



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

**MICROWAVE**

**GROUP A9**

<b>SENO PAMUNGKAS RAHMAN</b>	<b>2106731586</b>
<b>AMRITA DEVIAYU TUNJUNGBIRU</b>	<b>2106636584</b>
<b>SULTHAN SATRYA YUDHA DARMAWAN</b>	<b>2106731560</b>
<b>AHMAD RIFQI FADHLURRAHMAN</b>	<b>2106731301</b>

## PREFACE

Assalamualaikum warahmatullahi wabarakatuh,

Perkenalkan kami dari Kelompok SSF A9 dari jurusan Teknik Komputer UI angkatan 2021. Pertama-tama saya ingin menghaturkan terima kasih kepada Tuhan Yang Maha Esa karena atas rahmat dan hidayah-Nya kami bisa diberikan hidup yang nikmat dan bahagia. Tentunya, kami juga ingin mengucapkan **terima kasih** kepada orang tua kami, dosen kami, dan teman-teman kami yang telah **mendukung** dan **memotivasi** kami dalam melaksanakan proyek akhir SSF ini.

Pada saat ini, kita sedang menghadapi permasalahan kehidupan yang mewajibkan kita semua untuk bergerak secara cepat dan efisien. Hal ini dilakukan agar kita semua bisa meraih banyak hal yang kita idamkan. **Dengan berkaca pada permasalahan ini**, kami dari kelompok SSF A9 ingin turut membantu dengan membuat suatu alat untuk menjaga kesinambungan efektivitas kerja dari setiap manusia dalam bentuk *Microwave* untuk memasak makanan dengan mudah tanpa kehilangan banyak waktu. Kami harap produk yang kami buat ini bisa benar-benar berkontribusi dalam kehidupan nyata.

Depok, 15 Mei, 2023

Group

## TABLE OF CONTENTS

<b>BAB 1.....</b>	<b>4</b>
<b>PENDAHULUAN.....</b>	<b>4</b>
1.1 LATAR BELAKANG.....	4
1.3 TUJUAN.....	5
1.4 PERAN DAN TANGGUNG JAWAB.....	5
1.5 TIMELINE AND MILESTONES.....	5
<b>BAB 2.....</b>	<b>7</b>
<b>IMPLEMENTASI.....</b>	<b>7</b>
2.1 DESAIN HARDWARE DAN SKEMA.....	7
2.2 PENGEMBANGAN SOFTWARE.....	7
2.3 INTEGRASI SOFTWARE DAN HARDWARE.....	8
<b>BAB 3.....</b>	<b>9</b>
<b>PENGUJIAN DAN EVALUASI.....</b>	<b>9</b>
3.1 PENGUJIAN.....	9
3.2 HASIL.....	9
3.3 EVALUASI.....	10
<b>BAB 4.....</b>	<b>11</b>
<b>KESIMPULAN.....</b>	<b>11</b>

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 LATAR BELAKANG**

Dalam kehidupan sehari-hari, masyarakat modern sering dihadapkan pada tantangan kesibukan dan kekurangan waktu. Kehidupan yang serba cepat memaksa kita untuk menyelesaikan banyak tugas dalam waktu yang terbatas, termasuk memasak dan menghangatkan makanan. Di tengah kesibukan ini, muncul kebutuhan akan solusi yang praktis, cepat, dan efisien.

Kehidupan modern ditandai dengan jadwal yang padat dan tekanan waktu yang tinggi. Banyak orang yang sibuk dengan pekerjaan, pendidikan, dan tanggung jawab lainnya. Di sisi lain, kebutuhan akan makanan yang sehat dan bergizi juga penting untuk menjaga kesehatan dan kesejahteraan. Namun, memasak makanan membutuhkan waktu dan tenaga yang tidak selalu tersedia dalam rutinitas sehari-hari yang padat. Akibatnya, masyarakat sering kali menghadapi dilema antara waktu yang terbatas dan keinginan untuk mengkonsumsi makanan sehat. Banyak orang cenderung memilih solusi instan seperti makanan siap saji yang kurang sehat dan bergizi. Hal ini bisa berdampak negatif pada kesehatan dan kualitas hidup secara keseluruhan.

Oleh karena itu, microwave hadir sebagai solusi praktis dan efisien untuk memasak dan menghangatkan makanan. Microwave ini menggunakan button, timer, dan sensor DHT11 untuk memanaskan makanan dengan cepat. Microwave ini dirancang untuk mempercepat proses memasak, sehingga sangat cocok bagi mereka yang memiliki jadwal yang padat.

#### **1.2 SOLUSI YANG DIAJUKAN**

*Microwave* bertujuan untuk membantu mempercepat proses memasak dan menghangatkan makanan. *Microwave* ini memiliki tiga buah button untuk mengaktifkan *timer* yang bekerja pada *microwave* (10 menit, 20 menit, 30 menit) serta 2 buah button untuk mengaktifkan atau menonaktifkan *microwave*. *Microwave* ini juga mampu menampilkan *timer* yang bekerja serta suhu yang terjadi di dalam *microwave*. Hasil *timer* dan suhu ini akan ditampilkan menggunakan MAX7219.

### 1.3 TUJUAN

Tujuan dari pembuatan proyek akhir dari Sistem Siber Fisik ini adalah:

1. Mengimplementasikan materi-materi dari berbagai modul yang telah diberikan selama praktikum dan kelas berlangsung
2. Membuat rangkaian yang dapat mendeteksi suhu dalam sebuah ruangan (pada kasus ini, Microwave) dan menampilkannya
3. Mengimplementasikan *interrupt* pada rangkaian yang bisa menampilkan timer dengan *button*.
4. Mengimplementasikan *interrupt* pada rangkaian dengan *button* START dan STOP.

### 1.4 PERAN DAN TANGGUNG JAWAB

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

Roles	Responsibilities	Person
<i>Code writer</i>	Menulis kode, Merancang rangkaian hardware, Membuat Proteus	Seno Pamungkas Rahman
<i>Paper writer</i>	Menulis laporan proyek, Membuat flowchart, Membuat PPT	Amrita Deviayu Tunjungbiru
<i>Paper writer</i>	Menulis laporan proyek, Membuat README.md, Membuat PPT	Sulthan Satrya Yudha Darmawan
<i>Code writer</i>	Menulis kode, Merancang rangkaian hardware, Men- <i>testing</i> rangkaian	Ahmad Rifqi Fadhlurrahman

