CYBER-PHYSICAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

Fire Alarm

GROUP A8

Laode Alif Ma'sum Sidrajat Raja Ika	**2106731213**
Shabrina Kamiliya Wiyana	**2106733894**
Leonardo Jeremy Pong Pare Munda	**2106707914**
Fatima Khairunnisa	**2106651515**

PREFACE

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena karunia dan rahmat-Nya sehingga kami, kelompok A8 dapat menyelesaikan proyek akhir praktikum Perancangan Sistem Digital dengan judul "*Fire Alarm*" tepat pada waktunya. Kami kelompok A8 berusaha semaksimal mungkin agar laporan yang kami buat dapat bermanfaat dalam bidang keamanan dan pertahanan.

Adapun tujuan dari penulisan laporan ini, yakni untuk memenuhi tugas akhir praktikum Perancangan Sistem Digital (PSD). Kami ingin mengucapkan terima kasih kepada Bapak Dr. lupa nma panjangnya siapa, ST. MT yang telah membimbing mata kuliah Perancangan Sistem Digital, serta para asisten laboratorium digital yang telah membimbing kami selama praktikum hingga proyek akhir ini. Kami juga mengucapkan terima kasih kepada Kak Raihan yang telah membantu kami dalam mengerjakan proyek akhir ini.

Kami menyadari laporan yang kami susun ini masih jauh dari kata sempurna. Oleh karena itu, kritik serta saran kami butuhkan untuk menyempurnakan laporan ini.

Depok, December 16, 2023

Group A8

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Faktanya, sistem peringatan kebakaran telah ada sejak lama, namun masih kurang memadai dalam penanganannya. Saat ini, sudah semestinya kita tidak lagi mengandalkan cara penanganan kebakaran konvensional. Oleh karena itu, perlu ada suatu perangkat yang dapat mendeteksi adanya suhu panas akibat kebakaran lebih awal, dan saat ini banyak dikembangkan dengan sistem peringatan yang berbeda. Penggunaan cara penanganan yang lama atau konvensional akan sangat sulit untuk menemukan sumber kebakaran, sehingga sudah seharusnya kita beralih ke cara penanganan yang lebih modern.

Tujuan utama dari teknologi adalah untuk memberikan kemudahan bagi manusia dalam melakukan pekerjaan. Dalam hal ini, teknologi dapat membantu pengguna dalam mencegah terjadinya kebakaran dan memberikan informasi yang jelas tentang lokasi terdeteksi melalui suhu api yang panas serta sensor alarm dan air akan keluar secara otomatis untuk memadamkan api.

Pada umumnya, alarm diartikan sebagai peringatan yang diberikan melalui bunyi, seperti suara sirene, bel, atau sejenisnya. Jika terjadi kebakaran di gedung bertingkat yang memiliki beberapa ruangan, seperti hotel, apartemen, kantor, dan bangunan lainnya, diperlukan perangkat yang dapat secara otomatis mendeteksi suhu api kebakaran yang akan memadamkan api tersebut melalui air yang dikeluarkan. Dalam tugas akhir ini, kelompok A8 membuat *fire alarm* sebagai alternatif yang lebih praktis.

1.2 PROPOSED SOLUTION

Fire alarm merupakan alat yang digunakan untuk mendeteksi adanya kebakaran dengan menggunakan sensor suhu dan kelembaban, yaitu DHT11. Sensor tersebut membaca suhu ruangan secara terus-menerus untuk mendeteksi adanya kenaikan suhu yang signifikan, yang menjadi tanda awal terjadinya kebakaran. Ketika terindikasi adanya kebakaran, maka sistem akan memberikan peringatan melalui lampu yang berkedip sebagai warning lights dan buzzer yang menyala sebagai peringatan suara. Selain memberikan peringatan, alat ini juga

dilengkapi dengan sistem pemadam kebakaran otomatis. Alat akan menyalakan air menggunakan servo untuk memadamkan api.

Pada sistem ini, alarm berperan sebagai slave device yang menerima perintah dari sensor yang berperan sebagai master device. Sensor akan memberikan informasi mengenai kenaikan suhu dan indikasi kebakaran ke alarm, dan alarm akan memberikan peringatan dan mengaktifkan sistem pemadam kebakaran secara otomatis. Hal ini memungkinkan untuk tindakan pencegahan yang lebih cepat dan efektif dalam menanggulangi kebakaran.

Penggunaan sensor suhu dan kelembaban DHT11 sebagai detektor kebakaran merupakan pilihan yang tepat karena sensor tersebut dapat membaca perubahan suhu dan kelembaban secara akurat dan responsif. Sensor ini memiliki tingkat keakuratan yang tinggi dan memungkinkan untuk mendeteksi adanya kebakaran dengan cepat dan efektif.

Selain itu, adanya fitur pemadam kebakaran otomatis juga sangat penting dalam menjaga keselamatan dan mencegah kerugian akibat kebakaran. Dalam situasi kebakaran, waktu sangat krusial dan setiap detik sangat berharga. Dengan adanya sistem pemadam kebakaran otomatis, tindakan pemadaman api dapat dilakukan secara cepat dan efektif, sehingga kerugian yang ditimbulkan dapat diminimalkan.

Secara keseluruhan, Fire alarm yang dilengkapi dengan sensor suhu dan kelembaban DHT11 serta sistem pemadam kebakaran otomatis merupakan solusi yang tepat dalam mengatasi bahaya kebakaran. Alat ini sangat efektif dalam mendeteksi kebakaran secara dini dan memberikan peringatan yang cepat, serta mampu mengaktifkan sistem pemadam kebakaran secara otomatis untuk mengurangi risiko dan kerugian akibat kebakaran.

1.3 ACCEPTANCE CRITERIA

Kriteria penerimaan proyek ini adalah sebagai berikut:

1. Membuat suatu rangkaian yang dilengkapi dengan sensor DHT11, buzzer, dan servo.
2. Melakukan integrasi antara sensor dan Arduino untuk memungkinkan program berjalan dan dapat mengaktifkan LED serta Buzzer.
3. Menggunakan mikrokontroler Arduino sebagai sistem siber-fisik untuk merancang alat deteksi kebakaran.
4. Membuat penyalur air yang dilengkapi dengan servo untuk memadamkan api.

