



EMBEDDED SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

PACEGUARD: HIGH PRECISION RUNNING
PACE MONITORING SYSTEM

GROUP PA-MBD-11

Anthonius Hendhy Wirawan	2306161795
Javana Muhammad Dzaki	2306161826
Maxwell Zefanya Ginting	2306221200
Ruben Kristanto	2306214624

PREFACE

Puji syukur kehadiran Tuhan Yang Maha Esa, atas rahmat dan karunia-Nya sehingga proyek akhir praktikum Embedded System yang berjudul PaceGuard ini dapat diselesaikan dengan baik. Proyek ini dirancang sebagai solusi berbasis *microcontroller* ATmega328P pada Arduino yang bertujuan untuk mendukung kegiatan lari dengan presisi tinggi, khususnya dalam hal pengukuran kecepatan pelari secara real-time.

Berbeda dari solusi umum yang cenderung kompleks dan mahal, PaceGuard berfokus pada akurasi tinggi dalam pengukuran waktu menggunakan Infrared Sensor. Sistem ini dirancang untuk mencatat waktu secara presisi saat pelari melewati titik deteksi tertentu, sehingga diharapkan dapat membantu atlet meningkatkan performa lari mereka, *cause every milisecond counts*.

Seluruh sistem kami kembangkan menggunakan bahasa Assembly AVR melalui Arduino IDE, untuk mendapatkan kontrol yang lebih presisi terhadap ATmega dan efisiensi maksimal dalam eksekusi program. Melalui proyek ini, kami berharap PaceGuard dapat menjadi kontribusi nyata dalam pengembangan *embedded system* presisi tinggi, serta menjadi referensi untuk proyek-proyek serupa di masa mendatang.

Depok, 16 Mei, 2025

Group PA11

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

- 1.1 Background
- 1.2 Project Description
- 1.3 Objectives
- 1.4 Roles and Responsibilities

CHAPTER 2: IMPLEMENTATION

- 2.1 Equipment
- 2.2 Implementation
- 2.3 Schematics

CHAPTER 3: TESTING AND ANALYSIS

- 3.1 Testing results and expectations
- 3.2 analysis

CHAPTER 4: CONCLUSION

REFERENCES

APPENDICES

- Appendix A: Project Progress & Documentation

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Dalam dunia olahraga lari, terutama di level kompetitif dan high-performance, presisi waktu adalah segalanya. Perbedaan satu milidetik saja bisa menjadi penentu antara kemenangan dan kekalahan. Ketika seorang pelari mendekati garis finish, semua latihan, strategi pacing, dan daya tahan bermuara pada satu hal: seberapa cepat mereka menyelesaikan lintasan. Oleh karena itu, sistem pencatatan waktu yang akurat dan responsif bukan hanya pelengkap, melainkan komponen krusial dalam evaluasi dan pengembangan performa atletik.

Namun, teknologi pengukuran waktu yang tersedia di pasaran saat ini sering kali tidak terjangkau, bersifat kompleks, atau kurang cocok untuk lingkungan pelatihan mandiri maupun komunitas lari. Banyak pelari, terutama mereka yang berlatih untuk long-distance atau marathon, mengandalkan strategi pacing, yaitu menjaga kecepatan konstan sepanjang lintasan, untuk mengoptimalkan energi dan menjaga performa. Tanpa sistem yang akurat untuk mengukur waktu dan memberi umpan balik secara langsung, pelari kesulitan mengetahui apakah mereka telah mempertahankan pace yang ideal atau tidak.

PaceGuard hadir sebagai jawaban atas tantangan ini. Sistem ini memungkinkan pelari untuk memantau waktu tempuh di titik-titik tertentu secara otomatis, membantu mereka melatih konsistensi kecepatan dan mengoptimalkan strategi pacing. Tidak hanya bagi atlet profesional, solusi ini juga sangat relevan bagi pelatih, komunitas lari, dan individu yang ingin meningkatkan performa dengan cara yang terukur, presisi, dan terjangkau.

Dengan menggunakan teknologi sensor inframerah dan platform Arduino yang diprogram menggunakan bahasa Assembly AVR, PaceGuard menawarkan akurasi tinggi, waktu respons cepat, dan kinerja yang dapat diandalkan bahkan di kondisi pelatihan outdoor. Di era di mana setiap detik memiliki arti besar dalam dunia olahraga, PaceGuard menjadikan waktu sebagai sahabat, bukan sekadar angka.

