

TUGAS BESAR MIKROPROSESOR DAN ANTARMUKA

Reaction Game Berbasis STM32F103

Modul SysTick & Hal_Delay



Disusun Oleh :

Marcellinus Geofani Sihaloho 1103223171

**Teknik Komputer
Fakultas Teknik Elektro
Universitas Telkom
2024**

A. Latar Belakang

Pada permainan reaction game ini, pengelolaan waktu menjadi aspek penting untuk memastikan pengalaman bermain yang lancar. Permainan ini mengandalkan presisi waktu untuk mengukur kecepatan reaksi pemain setelah semua LED mati. Untuk itu, dua modul utama digunakan, yaitu SysTick dan HAL_Delay, yang bekerja bersama untuk mendukung mekanisme waktu dalam permainan.

Modul SysTick berfungsi sebagai dasar penghitungan waktu dengan menghasilkan interrupt periodik setiap 1 milidetik. Interrupt ini digunakan untuk memperbarui counter waktu internal (`uwTick`) yang menjadi acuan bagi seluruh operasi waktu di aplikasi. Dengan presisi hingga 1 ms, SysTick memungkinkan pengukuran waktu yang cukup akurat untuk kebutuhan permainan ini.

Sementara itu, modul HAL_Delay memanfaatkan mekanisme SysTick untuk menyediakan waktu tunda (delay) pada berbagai tahapan permainan. Fungsi ini digunakan untuk:

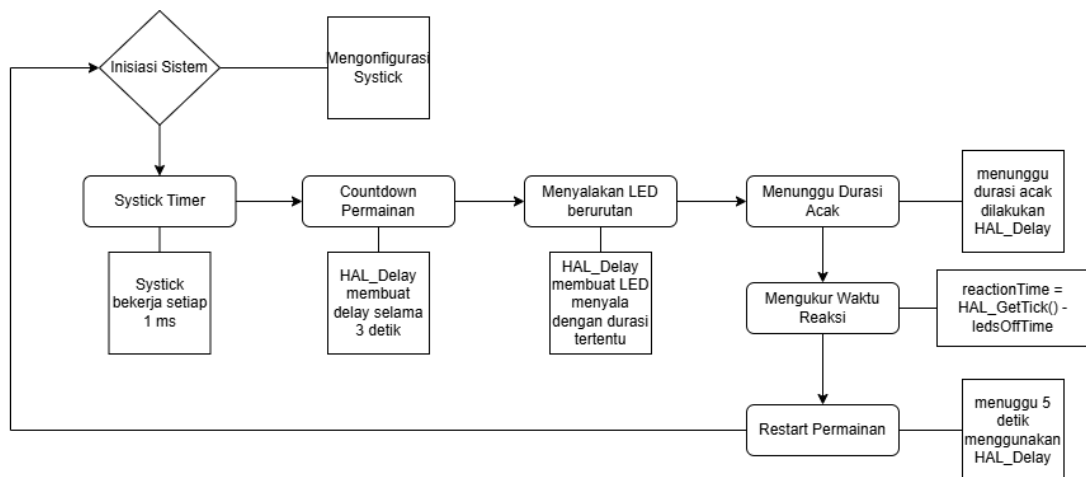
1. Memberikan countdown sebelum permainan dimulai.
2. Mengatur durasi nyala LED dan bunyi buzzer.
3. Mengimplementasikan waktu tunda acak sebelum pemain bereaksi.
4. Memberikan jeda sebelum permainan dimulai ulang.

Kombinasi SysTick dan HAL_Delay memberikan solusi yang efektif untuk mengelola waktu dalam reaction game. Dengan pendekatan ini, permainan dapat berjalan dengan baik tanpa memerlukan mekanisme pengelolaan waktu yang lebih kompleks, seperti multitasking atau timer non-blocking.