5. Memberikan pemberitahuan kepada pengguna ketika sensor berhasil mendeteksi suhu tinggi yang mencurigakan kebakaran melalui penanda visual berupa lampu yang berkedip sebagai warning light, dan peringatan suara berupa buzzer yang menyala untuk memberikan peringatan.

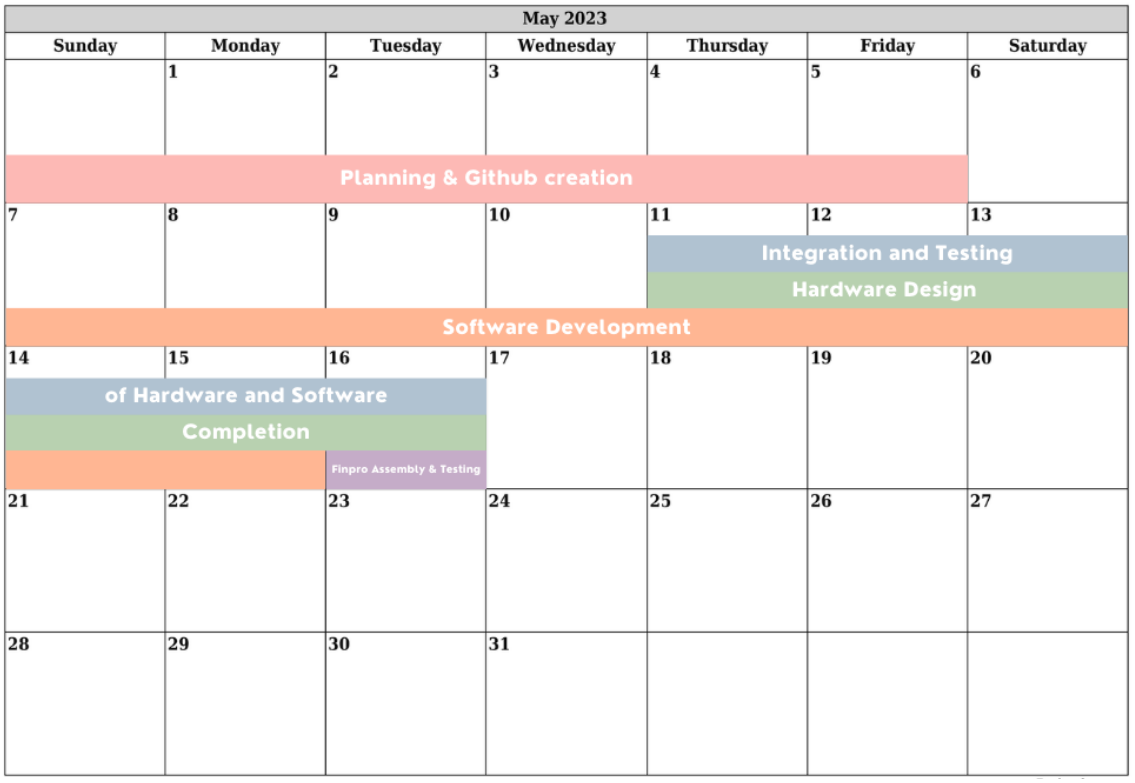
1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Membantu membuat kode program untuk membaca data sensor	Role 1 responsibilities	Nisa
Membuat keseluruhan kerangka code, membuat code untuk menyalakan LED serta buzzer dan membuat prototype rangkaian proteus	Role 2 responsibilities	Alif
Membantu membuat kode program untuk menggerakan DC Water Pump untuk mematikan sumber api	Role 3 responsibilities	Shabrina
Membantu membuat code untuk menyambungkan master device dan slave device	Role 4 responsibilities	Leonardo
Membuat laporan	Role 5 responsibilities	Semua

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES



Insert Gantt Chart here. The Gantt Chart should consist of date interval for:

a) Hardware Design completion: 11 Mei - 16 Mei 2023

b) Software Development: 7 Mei - 15 Mei 2023

c) Integration and Testing of Hardware and Software: 11 Mei - 16 Mei 2023

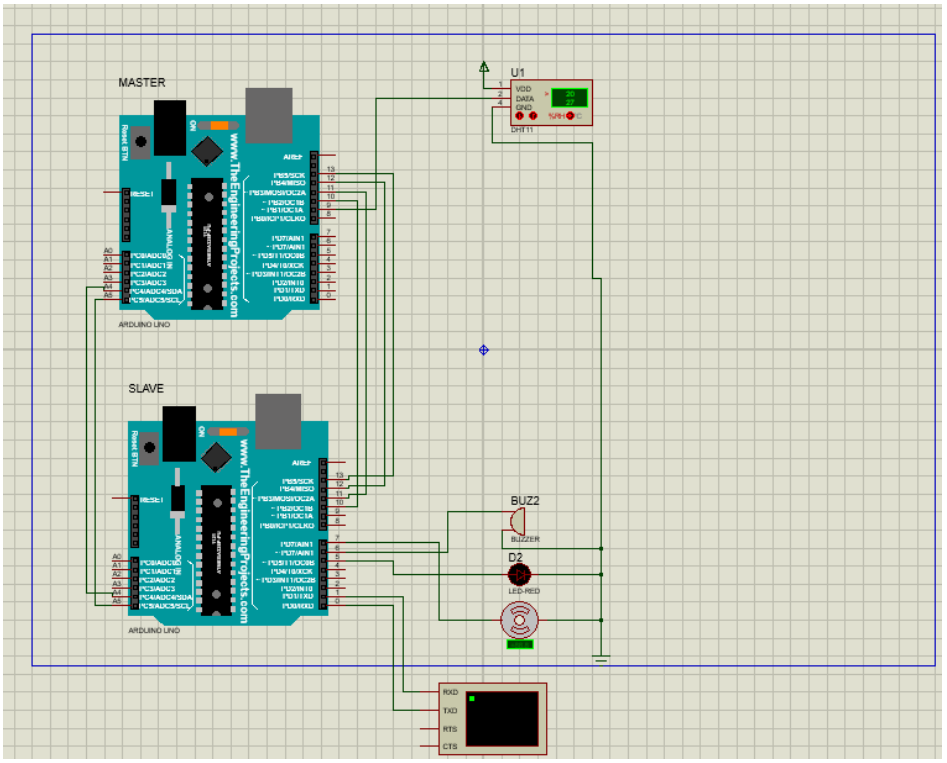
d) Final Product Assembly and Testing:16 Mei 2023

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

Pembuatan alat dilakukan dengan perancangan skematik melalui proteus, selanjutnya setelah prototype berhasil bekerja sesuai ketentuan, dilakukan perancangan secara langsung. Langkah ini dilakukan dalam rangka untuk mencegah kerusakan pada rangkaian asli.