1.2 PROJECT DESCRIPTION

PaceGuard merupakan sistem pencatatan waktu otomatis berbasis *microcontroller* ATmega328P pada Arduino UNO R3 yang menggunakan sensor inframerah untuk mendeteksi saat pelari melewati titik tertentu di lintasan. Setiap pelintas sensor akan memicu pencatatan waktu dengan presisi tinggi, yang kemudian diproses menggunakan kode berbasis Assembly AVR untuk efisiensi dan keakuratan maksimal. Data waktu yang dikumpulkan akan ditampilkan pada layar dan juga disimpan untuk kebutuhan evaluasi performa atlet, karena setiap milidetik dapat menentukan kemenangan.

1.3 OBJECTIVES

Objektif dari proyek ini adalah sebagai berikut:

1. Mengembangkan sistem pengukuran waktu presisi tinggi untuk olahraga lari menggunakan mikrokontroler.
2. Menerapkan pemrograman low-level (Assembly AVR) pada Arduino untuk efisiensi sistem dan pengendalian akurat.
3. Mendesain sistem yang mampu mendeteksi dan mencatat waktu secara otomatis melalui sensor inframerah.
4. Mengaplikasikan pemahaman teoritis mengenai *embedded system* pada proyek nyata sebagai sebuah solusi

1.4 ROLES AND RESPONSIBILITIES

Berikut ini adalah pembagian tugas dan kewajiban pada kelompok PA11 kami;

Roles	Responsibilities	Person
Role 1	Kode Serial communication to serial monitor & algoritma aritmatika dan konversi	Anthonius Hendhy Wirawan

Role 2	PPT, Laporan Master, Designs, AVR Code Base, Half Readme,	Javana Muhammad Dzaki
Role 3	Kode LCD dan Timer, Rangkaian Proteus, README pt. i-iii	Maxwell Zefanya
Role 4	kode dari distance set hingga led sebelum timer start. referensi laporam	Ruben Kristanto

Table 1. Roles and Responsibilities

CHAPTER 2

IMPLEMENTATION

2.1 EQUIPMENT

No.	Nama	QTY
1	Arduino IDE	Software
2	Proteus 8 Professional	Software
3	Breadboard small	1x
4	Arduino UNO R3	1x
5	KY-032 Infrared Sensor	1x
6	16x2 LCD with I2C adapter	1x
7	LED (Green Red Yellow)	3x
8	Button	3x
9	Resistor	7x
10	Kabel jumper	Secukupnya

Project Repository

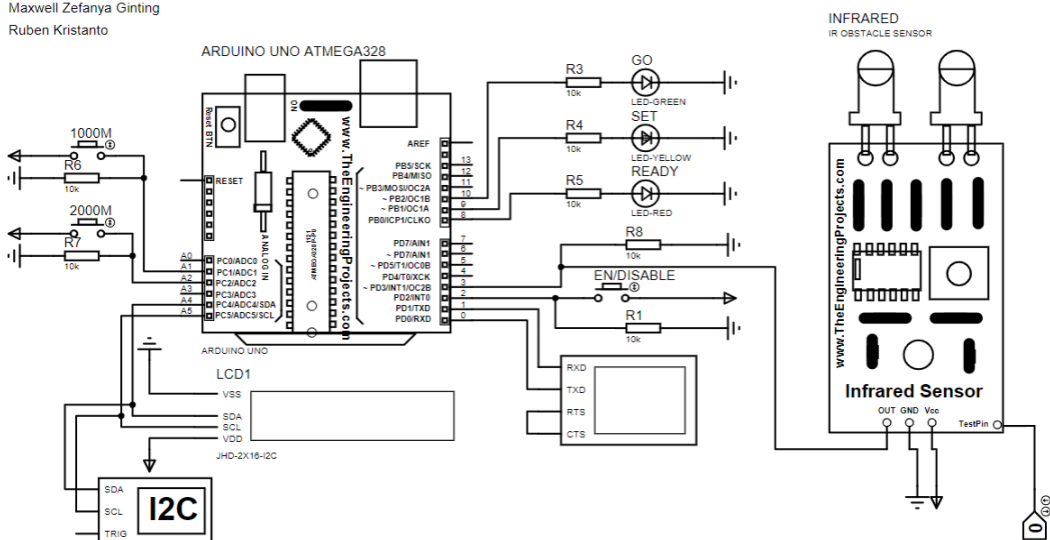
<https://github.com/Maxwell-Zefanya/MBD-11-Finpro>