Table 1. Roles and Responsibilities

## 1.5 TIMELINE AND MILESTONES

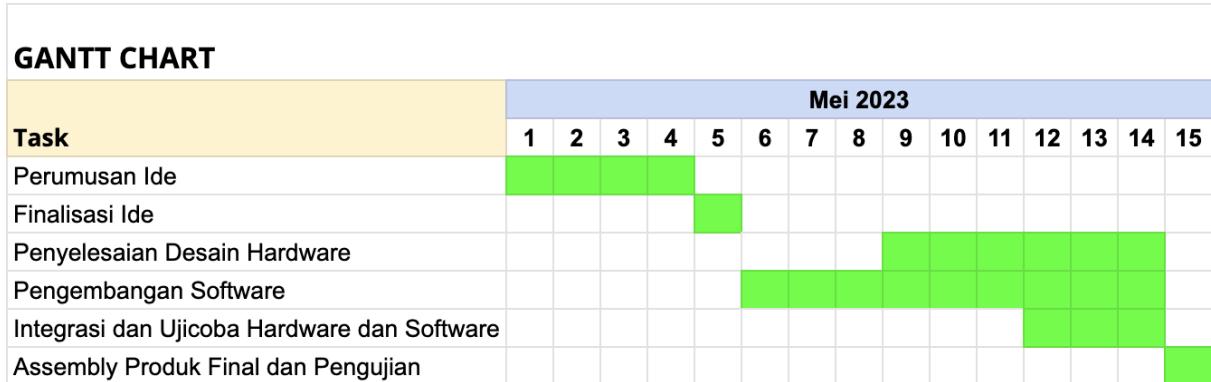


Fig.1 Gantt Chart

- a) Perumusan Ide : Pencapaian yang menunjukkan tanggal merumuskan ide
- b) Finalisasi Ide : Pencapaian yang menunjukkan tanggal finalisasi ide dengan Aslab
- c) Penyelesaian Desain Hardware : Pencapaian yang menunjukkan tanggal atas penggerjaan desain *hardware* secara virtual maupun fisik
- d) Pengembangan Software : Pencapaian yang menunjukkan tanggal atas penggerjaan pengembangan kode Assembly untuk proyek akhir
- e) Integrasi dan Ujicoba Hardware dan Software : Pencapaian yang menunjukkan tanggal atas penggerjaan menggabungkan hasil pekerjaan *hardware* dan *software*
- f) Assembly Produk Final dan Pengujian : Pencapaian yang menunjukkan tanggal untuk menguji hasil integrasi dan finalisasi rangkaian

## BAB 2

### IMPLEMENTASI

#### 2.1 DESAIN HARDWARE DAN SKEMA

Untuk memenuhi kriteria proyek akhir Praktikum Sistem Siber Fisik, maka pemrograman software dilakukan dalam bahasa Assembly untuk merancang program microwave ini. Sebelum menuliskan kode dalam bahasa Assembly, dibutuhkan implementasi rangkaian secara virtual. Dalam hal ini, kami menggunakan software Proteus untuk merangkai rangkaian *microwave* berdasarkan *flowchart* yang sudah kami buat.

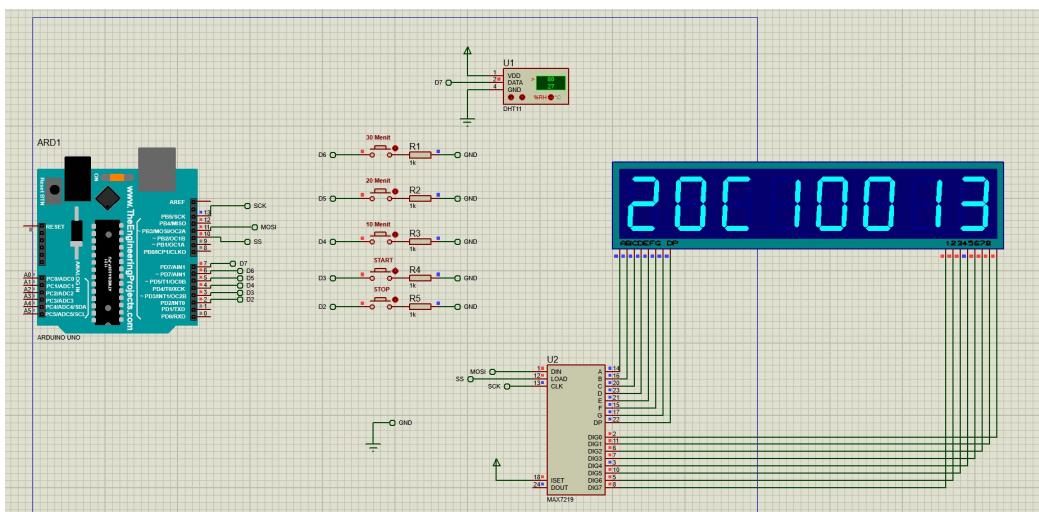
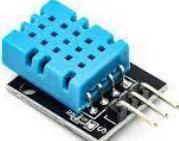


Fig.2 Proteus Design

Gambar diatas merupakan implementasi rangkaian secara virtual berdasarkan *flowchart* yang sudah kami buat. Pada visualisasi ini, kelompok kami menggunakan sebanyak 4 button dengan rincian setiap button berperan sebagai: Button Stop, Timer 10 menit, Timer 20 menit, dan Timer 30 menit. Penggunaan button merupakan implementasi Interrupt dari Modul 7. Setelah itu, digunakan serial monitor yang memiliki fungsi untuk *display* suhu secara real time dan juga memperlihatkan countdown dari timernya. Penggunaan Serial Port merupakan implementasi dari modul 4. Sensor DHT digunakan untuk melakukan pengecekan suhu di dalam microwave (implementasi Sensor dari modul 9). Timer juga digunakan sebagai penghitung countdown (implementasi Timer dari Modul 6). Seluruh komponen tersebut akan diprogramkan menggunakan Assembly Language (Modul 2). Serial Monitor juga akan melakukan perhitungan pada timer serta ASCII pada serial Monitor (Implementasi modul 5). Untuk memenuhi Implementasi dari Modul 8, akan dilakukan komunikasi antara dua Arduino untuk melakukan transfer data dari button.

Microwave pada proyek ini memiliki 5 button yaitu START, STOP, 10 menit, 20 menit, dan 30 menit sebagai timer untuk berapa lama microwave akan aktif. Pengguna bisa memulai *microwave* dengan menekan tombol START lalu menekan pilihan *timer* yang tersedia. Setelah ditekan, maka akan muncul *timer* yang berjalan sekaligus suhu *microwave* saat dijalankan. Namun, karena tidak ada implementasi *heater* yang terjadi, maka sensor DHT11 hanya akan mendeteksi suhu di ruangan pengguna berada.

Alat dan bahan:

Hardware			Software
Jumlah	Barang	Gambar	
1	Arduino		Arduino Uno
1	DHT11		Proteus
4	Button		Github
5	Resistor		Draw.io
	Kabel Jumper		

1	MAX7219		wokwi.com
	Laptop		Visual Studio code

## 2.2 PENGEMBANGAN SOFTWARE

Untuk memenuhi kriteria proyek akhir Praktikum Sistem Siber Fisik, maka pemrograman software dilakukan dalam bahasa Assembly untuk merancang program *microwave* ini. Bahasa Assembly adalah sebuah bahasa pemrograman rendah (low-level) yang digunakan untuk memprogram komputer, khususnya mikroprosesor dan mikrokontroler. Bahasa ini memiliki keterkaitan yang erat dengan bahasa mesin (machine language) yang merupakan bahasa yang dipahami langsung oleh komputer. Sebelum memulai pembuatan kode dalam bahasa Assembly, flowchart harus dibuat sebagai acuan pembuatan kode.