Gbr 1. Skematik Rangkaian *Fire Alarm*

Alat yang dibutuhkan untuk membuat *Fire Alarm* diantaranya adalah :

a. Arduino

Arduino merupakan microcontroller yang digunakan sebagai platform yang menghubungkan hardware dan software. Arduino ini di program dengan menggunakan bahasa assembly untuk membaca sensor DHT11 yang akan mendeteksi temperatur panas, lalu data yang didapatkan dari sensor akan dikirimkan ke arduino yang berperan sebagai slave, dimana pada arduino slave

ini akan mengambil keputusan berdasarkan kondisi yang dispesifikasikan, seperti menyalakan buzzer dan menyalakan dinamo untuk mematikan api.

b. LED

LED pada perangkat ini digunakan sebagai alarm, yang akan memberikan peringatan bahwa telah terjadi peningkatan suhu pada ruangan. Pada keadaan ini LED akan blinking sebagai peringatan untuk mengevakuasi diri dari ruangan.

c. Buzzer

Buzzer digunakan sebagai alarm yang berfungsi untuk memberikan informasi berupa sound yang terdengar, sebagai peringatan kepada penghuni ruangan untuk segera mengevakuasi diri.

d. DHT11

DHT11 merupakan sensor suhu dan kelembaban, sensor ini berfungsi untuk mendapatkan data dari lingkungan lalu kemudian mengirimkan datanya ke arduino untuk pengambilan keputusan berdasarkan kondisi yang telah dispesifikasikan.

e. Breadboard

Pada praktikum ini kami menggunakan breadboard berukuran kecil, sehingga dapat masuk kedalam rangkaian. Breadboard ini berfungsi sebagai tempat untuk merangkai perangkat, dan menyatukannya tanpa perlu menggunakan solder.

f. Jumper

Jumper merupakan kabel atau penghubung yang digunakan untuk menghubungkan bagian - bagian perangkat electronic. Jumper ini menghubungkan perangkat agar aliran elektronik dapat mengalir ke tempat semestinya.

g. Power Supply

Power supply pada rangkaian ini menggunakan powerbank, yang berfungsi untuk pemberi energi elektrik lebih tepatnya ke arduino, sehingga arduino dapat menjalankan kerjanya sebagai otak dari perangkat.

h. DC Water Pump

DC Water Pump pada perangkat ini digunakan untuk menarik dan memindahkan air dari sumber penyimpanan, dan nantinya akan dikirimkan melalui selang ke tempat api berada.

i. Button

Komponen Button berfungsi sebagai interrupt yang akan mematikan proses pada alat.

j. Selang

Selang pada *Fire Alarm* ini berfungsi untuk mengalirkan air yang ditarik melalui DC Water Pump ke tempat api berada.