B. Alat dan Bahan

1. STM32F103 Blue Pill
2. Breadboard
3. LED Display
4. LED
5. Buzzer Passive
6. Kabel Jumper
7. Kabel Connector
8. Switch Button

C. Cara Kerja



1. Inisialisasi:
HAL_SYSTICK_Config() mengatur timer SysTick untuk menghasilkan interrupt setiap 1 ms. Counter waktu di dalam HAL library mulai bekerja.
2. Membuat Delay:
Ketika HAL_Delay(ms) dipanggil, fungsi mencatat waktu awal (tickstart) dari counter waktu. SysTick terus memicu interrupt setiap 1 ms, memperbarui counter waktu. Fungsi HAL_Delay memeriksa apakah durasi waktu yang diminta telah berlalu berdasarkan nilai counter waktu.
3. Fungsi Timer Lainnya:
Selain untuk delay, interrupt SysTick juga dapat digunakan untuk fungsi lain, seperti mengatur flag untuk tugas periodik atau fungsi real-time lainnya.

D. Hasil Implementasi

1. Inisialisasi Sistem
 - Sistem memulai dengan fungsi HAL_Init() untuk mengatur konfigurasi awal hardware abstraction layer (HAL).
 - Fungsi GPIO_Init() mengatur pin GPIO untuk tombol, LED, dan buzzer. Pin tombol diatur sebagai input dengan interrupt, sedangkan pin LED dan buzzer diatur sebagai output.
 - Fungsi Timer_Init() mengatur SysTick untuk menghasilkan interrupt setiap 1 ms. Hal ini dilakukan dengan mengonfigurasi SysTick melalui HAL_SYSTICK_Config().
2. SysTick Timer
 - Setelah inisialisasi, timer SysTick mulai bekerja di latar belakang.
 - Setiap 1 ms, interrupt SysTick dipicu. Dalam interrupt ini, HAL library memperbarui counter global yang digunakan untuk mencatat waktu (HAL_GetTick()).

- Counter ini digunakan untuk pengukuran waktu, termasuk dalam HAL_Delay().

3. Loop Utama

- **Menunggu Tombol Ditekan**
Program memasuki loop utama di mana ia terus memeriksa apakah tombol telah ditekan. Tombol ini diawasi oleh interrupt eksternal yang mengatur flag btnPressed jika tombol ditekan. Jika tombol tidak ditekan, sistem tetap menunggu.
- **Countdown Permainan**
Setelah tombol ditekan, permainan dimulai dengan countdown 3 detik. Countdown ini dilakukan menggunakan HAL_Delay(3000), yang membuat program menunggu selama 3 detik.
- **Menyalakan LED Secara Berurutan**
LED dinyalakan satu per satu dalam urutan tertentu. Setiap LED tetap menyala untuk durasi tertentu (LED_DELAY), dan penundaan antar LED dilakukan dengan HAL_Delay(). Jika tombol ditekan selama proses ini, sistem menganggap pemain menekan terlalu cepat dan memainkan melodi kekalahan.
- **Menunggu Durasi Acak**
Setelah semua LED menyala, sistem menunggu durasi acak sebelum mematikan semua LED. Durasi ini diambil dari rentang MIN_WAIT_DELAY hingga MAX_WAIT_DELAY. Penundaan dilakukan dengan HAL_Delay(waitDelay).

4. Mengukur Waktu Reaksi

- **Mematikan LED**
Ketika durasi acak selesai, semua LED dimatikan, dan waktu saat ini dicatat menggunakan HAL_GetTick().
- **Menunggu Tombol Ditekan**
Program memasuki loop di mana ia menunggu tombol ditekan. Saat tombol ditekan, waktu saat itu dicatat lagi, dan waktu reaksi dihitung sebagai:
 $reactionTime = HAL_GetTick() - ledsOffTime.$
- **Mengevaluasi Rekor**
Jika waktu reaksi pemain lebih cepat dari rekor sebelumnya, waktu tersebut disimpan sebagai rekor baru, dan melodi kemenangan dimainkan. Jika tidak, melodi kekalahan dimainkan.

5. Restart Permainan

- Setelah melodi selesai dimainkan, program menunggu 5 detik menggunakan HAL_Delay(5000) sebelum memulai ulang permainan. Sistem kembali ke loop utama untuk menunggu tombol ditekan lagi.

E. Kesimpulan

SysTick bekerja di latar belakang untuk menghasilkan interrupt setiap 1 ms, yang digunakan untuk mencatat waktu dan menjalankan fungsi delay. HAL_Delay adalah fungsi blocking yang menggunakan counter waktu dari SysTick untuk menciptakan penundaan yang akurat. Dengan kombinasi keduanya, sistem dapat menampilkan urutan permainan, menghitung waktu reaksi, dan menunda proses sesuai kebutuhan.