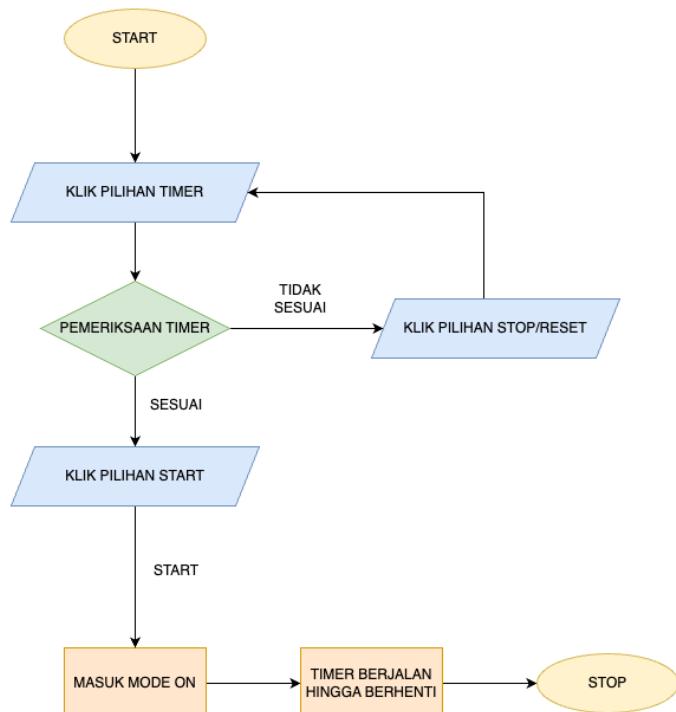


Fig.3 Flowchart

Gambar diatas merupakan alur kerja program yang divisualisasi dengan *flowchart* dengan software Draw.io. Awal alur *Microwave* ini bekerja adalah dengan memilih pilihan antara START (Mulai) atau STOP (Berhenti) pada button. Kemudian apabila button START dipilih, maka akan lanjut ke tahapan berikutnya yaitu memilih pilihan Timer (5, 10, atau 15 menit). Jika sudah memilih pilihan, maka *microwave* akan bekerja dan timer akan berjalan hingga berhenti dan masuk ke mode off. Jika pilihan button diawal adalah STOP, maka *microwave* tidak akan bekerja.

Merunut pada flowchart yang telah dibuat, rangkaian *microwave* dibuat dengan menggunakan software Arduino IDE dengan menggunakan bahasa Assembly. Berikut kode lengkap untuk proyek *microwave* ini :

```
#define __SFR_OFFSET 0x00
#include "avr/io.h"
#include "avr/interrupt.h"

.global main

.org 0x0000
RJMP main
.org 0x0002
RJMP stop

main:
RCALL init_interrupt
RCALL init_SPI

loop:
RCALL shutdown_MAX
RCALL SPI_MAX7219_init
RJMP option

backtoStart:
RJMP loop

option:
SBIC PIND,4
RJMP choose10
SBIC PIND,5
RJMP choose20
SBIC PIND,6
RJMP choose30
RJMP option

waitStart:
SBIC PIND,3
RJMP uptime_logic ;start counting
RJMP waitStart
```

```

choose10:
LDI R25,1
MOV R10,R25
LDI R17, 0x05
LDI R18, 1
RCALL send_bytes
RJMP waitStart

choose20:
LDI R25,2
MOV R10,R25
LDI R17, 0x05
LDI R18, 2
RCALL send_bytes
RJMP waitStart

choose30:
LDI R25,3
MOV R10,R25
LDI R17, 0x05
LDI R18, 3
RCALL send_bytes
RJMP waitStart

-----  

SPI_MAX7219_init:  

;-----  

.equ SCK, 5 ;clk 13
.equ MOSI, 3 ;din 11
.equ SS, 2 ;cs 10
;-----  

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
;-----  

LDI R17, 0x0A ;set segment intensity (0 to 15)
LDI R18, 5 ;intensity level = 8
RCALL send_bytes ;send command & data to MAX7219
;-----  

LDI R17, 0x09 ;set decoding mode command
LDI R18, 0b11011111 ;decoding byte
RCALL send_bytes ;send command & data to MAX7219
;-----  

LDI R17, 0x0B ;set scan limit command
LDI R18, 0x07 ;8 digits connected to MAX7219
RCALL send_bytes ;send command & data to MAX7219
;-----  

LDI R17, 0x0C ;set turn ON/OFF command
LDI R18, 0x01 ;turn ON MAX7219
RCALL send_bytes ;send command & data to MAX7219

```

```
;  
RET  
=====  
  
uptime_logic:  
LDI R17,0x05  
MOV R18,R10  
RCALL send_bytes  
  
RCALL disp_space  
LDI R25, 0  
LDI R17, 0x04  
MOV R18, R25  
RCALL send_bytes  
  
agn3:  
LDI R29,0  
LDI R17, 0x03  
MOV R18, R29  
RCALL send_bytes  
  
agn2:  
LDI R30,0  
LDI R17,0x02  
MOV R18, R30  
RCALL send_bytes  
  
agn1:  
LDI R31,0  
loopA:  
LDI R17, 0x01  
MOV R18, R31  
RCALL send_bytes  
INC R31  
CPI R31, 11  
BREQ inc_MSD  
RCALL delay_timer1  
RJMP loopA  
  
inc_MSD:  
RCALL DHT11_sensor  
LDI R17, 0x02  
INC R30  
CPI R30,6  
BREQ inc_min  
MOV R18,R30  
RCALL send_bytes  
RJMP agn1  
  
inc_min:  
LDI R17,0x03  
INC R29  
CPI R29,10
```

```

BREQ inc_min2
MOV R18,R29
RCALL send_bytes
RJMP agn2

inc_min2:
LDI R17,0x04
INC R25
CP R25,R10      ;CP R25,R15
BREQ back
MOV R18,R25
RCALL send_bytes
RJMP agn3

back:
JMP backtoStart

send_bytes:
CBI PORTB, SS    ;enable slave device MAX7219
OUT SPDR, R17    ;transmit command
;-----
I2:
IN R19, SPSR
SBRS R19, SPIF   ;wait for byte transmission
RJMP I2          ;to complete
;-----
OUT SPDR, R18    ;transmit data
;-----
I3:
IN R19, SPSR
SBRS R19, SPIF   ;wait for byte transmission
RJMP I3          ;to complete
;-----
SBI PORTB, SS    ;disable slave device MAX7219
RET

delay_timer1: ; 1 sec delay via timer1
.EQU value, 62498 ;62498
;-----
CLR R20 ; Clear R20 to be 0
STS TCNT1H, R20 ; Store R20 to High Byte TCNT1
STS TCNT1L, R20 ;initialize counter TCNT1 = 0
LDI R20, hi8(value) ; Store high byte from value to R20
STS OCR1AH, R20 ; Store Content from R20 to OCR1AH
LDI R20, lo8(value) ; Store low byte from value to R20
STS OCR1AL, R20 ; Store Content from R20 to OCR1AL
;-----
CLR R20 ; Clear R20 to be 0
STS TCCR1A, R20 ; Store content from R20 to TCCR1A
LDI R20, 0b00001100 ; Store a byte to R20
STS TCCR1B, R20 ; CTC mode, prescaler = 256
;-----