Embedded System Final Project G11

Anthonius Hendhy Wirawan
Javana Muhammad Dzaki
Maxwell Zefanya Ginting
Ruben Kristanto

PaceGuard Connection Diagram

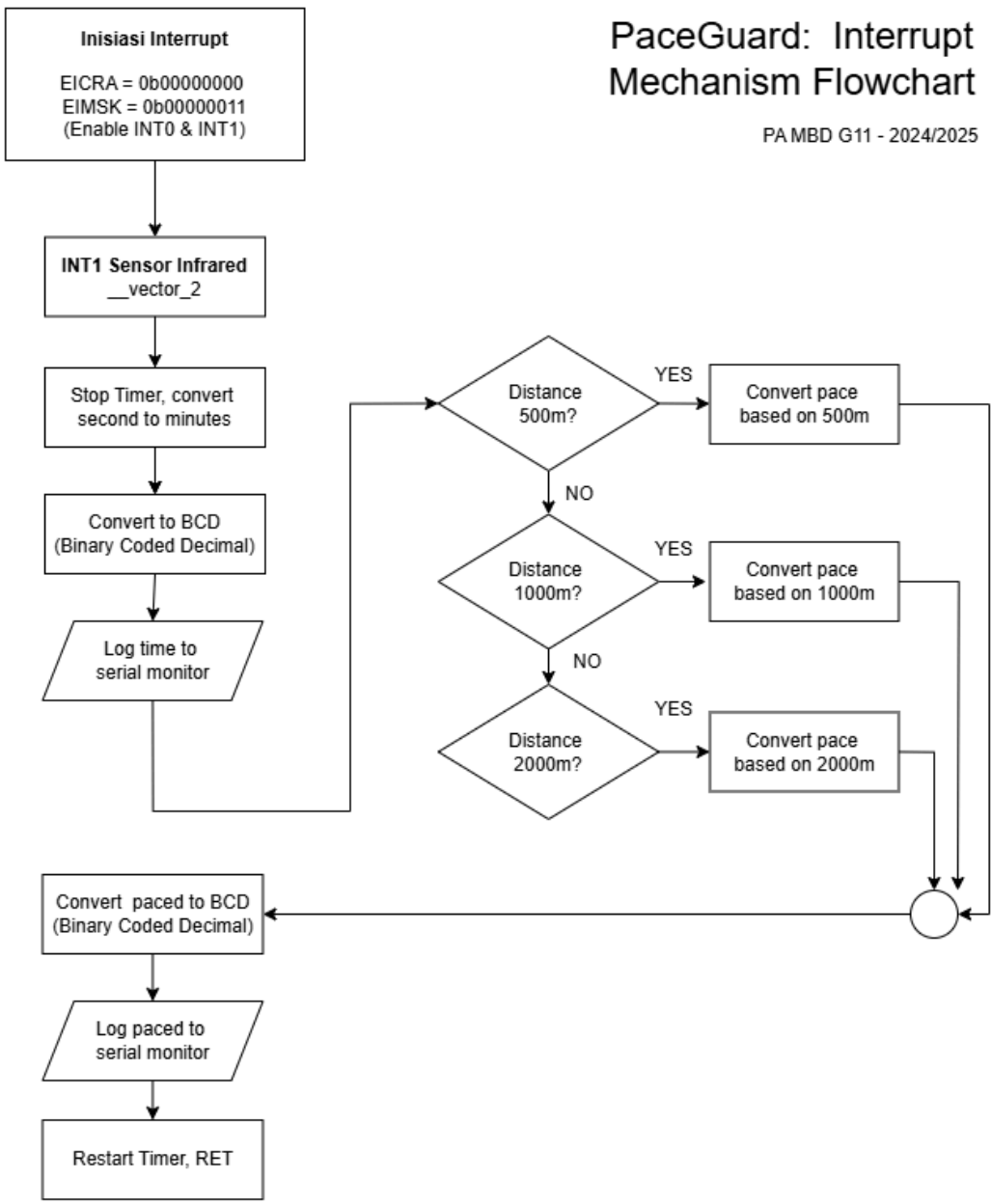
High Precision Running Pace Monitoring System