2.2 SOFTWARE DEVELOPMENT

Master's Code

```
;-----------------------
; MASTER CODE
;-----------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"


.global main
.global again



main:
```

```
;----------------------------------------------------------------

       RCALL I2C_init      ;initialize TWI module

;----------------------------------------------------------------

       LDI R26, 0xA0       ;inital byte to be transmitted (on LED
in slave PD)

       RCALL init_serial


     again:

;----------------------------------------------------------------

       ;Sensor Read

;----------------------------------------------------------------

       SBI DDRB, 1 ;pin PB0 as o/p

       CBI PORTB, 1 ;first, send low pulse

       RCALL delay_20ms ;for 20ms

       SBI PORTB, 1 ;then send high pulse

;----------------------------------------------------------------

       ;wait for response signal

       ;---------------

       CBI DDRB, 1 ;pin PB0 as i/p

       w1: SBIC PINB, 1

       RJMP w1 ;wait for DHT11 low pulse

       w2: SBIS PINB, 1

       RJMP w2 ;wait for DHT11 high pulse
```

```
        w3: SBIC PINB, 1
        RJMP w3 ;wait for DHT11 low pulse
      ;-------------------------------------------------------------
-----
         RCALL  DHT11_reading  ;read  humidity  (1st  byte  of  40-bit
data)
        MOV R19, R18
        RCALL DHT11_reading
        RCALL DHT11_reading ;read temp (3rd byte of 40-bit data)
      ;-------------------------------------------------------------
-----
      ;-----------------------------------------------------
      ; I2C for Sensor Read Result
      ;-----------------------------------------------------
      sendResult:
        RCALL I2C_start     ;transmit START condition
        LDI R27, 0b10010000 ;SLA(1001000) + W(0)
        RCALL I2C_write     ;write slave address SLA+W
        MOV R26, R18
        RCALL I2C_write2    ;write data byte
        RCALL I2C_stop      ;transmit STOP condition

        ;-----------------------------------------------------

        RCALL delay_50usec    ;delay 50usec second
        RJMP again          ;repeat transmission
```

```
;============================================================

     delay_timer0: ;50 usec delay via Timer 0
       ;-----------------------------------------------------
       CLR R20
       OUT TCNT0, R20 ;initialize timer0 with count=0
       LDI R20, 100
       OUT OCR0A, R20 ;OCR0 = 100
       LDI R20, 0b00001010
       OUT TCCR0B, R20 ;timer0: CTC mode, prescaler 8
       ;-----------------------------------------------------
       lo2: IN R20, TIFR0 ;get TIFR0 byte & check
       SBRS R20, OCF0A ;if OCF0=1, skip next instruction
       RJMP lo2 ;else, loop back & check OCF0 flag
       ;-----------------------------------------------------
       CLR R20
       OUT TCCR0B, R20 ;stop timer0
       ;-----------------------------------------------------
       LDI R20, (1<<OCF0A)
       OUT TIFR0, R20 ;clear OCF0 flag
       RET

     delay_50usec: ;50 usec delay via timer1
       ;-----------------------------------------------------
       CLR R20
```

```
  OUT TCNT0, R20 ;initialize timer0 with count=0
  LDI R20, 100
  OUT OCR0A, R20 ;OCR0 = 100
  LDI R20, 0b00001010
  OUT TCCR0B, R20 ;timer0: CTC mode, prescaler 8
  ;-----------------------------------------------------------
  LOOP2: IN R20, TIFR0 ;get TIFR0 byte & check
  SBRS R20, OCF0A ;if OCF0=1, skip next instruction
  RJMP LOOP2 ;else, loop back & check OCF0 flag
  ;-----------------------------------------------------------
  CLR R20
  OUT TCCR0B, R20 ;stop timer0
  ;-----------------------------------------------------------
  LDI R20, (1<<OCF0A)
  OUT TIFR0, R20 ;clear OCF0 flag
  RET

I2C_init:
  LDI R21, 0
  STS TWSR, R21   ;prescaler 1
  LDI R21, 35     ;division factor = 35
  STS TWBR, R21   ;
  LDI R21, (1<<TWEN)
  STS TWCR, R21   ;enable TWI
  RET
```

```
I2C_start:
  LDI R21, (1<<TWINT)|(1<<TWSTA)|(1<<TWEN)
  STS TWCR, R21         ;transmit START condition
  wt1:
    LDS R21, TWCR
    SBRS R21, TWINT     ;TWI interrupt = 1?
    RJMP wt1            ;no, wait for end of transmission
  RET

I2C_write:
  STS TWDR, R27         ;copy SLA+W into data register
  LDI R21, (1<<TWINT)|(1<<TWEN)
  STS TWCR, R21         ;transmit SLA+W
  wt2:
    LDS R21, TWCR
    SBRS R21, TWINT
    RJMP wt2            ;wait for end of transmission
  RET

I2C_write2:
  STS TWDR, R26         ;copy data into data register
  LDI R20, (1<<TWINT)|(1<<TWEN)
  STS TWCR, R20         ;transmit data
  wt22:
    LDS R20, TWCR
    SBRS R20, TWINT
```

```
    RJMP wt22            ;wait for end of transmission
  RET

I2C_stop:
  LDI R21, (1<<TWINT)|(1<<TWSTO)|(1<<TWEN)
  STS TWCR, R21          ;transmit STOP condition
  RET

DHT11_reading:
  LDI R17, 8 ;set counter for receiving 8 bits
  CLR R18 ;clear data register
;-----------------------------------------------------
w4:
  SBIS PINB, 1
  RJMP w4 ;detect data bit (high pulse)
  RCALL delay_timer0 ;wait 50us & then check bit value
  ;-----------------------------------------------------
  SBIS PINB, 1 ;if received bit=1, skip next inst
  RJMP skp ;else, received bit=0, jump to skp
  SEC ;set carry flag (C=1)
  ROL R18 ;shift in 1 into LSB data register
  RJMP w5 ;jump & wait for low pulse
  skp:LSL R18 ;shift in 0 into LSB data register
  ;-----------------------------------------------------
  w5: SBIC PINB, 1
  RJMP w5 ;wait for DHT11 low pulse
```

```
      ;-----------------------------------------------------
      DEC R17 ;decrement counter
      BRNE w4 ;go back & detect next bit
      RET ;return to calling subroutine
    ;=============================================================
=====
    ;delay subroutines
    ;=============================================================
=====
    delay_20ms: ;delay 20ms
      LDI R21, 255
      l3: LDI R22, 210
      l4: LDI R23, 2
      l5: DEC R23
      BRNE l5
      DEC R22
      BRNE l4
      DEC R21
      BRNE l3
      RET



    init_serial:
      CLR R24
      STS UCSR0A, R24   ; clear UCSR0A register
      STS UBRR0H, R24   ; clear UBRR0H register
      LDI R24, 103      ; store in UBRR0L 103 value
```

```
        STS UBRR0L, R24   ; to set baud rate 9600
        LDI R24, 1<<RXEN0 | 1<<TXEN0   ;enable RXB & TXB
        STS UCSR0B, R24
        LDI R24, 1<<UCSZ00 | 1<<UCSZ01 ;asynch, no parity, 1 stop,
8 bits
        STS UCSR0C, R24
        RET

      LCD_buffer:
        LDS R28, UCSR0A
        SBRS R28, UDRE0 ;test data buffer if data can be sent
        RJMP LCD_buffer
        RET

      ASCII_MSD: ;Proses mendapatkan ASCII dari MSD
        MOV R23, R16 ;save copy of result
        ANDI R16, 0xF0 ;extract & swap high-nibble
        SWAP R16
        SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
        MOV R28, R16 ;save a copy of high-byte result
        SUBI R28, 58 ;if +ve
        BRPL A_F_D1 ;branch & add 7 to get ASCII A to F
      lo1:
        RET

      ASCII_LSD: ;Proses mendapatkan ASCII dari LSD
```

```
  MOV R16, R23 ;restore copy of result
  ANDI R16, 0x0F ;extract low-nibble
  SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
  MOV R28, R16 ;save a copy of high-byte result
  SUBI R28, 58 ;if +ve
  BRPL A_F_D0 ;branch & add 7 to get ASCII A to F
loop2:
  RET

A_F_D1:
  SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
  RJMP lo1
A_F_D0:
  SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
  RJMP loop2

delay_sec: ;1s delay
  LDI R20, 255
loop4:
  LDI R21, 255
loop5:
LDI R22, 82
loop6:
  DEC R22
  BRNE loop6
  DEC R21
```

```
        BRNE loop5
        DEC R20
        BRNE loop4
        RET
```

