations[] = {300, 300, 300, 300, 300};
playMelody(loseMelody, loseDurations, 5);
```

Ketika pemain menekan tombol terlalu cepat (sebelum waktu yang tepat), sistem memainkan melodi kekalahan menggunakan buzzer. Fungsi playMelody() digunakan dengan parameter loseMelody[] dan loseDurations[] untuk memainkan melodi tertentu.

- *Feedback* beep selama LED menyala

```
HAL_GPIO_WritePin(GPIOC, BUZZER_PIN, GPIO_PIN_SET);
HAL_Delay(200);
HAL_GPIO_WritePin(GPIOC, BUZZER_PIN, GPIO_PIN_RESET);
```

Setiap kali LED menyala, buzzer juga memberikan *feedback* berupa suara beep. Buzzer dinyalakan selama durasi beep (200 ms) dan dimatikan setelahnya.

- *Feedback* kemenangan

```
uint16_t winMelody[] = {262, 294, 330, 349, 392, 440};
uint16_t winDurations[] = {300, 300, 300, 300, 300, 300};
playMelody(winMelody, winDurations, 6);
```

Ketika pemain berhasil menekan tombol dengan waktu reaksi tercepat, buzzer memainkan melodi kemenangan. Fungsi playMelody() digunakan dengan parameter winMelody[] dan winDurations[] untuk memainkan melodi tertentu yang menunjukkan kemenangan.

- Buzzer sebagai penanda

```
HAL_GPIO_WritePin(GPIOC, BUZZER_PIN, GPIO_PIN_SET);
```

Setelah LED mati, buzzer menghasilkan suara tanpa henti untuk memberi tahu pemain bahwa mereka bisa menekan tombol. Suara buzzer tetap aktif selama menunggu pemain menekan tombol.

- Menghentikan buzzer setelah tombol ditekan

```
HAL_GPIO_WritePin(GPIOC, BUZZER_PIN, GPIO_PIN_RESET);
```

Setelah pemain menekan tombol, suara buzzer dimatikan untuk menghentikan *feedback* suara tersebut.

E. Hasil Implementasi

Setelah implementasi selesai, hasil yang didapatkan sesuai dengan yang diharapkan:

1. Melodi Kemenangan dan Kekalahan:
   Ketika pemain menekan tombol terlalu cepat, buzzer memutar melodi kekalahan. Jika pemain berhasil menekan tombol dengan waktu reaksi tercepat, buzzer memainkan melodi kemenangan.

2. *feedback* Suara (Beep):

   Buzzer memberikan suara beep yang jelas setiap kali LED menyala. Suara beep ini memberi tahu pemain bahwa mereka harus siap menekan tombol setelah LED dimatikan.

3. Keberhasilan Melodi dan Feedback Suara:

   Semua nada diputar dengan baik, dan suara beep memberikan *feedback* yang jelas kepada pemain. *feedback* suara berfungsi dengan baik dan memberikan informasi yang tepat selama permainan.

F. Kesimpulan

Pemutaran melodi dan *feedback* suara yang diterapkan pada sistem permainan berbasis STM32F103 berjalan dengan baik. Buzzer berhasil memainkan melodi kekalahan dan kemenangan sesuai dengan kondisi permainan, serta memberikan suara beep yang jelas saat LED menyala. Sistem ini efektif memberikan *feedback* yang diperlukan untuk meningkatkan pengalaman pemain dalam permainan reaksi. Penggunaan buzzer sebagai *feedback* suara membuat permainan lebih interaktif dan menyenangkan, serta memberikan respons yang cepat kepada pemain.