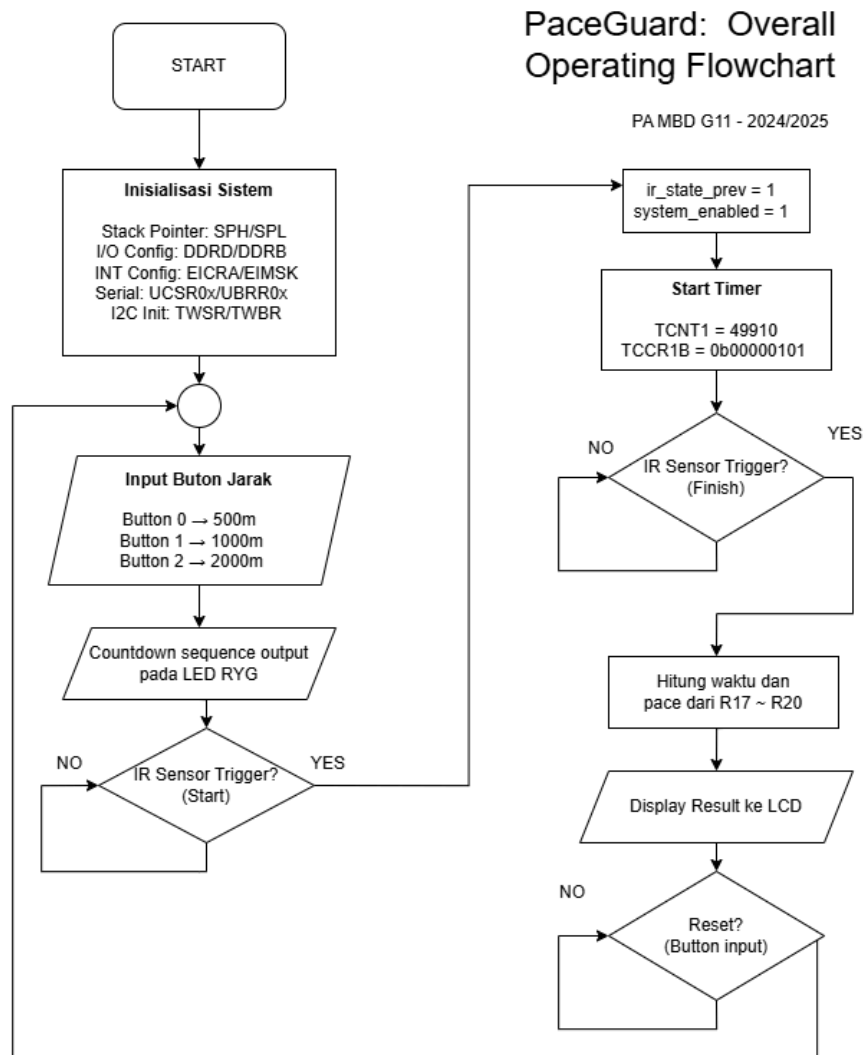
PaceGuard Schematic Diagram

PaceGuard: Interrupt Mechanism Flowchart

PAMBD G11 - 2024/2025



PaceGuard Interrupt Mechanism Flowchart



PaceGuard Overall Operating Flowchart

2.2 IMPLEMENTATION

PaceGuard memiliki program software yang dibuat secara khusus untuk keperluan pacing. Software tersebut ditulis sepenuhnya dalam bahasa assembly, dan ditargetkan untuk microprocessor ATmega328P yang terdapat pada Arduino UNO R3. Alur kerja program bisa dibagi menjadi 7 bagian utama, sesuai dengan gambar flowchart di atas. Arduino yang ada pada sistem bisa mendapatkan program dengan pertama mengcompile (verify) kode kemudian menguploadnya menggunakan software Arduino IDE. Secara lebih lengkap, alur kerja dari program adalah sebagai berikut:

a. Setup

Fase setup dilakukan untuk memastikan Arduino bisa berkomunikasi dengan komponen eksternal secara benar dan efisien. Kode pada fase ini sebagian besar terdapat pada label main. Hal-hal yang diatur pada fase setup adalah sebagai berikut;