Slave's Code

Slave Code

```
;-----------------------
; SLAVE CODE
;-----------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"

.global main

main:
  RCALL init_serial
  LDI R21, 0xF0
  OUT DDRD, R21 ;port D[7:4] is o/p

again:
  ;-------------------------------------------------------
  ; I2C read
  ;-------------------------------------------------------
  RCALL I2C_init
  RCALL I2C_listen
  RCALL I2C_read
  MOV R16, R27
  RCALL checkTemp

printTemperature:
  ;-------------------------------------------------------
```

```
  ; Print to serial monitor
  ;-----------------------------------------------------
  RCALL ASCII_MSD    ; Mendapatkan ASCII dari MSD
  RCALL LCD_buffer
  STS UDR0, R16

  RCALL ASCII_LSD    ;Mendapatkan ASCII dari LSD
  RCALL LCD_buffer   ;Subroutine untuk mengecek FLAG UDRE0
  STS UDR0, R16      ;print LSD

  LDI R16, '|'       ;
  RCALL LCD_buffer   ;Subroutine untuk mengecek FLAG UDRE0
  STS UDR0, R16

  RJMP again


checkTemp:
  LDI R30,0x30
  CP R27, R30
  BRSH fireDetected
  BRLT noFire
  RET

fireDetected:
  SBI PORTD, 6 ;turn on buzzer
  RCALL myDelay ;delay between led switch
  CBI PORTD, 5 ;turn on led
  RCALL myDelay ;delay between led switch
  SBI PORTD, 5 ;turn on led
  SBI PORTD, 7 ;turn on dc water pump
  RET
;---------------------------------------------------------------
myDelay: ;3-level nested loop delay
```

```
  LDI R20, 255 ;outer loop counter
c1:
  LDI R21, 255 ;mid loop counter
c2:
  LDI R22, 20 ;inner loop counter
;1s delay = 255 * 255 * 82 * 3 = 15.996.150 cycle = 1 s
c3:
  DEC R22 ;decrement inner loop
  BRNE c3 ;Branch ke l3 jika R22 != 0
  DEC R21 ;decrement mid loop
  BRNE c2 ;Branch ke l2 jika R21 != 0
  DEC R20 ;decrement outer loop
  BRNE c1 ;`Branch ke l1 jika R20 != 0
  RET ;return

noFire:
  CBI PORTD, 6
  CBI PORTD, 5
  CBI PORTD, 7
  RCALL delay_sec
  RET

init_serial:
  CLR R24
  STS UCSR0A, R24   ; clear UCSR0A register
  STS UBRR0H, R24   ; clear UBRR0H register
  LDI R24, 103      ; store in UBRR0L 103 value
  STS UBRR0L, R24   ; to set baud rate 9600
  LDI R24, 1<<RXEN0 | 1<<TXEN0   ;enable RXB & TXB
  STS UCSR0B, R24
   LDI  R24,  1<<UCSZ00  |  1<<UCSZ01  ;asynch,  no  parity,  1  stop,  8
bits
  STS UCSR0C, R24
  RET
```

```
LCD_buffer:
  LDS R28, UCSR0A
  SBRS R28, UDRE0 ;test data buffer if data can be sent
  RJMP LCD_buffer
  RET

ASCII_MSD: ;Proses mendapatkan ASCII dari MSD
  MOV R23, R16 ;save copy of result
  ANDI R16, 0xF0 ;extract & swap high-nibble
  SWAP R16
  SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
  MOV R28, R16 ;save a copy of high-byte result
  SUBI R28, 58 ;if +ve
  BRPL A_F_D1 ;branch & add 7 to get ASCII A to F
lo1:
  RET

ASCII_LSD: ;Proses mendapatkan ASCII dari LSD
  MOV R16, R23 ;restore copy of result
  ANDI R16, 0x0F ;extract low-nibble
  SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
  MOV R28, R16 ;save a copy of high-byte result
  SUBI R28, 58 ;if +ve
  BRPL A_F_D0 ;branch & add 7 to get ASCII A to F
lo2:
  RET

A_F_D1:
  SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
  RJMP lo1
A_F_D0:
  SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
  RJMP lo2
```

```
delay_sec: ;1s delay
  LDI R20, 255
  l4: LDI R21, 255
  l5: LDI R22, 82
  l6: DEC R22
  BRNE l6
  DEC R21
  BRNE l5
  DEC R20
  BRNE l4
  RET

I2C_init:
  LDI R21, 0b10010000
  STS TWAR, R21 ;store slave address 0b10010000
  LDI R21, (1<<TWEN)
  STS TWCR, R21 ;enable TWI
  LDI R21, (1<<TWINT)|(1<<TWEN)|(1<<TWEA)
  STS TWCR, R21 ;enable TWI & ACK
  RET

I2C_listen:
  LDS R21, TWCR
  SBRS R21, TWINT
  RJMP I2C_listen ;wait for slave to be addressed
  RET

I2C_read:
  LDI R21, (1<<TWINT)|(1<<TWEA)|(1<<TWEN)
  STS TWCR, R21 ;enable TWI & ACK
  ;-------------------------------------------------------
  wt: LDS R21, TWCR
  SBRS R21, TWINT
```

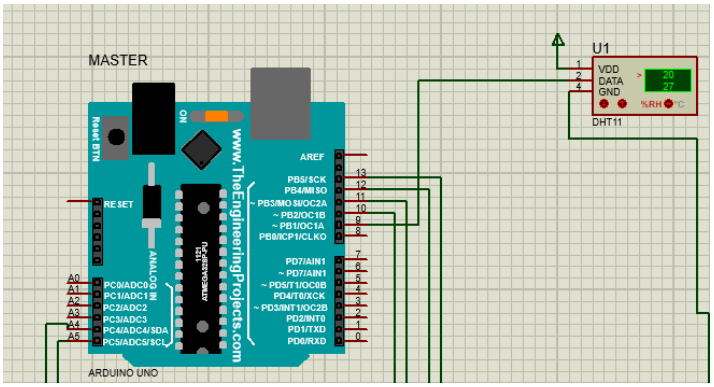
```
RJMP wt ;wait for data byte to be read
;-----------------------------------------------------------
LDS R27, TWDR ;store received byte
RET
```

Pada master code ini, arduino yang berperan sebagai master device berfungsi untuk menangkap sinyal dari lingkungan melalui sensor DHT11, lalu mengirimkan datanya ke slave device. Master device akan mengirimkan start condition sehingga I2C akan terhubung, dan memasukan slave addressnya. Master device ini juga menggunakan protokol SPI agar dapat berkomunikasi dengan slave device. Pada code yang dilampirkan tersebut terdiri dari berbagai macam function dan subroutine, yang digunakan untuk menginisialisasi protocol seperti I2C,dan SPI, lalu membaca data dari DHT11, mengatur DC motor, dan menangani komunikasi serial.

Master device mengirimkan data dan Slave device menerima data, data yang dikirimkan ini disinkronisasikan dengan menggunakan clock. Protokol I2C bekerja dengan mengirimkan data secara serial melalui jalur SDA yang disinkronisasi dengan sinyal clock pada jalur SCL, dimana sinyal clock ini dikendalikan oleh master untuk berkomunikasi pada slave. Master device akan mengirimkan alamat slave yang dituju beserta bit read/write setelah kondisi start, selanjutnya akan dilakukan pertukaran data frame yang ditentukan oleh bit read/write. Secara keseluruhan master device ini berfungsi untuk membaca data dari sensor lalu mengirimkan data yang telah dikonversikan menjadi digital ke slave device.