```

```

lo4:
IN R20, TIFR1 ;get TIFR1 byte & check
SBRS R20, OCF1A ;if OCF1=1, skip next instruction
RJMP lo4 ;else, loop back & check OCF flag
;-----
LDI R20, 1<<OCF1A
OUT TIFR1, R20 ;clear OCF1A and OCF1B flag
;-----
CLR R20
STS TCCR1B, R20 ;stop timer0
RET

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 64
;-----
I2:
IN  R20, TIFR0   ;get TIFR0 byte & check
SBRS R20, OCF0A  ;if OCF0=1, skip next instruction
RJMP I2          ;else, loop back & check OCF0 flag
;-----
CLR R20
OUT TCCR0B, R20   ;stop timer0
;-----
LDI R20, (1<<OCF0A)
OUT TIFR0, R20   ;clear OCF0 flag
RET

convtemp:          ;convert and send temperature
;-----
CLR R26          ;set counter1, initial value 0 , ini digit ratusan atau puluhan?
CLR R27          ;set counter2, initial value 0 , ini digit puluhan atau satuan?
;-----
I70:
CPI R28, 100     ;compare R28 with 100
Ret:
BRMI I80         ;jump when R28 < 100
INC R26          ;increment counter1 by 1
SUBI R28, 100    ;R28 = R28 - 100
RJMP I70
;-----
I80:
CPI R28, 10      ;compare R28 with 10
BRMI I80         ;jump when R28 < 10
INC R27          ;increment counter2 by 1
SUBI R28, 10      ;R28 = R28 - 10
RJMP I80
RET
;-----
```

6+5+4+3+2+1

```

dsp:
MOV R18, R27
LDI R17, 0x08 ;select digit, R29 ganti dengan 0x01
RCALL send_bytes ;send command & data to MAX7219
;-----
MOV R18, R28
MOV R17, 0x07 ;select digit, R30 ganti dengan 0x02
RCALL send_bytes ;send command & data to MAX7219
;-----
RET

DHT11_sensor:
;-----
;agn:RCALL delay_2s ;wait 2s for DHT11 to get ready, bisa diganti ke 3s, mungkin ini
gabutuh
;-----
;start_signal
;-----
SBI DDRD, 7 ;pin PD7 as o/p
CBI PORTD, 7 ;first, send low pulse
RCALL delay_20ms ;for 20ms
SBI PORTD, 7 ;then send high pulse
;-----
;responce signal
;-----
CBI DDRD, 7 ;pin PD7 as i/p
w1:
SBIC PIND, 7
RJMP w1 ;wait for DHT11 low pulse
w2: SBIS PIND, 7
RJMP w2 ;wait for DHT11 high pulse
w3:
SBIC PIND, 7
RJMP w3 ;wait for DHT11 low pulse
;-----
RCALL DHT11_reading ;read humidity (1st byte of 40-bit data)
MOV R13, R24
RCALL DHT11_reading
RCALL DHT11_reading ;read temp (3rd byte of 40-bit data)
;-----
;convert temp & humidity bytes to decimal & display on MAX7219
;-----
MOV R28, R24 ;4 line ini bakal ga butuh
;LDI R29, 0x07
;LDI R30, 0x06
;-----
RCALL convtemp ;temp in decimal
;-----
;MOV R28, R25
;LDI R29, 0x02
;LDI R30, 0x01
;-----
;RCALL convtemp ;humidity in decimal
;-----

```

```

;RJMP agn      ;go back & get another sensor reading
RET
;=====
DHT11_reading:
LDI R16, 8      ;set counter for receiving 8 bits
CLR R24      ;clear data register
;-----
w4:
SBIS PIND, 7
RJMP w4      ;detect data bit (high pulse)
RCALL delay_timer0 ;wait 50us & then check bit value
;-----
SBIS PIND, 7      ;if received bit=1, skip next inst
RJMP skp      ;else, received bit=0, jump to skp
SEC      ;set carry flag (C=1)
ROL R24      ;shift in 1 into LSB data register
RJMP w5      ;jump & wait for low pulse
skp:
LSL R24      ;shift in 0 into LSB data register
;-----
w5:
SBIC PIND, 7
RJMP w5      ;wait for DHT11 low pulse
;-----
DEC R16      ;decrement counter
BRNE w4      ;go back & detect next bit
RET      ;return to calling subroutine
;=====
;delay subroutines
;=====
delay_20ms:      ;delay 20ms
LDI R21, 255
I3:
LDI R22, 210
I4:
LDI R23, 2
I5:
DEC R23
BRNE I5
DEC R22
BRNE I4
DEC R21
BRNE I3
RET

init_interrupt:
// Initialize pins D2 to D6 as input pullup
LDI r16, 0b00000000 ; set the bit pattern for pins D2 to D6 as input with pull-up resistors
enabled
OUT DDRD, r16      ; set the DDRD register to make the pins inputs
LDI r16, 0b01111100 ; set the bit pattern for pins D2 to D6 as input with pull-up resistors
enabled
OUT PORTD, r16      ; set the PORTD register to enable pull-up resistors on the input pins
CLR R16

```

```

LDI R16, (1 << ISC01) ; Falling Edge mode source = INT0 and INT1?
STS EICRA, R16
SBI EIMSK, INT0          ; Enable INT0
SEI                      ; Enable global interrupts
RET

disp_space:
LDI R17,0x06
LDI R18,0x4E
RCALL send_bytes
RET

stop:
SBI EIFR, INTF0 ; Return from interrupt
RETI

shutdown_MAX:
LDI R17,0x09    ;set decoding mode command
LDI R18,0x00    ;decoding byte
RCALL send_bytes ;send command & data to MAX7219
LDI R17,0x01
LDI R18,0x00
RCALL send_bytes
LDI R17,0x02    ;set decoding mode command
LDI R18,0x00    ;decoding byte
RCALL send_bytes ;send command & data to MAX7219
LDI R17,0x03
LDI R18,0x00
RCALL send_bytes
LDI R17,0x04    ;set decoding mode command
LDI R18,0x00    ;decoding byte
RCALL send_bytes ;send command & data to MAX7219
LDI R17,0x05
LDI R18,0x00
RCALL send_bytes
LDI R17,0x06    ;set decoding mode command
LDI R18,0x00    ;decoding byte
RCALL send_bytes ;send command & data to MAX7219
LDI R17,0x07
LDI R18,0x00
RCALL send_bytes
LDI R17,0x08    ;set decoding mode command
LDI R18,0x00    ;decoding byte
RCALL send_bytes ;send command & data to MAX7219
RET

init_SPI:
.equ SCK, 5 ;clk 13
.equ MOSI, 3 ;din 11
.equ SS, 2 ;cs 10
-----

```

```

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
RET

```

Berdasarkan kode, fungsi **SPI\_MAX7219\_init** berfungsi untuk melakukan inisialisasi yang mengatur mode SPI dan mengirim beberapa perintah dan data ke MAX7219, chip yang mengendalikan *display* tujuh segmen. Fungsi ini mempersiapkan pengaturan awal sebelum *display* dapat digunakan. Fungsi **uptime\_logic** bertanggung jawab untuk mengatur dan mengirimkan data waktu (menit, detik) ke *display* MAX7219. Fungsi **inc\_MSD** berfungsi untuk mengatur peningkatan digit MSD (Most Significant Digit) yaitu detik hingga 60, fungsi **inc\_min** berfungsi untuk mengatur peningkatan digit menit hingga 10 menit, dan fungsi **inc\_min2** berfungsi untuk mengatur peningkatan digit menit puluhan sesuai *button* yang dipilih. Fungsi **send\_bytes** digunakan untuk mengirimkan byte perintah dan data ke MAX7219 melalui komunikasi SPI. Fungsi **delay\_timer1** digunakan untuk membuat penundaan selama 1 detik menggunakan Timer 1 dan diaplikasikan fungsi **uptime\_logic** untuk membuat timer detik berjalan naik. Fungsi **delay\_timer0** digunakan untuk membuat penundaan sebesar 10 detik menggunakan Timer 0 dan diaplikasikan ke dalam **DHT11\_reading** untuk memberikan delay dalam pembacaan sensor.

Fungsi **convtemp** bertanggung jawab untuk mengkonversi dan mengirimkan suhu dalam format desimal ke MAX7219. Fungsi ini melakukan konversi dari byte suhu menjadi digit-desimal dan mengirimkannya ke *display* tujuan. Konversi dilakukan dengan membagi byte suhu menjadi puluhan dan satuan, kemudian mengirimkan digit-desimal tersebut ke MAX7219. Fungsi **DHT11\_sensor** digunakan untuk membaca data dari sensor DHT11. Fungsi ini mengatur sinyal start dan response yang diperlukan untuk berkomunikasi dengan sensor. Setelah itu, fungsi membaca kelembaban dan suhu dari sensor DHT11 dan menyimpannya dalam register R13 dan R24. Fungsi **DHT11\_reading** digunakan untuk membaca bit-bit data dari sensor DHT11. Fungsi ini mengatur waktu dan membaca sinyal dari sensor untuk mendapatkan bit-bit data yang dikirimkan. Bit-bit tersebut kemudian dikonversi menjadi byte kelembaban dan suhu yang lebih mudah dipahami. Fungsi **delay\_20ms** digunakan untuk melakukan penundaan sebesar 20 milisekon dan diimplementasikan terhadap fungsi **DHT11\_sensor**. Fungsi **init\_interrupt** digunakan untuk menginisialisasi pengaturan interrupt pada mikrokontroler. Fungsi **choose10, choose20**,

**choose30** adalah fungsi yang dipanggil ketika pengguna memilih waktu timer berjalan. Fungsi **disp\_space** digunakan untuk menampilkan spasi pada *display* MAX7219. Fungsi **stop** digunakan untuk kembali ke kode setelah fungsi **init\_interrupt**. Fungsi **shutdown\_MAX** digunakan untuk mematikan *microwave* yang sedang berjalan dengan tidak menampilkan apapun pada MAX7219. Fungsi **init\_SPI** bertujuan untuk menginisialisasi modul SPI (Serial Peripheral Interface). Fungsi ini melakukan pengaturan beberapa register untuk mengkonfigurasi modul SPI sebagai master dengan prescaler  $fsck=fosc/16$ . Selain itu, fungsi ini juga mengatur intensitas segment, mode dekode, batas pemindaian, dan status on/off pada perangkat MAX7219 yang terhubung melalui SPI.

## 2.3 INTEGRASI SOFTWARE DAN HARDWARE

Tidak mungkin suatu rangkaian bisa bekerja tanpa adanya kode yang diunggah ke dalam Arduinonya. Proses integrasi antara *software* dan *hardware* hadir untuk mengatasi problematika tersebut. Dalam proses integrasi ini, rangkaian harus disesuaikan antara *input* dan *outputnya* dengan kode yang sudah dibuat. *Input button* untuk START atau untuk memulai *microwave* ditempatkan pada pin D nomor 3, sedangkan untuk *button STOP* atau untuk menghentikan berjalannya *microwave*, input ditempatkan pada pin D nomor 2. Untuk *button timer* 10 menit, pin ditempatkan pada pin D nomor 4. Untuk *button timer* 20 menit, pin ditempatkan pada pin D nomor 5. Dan untuk *button timer* 30 menit, pin ditempatkan pada pin D nomor 6. *Input* untuk sensor DHT11 ditempatkan pada pin D nomor 7. Semua *button* dipastikan terdapat resistor sebelum diseimbangkan dengan dikoneksikan terhadap *ground* (GND). Output dari rangkaian akan ditempatkan ke *display* MAX7219. MAX7219 dalam proyek ini bertindak sebagai *Slave*, sehingga mempunyai pin untuk *Slave Select* yaitu di pin B nomor 2, pin untuk MOSI SPI yaitu di pin B nomor 3, dan pin untuk SCK yaitu di pin B nomor 5.

Setelah memasangkan pin, langkah selanjutnya adalah untuk mengecek apakah kode berhasil saat dijalankan atau tidak. Merunut pada penjelasan kode yang sudah dibuat diatas, rangkaian ini akan memeriksa berapa derajat temperatur celcius dari sensor DHT11. Tentunya, sensor DHT11 tidak akan berjalan sebelum *button START* dan *button timer* ditekan. Kemudian, rangkaian ini akan menghitung maju menuju *button timer* yang ditekan untuk menandakan berapa lama *microwave* ini ingin digunakan. Setelah semua *button* diaktifkan, maka rangkaian akan menampilkan hasil dari sensor dan *timer* yang dipilih ke

*display* MAX7219. Setiap *button* yang ada pada rangkaian bersifat *input-pullup* dimana *input* akan diberikan *output* HIGH.

Untuk *display* pada MAX7219, 3 digit dari kiri digunakan untuk temperatur dimana 2 digit akan menampilkan angkanya dan 1 digit akan menampilkan huruf ‘C’ sebagai celcius. 4 digit dari kanan digunakan sebagai *timer* dimana digit 1 dan 2 dari kanan menampilkan waktu dalam satuan detik, dan digit 3 dan 4 menampilkan waktu dalam satuan menit. Namun, pada digit 5 dari kanan, MAX7219 akan menampilkan angka 1 (untuk 10 menit) atau 2 (untuk 20 menit) atau 3 (untuk 30 menit) sesuai *timer* yang dipilih dan setelah *timer* mencapai waktu dalam menit puluhan. Untuk tambahan, setelah *timer* mencapai batas waktu yang ditentukan oleh *button timer*, maka MAX7219 akan mendisplay angka 0 pada semua digitnya.

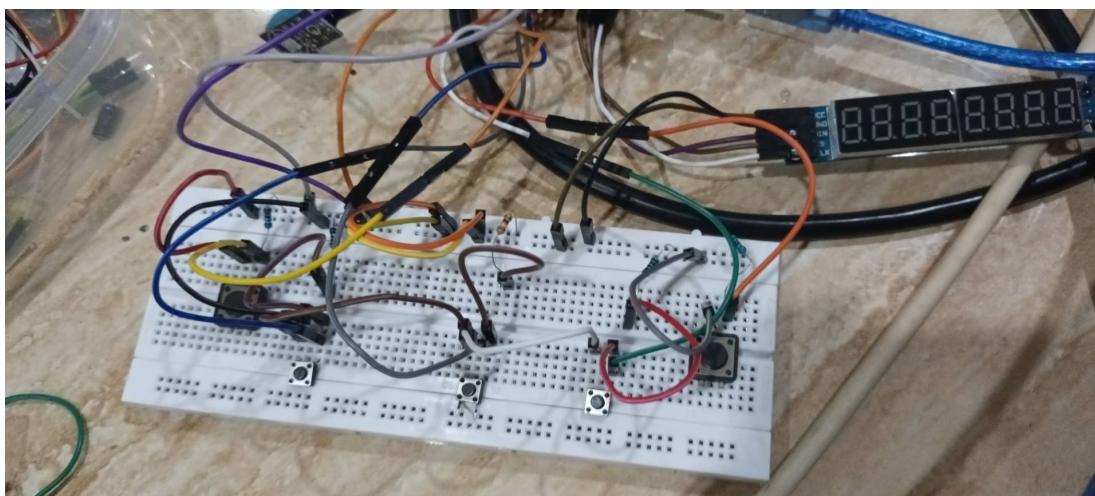


Fig.4 *Integration between Software and Hardware*

## **BAB 3**

### **PENGUJIAN DAN EVALUASI**

#### **3.1 PENGUJIAN**

Setelah melakukan integrasi hardware dengan software, rangkaian diuji dengan tujuan membuktikan apakah rangkaian akan berjalan sesuai dengan input kode yang telah diberikan. Rangkaian Microwave diuji dengan memilih durasi lamanya microwave akan berjalan melalui button dengan spesifikasi waktu antara 10 menit, 20 menit, dan 30 menit melalui. Apabila durasi yang dipilih tidak sesuai, button STOP dapat ditekan untuk menghentikan proses pemilihan durasi serta mengulang input waktu. Setelah durasi yang dipilih sesuai, Button START dapat ditekan untuk memulai Microwave bekerja. Selama proses tersebut berlangsung, MAX 7219 akan mendisplay Suhu dengan besaran derajat Celcius yang ditangkap oleh sensor DHT11 serta menampilkan display CountUp selama microwave bekerja sesuai dengan durasi yang dipilih sebelum menekan button START. Microwave akan berhenti bekerja apabila telah mencapai durasi yang telah ditentukan. Sebagai catatan, durasi tidak bisa diubah sebelum button STOP ditekan, maka dari itu alur bekerja nya microwave adalah memilih durasi melalui button TIMER, lalu menekan button START untuk mulai, dan button STOP digunakan untuk keperluan mengulang pemilihan durasi serta pemberhentian sebelum durasi timer tercapai.

#### **3.2 HASIL**

Setelah dilakukan pengujian, didapatkan hasil seperti yang diinginkan. *Button START* dan *button timer* bekerja dengan baik sehingga terdapat *display* pada MAX7219. MAX7219 menampilkan suhu *microwave* (pada pengujian : suhu ruangan) serta menampilkan waktu *timer* yang sedang menghitung naik sesuai dengan pilihan *timer* yang diinginkan diawal, yaitu 20 menit. Pada pengujian pertama, hasil yang ditampilkan adalah suhu senilai 30 derajat celcius dan *timer* 10 menit 40 detik.

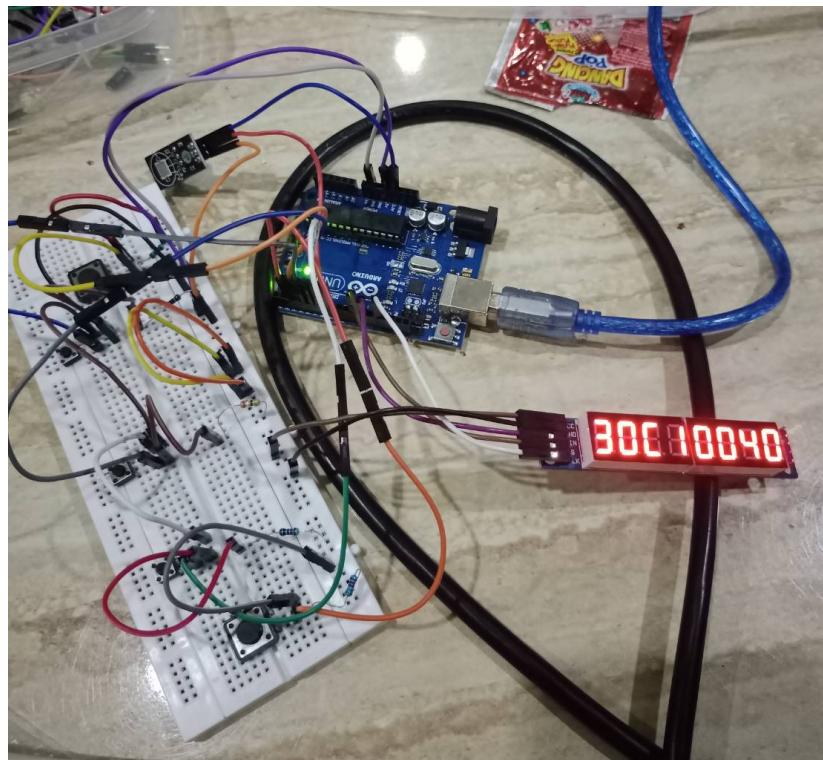


Fig.5 Testing Result Part 1

Dalam pengujian kedua, kami ingin melihat apakah *button* STOP atau RESET kami bekerja. Setelah diujikan, *button* STOP atau RESET berhasil bekerja dan didapatkan hasil yang diinginkan. MAX7219 menampilkan angka 0 untuk suhu maupun *timer*.