- Penyetelan stack pointer, diarahkan kepada bagian paling akhir RAM. Dilakukan karena program akan banyak melakukan operasi push dan pop
- Penyetelan semua button dan sensor sebagai input, dan semua LED sebagai output.
- Memastikan bahwa register yang dipakai untuk menyimpan waktu lap user sudah kosong.
- Mengaktifkan fungsionalitas interrupt untuk INT0 dan INT1.
- Mengaktifkan serial communication dengan baud rate 9600.
- Menyetel konfigurasi Two-Wire Interface (I2C) agar Arduino bekerja sebagai Master Transmitter. I2C akan digunakan untuk menampilkan pesan pada LCD.
- Print welcome message pada LCD menggunakan I2C. Penjelasan selengkapnya ada pada fase Display
- Kode setup sebagian besar distruktur secara in-line, yang berarti isi kode berada satu disamping yang lainnya. Bagian kode yang tidak in-line hanya terdapat pada saat menyetel I2C, dimana penyetelan dilakukan dengan instruksi CALL, yang akan memanggil function.

b. Select

Setelah segala aspek program berhasil disetel, maka program akan diam menunggu button press dari user. Fase ini bersifat opsional, karena dari kedua button, terdapat 3 opsi yang bisa dipilih oleh user. Bila user tidak menekan tombol select dan langsung menekan start, maka secara default program akan menggunakan jarak lari sepanjang 500m. Bila menekan, maka jarak lari yang digunakan sesuai dengan apa yang dipilih oleh user, baik 1000m atau 2000m.

Implementasi fase select berkaitan erat dengan fase start. Hal ini adalah karena meskipun seleksi menggunakan polling, button start akan memanggil start sequence

menggunakan interrupt. Oleh karena itu, user bisa langsung start (default 500m) tanpa perlu menekan button seleksi.

c. Start

Fungsi utama dari fase start adalah untuk memberikan aba-aba awal kepada pelari. Aba-aba diimplementasikan dalam bentuk LED sekuensial, dimana awalnya LED merah akan menyala, kemudian yang kuning, lalu yang hijau menyala untuk menandakan bahwa pelari bisa mulai berlari. Implementasi dari fase start cukup sederhana. Program hanya akan selang-seling mengubah register PORTB dan memanggil delay saja. Register PORTB diubah untuk menyalakan LED secara sekuensial, dan delay diimplementasikan menggunakan polling.

d. Run

Mulainya fase run ditandai dengan menyalnya LED hijau. Fase ini memiliki 2 bagian utama, yaitu bagian timer, dan bagian counter. Kode inti untuk fase run terdapat pada label loop, dimana loop akan memanggil timer dan menambah register X sebanyak 1. Instruksi BRTC berhubungan dengan fase record, dimana fase run akan terus berjalan selama fase record belum berjalan.

Implementasi timer dilakukan dengan melakukan polling. Caranya adalah program akan pertama menyetel TCNT (Timer Counter Register) sedemikian rupa sehingga timer akan overflow tepat pada saat waktu 1 detik. Artinya, timer akan mengeluarkan flag overflow setiap 1 detik. Flag inilah yang ditunggu program menggunakan polling, dimana setelah muncul baru lanjut ke penambahan counter detik.

Sesuai dengan namanya, counter detik bertugas untuk mencatat sudah berapa detik lamanya sejak awal lampu LED hijau menyala. Counter untuk detik berukuran 16-bit, dan disimpan pada register R26-R27, atau bisa disebut sebagai register X. Counter ini akan bertambah 1 angka setiap detiknya. Secara teori, timer bisa menghitung sampai

18 jam setiap lap. Rentang waktu tersebut dianggap sudah cukup, melihat PaceGuard hanya menerima jarak lari maksimum sepanjang 2000m.

e. Record

Fase record akan aktif secara bersamaan fase run mulai. Namun, fase ini hanya akan ada pada mode standby. Fase record baru mulai saat sensor infrared KY-032 mendeteksi adanya pelari yang melewati sensor tersebut. Sinyal yang datang dari KY-032 akan menyebabkan terjadinya interrupt pada Arduino. Program handler interrupt yang diimplementasikan akan melakukan 2 hal. Pertama, handler akan mematikan timer yang sedang berjalan, dimana hasil akan terdapat pada register X. Kedua, handler akan menyetel bit T pada SREG menjadi 1. Bit T disetel dengan tujuan program bisa keluar dari fase run, dan bisa mulai melakukan kalkulasi pace. Implementasi lengkap untuk handler dapat dilihat pada label `__vector_2`.