Pada code tersebut mengatur komunikasi serial pada slave device, yang mengkonversikan ASCII characters, menentukan delay, mengecheck temperature, mengontrol DC motor serta menggunakan procol I2C untuk komunikasi. Pada slave device ini terdapat sebuah metode **checkTemp** yang berfungsi untuk membandingkan nilai temperatur yang dideteksi oleh sensor DHT11, dengan temperatur yang threshold, lalu dengan metode **fireDetected** yang digunakan untuk melakukan task ketika api terdeteksi, yaitu akan menyalakan DC motor, dan apabila api tidak terdeteksi, dengan menggunakan metode **noFire** maka akan DC motor akan dimatikan.

2.3 HARDWARE AND SOFTWARE INTEGRATION



Gbr 2. Master Device

Pada Master device menggunakan PB1 untuk dihubungkan dengan port data pada DHT11, ketika MCU mengirimkan start signal MCU, DHT11 berubah dari low-power-consumption mode ke running-mode, Tanpa start signal dari MCU, DHT11 tidak akan memberikan signal response. setelah itu DHT11 menunggu MCU menyelesaikan signal, ketika telah selesai maka DHT akan merespon signal 40-bit data yang didalamnya berisikan informasi temperatur ke MCU. Proses pembacaan sensor ini dilakukan dengan potongan code dibawah

```
;Sensor Read

;-----------------------------------------------------------------

        SBI DDRB, 1 ;pin PB0 as o/p

        CBI PORTB, 1 ;first, send low pulse

        RCALL delay_20ms ;for 20ms

        SBI PORTB, 1 ;then send high pulse


;-----------------------------------------------------------------

        ;wait for response signal

        ;---------------

        CBI DDRB, 1 ;pin PB0 as i/p

        w1: SBIC PINB, 1

        RJMP w1 ;wait for DHT11 low pulse
```

```
        w2: SBIS PINB, 1
        RJMP w2 ;wait for DHT11 high pulse
        w3: SBIC PINB, 1
        RJMP w3 ;wait for DHT11 low pulse
     ;-----------------------------------------------------------
-----
         RCALL  DHT11_reading  ;read  humidity  (1st  byte  of  40-bit
data)
        MOV R19, R18
        RCALL DHT11_reading
        RCALL DHT11_reading ;read temp (3rd byte of 40-bit data)
     ;-----------------------------------------------------------
-----
        OUT PORTD, R18 ;o/p temp byte to port D
        MOV R16, R18
        RCALL LCD_buffer
        RCALL ASCII_MSD
        STS UDR0, R16
        RCALL LCD_buffer
        RCALL ASCII_LSD
        STS UDR0, R16
     ;-----------------------------------------------------------
     ; I2C for Button
     ;-----------------------------------------------------------
     RCALL I2C_start     ;transmit START condition
     LDI R27, 0b10010000 ;SLA(1001000) + W(0)
     RCALL I2C_write     ;write slave address SLA+W
```

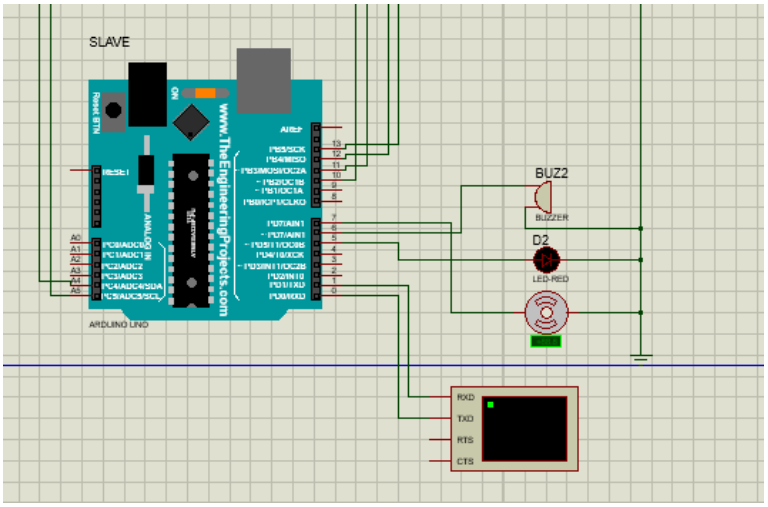
```
RCALL I2C_write2     ;write data byte

RCALL I2C_stop       ;transmit STOP condition
```

Code dibawah ini berfungsi untuk mengaktifkan protokol SPI yang nantinya akan digunakan untuk menghubungkan master device dengn slave device.

```
LDI R17, (1<<MOSI)|(1<<SCK)|(1<<SS)

  OUT DDRB, R17        ; set MOSI, SCK, SS as o/p

  LDI R17, (1<<SPE)|(1<<MSTR)|(1<<SPR0)

   OUT SPCR, R17         ; enable SPI as master, fsck=fosc/16,
mode 0

;-------------------------------------------------------------
--
```

Port yang digunakan pada master device, diantaranya adalah

- Port B, yang digunakan untuk komunikasi dengan slave device dengan menggunakan protokol SPI, dengan menggunakan pin pin sebagai berikut:
    - PB3: MOSI, Output dari master dilakukan transmisi data ke slave
    - PB5 : SCK, Output clock signal dari master untuk mensinkronisasikan transfer data.
    - PB2 : Slave Select, Output dari master untuk memilih slave device.