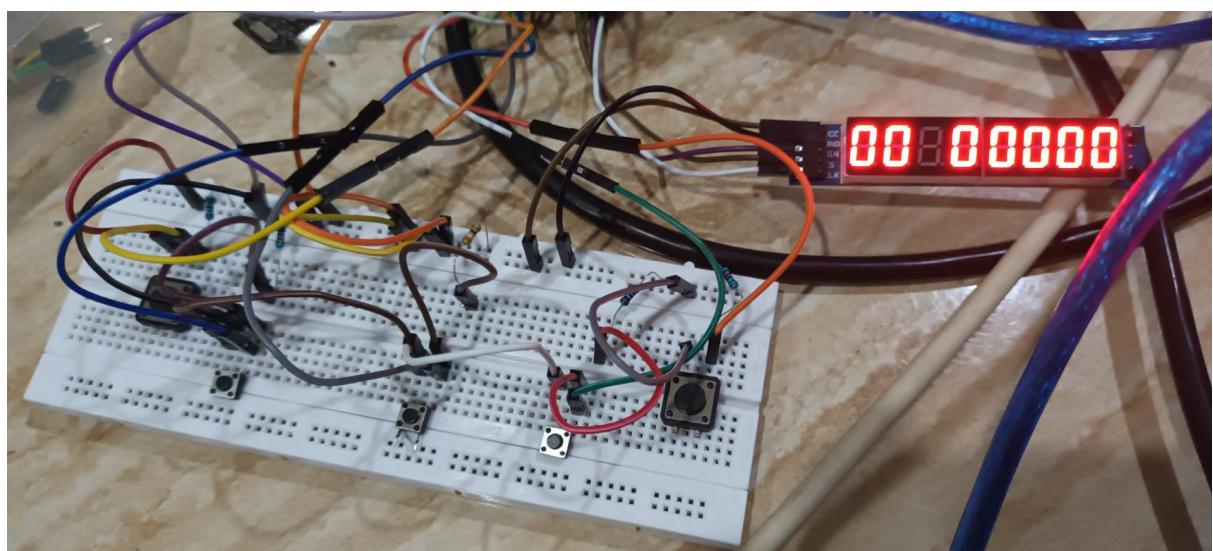


Fig.6 Testing Result Part 2

### 3.3 EVALUASI

Evaluasi yang timbul dari adanya proyek akhir Sistem Siber Fisik ini dipisahkan menjadi dua: evaluasi dari kinerja selama pengerjaan proyek dan evaluasi dari hasil yang didapatkan. Untuk kinerja selama pengerjaan proyek, kelompok kami memiliki kerjasama kelompok yang cukup bagus sehingga proses pengerjaan tidak berantakan. Pembagian pekerjaan pun cukup seimbang dan waktu untuk menyelesaikan semua kebutuhan proyek bisa terlaksana. Dari segi hasil yang didapatkan, kelompok kami berhasil mendapatkan hasil sesuai tujuan dan harapan kami. Implementasi seluruh modul pun juga sudah dilaksanakan dalam pembuatan kode Assembly sehingga proyek kami bisa dinyatakan memenuhi semua syarat proyek akhir Sistem Siber Fisik. Apabila ada yang perlu ditambahkan, maka kelompok kami mengusulkan untuk menggunakan *heater* untuk mensimulasikan *microwave* secara nyata dalam skala kecil. Kemudian, penggunaan LED untuk menandakan apakah *microwave* sedang berjalan atau tidak juga bisa ditambahkan untuk menambah keamanan pada replikasi *microwave* ini.

## **BAB 4**

### **KESIMPULAN**

Microwave merupakan barang yang digunakan dalam kehidupan sehari-hari yang fungsi utamanya adalah memanaskan makanan. Proyek ini mengimplementasikan cara kerja microwave dengan melakukan integrasi hardware dan software seputar penggunaan Arduino Uno. Kode pada proyek ini dirancang dengan menggunakan bahasa Assembly dengan memperhatikan implementasi Sembilan modul yang telah dipaparkan selama Praktikum Sistem Siber Fisik Laboratorium Digital Fakultas Teknik Universitas Indonesia berlangsung dengan fokus pada penggunaan Sensor DHT11, *Timer*, implementasi serial monitor, serta implementasi *Interrupt*. Lalu Hardware dirangkai dengan menggunakan alat dan bahan yang telah tertera serta dilakukan integrasi antara hardware dan software dengan mengimplementasikan kode yang telah dibuat agar pengujian dapat dilaksanakan.

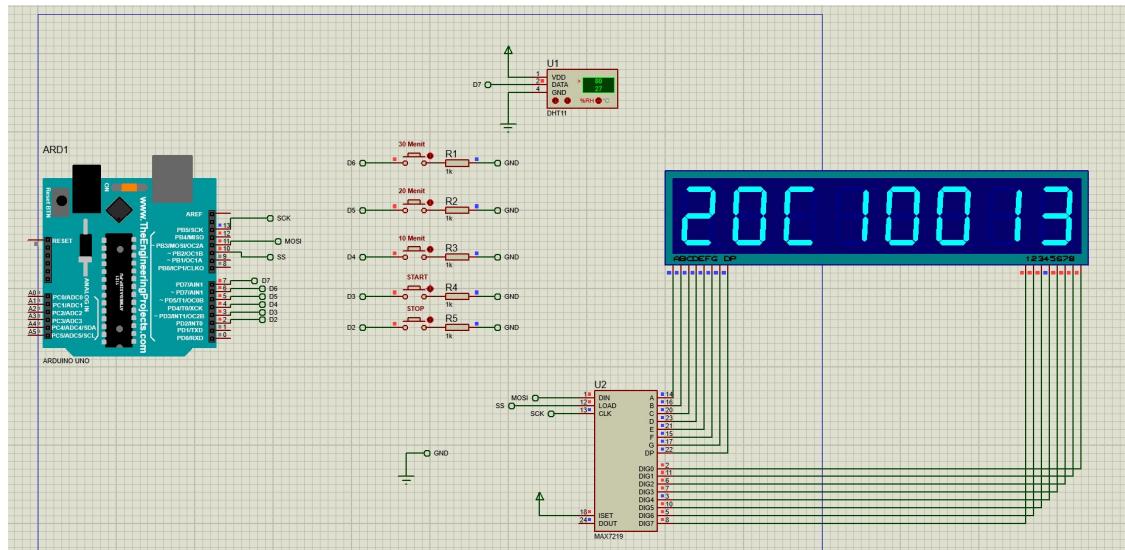
Berdasarkan hasil dari pengujian dan evaluasi, Proyek Microwave yang telah kelompok kami buat dapat mendeteksi suhu serta melakukan perhitungan waktu sebagaimana yang ditujukan dari tujuan awal pembuatan proyek. Maka dari itu, Proyek Microwave ini telah berjalan dengan baik.

## REFERENSI

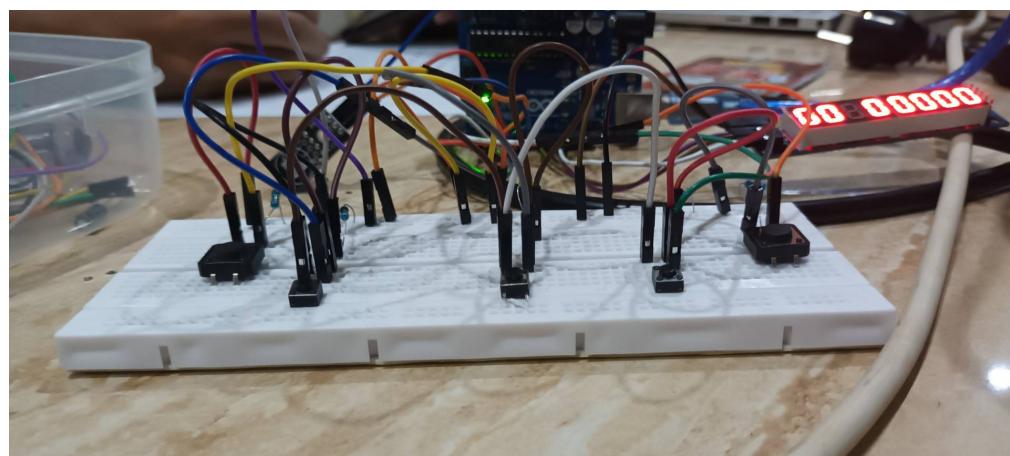
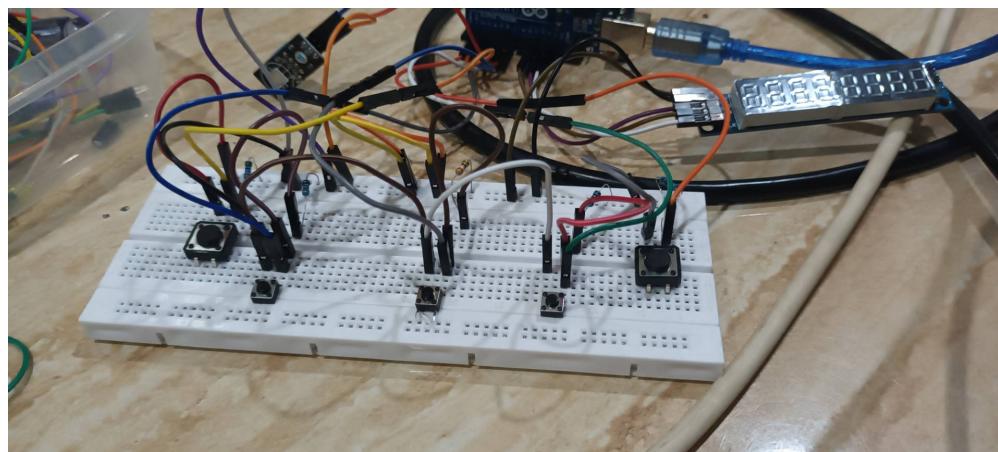
- [1] Harditya, M., & Faza, M. N. (n.d.). *Modul 2 SSF: Introduction to Assembly\_I\_O Programming*. [Online] Available: <https://classroom.google.com/u/0/c/NTkzNjAyMDgzODQw> [Accessed: May. 6, 2023].
- [2] Harditya, M., & Faza, M. N. (n.d.) *Modul 3 SSF: Analog to Digital Converter*. [Online] Available: <https://classroom.google.com/u/0/c/NTkzNjAyMDgzODQw/a/NTk1MTc3NTE0MzI5/details> [Accessed: May. 6, 2023].
- [3] Harditya, M., & Faza, M. N. (n.d.) *Modul 4 SSF: Serial Port*. [Online] Available: <https://classroom.google.com/u/0/c/NTkzNjAyMDgzODQw/m/NTk2NzYzMjQ2MTUx/details> [Accessed: May. 6, 2023].
- [4] Harditya, M., & Faza, M. N. (n.d.) *Modul 5 SSF: Aritmatika*. [Online] Available: [https://emas2.ui.ac.id/pluginfile.php/3744299/mod\\_resource/content/3/Modul%205%20SSF%20Aritmatika.pdf](https://emas2.ui.ac.id/pluginfile.php/3744299/mod_resource/content/3/Modul%205%20SSF%20Aritmatika.pdf) [Accessed: May. 6, 2023].
- [5] Harditya, M., & Faza, M. N. (n.d.) *Modul 6 SSF: Timer*. [Online] Available: [https://emas2.ui.ac.id/pluginfile.php/3754317/mod\\_resource/content/1/Modul%206%20SSF%20Timer.pdf](https://emas2.ui.ac.id/pluginfile.php/3754317/mod_resource/content/1/Modul%206%20SSF%20Timer.pdf) [Accessed: May. 6, 2023]
- [6] Harditya, M., & Faza, M. N. (n.d.) *Modul 7 SSF: Interrupt*. [Online] Available: [https://emas2.ui.ac.id/pluginfile.php/3767711/mod\\_resource/content/3/Modul%207%20SSF%20Interrupt.pdf](https://emas2.ui.ac.id/pluginfile.php/3767711/mod_resource/content/3/Modul%207%20SSF%20Interrupt.pdf) [Accessed: May. 6, 2023]
- [7] Harditya, M., & Faza, M. N. (n.d.) *Modul 8 SSF: I2C & SPI*. [Online] Available: [https://emas2.ui.ac.id/pluginfile.php/3797142/mod\\_resource/content/1/Modul%208%20SSF%20I2C%20%20SPI.pdf](https://emas2.ui.ac.id/pluginfile.php/3797142/mod_resource/content/1/Modul%208%20SSF%20I2C%20%20SPI.pdf) [Accessed: May. 6, 2023]
- [8] Harditya, M., & Faza, M. N. (n.d.) *Modul 9 SSF: Sensor Interfacing*. [Online] Available: [https://emas2.ui.ac.id/pluginfile.php/3797143/mod\\_resource/content/1/Modul%209%20SSF%20Sensor%20Interfacing.pdf](https://emas2.ui.ac.id/pluginfile.php/3797143/mod_resource/content/1/Modul%209%20SSF%20Sensor%20Interfacing.pdf) [Accessed: May. 6, 2023].
- [9] Kuzechie, A. *Creating Arduino Library for MAX7219 7-Segment Display*. [Online] Available: <https://akuzechie.blogspot.com/2022/08/creating-arduino-library-for-max7219-7.html> [Accessed: May. 6, 2023].

## APPENDICES

## Appendix A: Project Schematic in Proteus



## Appendix B : Project Hardware



## Appendix C: Documentation