Kalkulasi pace perlu dilakukan karena pace akan ditampilkan dalam format menit:detik, dimana format ini berlawanan dengan format register X yang hanya menyimpan waktu dalam detik. Kalkulasi pace terdapat pada label `hitung`, dimana luarannya adalah register yang menyimpan menit (R17), dan yang menyimpan detik (R18). Kalkulasi diimplementasikan dengan mengurangi register X dengan angka 60. Bila hasil X tidak negatif, maka R17 akan ditambah 1, bila iya, maka X akan ditambah 60 dan dipindahkan kedalam R18.

f. Display

Fase display sebenarnya banyak dijalankan, bukan hanya setelah record saja. Namun, fase display yang ditampilkan pada flowchart merujuk kepada display hasil pace dan laptime, yang akan menggunakan Serial Monitor. Meskipun begitu, fase ini tetap juga menggunakan LCD dengan I2C module. Oleh karena itu, fase display bisa dibagi menjadi 2 bagian, yaitu bagian Serial Monitor dan bagian LCD.

g. Serial Monitor

Fase yang ada pada flowchart merujuk kepada bagian Serial Monitor. Fase ini dimulai setelah program selesai mengonversi counter menjadi menit dan detik. Hasil yang ada pada register R17 dan R18 masih dalam bentuk biner, bukan format ASCII. Oleh karena itu, perlu dilakukan semacam konversi untuk mengubah angka biner menjadi format ASCII.

Program mengubah biner ke ASCII dengan pertama mengubah biner ke desimal. Caranya mirip seperti kalkulasi pace, dengan perbedaan register dikurangi dengan angka 10. Konversi akan menghasilkan 2 luaran, yaitu desimal puluhan dan satuan. Setelah konversi berlangsung, barulah semua hasil desimal dikonversikan menjadi ASCII dengan menambahkan nilai dengan 48.

Barulah setelah konversi selesai dilakukan kedua angka tersebut di-print ke serial monitor. Implementasinya adalah dengan menggunakan register UDR0, dimana data ASCII yang didapatkan dikirimkan ke UDR0 untuk kemudian dikirimkan ke serial monitor dan akhirnya di-print.

h. LCD

LCD hanya digunakan untuk mengirimkan beberapa informasi penting saja kepada pelari. Pesan yang ingin dikirimkan sudah ada dari awal sebagai data pada program. Pesannya sendiri berupa null-terminated string, dimana setiap kalimat diakhiri dengan karakter null. Program akan terus mengirimkan data string tersebut kepada LCD sampai ketemu nilai 0.

LCD bisa bekerja menggunakan modul I2C yang sudah ditempelkan. Secara garis besar, komunikasi I2C terjadi atas 4 tahap, yaitu setup, start, write & acknowledge, dan stop. Tahap setup sudah terjadi pada fase setup. Tahap start dilakukan dengan mengisi bit TWINT, TWSTA, TWEN dan TWEA kedalam TWCR. Tahap write dilakukan dengan mengisi TWDR dengan data yang diinginkan, dalam kasus ini

string message. Terakhir, tahap stop dilakukan dengan mengisi TWINT, TWSTO, TWEN kedalam TWCR.

Perlu diketahui juga bahwa terdapat pola pengiriman sekuensial khusus yang perlu dilakukan saat ingin berinteraksi dengan LCD. Pola ini dilakukan supaya LCD bisa disetel dengan benar dan proses pengiriman data string bisa terjadi tanpa masalah.

i. Reset

Fase reset hanya akan berjalan bila tombol enable dipencet setelah fase start berlangsung. Fase ini adalah fase yang paling sederhana, dimana secara implementasi hal yang dilakukan hanyalah mengubah lap-time menjadi 0 dan menambahkan 1 ke dalam lap counter.

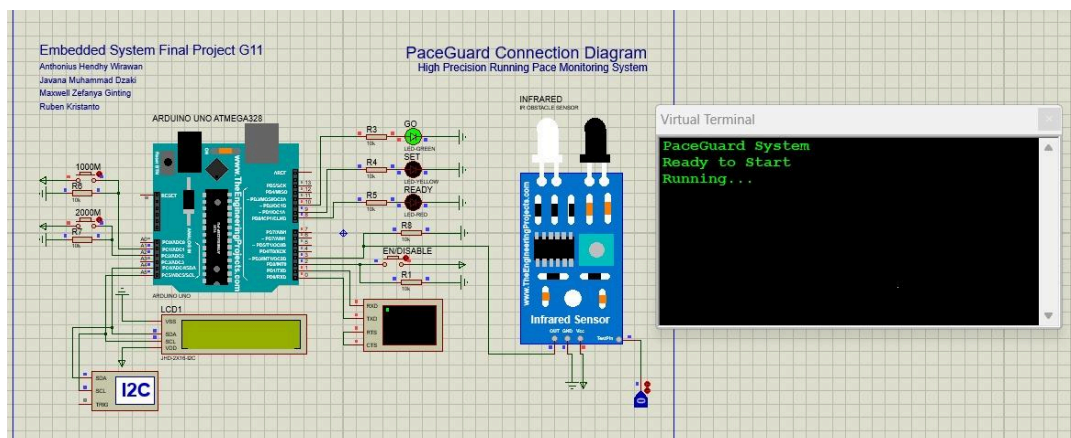
CHAPTER 3

Testing Results and Expectations

3.1 Expectations

hasil yang idealnya didapatkan dari Running Pace Monitoring System ini adalah berfungsinya keseluruhan dari sistem, dimana user pertama tama dapat memilih jarak dari track (1000m atau 2000m), kemudian memulai lap, akan ada indikator untuk memulai timer, dimana akan dinyalakan lampu LED secara bergantian, yang masing masing berwarna merah, kuning, dan hijau. Ketika lampu hijau menyala, maka timer akan dimulai, dan akan menyimpan nilai waktu yang telah berlalu di register. Setelah user melewati dan memicu sensor infrared, maka akan terpicu interrupt, dimana akan menghentikan timer, mengolah data waktu yang didapatkan, seperti mengubah data detik menjadi format menit:detik, mengonversi valuenya berdasarkan jarak untuk menghitung pace, serta mengubah value hasil perhitungan menjadi bentuk BCD dan ASCII untuk ditampilkan pada serial monitor. Sebagai hasil akhir dari sistem ini, akan ditampilkan waktu yang dihabiskan oleh user untuk berlalu tiap lapnya (dalam bentuk menit:detik) serta pace dari user selama berlari.

3.2 RESULT - PACEGUARD



3.6 ANALYSIS

implementasi software yang kami terapkan kepada rangkaian di simulasi menghasilkan hasil yang cukup memuaskan. Hampir semua fitur yang ada pada spesifikasi awal bisa berjalan dengan baik. Pengecualian terdapat pada LCD I2C, yang meskipun bisa mengeprint karakter di layar LCD, karakter yang ditampilkan tidak sesuai dengan keinginan.

Ketidakbisaannya LCD dalam mengeprint karakter yang benar terdapat pada kesalahan di kode. Hal ini adalah karena datasheet yang diambil sebagai referensi dari internet memiliki banyak data dan cara penggunaan yang kontradiktif. Alhasil, baik setup maupun pengiriman data karakter menjadi tidak sesuai spesifikasi yang ada khusus display LCD tersebut

CHAPTER 4

CONCLUSION

- PaceGuard berhasil memampukan pelari untuk mengatur pace sesuai dengan kebutuhannya serta menampilkan data lap yang akurat. Dengan menggunakan interrupt,
- Akurasi dan koresponsifan PaceGuard dapat dicapai berkat penggunaan interrupt dan kode assembly yang efisien
- Penggunaan LCD dan Serial display bisa memungkinkan pelari untuk melihat informasi yang penting mengenai kegiatan larinya

REFERENCES

M. Harditya and M. N. Faza, Lab Module: Fundamental of Digital System. Jakarta, Indonesia: Universitas Indonesia, 2024.

Digital Laboratory, Modul 8: SPI & I2C. Jakarta, Indonesia: Universitas Indonesia, 2024.

Digital Laboratory, Modul 7: Interrupt. Jakarta, Indonesia: Universitas Indonesia, 2024.

Digital Laboratory, Modul 4: Serial Port. Jakarta, Indonesia: Universitas Indonesia, 2024.

Microchip Technology Inc. (2015). ATmega328P: 8-bit AVR microcontroller with 32K bytes In-System Programmable Flash (Rev. 7810D–AVR–01/15).

Waveshare Electronics Co., Ltd., "LCD1602 Datasheet," Version 1.0, Oct. 2020. [Online]. Available: https://www.waveshare.com/datasheet/LCD_en_PDF/LCD1602.pdf

APPENDICES

Appendix A: Project Progress