Gbr 3. Slave Device

Sedangkan slave device menggunakan port D, yang digunakan untuk outputting signal dan mengatur slave device. Pin yang digunakan , diantaranya adalah

```
checkTemp:
  LDI R30,0x30
  CP R18, R30
  BRSH fireDetected
  BRLT noFire
  RET
```

Potongan code diatas ini berfungsi untuk memeriksa temperature, dengan membandingkan nilai temperatur yang didapatkan dari sensor DHT11 dengan suhu yang telah normal yang telah dispesifikasikan, apabila suhunya melewati batasnya, maka akan jump ke method **fireDetected**

```
fireDetected:
  SBI PORTD, 6
  RCALL myDelay ;Memanggil delay untuk pergantian nyala lampu
  CBI PORTD, 5
  RCALL myDelay ;Memanggil delay untuk pergantian nyala lampu
  SBI PORTD, 5
  SBI PORTD, 7
;-------------------------------------------------------------
----
myDelay: ;3-level nested loop delay
  LDI R20, 255 ;outer loop counter
c1: LDI R21, 255 ;mid loop counter
c2: LDI R22, 20 ;inner loop counter
;1s delay = 255 * 255 * 82 * 3 = 15.996.150 cycle = 1 s
c3:
```

```
DEC R22 ;decrement inner loop
BRNE c3 ;Branch ke l3 jika R22 != 0
DEC R21 ;decrement mid loop
BRNE c2 ;Branch ke l2 jika R21 != 0
DEC R20 ;decrement outer loop
BRNE c1 ;`Branch ke l1 jika R20 != 0
RET ;return
```

Method ini berfungsi ketika suhu meningkat maka dengan method ini akan menyalakan buzzer yang memberikan effect suara sebagai peringatan, blinking LED, dan menyalakan DC water pump untuk menarik air dari water bank dan menyalurkannya ke selang untuk selanjutnya diarahkan ke sumber panas.

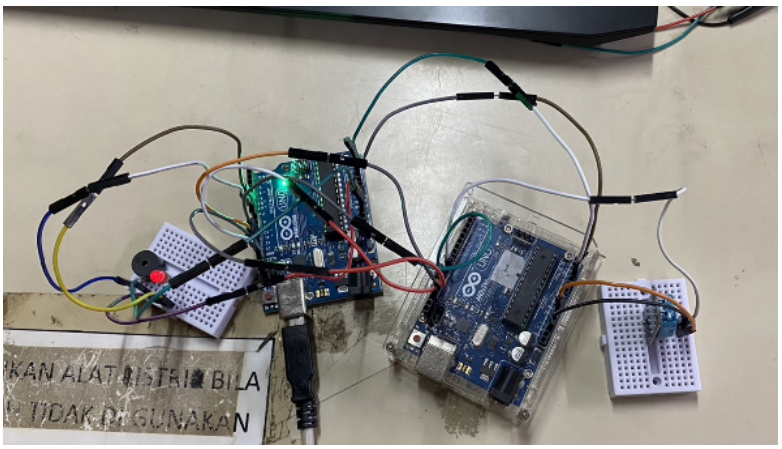
- PD7, pin ini dihubungkan ke DC Water Pump untuk menyalakan pompa yang akan digunakan untuk menarik air dari water bank ke sumber api.
- PD6, pin ini dihubungkan ke buzzer, yang mana berguna untuk memberikan alarm berupa sound yang dapat didengar, sebagai tanda bahwa telah terjadi peningkatan suhu secara significant.
- PD5, pin ini dihubungkan ke LED yang digunakan untuk memberikan warning light, yang menginformasikan bahwa telah terjadi peningkatan suhu pada ruangan, dan penghuni harus mengevakuasi diri secepatnya.

CHAPTER 3

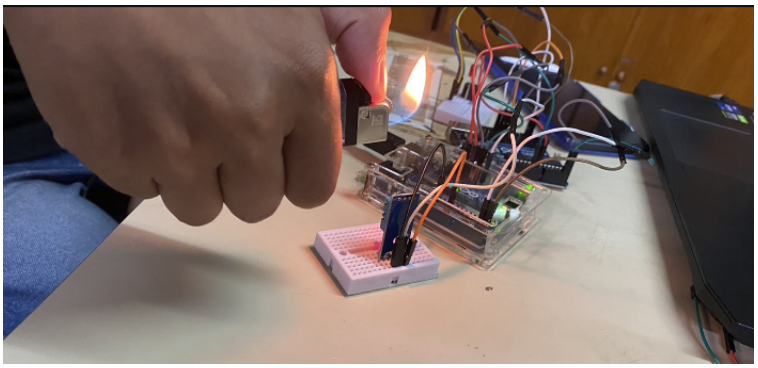
TESTING AND EVALUATION

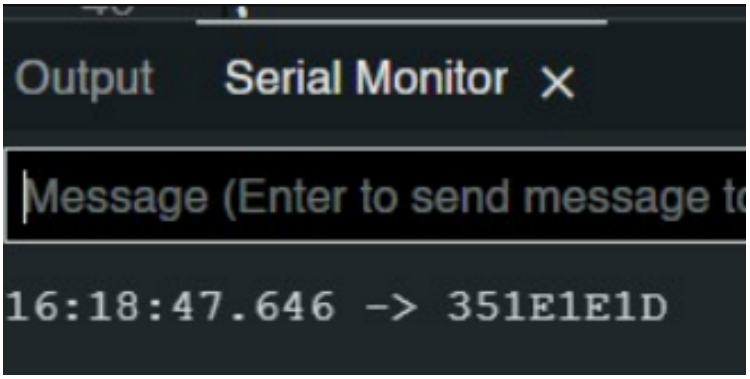
3.1 TESTING

- Pada kondisi suhu normal serial monitor membaca suhu yang didapatkan dari sensor DHT11 adalah 1A hexadecimal yang mana merupakan 26 derajat celcius.



- Pada kondisi panas (sensor DHT11 didekatkan dengan sumber panas), serial monitor didapat dari sensor DHT11 ialah 35 dalam hexadecimal, atau 53 derajat celcius. Selain itu Buzzer mengeluarkan bunyi, LED blinking dan DC water pump menarik air dari water bank.





3.2 RESULT

Berdasarkan hasil percobaan di atas, telah terbukti bahwa rangkaian berhasil memenuhi tujuannya yaitu memberi peringatan dan melakukan tindakan preventif terhadap potensi terjadinya kebakaran. Rangkaian akan berusaha memadamkan api ketika lonjakan suhu pada ruangan terjadi dengan cukup signifikan yaitu sekitar 48 derajat celcius yang berarti telah terjadi peningkatan suhu sekitar 20 derajat celcius dari suhu ruangan normal. Peletakkan alat pemadam juga sudah cukup efektif dalam memadamkan api yang berada pada ruang lingkup sensor DHT11.

