



Embedded System Final Project G11

PACEGUARD: HIGH PRECISION RUNNING PACE MONITORING SYSTEM

Antonius Hendhy Wirawan
Javana Muhammad Dzaki
Maxwell Zefanya Ginting
Ruben Kristanto

2306161795
2306161826
2306221200
2306214624

PACEGUARD



Where Miliseconds Matter.

TABLE OF CONTENTS

01 Pendahuluan

Latar Belakang

Rumusan Masalah

Mengukur Waktu

Menghitung Pace

Interrupt ISR INT1

LCD & I2C

02 Landasan Teori

Konsep Pacing dalam Lari/Running

Assembly AVR pada Arduino

05 Testing dan Analysis

Rangkaian Fisik

Hasil Pengujian

Analisis Performa

03 Implementasi Sistem

Schematic Diagram

Komponen Hardware

Overall Operation

Mekanisme Interrupt

06 Penutup

Kesimpulan

Saran

04 Implementasi AVR

Inisialisasi Sistem

07 Referensi

Pendahuluan Proyek



Latar Belakang

Dalam dunia olahraga lari, presisi waktu merupakan faktor yang sangat krusial. Perbedaan satu milidetik dapat menentukan kemenangan atau kekalahan seorang atlet. Sistem pencatatan waktu yang akurat bukan hanya pelengkap, melainkan komponen vital dalam evaluasi performa atletik.

PaceGuard hadir sebagai solusi atas tantangan ini, memungkinkan pelari memantau waktu tempuh di titik-titik tertentu secara otomatis sehingga dapat melatih konsistensi kecepatan dengan cara yang terukur, presisi, dan terjangkau.

Rumusan Masalah

PaceGuard hadir sebagai solusi atas tantangan dalam dunia running. Untuk melanjutkan perancangan, kami menyusun kerangka rumusan masalah sebagai berikut;

- 1** Bagaimana mengimplementasikan teknologi deteksi otomatis untuk menghitung waktu tempuh dengan minim intervensi manual?
- 2** Bagaimana merancang sistem timing yang dapat memberikan feedback real-time kepada pelari terkait pace mereka?
- 3** Bagaimana cara menghitung waktu pada ATMega328P menggunakan Assembly AVR untuk memastikan akurasi hingga level milisekon?

Landasan Teori

What is Pacing?



Dalam dunia lari, pacing ada strategi untuk mendistribusikan energi secara optimal selama aktivitas fisik berkelanjutan. Pacing ini sangat penting, karena merupakan faktor utama yang menentukan performa dan hasil akhir kompetisi.

Distribusi energi yang tepat sepanjang track memungkinkan pelari menghindari kelelahan terlalu cepat yang dapat mengganggu performa di bagian akhir perlombaan. Pacing berkaian erat dengan kecepatan yang menjadi kunci sukses dalam kompetisi lari jarak menengah dan jauh.

Pace per km	1 st km	2 nd km	3 rd km	4 th km	5 th km	Finish time
6:00	6:00	12:00	18:00	24:00	30:00	30:00
5:30	5:30	11:00	16:30	22:00	27:30	27:30
5:00	5:00	10:00	15:00	20:00	25:00	25:00

Assembly AVR Pada Arduino

Assembly AVR menawarkan paradigma eksekusi yang sangat berbeda dibandingkan bahasa pemrograman tingkat tinggi. Pada **PaceGuard**, pemilihan Assembly AVR didasarkan pada pertimbangan teknis yang mendalam untuk mencapai akurasi dan efisiensi maksimal.

Eksekusi instruksi dalam Assembly AVR memungkinkan prediktabilitas timing yang presisi karena setiap instruksi memiliki jumlah siklus clock yang tetap dan terdokumentasi. Karakteristik ini menjadi krusial dalam aplikasi pengukuran waktu presisi tinggi seperti di PaceGuard.

$$\begin{aligned} CPU_Time &= CPU_Clock_Cycles * Clock_Cycle_Time \\ &= \frac{CPU_Clock_Cycles}{Clock_Rate} \end{aligned}$$



Implementasi Sistem

Schematic Diagram

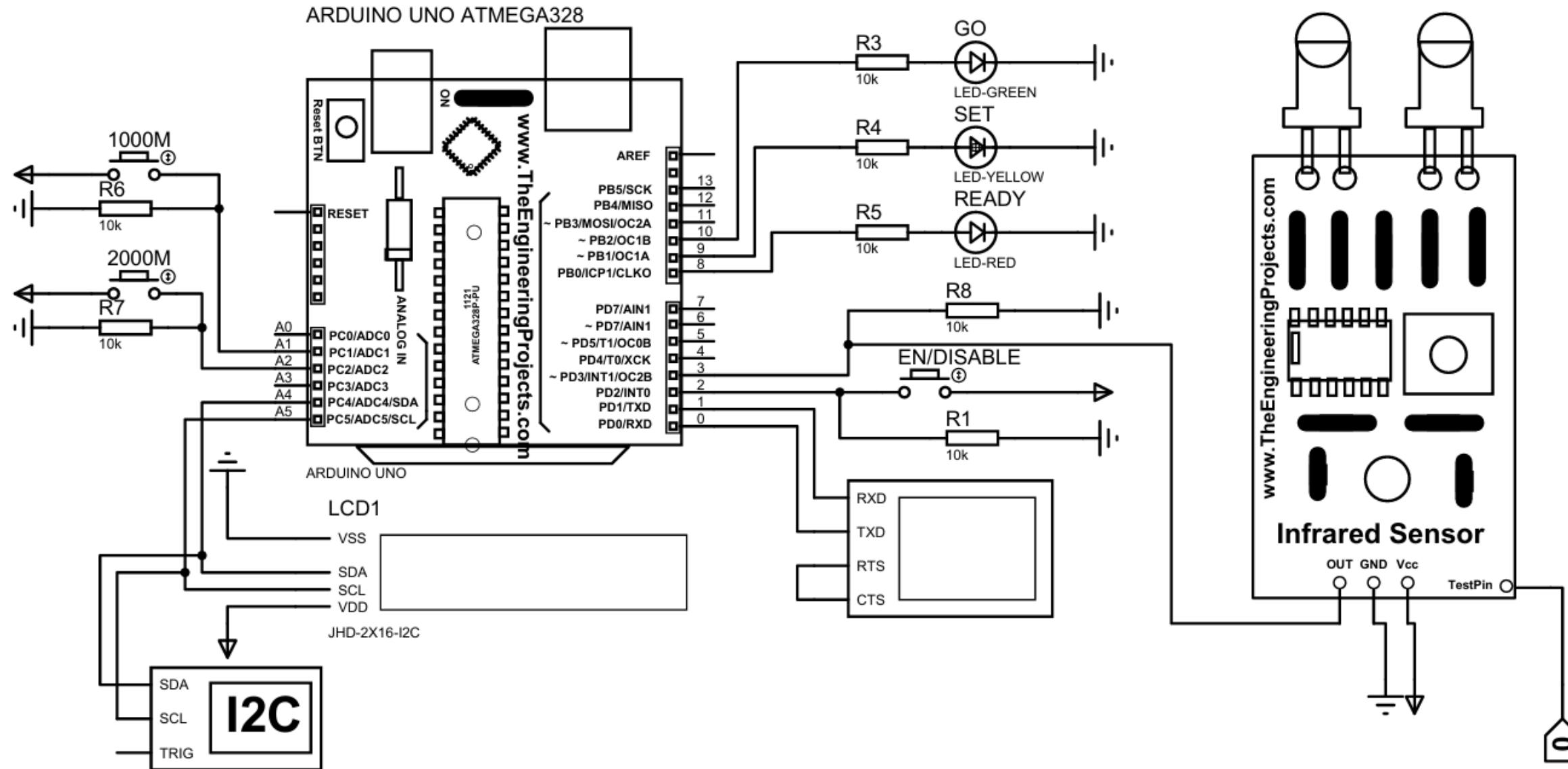
Embedded System Final Project G11

Anthonius Hendhy Wirawan

Javana Muhammad Dzaki

Maxwell Zefanya Ginting

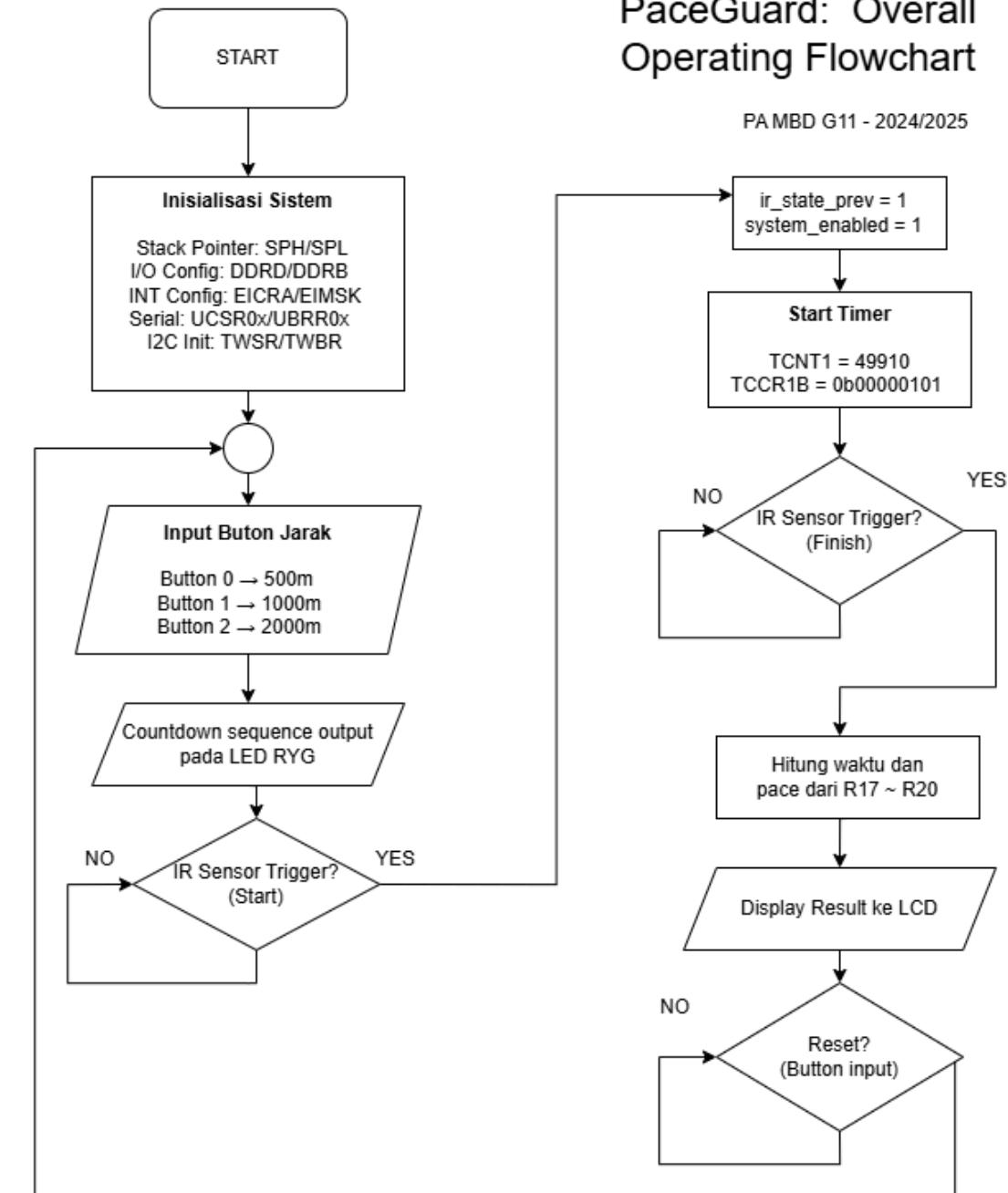
Ruben Kristanto



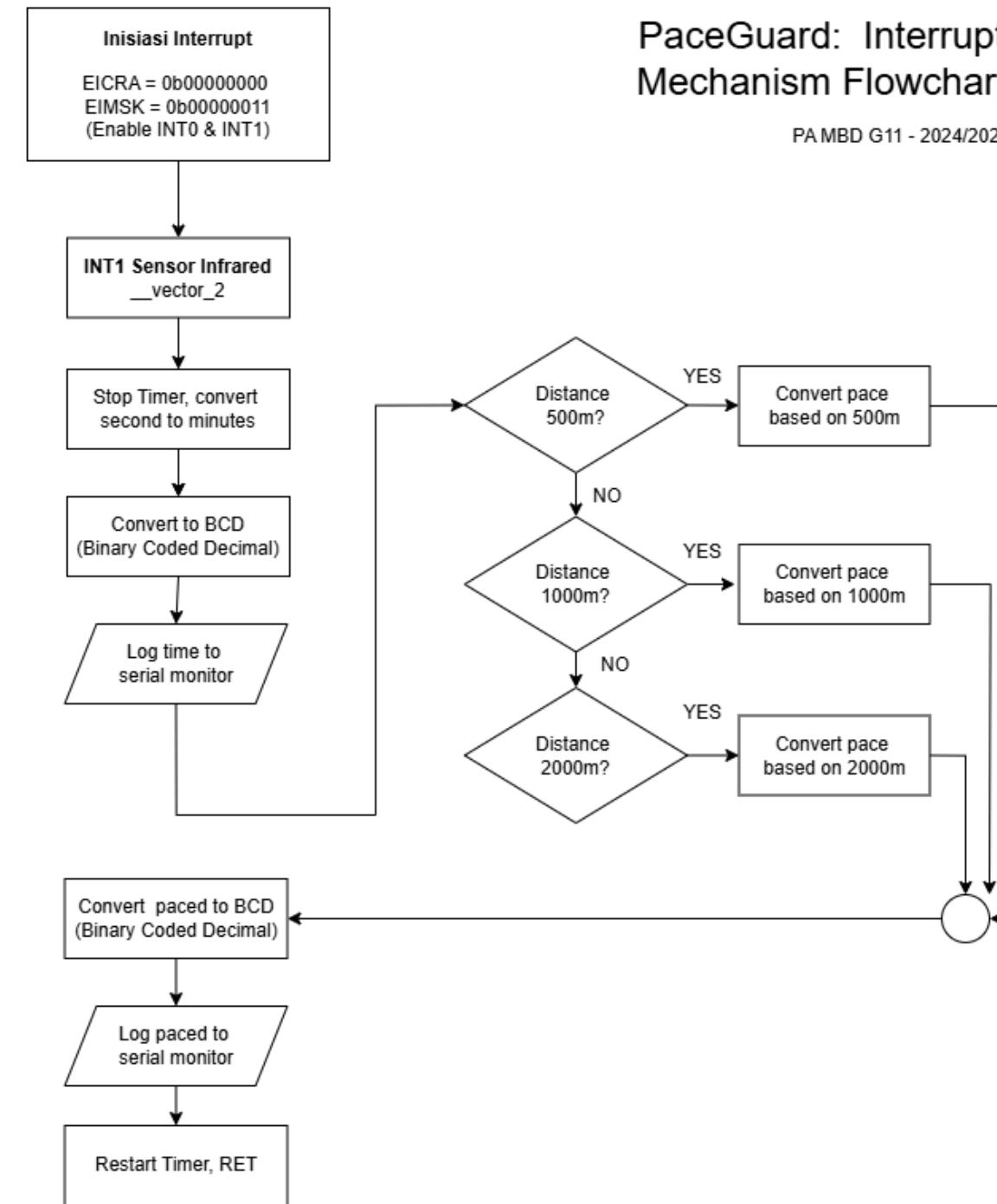
Komponen Hardware

Komponen	Jumlah
Arduino UNO R3 (ATMega328P)	1x
KY-032 Infrared Sensor	1x
16x2 LCD with I2C adapter	1x
LED RYG	3x (1 Each)
Push Button	3x
Resistor	7x
Breadboard Small	1x
Jumper	Secukupnya

Overall Operation



Mekanisme Interrupt



```
l      _start
on
.on .text
    rax, 1
    rdi, 1
    rsi, message
    rdx, 13
11
    rax, 60
    rdi, rdi
11
on .data
    "Hello, Wo
```

Implementasi AVR Program

AVR: Inisialisasi Sistem

```
; Code Segment
.section .text
main:
    ; Initialize Stack Pointer
    ldi r18, hi8(RAMEND)
    STS SPH, r18
    ldi r18, lo8(RAMEND)
    STS SPL, r18

    ; Digital: All as i/p
    LDI    R16, (0<<BTN_ENA) || (0<<BTN_RES)
    OUT   DDRD, R16

    ; LED: All as o/p
    LDI    R16, (1<<LED_RED) || (1<<LED_YEL) || (1<<LED_GRE)
    OUT   DDRB, R16

    ; Sensor: All as i/p
    LDI    R16, (0<<SENSOR_PIN)
    OUT   DDRD, R16

    ; Timer: Clear register
    CLR   R26
    CLR   R27

    ; All: Enable interrupt
    LDI R16, (0<<ISC11) || (0<<ISC10) || (0<<ISC01) || (0<<ISC00)
        STS EICRA, R16      ;Interrupts on low
    LDI R16, (1<<INT0) || (1<<INT1)
        OUT EIMSK, R16      ;Enable INT0 and INT1
```

AVR: Mengukur Waktu

```
loop:  
    RCALL start_timer  
    ADIW X, 1 ; Add 1 to R27:R26  
    BRTC loop  
  
hitung:  
    RCALL convert_to_minute  
    RCALL convert_to_BCD ;ubah detik ke bentuk BCD  
    MOV R18, R17  
    RCALL convert_to_BCD ;ubah menit ke bentuk BCD  
    //tampilkan waktu per lab ke serial monitor  
    LDI R30, lo8(message_time)  
    LDI R31, hi8(message_time)
```

AVR:

Menghitung

Pace

```
;=====;  
;~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ;  
convert_disctance:  
    CPI R16, 0  
    BREQ dis_500m  
    CPI R16, 1  
    BREQ dis_1000m  
    CPI R16, 2  
    BREQ dis_2000m  
;~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ;
```

AVR: Interrupt ISR INT1

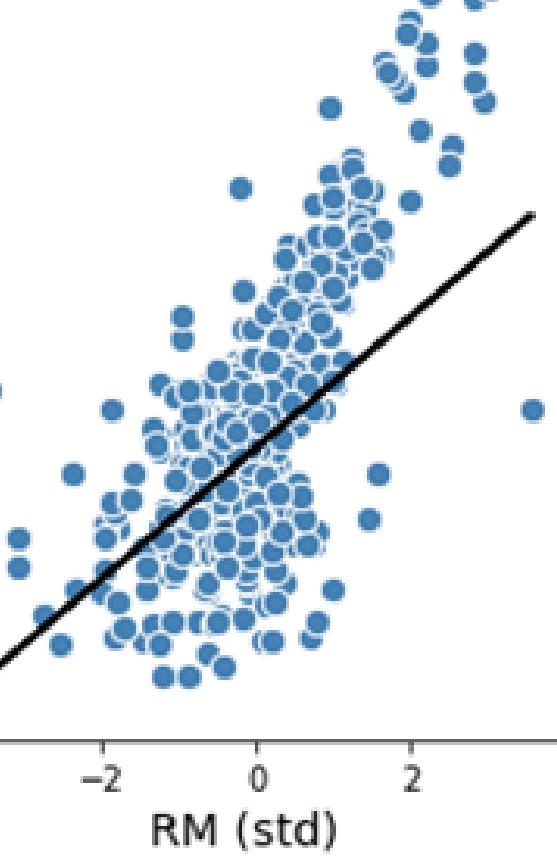
```
; Interrupt for sensor
; Purpose: stop timer
_vector_2:
    ;Disable clock
    LDI R28, 0b00000000
    STS TCCR1B, R28
    LDI R28, 1<<TOV1

    ;Clear TOV1 flag
    OUT TIFR1, R28
    LDI R28, 0b00000000
    SET

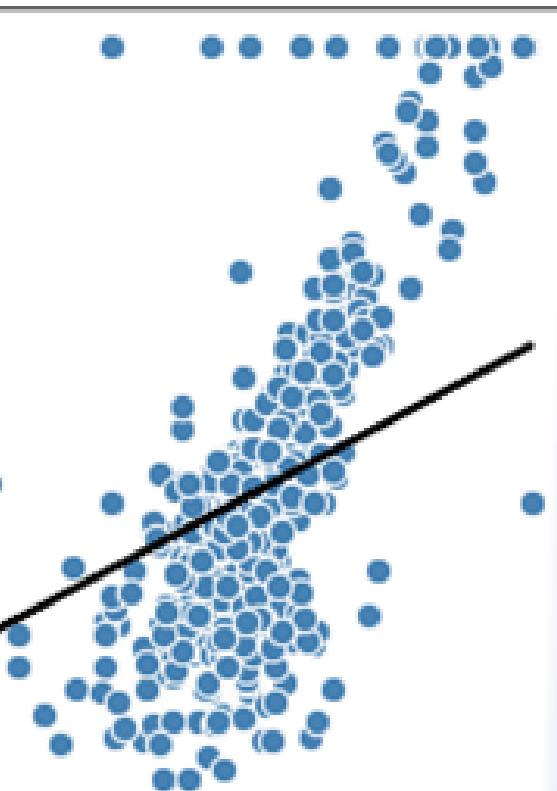
RETI
```

AVR: I2C & LCD

```
lcd_print:  
    RCALL I2C_start  
    LDI R22, 0x20  
    RCALL I2C_write  
  
    LDI R22, 0b10000000 ; Set slave address  
  
    Initialization  
    RCALL I2C_write  
    LDI R22, 0b00111100  
    RCALL I2C_write  
    CALL delay_4200us  
  
    LDI R22, 0b10000000 ;  
    Initialization  
    RCALL I2C_write  
    LDI R22, 0b00111100  
    RCALL I2C_write  
    CALL delay_120us  
  
    LDI R22, 0b10000000 ; Step 1: Function set  
    RCALL I2C_write  
    LDI R22, 0b00110000 ; 1-line mode,  
    display ON  
    RCALL I2C_write  
  
    LDI R22, 0b10000000 ; Step 2: Display ON  
    RCALL I2C_write  
    LDI R22, 0b00001100 ; Display ON, cursor  
    ON, blink OFF  
    RCALL I2C_write
```

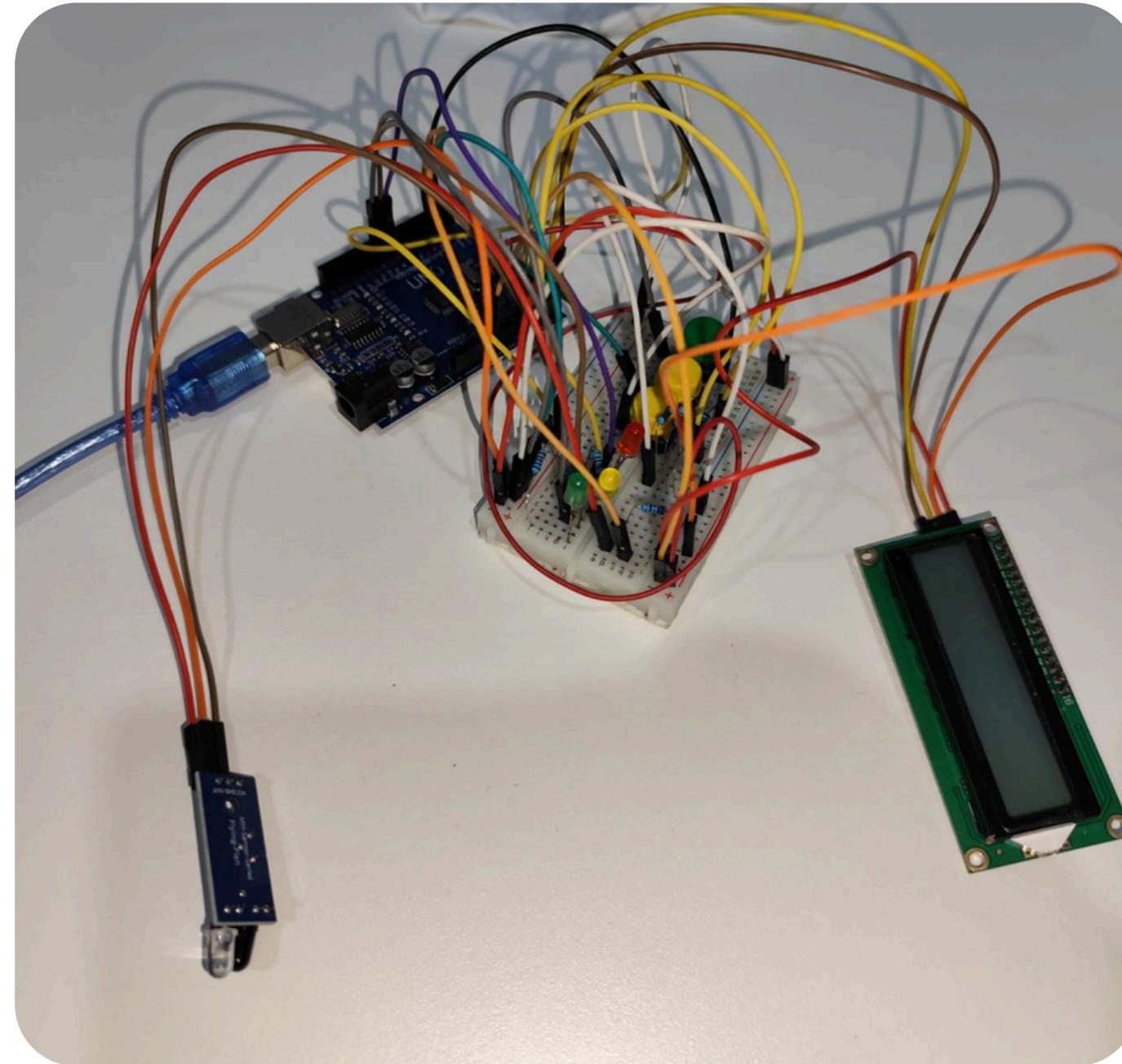


$$\eta = 10^{-5}, R^2 = 0.308$$

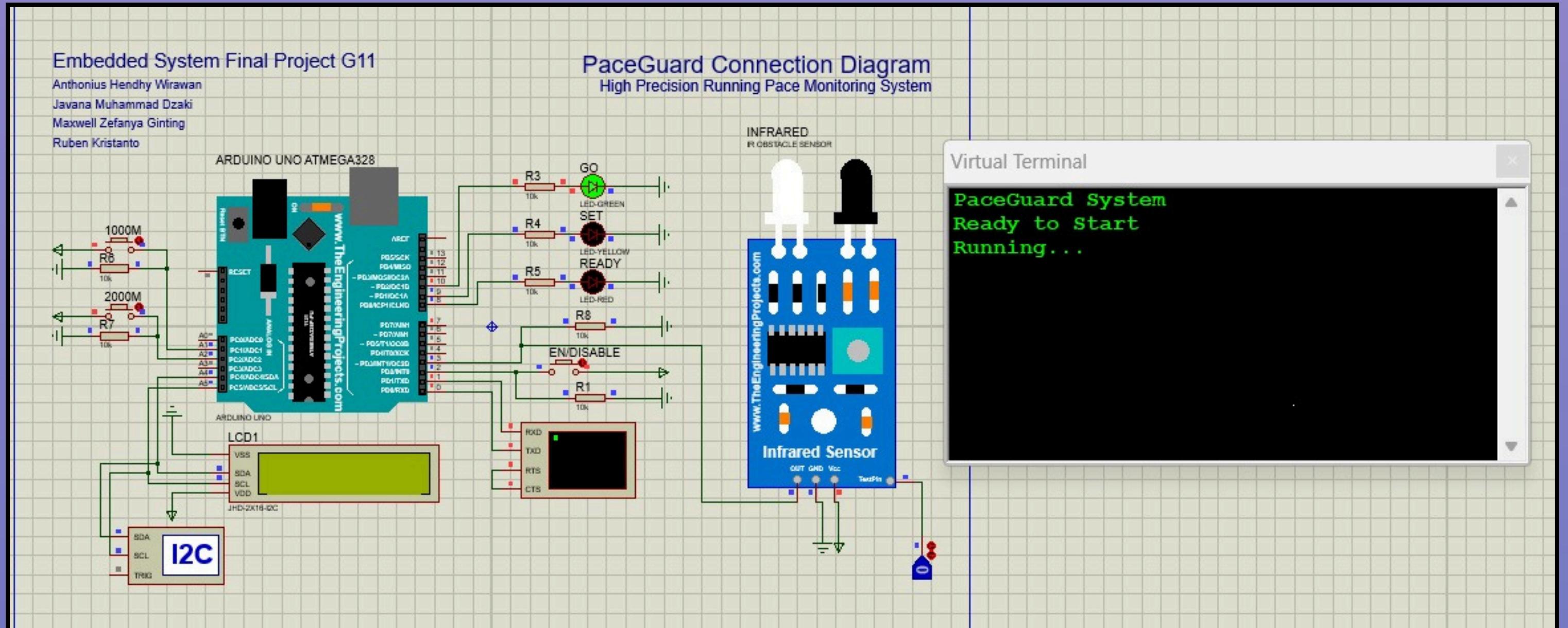


Testing dan Analysis

Rangkaian Fisik



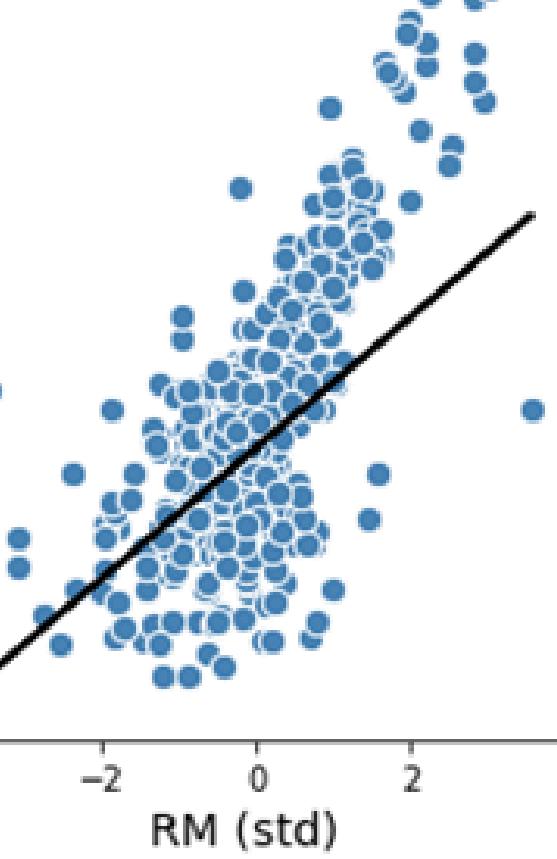
Hasil Pengujian



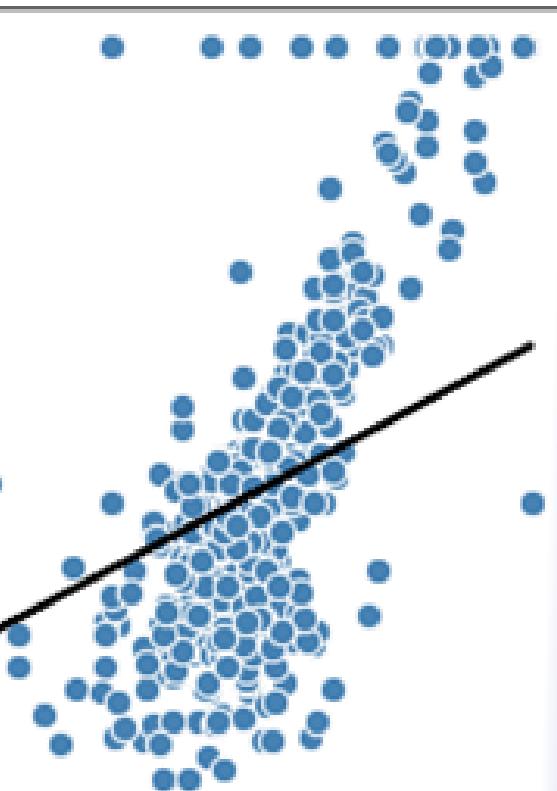
Analisis Performa

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



$$\eta = 10^{-5}, R^2 = 0.308$$



Penutup

Kesimpulan

1

PaceGuard berhasil memampukan pelari untuk mengatur pace sesuai dengan kebutuhannya serta menampilkan data lap yang akurat

2

Akurasi dan keresponsifan PaceGuard dapat dicapai berkat penggunaan interrupt dan kode assembly yang efisien

3

Penggunaan LCD dan Serial display bisa memungkinkan pelari untuk melihat informasi yang penting mengenai kegiatan larinya

Saran

Penggunaan assembly ketimbang C bisa membuat sistem lebih responsif

Sistem PaceGuard bisa dibuat menjadi lebih hemat daya dan tempat bila menggunakan IC ketimbang rangkaian Arduino + Breadboard

Referensi

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

Thank You!

PACEGUARD

Where Miliseconds Matter.