3.3 EVALUATION

Rangkaian di atas telah cukup baik dalam menjalankan tujuannya, tetapi masih terdapat beberapa hal yang dapat ditingkatkan untuk meningkatkan fungsionalitas rangkaian. Salah satunya adalah dengan menggunakan sensor yang lebih sesuai untuk mendeteksi kebakaran. DHT11 sudah cukup baik dalam mendeteksi keberadaan api berdasarkan kenaiakan suhu, tetapi sensor tersebut masih memiliki beberapa masalah seperti cukup lambatnya suhu pada sensor untuk kembali ke suhu normal sehingga ketika api sudah berhasil dipadamkan, sensor masih akan tetap membaca bahwa keadaan di sekitarnya masih cukup panas.

Hal berikutnya yang dapat diperbaiki pada rangkaian adalah penempatan alatnya yang kurang aman. Pada rangkaian di atas, peletakkan sensor dan alat pemadam yang akan memancarkan air untuk memadamkan api terlalu dekat dan tidak memiliki proteksi sehingga ada sedikit kemungkinan air tersebut dapat merusak sensor jika aliran air tidak mengarah ke tempat yang seharusnya. Untuk mengatasi masalah tersebut, posisi peletakkan sensor dapat

diperbaiki atau diberi proteksi tambahan sehingga kemungkinan terkena air dari alat pemadam dapat dikurangi.

CHAPTER 4

CONCLUSION

Berdasarkan hasil pengembangan proyek fire alarm dengan menggunakan sensor DHT11, buzzer, servo, dan mikrokontroler Arduino, dapat disimpulkan bahwa sistem ini dapat berfungsi dengan baik dalam mendeteksi suhu panas yang dapat mengindikasikan adanya kebakaran di suatu ruangan. Sistem ini menggunakan sensor DHT11 untuk membaca suhu ruangan dan mengirimkan data ke mikrokontroler Arduino untuk diproses.

Saat suhu ruangan meningkat secara signifikan dan terindikasi adanya kebakaran, sistem akan memberikan peringatan dengan cara menyalakan lampu sebagai warning lights dan buzzer untuk memberikan peringatan suara. Selain itu, sistem juga dilengkapi dengan penyalur air yang menggunakan servo untuk memadamkan api yang terdeteksi.

Dalam pengembangan proyek ini, telah dilakukan implementasi beberapa kriteria penerimaan proyek seperti menerapkan sistem siber-fisik dalam bentuk mikrokontroler Arduino, membuat rangkaian yang dilengkapi dengan sensor gerak dan jarak untuk mendeteksi pergerakan pada pintu secara otomatis, mengimplementasikan protokol serial pada Arduino yaitu Serial Peripheral Interface (SPI), serta mengintegrasikan sensor dan komponen lainnya agar dapat menjalankan program sehingga mampu mengaktifkan LED dan buzzer.

Dengan demikian, proyek fire alarm ini dapat menjadi solusi yang efektif dalam mengantisipasi dan mencegah kebakaran di suatu ruangan dengan memberikan peringatan dini dan tindakan pencegahan yang cepat. Diharapkan pengembangan sistem ini dapat terus dilakukan dengan meningkatkan performa dan fungsionalitasnya untuk menghadapi tantangan masa depan dalam bidang keamanan dan keselamatan.

REFERENCES

[1] Anas Kuzechie, “Assembly via Arduino (part 17) - Programming I2C,” YouTube. Nov. 23, 2021 [Online]. Available: https://www.youtube.com/watch?v=N0tmYhU9sN8&list=PL09ZAP7_T_LmlX5vctZV4PFfZwMNzjX1F&index=18. [Accessed: May 16, 2023]

[2] Anas Kuzechie, “Assembly via Arduino (part 19) - DHT11 Sensor,” YouTube. Nov. 29, 2021 [Online]. Available: https://www.youtube.com/watch?v=vnLpzvkCUq8&list=PL09ZAP7_T_LmlX5vctZV4PFfZwMNzjX1F&index=19. [Accessed: May 16, 2023]

[3] Anas Kuzechie, “Assembly via Arduino (part 6) - ADC Hex Value on LCD,” YouTube. Oct. 15, 2021 [Online]. Available: https://www.youtube.com/watch?v=alLuNNQdpCU&list=PL09ZAP7_T_LmlX5vctZV4PFfZwMNzjX1F&index=7. [Accessed: May 16, 2023]

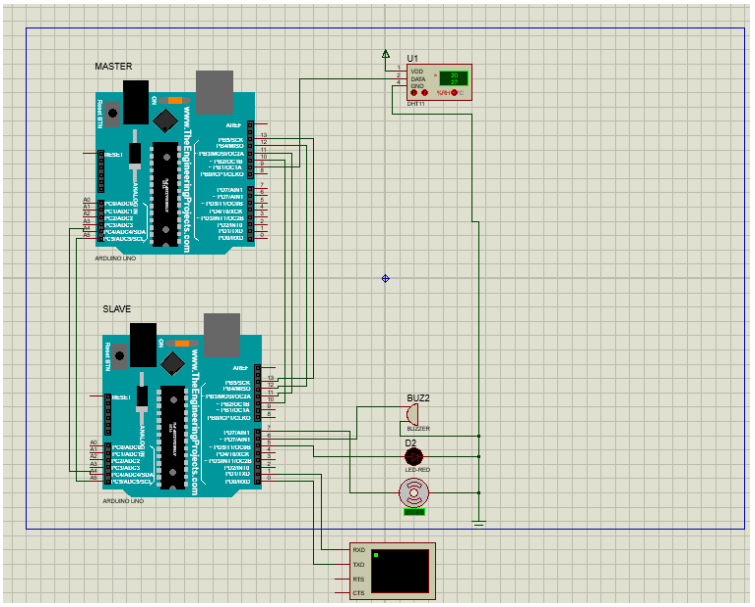
[4] “ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET” [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[5] “EMAS2: Log in to the site,” emas2.ui.ac.id. [Online]. Available: https://emas2.ui.ac.id/pluginfile.php/3797142/mod_resource/content/1/Modul%208%20SSF_%20I2C%20_%20SPI.pdf

[6] “EMAS2: Log in to the site,” emas2.ui.ac.id. [Online]. Available: https://emas2.ui.ac.id/pluginfile.php/3797143/mod_resource/content/1/Modul%209%20SSF-%20Sensor%20Interfacing.pdf

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation