



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

Automatic House Gate

GROUP B5

Fahrezy H.	2106731466
Fayza Nirwasita	2106635700
Ivan Indrastata Ramadhan	2106706981
Mochammad Dyenta Dwiantomitara	2106731245

PREFACE

Puji syukur atas kehadirat Tuhan Yang Maha Esa karena berkat dan rahmat-Nya, kami dapat menyusun laporan praktikum. Laporan ini disusun dengan tujuan implementasi pemrograman Arduino menggunakan bahasa Assembly dapat diterapkan dalam masalah kehidupan sehari-hari. Laporan ini membahas penerapan pagar otomatis untuk transportasi, yang merupakan salah satu permasalahan kehidupan sehari-hari yang dapat diimplementasikan teknologi Arduino.

Penulisan laporan bertujuan memenuhi objektif proyek akhir praktikum Sistem Siber Fisik. Selain itu, laporan dibuat untuk menunjukkan implementasi teknologi yang sebelumnya dipelajari selama masa pembelajaran mata kuliah Sistem Siber Fisik. Diharapkan dari penulisan laporan ini, tujuan penulis dapat tersampaikan dan dapat diterapkan dalam kehidupan nyata secara aplikatif, solutif, dan efisien.

Penulis mengucapkan terima kasih kepada asisten laboratorium selaku pembimbing praktikum perancangan sistem digital. Penulis juga mengucapkan terima kasih kepada teman-teman dan orang tua dalam membantu baik dukungan dan materi untuk membuat laporan ini. Penulis menyadari bahwa terdapat kekurangan dalam menyusun laporan. Maka dari itu, penulis memohon kritik dan saran yang membangun untuk menyempurnakan dalam pembuatan makalah berikutnya. Harapan penulis adalah dengan makalah ini semoga pembaca memahami dan menerapkan ilmu dasar Arduino dengan baik.

Depok, May 13, 2023

Group B5

TABLE OF CONTENTS

CHAPTER 1.....	4
INTRODUCTION.....	4
1.1 PROBLEM STATEMENT.....	4
1.3 ACCEPTANCE CRITERIA.....	6
1.4 ROLES AND RESPONSIBILITIES.....	6
1.5 TIMELINE AND MILESTONES.....	7
CHAPTER 2.....	9
IMPLEMENTATION.....	9
2.1 HARDWARE DESIGN AND SCHEMATIC.....	10
2.2 SOFTWARE DEVELOPMENT.....	11
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	13
CHAPTER 3.....	14
TESTING AND EVALUATION.....	15
3.1 TESTING.....	15
3.2 RESULT.....	16
3.3 EVALUATION.....	16
CHAPTER 4.....	19
CONCLUSION.....	19

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Internet of Things (IoT) mengacu pada jaringan perangkat fisik yang saling terhubung yang tertanam dengan sensor, perangkat lunak, dan kemampuan konektivitas yang memungkinkan mereka untuk mengumpulkan dan bertukar data melalui internet. IoT telah menjadi suatu tren dalam kehidupan kita sehari-hari, mengubah cara kita berinteraksi dengan teknologi dan dunia di sekitar kita. IoT telah membuka peluang untuk inovasi, otomatisasi, dan efisiensi di berbagai bidang, salah satunya adalah pagar otomatis untuk kendaraan.

Terutama pada kota-kota besar di mana hampir semua orang memiliki kendaraan pribadi seperti mobil. Sehingga akan dibutuhkan lahan untuk memarkirkan kendaraan pribadi tersebut agar aman dan tidak mengganggu jalan. Umumnya masing-masing kendaraan pribadi memiliki garasi atau lahan sendiri di halaman rumah, atau menyewa suatu lahan parkir besar untuk memarkirkan mobilnya di sana. Mobil merupakan suatu barang yang cukup mewah, sehingga dibutuhkan jaminan keamanan dan kenyamanan pengendara untuk meninggalkan mobilnya di lahan yang disediakan.

Seiring dengan perkembangan zaman, operasi manual di mana membuka gerbang pagar parkir menyadarkan bahwa keberadaan IoT memberikan metode yang lebih efisien dan nyaman untuk pengendara. Di mana dengan pagar parkir manual masih memiliki kekurangan, seperti keamanan dan terutama saat kondisi cuaca buruk perlu melakukan pengoperasian manual di mana pengendara harus turun dari mobil terlebih dahulu, di mana menyulitkan pengendara. Maka untuk mengatasi permasalahan tersebut, teknologi gerbang pagar otomatis untuk lahan parkir dapat diaplikasikan.

Tujuan dari proyek ini adalah untuk membuat sistem yang secara otomatis membuka dan menutup gerbang garasi berdasarkan input deteksi objek oleh sensor. Sistem ini dapat mendeteksi keberadaan kendaraan dan merespons dengan tepat, memberikan kenyamanan dan keamanan, kemudian membuka gerbang secara otomatis.

1.2 PROPOSED SOLUTION

Proyek berjudul *Automatic House Gate*, yaitu membuat gerbang otomatis yang akan mendeteksi objek dalam jangkauan yang ditetapkan dan mengatur keluar masuk kendaraan seperti mobil memasuki lahan untuk memarkir. *Automatic House Gate* bekerja secara *stopping system* di mana menghentikan kendaraan menggunakan *stopper* yang sepenuhnya otomatis dan diaktifkan ketika ada kendaraan yang lewat di depan sensor, atau beberapa waktu diaktifkan melalui tombol. Tools yang digunakan adalah Arduino IDE, Arduino Uno, dan rangkaian elektronika.

Pada proyek kali ini sistem akan mendeteksi adanya suatu objek hambatan, yaitu kendaraan mobil saat dalam jangkauan tertentu dari *stopper*. Pendeksi menggunakan sensor ultrasonik, yaitu HC-SR04. Sistem secara keseluruhan melakukan empat mekanisme, yaitu: sensor HC-ST04 mengukur jarak antara sensor dengan objek penghalang (mobil), mendeteksi kendaraan tersebut, *gate control*, serta timing dan delay agar mekanisme sistem berjalan dengan baik.

Pertama, sensor HC-SR04 akan mengukur jarak antara dirinya dan kendaraan yang mendekat dengan menggunakan gelombang ultrasonik. Hal ini memungkinkan sistem untuk menentukan jarak kendaraan. Kemudian kode yang diimplementasikan dalam Arduino akan mendeteksi kendaraan tersebut, dengan cara membaca jarak yang diukur oleh sensor HC-SR04. Jika jarak yang diukur berada dalam ambang batas yang telah ditentukan (menunjukkan keberadaan kendaraan), akan mengidentifikasi bahwa ada kendaraan yang datang atau pergi.

Kemudian gerbang akan diatur ketika sebuah kendaraan mendekati gerbang masuk, dan jarak yang diukur oleh sensor menunjukkan keberadaannya, Arduino mengirimkan sinyal ke motor untuk membuka gerbang gerbang pagar, yang memungkinkan kendaraan untuk masuk. Sebaliknya, ketika kendaraan terdeteksi meninggalkan tempat parkir, Arduino memberi sinyal kepada motor untuk membuka gerbang agar kendaraan dapat keluar.

Selain itu pada kode Arduino juga mengimplementasikan waktu timing dan delay. Hal ini untuk memastikan kelancaran dan mencegah tabrakan, sistem dapat menggabungkan mekanisme pengaturan waktu dan penundaan. Misalnya, setelah kendaraan masuk, gerbang dapat tetap terbuka untuk jangka waktu tertentu agar kendaraan dapat melewati sepenuhnya sebelum ditutup. Demikian pula, setelah kendaraan keluar, mungkin ada sedikit delay sebelum gerbang ditutup kembali untuk menghindari halangan yang tidak diinginkan.

1.3 ACCEPTANCE CRITERIA

Tujuan dari pembuatan proyek akhir adalah sebagai berikut:

1. Dapat mengimplementasikan materi dan modul-modul praktikum Sistem Siber Fisik
2. Dapat membuat sebuah sistem pendekripsi objek dan pagar otomatis
3. Dapat mengimplementasikan sistem Arduino menggunakan kode Assembly

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Anggota	Sketch Design Menentukan Ide Proyek Kode Laporan	Fahrezy H.
Anggota	Membuat laporan Menentukan Ide Proyek Rangkaian	Fayza Nirwasita
Anggota	Kode Rangkaian Laporan	Ivan Indrastata Ramadhan
Anggota	Rangkaian Laporan PPT	Mochammad Dyenta Dwiantomitara

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

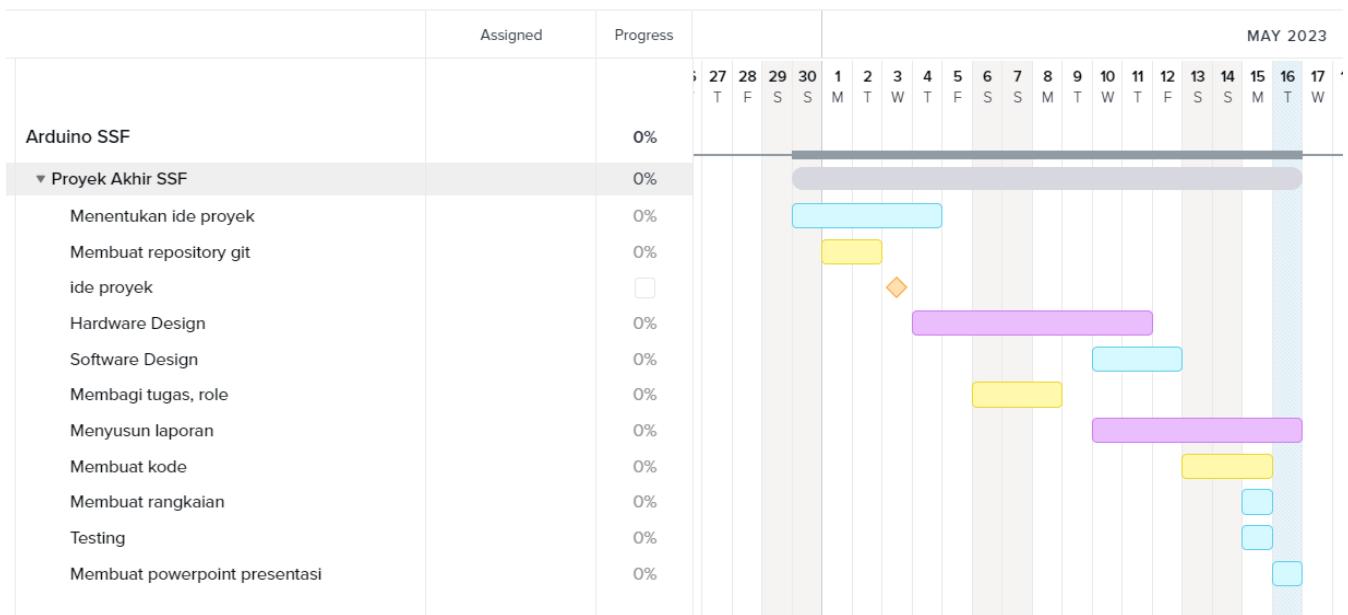


Figure. Project Timeline in Gantt Chart

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

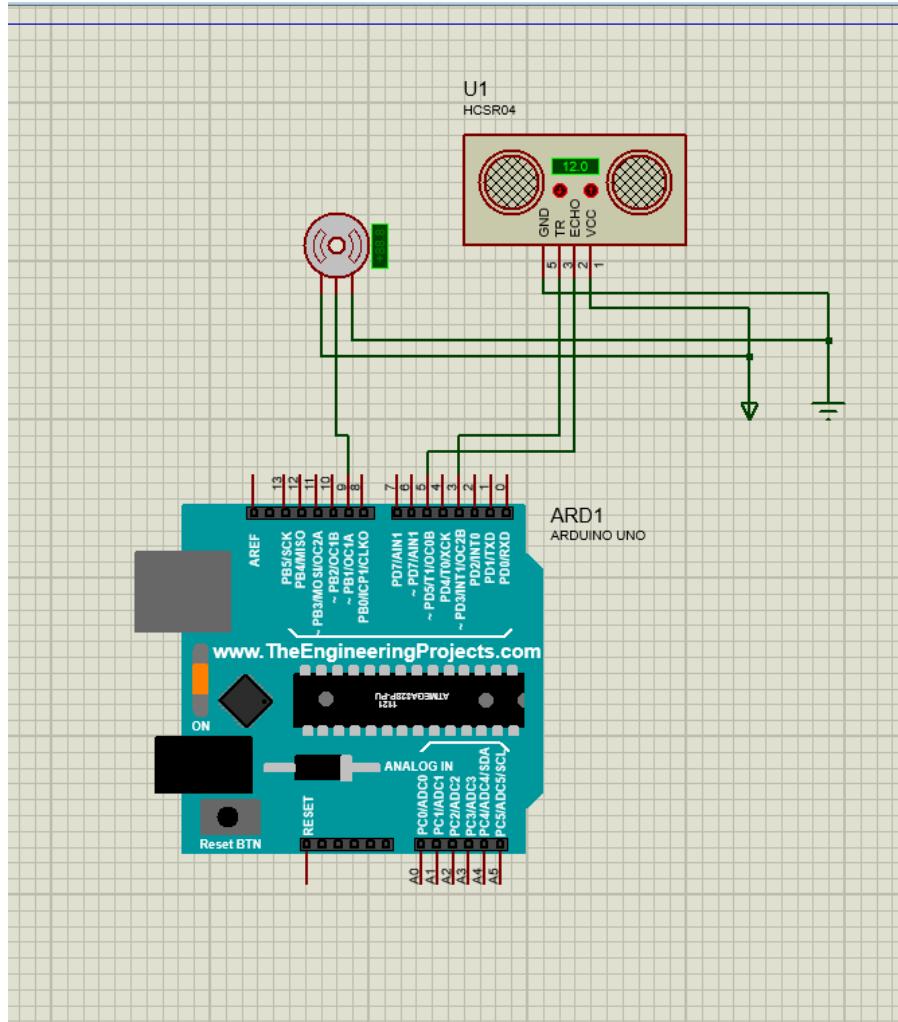


Figure. Schematic Hardware on Proteus

Desain hardware dilakukan pertama dengan menyusun rangkaian pada simulator secara software-based. Software untuk membangun hardware adalah proteus, menjadi prototipe dari skema rangkaian sebelum diimplementasikan secara langsung ke Arduino UNO secara fisik. Desain dan skema hardware dilakukan pada software terlebih dahulu bertujuan mengurangi biaya, meningkatkan efisiensi, dan memudahkan konfigurasi dan modifikasi komponen-komponen yang akan disusun

secara physical. Pada skema komponen-komponen yang digunakan di antaranya satu buah Arduino UNO, satu buah servo motor, dan satu buah sensor ultrasonic HC-SR04. Selain itu juga konektor antara komponen digunakan sesuai kebutuhan, yaitu kabel, di mana pada konfigurasi secara fisik Arduino menggunakan jumper.

Berikut adalah penjelasan lebih mendalam mengenai komponen skema hardware yang digunakan:

1. Arduino Uno:

papan mikrokontroler yang berbasis mikrokontroler ATmega328P. Di mana tipe ini salah satu papan Arduino yang paling umum digunakan untuk membangun Internet of Things atau prototyping. Papan memiliki 14 pin input/output digital (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, kristal kuarsa 16 MHz, koneksi USB, dan colokan listrik. Komponen-komponen input/output yang dibutuhkan akan dihubungkan melalui pin pada Arduino UNO. Begitu juga power dan ground.

2. Mini Motor Servo:

Jenis motor yang digunakan untuk mengontrol posisi suatu objek. Motor terdiri motor DC, rangkaian roda gigi, dan sirkuit kontrol. Motor servo akan diatur posisi rotasinya dengan mengirimkan sinyal ke sirkuit kontrol motor. Saat objek dideteksi mendekat dalam jarak yang ditentukan, motor akan menggerakkan posisi sebesar 90 derajat. Saat objek menjauh, maka motor kembali ke posisi semula yaitu 0 derajat.

3. Sensor Ultrasonik HC-SR04

Sensor ultrasonik yang digunakan dalam proyek untuk mengukur jarak. Sensor bekerja dengan mengirimkan gelombang suara berfrekuensi tinggi dan mengukur waktu yang dibutuhkan gelombang untuk memantul kembali. Sensor dapat mendeteksi jarak dari 2 cm hingga 400 cm dengan akurasi hingga 3 mm. Dalam proyek, fungsi dalam sensor melakukan pengukuran durasi pulsa ultrasonik, di mana durasi dapat diubah menjadi jarak menggunakan kecepatan suara.

2.2 SOFTWARE DEVELOPMENT

Kode program untuk mengatur arduino dalam melaksanakan sistem otomasi gerbang untuk parkir. Di mana gerbang tersebut akan terbuka dengan sensor benda yang ada didepannya. Kode di bawah merupakan contoh program untuk mengontrol sebuah servo motor dan mengukur jarak menggunakan sensor ultrasonik.

```
1 #define __SFR_OFFSET 0x00
2 #include "avr/io.h"
3
4 .global main
5
6 main:
7     SBI DDRB, 1      ; Mengatur Pin PB1 (Pin 9 pada arduino) sebagai output
8     CBI DDRB, 0      ; Mengatur pin PB0 (Pin 8 pada arduino) sebagai input
9     RCALL init_serial ;subroutine untuk inisialisasi serial
10    sei
11
12 loop: ; Subroutine yang akan dilakukan berulang-ulang
13
14 ; Proses mendapatkan jarak dari sensor HC-SR04
15     SBI PORTB, 1
16     RCALL delay_timer0_SENSOR
17     CBI PORTB, 1      ;send 10us high pulse to sensor
18
19     RCALL echo_PW      ;Subroutine untuk membaca sensor
20
21     RCALL byte2decimal ;subroutine untuk menghitung jarak yang terbaca
22     RCALL LCD_buffer ;subroutine untuk mengecek Flag UDRE0
23     LDI R26, 0x0A
24     STS UDR0, R26 ;print newline
25     RCALL LCD_buffer ;Subroutine untuk mengecek Flag UDRE0
26     LDI R26, 0x0D
27     STS UDR0, R26 ;print carriage return
28     RJMP loop
```

```
init_serial:
CLR R24 ; Meng-clear R24 untuk meng-clear UCSR0A dan UBRR0H
STS UCSR0A, R24 ; Menyimpan isi dari R24 yang sudah di-clear ke UCSR04
STS UBRR0H, R24 ; Menyimpan isi dari R24 yang sudah di-clear ke UBRR0H
LDI R24, 103 ; Mengisi R24 untuk mengisi UBRR0L
STS UBRR0L, R24 ; Menetapkan baud rate dengan menyimpan isi dari R24
LDI R24, 1<<RXEN0 | 1<<TXEN0 ; Meng-enable RXB dan TXB
STS UCSR0B, R24 ; Menyimpan isi dari R24
LDI R24, 1<<UCSZ00 | 1<<UCSZ01 ; Async, no parity, 1 stop, 8 bits
STS UCSR0C, R24; Menyimpan isi dari R24
RET
```

```

42 echo_PW: Subroutine untuk membaca sensor
43 ;-----
44 LDI R20, 0b00000000
45 STS TCCR1A, R20      ;Timer 1 normal mode
46 LDI R20, 0b11000101 ;set for rising edge detection &
47 STS TCCR1B, R20      ;prescaler=1024, noise cancellation ON
48 ;-----
49 l1: IN   R21, TIFR1
50 SBRS R21, ICF1
51 RJMP l1                ;loop until rising edge is detected
52 ;-----
53 LDS  R16, ICR1L      ;store count value at rising edge
54 ;-----
55 OUT  TIFR1, R21      ;clear flag for falling edge detection
56 LDI  R20, 0b10000101
57 STS  TCCR1B, R20      ;set for falling edge detection
58 ;-----
59 l2: IN   R21, TIFR1
60 SBRS R21, ICF1
61 RJMP l2                ;loop until falling edge is detected
62 ;-----
63 LDS  R28, ICR1L      ;store count value at falling edge
64 ;-----
65 SUB  R28, R16          ;count diff R22 = R22 - R16
66 OUT  TIFR1, R21      ;clear flag for next sensor reading
67    RET
68

```

```

68
69 byte2decimal: ;Subroutine untuk menghitung jarak yang terbaca pada sensor
70 ;-----
71    CLR  R26            ;set counter1, initial value 0
72    CLR  R27            ;set counter2, initial value 0
73    ;
74 l70: CPI  R28, 100      ;compare R28 with 100
75 Ret: BRMI 180           ;jump when R28 < 100
76    INC  R26            ;increment counter1 by 1
77    SUBI R28, 100          ;R28 = R28 - 100
78    RJMP l70
79    ;
80 l80: CPI  R28, 10          ;compare R28 with 10
81    BRMI dsp             ;jump when R28 < 10
82    INC  R27            ;increment counter2 by 1
83    SUBI R28, 10          ;R28 = R28 - 10
84    RJMP l80
85    ;
86 dsp:

```

```
87    CPI R27, 1
88    BRSH close ;Jika jarak lebih dari 10 cm, branch ke close
89    CPI R27, 0
90    BREQ open ;Jika jarak kurang dari 10 cm, branch ke open
91    RET
92
93    close: ;Subroutine yang dijalankan ketika jarak < 10 cm
94        LDI R30, lo8(message_close) ; Me-load low-byte dari pesan yang diinginkan ke Z
95        LDI R31, hi8(message_close) ; Me-load high-byte dari pesan yang diinginkan ke Z
96        RCALL agn
97        LDI R26, 1
98        LDI ZL, lo8(rotate_pos_close) ;Berisi derajat dari perputaran servo
99        LDI ZH, hi8(rotate_pos_close)
100       RCALL loop_servo ;Subroutine untuk memutar servo
101      RET
102
103    open: ;Subroutine yang dijalankan ketika jarak > 10 cm
104        LDI R30, lo8(message_open) ; Me-load low-byte dari pesan yang diinginkan ke Z
105        LDI R31, hi8(message_open) ; Me-load high-byte dari pesan yang diinginkan ke Z
106        RCALL agn
107        LDI R26, 1
108        LDI ZL, lo8(rotate_pos_open);Berisi derajat dari perputaran servo
109        LDI ZH, hi8(rotate_pos_open)
110        RCALL loop_servo
111      RET
112
113
```

```

114 agn: ;Subroutine untuk menampilkan pesan di Serial Monitor
115     LPM R29, Z+ ; Me-load satu char dari strings yang disimpan di program memory Z
116     CPI 29, 0 ; Meng-compare apakah R19 berisi 0
117     BREQ ext ; Jika nol, program akan branch ke ext
118     RCALL LCD_buffer
119     STS UDR0, R29 ; Menyimpan char ke UDR0 yang akan ditampilkan di serial monitor
120     RJMP agn ; kembali ke agn sampai R19 bernilai 0
121 ext:
122     RET
123
124 loop_servo: ;Subroutine untuk memutar Servo
125     LPM R24, Z+ ;load rotation pos
126     RCALL rotate_servo ;& rotate servo
127     DEC R26
128     BRNE loop_servo
129     RET ; kembali
130
131 rotate_servo:
132 ;-----
133     LDI R20, 10 ;count to give enough cycles of PWM
134 l3: SBI PORTB, 4
135     RCALL delay_timer0_SERVO
136     CBI PORTB, 4 ;send msec pulse to rotate servo
137     RCALL delay_20ms ;wait 20ms before re-sending pulse
138     DEC R20
139     BRNE l3 ;go back & repeat PWM signal
140     RCALL delay_ms ;0.5s delay
141     RET ;& return to main subroutine

```

```

143 message_close: ;Pesan ketika jarak < 10 cm
144     .ascii "Pintu Tertutup..." ; Pesan yang diinginkan
145     .byte 0
146
147 rotate_pos_close: ;Besar derajat servo ketika jarak < 10 cm
148     .byte 40,40 ;Derajat yang diinginkan
149
150 message_open: ;Pesan ketika jarak > 10 cm
151     .ascii "Pintu Terbuka.." ; Pesan yang diinginkan
152     .byte 0
153
154 rotate_pos_open: ;Besar derajat servo ketika jarak > 10 cm
155     .byte 100,100 ;Derajat yang diinginkan
156
157 ;=====
158 ; Subroutine2 yang berhubungan dengan delay
159 ;=====
160

```

```

161    delay_timer0_SENSOR:           ;10 usec delay via Timer 0
162    ;-----
163    CLR   R20
164    OUT   TCNT0, R20      ;initialize timer0 with count=0
165    LDI   R20, 20
166    OUT   OCR0A, R20     ;OCR0 = 20
167    LDI   R20, 0b00001010
168    OUT   TCCR0B, R20    ;timer0: CTC mode, prescaler 8
169    ;
170    l0: IN    R20, TIFR0    ;get TIFR0 byte & check
171    SBRS  R20, OCF0A      ;if OCF0=1, skip next instruction
172    RJMP  l0              ;else, loop back & check OCF0 flag
173    ;
174    CLR   R20
175    OUT   TCCR0B, R20    ;stop timer0
176    ;
177    LDI   R20, (1<<OCF0A)
178    OUT   TIFR0, R20      ;clear OCF0 flag
179    RET
180
181    ;=====
182    delay_timer0_SERVO:        ;delay via Timer0
183    ;-----
184    CLR   R21
185    OUT   TCNT0, R21      ;initialize timer0 with count=0
186    MOV   R21, R24
187    OUT   OCR0A, R21
188    LDI   R21, 0b00001100
189    OUT   TCCR0B, R21    ;timer0: CTC mode, prescaler 256
190    ;

```

```

191  loop2: IN    R21, TIFR0      ;get TIFR0 byte & check
192  |     SBRS R21, OCF0A      ;if OCF0=1, skip next instruction
193  |     RJMP loop2          ;else, loop back & check OCF0 flag
194  ;
195  CLR   R21
196  OUT   TCCR0B, R21       ;stop timer0
197  ;
198  LDI   R21, (1<<OCF0A)
199  OUT   TIFR0, R21        ;clear OCF0 flag
200  RET
201
202  delay_20ms:             ;delay 20ms
203  |     LDI   R21, 255
204  |     l7: LDI   R22, 210
205  |     l5: LDI   R23, 2
206  |     l6: DEC  R23
207  |     BRNE R22, l6
208  |     DEC   R22
209  |     BRNE R21, l5
210  |     DEC   R21
211  |     BRNE R22, l7
212  |     RET
213

```

```

214  delay_ms:
215  ;
216  ldi r25, hi8(200) ; Memasukkan nilai delay dalam satuan ms ke r25
217  ldi r24, lo8(200) ; Memasukkan nilai delay dalam satuan ms ke r24
218  outerLoop:
219  ; karena innerLoop berjalan selama 4 cycle, maka di-loop 4000x sehingga menghasilkan
220  ; 1 millisecond (4 x 4000 : 16Mhz = 10^-3)
221  ldi r31, hi8(4000) ; mengisi r30 dan r31 dengan 4000
222  ldi r30, lo8(4000)
223  innerLoop:
224  sbiw r30,1 ; Mengurangi isi r30 dan r31 sebanyak 1
225  brne innerLoop ; Jika hasil pengurangan baris sebelumnya tidak menghasilkan nol,
226  ; innerLoop akan terus diulang
227  ;----- (end of innerLoop)
228
229  sbiw r24, 1 ; Mengurangi r24 dan r25 sebanyak 1
230  brne outerLoop; Jika hasil pengurangan baris sebelumnya tidak menghasilkan nol,
231  ; maka outerLoop akan terus diulang dan innerLoop akan dijalankan kembali
232  RET
233  ;
234  ;
235  ;=====
236  ;Subroutine2 yang berhubungan dengan proses menampilkan pesan di Serial monitor
237  ;=====
238  LCD_buffer:
239  LDS R17, UCSR0A
240  SBRS R17, UDRE0 ;test data buffer if data can be sent
241  RJMP LCD_buffer
242  RET
243

```

```

244 ASCII_MSD: ; Proses mendapatkan ASCII dari MSD
245     MOV R23, R16 ;save copy of result
246     ANDI R16, 0xF0 ; extract & swap high-nibble
247     SWAP R16
248     SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
249     MOV R28, R16 ;save a copy of high-byte result
250     SUBI R28, 58 ;if +ve
251     BRPL A_F_D1 ;branch & add 7 to get ASCII A to F
252 14: RET
253
254 ASCII_LSD: ; Proses mendapatkan ASCII dari LSD
255     MOV R16, R23 ;restore copy of result
256     ANDI R16, 0x0F ;extract low-nibble
257     SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
258     MOV R28, R16 ;save a copy of high-byte result
259     SUBI R28, 58 ;if +ve
260     BRPL A_F_D0 ;branch & add 7 to get ASCII A to F
261
262 19: RET
263 A_F_D1:
264     SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
265     RJMP 14
266 A_F_D0:
267     SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
268     RJMP 19

```

Figure. Code in C program

Kode ini mengimplementasikan program yang berinteraksi dengan sensor jarak ultrasonik (HC-SR04) dan motor servo untuk mengontrol pintu berdasarkan jarak yang terdeteksi.

1. Menginisialisasi konfigurasi dan pengaturan yang diperlukan untuk mikrokontroler, termasuk mengatur pin sebagai input atau output, menginisialisasi komunikasi serial, dan mengaktifkan interupsi
2. Menyiapkan loop yang berulang kali melakukan proses pengukuran jarak
3. Mengirimkan pulsa tinggi $10\mu\text{s}$ ke sensor ultrasonik untuk memicu pengukuran jarak.
4. Mengukur lebar pulsa gema dari sensor ultrasonik untuk menghitung jarak.
5. Mengubah jarak yang diukur menjadi representasi desimal.
6. Mencetak nilai jarak ke monitor serial.
7. Menentukan apakah jarak yang diukur lebih besar atau kurang dari 10 cm.
8. Jika jaraknya kurang dari 10 cm, maka akan mengirimkan pesan "Pintu Tertutup..." ke monitor serial dan memutar servo ke posisi tertentu (tutup).

9. Jika jaraknya lebih besar dari 10 cm, program akan mengirimkan pesan "Pintu Terbuka..." ke monitor serial dan memutar servo ke posisi lain (buka).
10. Program ini mencakup berbagai subrutin untuk menangani penundaan, komunikasi serial, dan kontrol servo.

Flowchart:

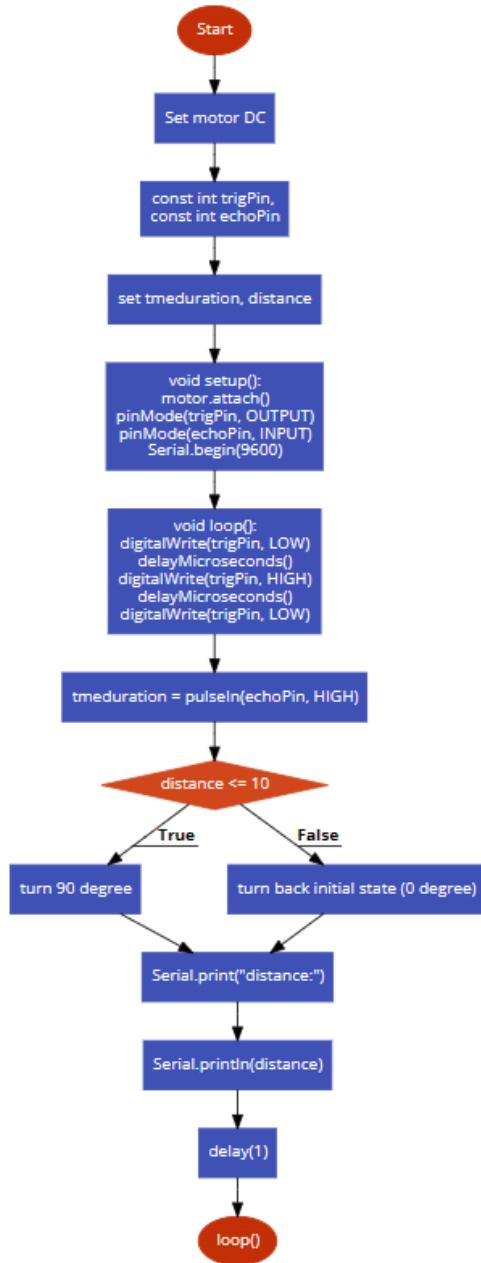


Figure. Flowchart

2.3 HARDWARE AND SOFTWARE INTEGRATION

Rangkaian pertama dibuat menggunakan kode C dan simulasi arduino untuk mendapatkan konsep dari sistem yang ingin dibuat. Di gambar di bawah ini adalah kondisi

ketika mendeteksi objek di bawah jarak 10 cm, lalu gambar kedua menunjukkan kondisi ketika objek menjauh (jarak > 10 cm).

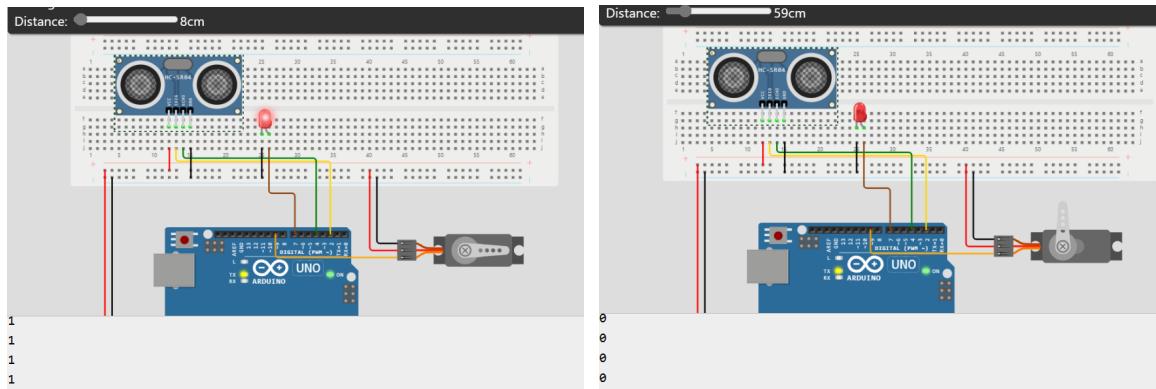


Figure. Project Implementation on wokwi.com

Kode implementasi C:

```

1 #include <Servo.h>
2
3 Servo myservo;
4 const int trigPin = 2;
5 const int echoPin = 4;
6 long tmeduration;
7 int distance;
8 const int LED_PIN = 7;
9
10 void setup() {
11     myservo.attach(9);
12     pinMode(trigPin, OUTPUT);
13     pinMode(echoPin, INPUT);
14     pinMode(LED_PIN, OUTPUT);
15     Serial.begin(9600);
16 }
17
18 void loop() {
19     digitalWrite(trigPin, LOW);
20     delayMicroseconds(2);
21     digitalWrite(trigPin, HIGH);
22     delayMicroseconds(10);
23     digitalWrite(trigPin, LOW);
24
25     tmeduration = pulseIn(echoPin, HIGH);
26     distance = (0.034 * tmeduration) / 2;
27
28     if (distance <= 10) {
29         Serial.println("1");
30         digitalWrite(LED_PIN, HIGH);
31         myservo.write(90);
32     } else {
33         Serial.println("0");
34         digitalWrite(LED_PIN, LOW);
35         myservo.write(0);
36     }
37
38     delay(1);
39 }
40

```

Figure. Project Code Implementation in C program

Hasil rangkaian menunjukkan implementasi konsep dapat dilakukan pada kehidupan nyata. Di mana sistem akan mendeteksi objek penghalang (mobil) yang ditangkap oleh sensor ultrasonik HC-SR04 yang mengukur jarak objek tersebut. Kemudian jika jarak objek sangat dekat (<10 cm) menandakan objek itu mendekat ke gate, sehingga servo motor akan memutar rotasi sebesar 90 derajat. Saat objek menjauh maka servo motor kembali ke posisi semula. Saat objek mendekat, maka lampu LED menjadi nyala dan pada serial monitor menampilkan angka ‘1’. Sedangkan saat objek menjauh (>10 cm) maka LED akan mati dan pada serial monitor menampilkan angka ‘0’.

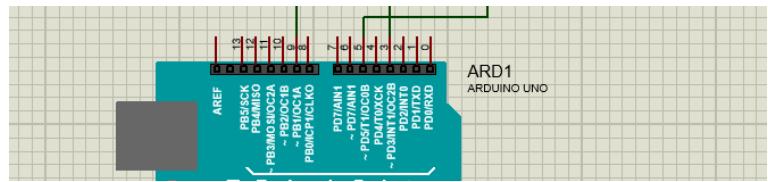


Figure. Pins from Arduino

Kemudian rangkaian diimplementasikan melalui simulasi pada proteus. Rangkaian yang menghubungkan output dengan arduino semuanya menggunakan pin digital. Di mana pin 3 Digital (PD5) terhubung pada HC-SR04 yaitu sensor ultrasonik. Kemudian pin 9 digital (PB1) terhubung pada Micro Servo jenis SG90. Sensor dan Micro Servo masing-masing terhubung pada GND dan VCC pada arduino board.

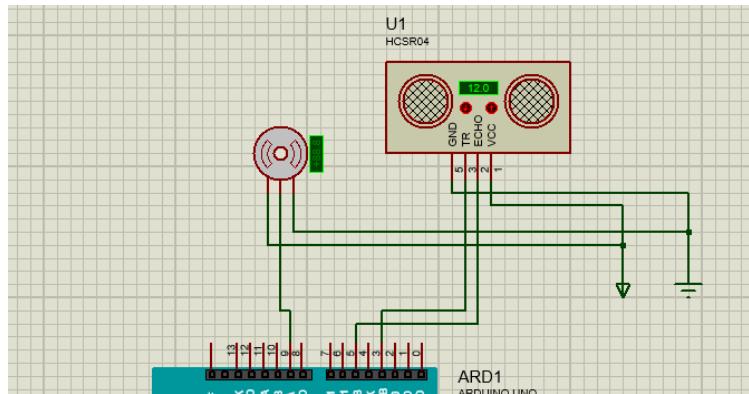


Figure. Components of the System

Salah satu komponen yang terhubung ke Arduino UNO adalah sensor ultrasonik HC-SR04. Sensor ultrasonik HC-SR04 bekerja dengan memancarkan gelombang suara dan mengukur waktu yang dibutuhkan gelombang untuk memantul kembali. Ketika dipicu, sensor ini akan mengirimkan gelombang ultrasonik dan menunggu gelombang tersebut mengenai suatu objek dan memantul kembali. Dengan mengukur waktu yang dibutuhkan, Arduino dapat menghitung jarak ke objek. Hal ini memungkinkan untuk dapat mendeteksi dan mengukur jarak, yaitu mendeteksi objek yang mendekat yaitu kendaraan seperti mobil.

Komponen selanjutnya adalah servo motor, menggunakan jenis SG90. Di mana jenis motor ini digunakan untuk mengontrol posisi objek dalam proyek Arduino. Motor bekerja dengan menerima sinyal dari Arduino yang menentukan posisi yang diinginkan. Dalam kode posisi motor yang digunakan adalah 0 derajat (posisi semula) dan 90 derajat. Motor servo akan bergerak sesuai sinyal yang diterimanya.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Dari kode yang sudah dibuat, di rangkailah sebuah rangkaian arduino yang mengimplementasikan sensor HC-SR04 yang akan menerima input berupa gelombang ultrasonik yaitu berupa jarak benda ke sensor> Input ini kemudian disalurkan melalui arduino dan dijadikan output berupa servo motor yang akan bergerak sesuai dengan program yaitu sebesar 90 derajat yang akan membuat pagar menjadi terbuka.

Dari Rangkaian tersebut kami melakukan testing rangkaian tersebut melalui proteus dan juga rangkaian fisik dengan arduino, sensor HC-SR04 dan servo motor dengan melakukan percobaan untuk membuka gerbang dengan menggunakan perantara jarak benda dengan sensor dimana jika ada benda yang mendekati sensor dengan jarak kurang lebih 10 cm maka sensor akan mengirimkan sinyal untuk menggerakkan servo motor untuk mengarah keatas atau 90 derajat.

```
init_serial:  
    CLR R24 ; Meng-clear R24 untuk meng-clear UCSR0A dan UBRR0H  
    STS UCSR0A, R24 ; Menyimpan isi dari R24 yang sudah di-clear ke UCSR04  
    STS UBRR0H, R24 ; Menyimpan isi dari R24 yang sudah di-clear ke UBRR0H  
    LDI R24, 103 ; Mengisi R24 untuk mengisi UBRR0L  
    STS UBRR0L, R24 ; Menetapkan baud rate dengan menyimpan isi dari R24  
    LDI R24, 1<<RXEN0 | 1<<TXEN0 ; Meng-enable RXB dan TXB  
    STS UCSR0B, R24 ; Menyimpan isi dari R24  
    LDI R24, 1<<UCSZ00 | 1<<UCSZ01 ; Async, no parity, 1 stop, 8 bits  
    STS UCSR0C, R24; Menyimpan isi dari R24  
    RET
```

```

42 echo_PW: Subroutine untuk membaca sensor
43 ;-----
44 LDI R20, 0b00000000
45 STS TCCR1A, R20      ;Timer 1 normal mode
46 LDI R20, 0b11000101 ;set for rising edge detection &
47 STS TCCR1B, R20      ;prescaler=1024, noise cancellation ON
48 ;-----
49 l1: IN   R21, TIFR1
50 SBRS R21, ICF1
51 RJMP l1                ;loop until rising edge is detected
52 ;-----
53 LDS  R16, ICR1L      ;store count value at rising edge
54 ;-----
55 OUT  TIFR1, R21      ;clear flag for falling edge detection
56 LDI  R20, 0b10000101
57 STS  TCCR1B, R20      ;set for falling edge detection
58 ;-----
59 l2: IN   R21, TIFR1
60 SBRS R21, ICF1
61 RJMP l2                ;loop until falling edge is detected
62 ;-----
63 LDS  R28, ICR1L      ;store count value at falling edge
64 ;-----
65 SUB  R28, R16          ;count diff R22 = R22 - R16
66 OUT  TIFR1, R21      ;clear flag for next sensor reading
67    RET
68

```

```

68
69 byte2decimal: ;Subroutine untuk menghitung jarak yang terbaca pada sensor
70 ;-----
71    CLR  R26            ;set counter1, initial value 0
72    CLR  R27            ;set counter2, initial value 0
73    ;
74 l70: CPI  R28, 100      ;compare R28 with 100
75 Ret: BRMI 180           ;jump when R28 < 100
76    INC  R26            ;increment counter1 by 1
77    SUBI R28, 100          ;R28 = R28 - 100
78    RJMP l70
79    ;
80 l80: CPI  R28, 10          ;compare R28 with 10
81    BRMI dsp             ;jump when R28 < 10
82    INC  R27            ;increment counter2 by 1
83    SUBI R28, 10          ;R28 = R28 - 10
84    RJMP l80
85    ;
86 dsp:

```

```
87    CPI R27, 1
88    BRSH close ;Jika jarak lebih dari 10 cm, branch ke close
89    CPI R27, 0
90    BREQ open ;Jika jarak kurang dari 10 cm, branch ke open
91    RET
92
93    close: ;Subroutine yang dijalankan ketika jarak < 10 cm
94        LDI R30, lo8(message_close) ; Me-load low-byte dari pesan yang diinginkan ke Z
95        LDI R31, hi8(message_close) ; Me-load high-byte dari pesan yang diinginkan ke Z
96        RCALL agn
97        LDI R26, 1
98        LDI ZL, lo8(rotate_pos_close) ;Berisi derajat dari perputaran servo
99        LDI ZH, hi8(rotate_pos_close)
100       RCALL loop_servo ;Subroutine untuk memutar servo
101      RET
102
103    open: ;Subroutine yang dijalankan ketika jarak > 10 cm
104        LDI R30, lo8(message_open) ; Me-load low-byte dari pesan yang diinginkan ke Z
105        LDI R31, hi8(message_open) ; Me-load high-byte dari pesan yang diinginkan ke Z
106        RCALL agn
107        LDI R26, 1
108        LDI ZL, lo8(rotate_pos_open);Berisi derajat dari perputaran servo
109        LDI ZH, hi8(rotate_pos_open)
110        RCALL loop_servo
111      RET
112
113
```

```

114 agn: ;Subroutine untuk menampilkan pesan di Serial Monitor
115     LPM R29, Z+ ; Me-load satu char dari strings yang disimpan di program memory Z
116     CPI 29, 0 ; Meng-compare apakah R19 berisi 0
117     BREQ ext ; Jika nol, program akan branch ke ext
118     RCALL LCD_buffer
119     STS UDR0, R29 ; Menyimpan char ke UDR0 yang akan ditampilkan di serial monitor
120     RJMP agn ; kembali ke agn sampai R19 bernilai 0
121 ext:
122     RET
123
124 loop_servo: ;Subroutine untuk memutar Servo
125     LPM R24, Z+ ;load rotation pos
126     RCALL rotate_servo ;& rotate servo
127     DEC R26
128     BRNE loop_servo
129     RET ; kembali
130
131 rotate_servo:
132 ;-----
133     LDI R20, 10 ;count to give enough cycles of PWM
134 l3: SBI PORTB, 4
135     RCALL delay_timer0_SERVO
136     CBI PORTB, 4 ;send msec pulse to rotate servo
137     RCALL delay_20ms ;wait 20ms before re-sending pulse
138     DEC R20
139     BRNE l3 ;go back & repeat PWM signal
140     RCALL delay_ms ;0.5s delay
141     RET ;& return to main subroutine

```

```

143 message_close: ;Pesan ketika jarak < 10 cm
144     .ascii "Pintu Tertutup..." ; Pesan yang diinginkan
145     .byte 0
146
147 rotate_pos_close: ;Besar derajat servo ketika jarak < 10 cm
148     .byte 40,40 ;Derajat yang diinginkan
149
150 message_open: ;Pesan ketika jarak > 10 cm
151     .ascii "Pintu Terbuka.." ; Pesan yang diinginkan
152     .byte 0
153
154 rotate_pos_open: ;Besar derajat servo ketika jarak > 10 cm
155     .byte 100,100 ;Derajat yang diinginkan
156
157 ;=====
158 ; Subroutine2 yang berhubungan dengan delay
159 ;=====
160

```

```

161    delay_timer0_SENSOR:           ;10 usec delay via Timer 0
162    ;-----
163    CLR   R20
164    OUT   TCNT0, R20      ;initialize timer0 with count=0
165    LDI   R20, 20
166    OUT   OCR0A, R20     ;OCR0 = 20
167    LDI   R20, 0b00001010
168    OUT   TCCR0B, R20    ;timer0: CTC mode, prescaler 8
169    ;
170    l0: IN    R20, TIFR0    ;get TIFR0 byte & check
171    SBRS R20, OCF0A       ;if OCF0=1, skip next instruction
172    RJMP l0               ;else, loop back & check OCF0 flag
173    ;
174    CLR   R20
175    OUT   TCCR0B, R20    ;stop timer0
176    ;
177    LDI   R20, (1<<OCF0A)
178    OUT   TIFR0, R20      ;clear OCF0 flag
179    RET
180
181    ;=====
182    delay_timer0_SERVO:        ;delay via Timer0
183    ;-----
184    CLR   R21
185    OUT   TCNT0, R21      ;initialize timer0 with count=0
186    MOV   R21, R24
187    OUT   OCR0A, R21
188    LDI   R21, 0b00001100
189    OUT   TCCR0B, R21    ;timer0: CTC mode, prescaler 256
190    ;

```

```

191  loop2: IN    R21, TIFR0      ;get TIFR0 byte & check
192  |     SBRS R21, OCF0A      ;if OCF0=1, skip next instruction
193  |     RJMP loop2          ;else, loop back & check OCF0 flag
194  ;
195  CLR   R21
196  OUT   TCCR0B, R21       ;stop timer0
197  ;
198  LDI   R21, (1<<OCF0A)
199  OUT   TIFR0, R21        ;clear OCF0 flag
200  RET
201
202  delay_20ms:             ;delay 20ms
203  |     LDI   R21, 255
204  |     l7: LDI   R22, 210
205  |     l5: LDI   R23, 2
206  |     l6: DEC  R23
207  |     BRNE R22, l6
208  |     DEC   R22
209  |     BRNE R21, l5
210  |     DEC   R21
211  |     BRNE R22, l7
212  |     RET
213

```

```

214  delay_ms:
215  ;
216  ldi r25, hi8(200) ; Memasukkan nilai delay dalam satuan ms ke r25
217  ldi r24, lo8(200) ; Memasukkan nilai delay dalam satuan ms ke r24
218  outerLoop:
219  ; karena innerLoop berjalan selama 4 cycle, maka di-loop 4000x sehingga menghasilkan
220  ; 1 millisecond (4 x 4000 : 16Mhz = 10^-3)
221  ldi r31, hi8(4000) ; mengisi r30 dan r31 dengan 4000
222  ldi r30, lo8(4000)
223  innerLoop:
224  sbiw r30,1 ; Mengurangi isi r30 dan r31 sebanyak 1
225  brne innerLoop ; Jika hasil pengurangan baris sebelumnya tidak menghasilkan nol,
226  ; innerLoop akan terus diulang
227  ;----- (end of innerLoop)
228
229  sbiw r24, 1 ; Mengurangi r24 dan r25 sebanyak 1
230  brne outerLoop; Jika hasil pengurangan baris sebelumnya tidak menghasilkan nol,
231  ; maka outerLoop akan terus diulang dan innerLoop akan dijalankan kembali
232  RET
233  ;
234  ;
235  ;=====
236  ;Subroutine2 yang berhubungan dengan proses menampilkan pesan di Serial monitor
237  ;=====
238  LCD_buffer:
239  LDS R17, UCSR0A
240  SBRS R17, UDRE0 ;test data buffer if data can be sent
241  RJMP LCD_buffer
242  RET
243

```

```

244 ASCII_MSD: ; Proses mendapatkan ASCII dari MSD
245     MOV R23, R16 ;save copy of result
246     ANDI R16, 0xF0 ; extract & swap high-nibble
247     SWAP R16
248     SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
249     MOV R28, R16 ;save a copy of high-byte result
250     SUBI R28, 58 ;if +ve
251     BRPL A_F_D1 ;branch & add 7 to get ASCII A to F
252 14: RET
253
254 ASCII_LSD: ; Proses mendapatkan ASCII dari LSD
255     MOV R16, R23 ;restore copy of result
256     ANDI R16, 0x0F ;extract low-nibble
257     SUBI R16, -48 ;R16 = R16 - (48) = R16 + 48
258     MOV R28, R16 ;save a copy of high-byte result
259     SUBI R28, 58 ;if +ve
260     BRPL A_F_D0 ;branch & add 7 to get ASCII A to F
261
262 19: RET
263 A_F_D1:
264     SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
265     RJMP 14
266 A_F_D0:
267     SUBI R16, -7 ;R16 = R16 - (7) = R16 + 7
268     RJMP 19

```

Program di atas adalah contoh program dalam bahasa Assembly untuk mengontrol pintu berdasarkan jarak yang terdeteksi oleh sensor HC-SR04. Program ini ditulis untuk platform Arduino menggunakan mikrokontroler AVR.

Pada bagian awal program, terdapat inisialisasi pin sebagai input dan output menggunakan instruksi SBI (Set Bit) dan CBI (Clear Bit) pada register DDRB. Kemudian dilakukan inisialisasi serial dengan memanggil subroutine init_serial yang mengatur baud rate, pengaturan mode komunikasi, dan enable RX/TX.

Selanjutnya, terdapat loop utama yang akan dieksekusi secara berulang. Di dalam loop ini, program akan melakukan pengukuran jarak menggunakan sensor HC-SR04. Proses pengukuran tersebut melibatkan instruksi SBI dan CBI untuk mengirim sinyal pulsa ke sensor, serta beberapa subroutine untuk membaca sinyal echo dari sensor, menghitung jarak yang terbaca, dan menampilkan hasilnya ke Serial Monitor.

Setelah mendapatkan jarak, program akan membandingkannya dengan batas tertentu (10 cm dalam contoh ini) untuk mengambil tindakan yang sesuai. Jika jarak lebih kecil dari batas, program akan menjalankan subroutine close yang menampilkan pesan "Pintu Tertutup..." dan

memutar servo ke posisi tertentu. Jika jarak lebih besar dari batas, program akan menjalankan subroutine open yang menampilkan pesan "Pintu Terbuka.." dan memutar servo ke posisi tertentu.

Selain itu, program juga memiliki beberapa subroutine pendukung, seperti delay_timer0_SENSOR dan delay_timer0_SERVO untuk menghasilkan delay menggunakan Timer0, serta subroutine LCD_buffer untuk memastikan data dapat dikirim ke Serial Monitor.

Pada bagian akhir program, terdapat beberapa subroutine tambahan seperti ASCII_MSD dan ASCII_LSD yang digunakan untuk mengkonversi bilangan biner ke ASCII untuk menampilkan jarak dalam bentuk angka di Serial Monitor.

Program ini dapat diubah dan disesuaikan dengan kebutuhan tertentu dalam mengontrol perangkat berdasarkan jarak yang terdeteksi oleh sensor.

3.2 RESULT

Hasil dari percobaan didapatkan bahwa baik di proteus maupun rangkaian fisik, rangkaian sudah berjalan sesuai kode program dan pengaturan rangkaian dan bekerja dengan baik karena sensor sudah bisa membaca jarak benda yang mendekat dan bisa mengirimkan sinyal ke servo motor untuk membuka gerbang dengan bergerak sebesar 90 derajat.

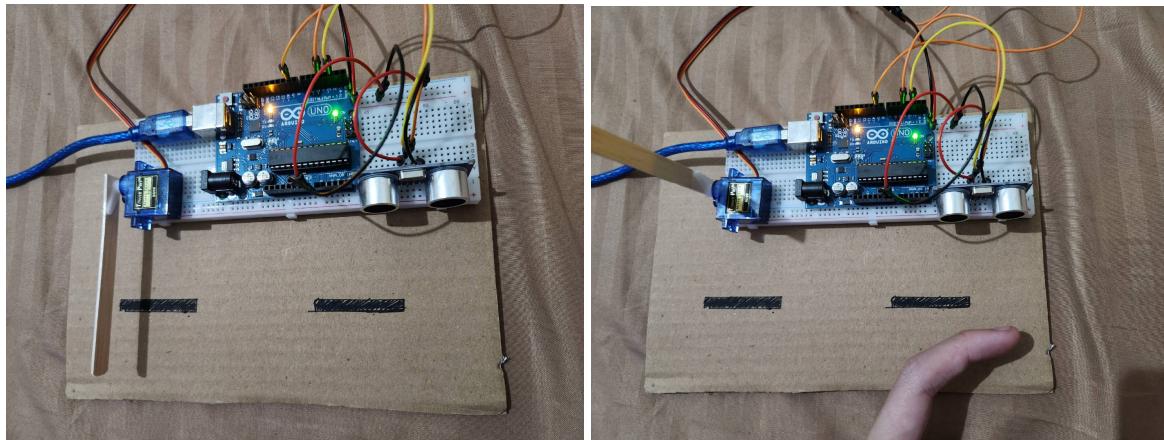


Fig 2.1 and 2.2 Testing Result

Pada Fig 2.1 Rangkaian berada pada keadaan default dimana sensor tidak membaca ada benda yang berada di dekat sensor sehingga sensor tidak mengirimkan sinyal ke servo motor yang membuat servo motor tidak melakukan pergerakan dan berada di keadaan default dan membuat pagar tertutup. Sedangkan, pada Fig 2.2 Rangkaian berada pada keadaan

menerima input yaitu input dari sensor yang mendeteksi bahwa terdapat benda didepannya sehingga sensor mengirimkan sinyal ke servo motor yang membuat servo motor bergerak sebesar 90 derajat yang membuat pagar terbuka.

3.3 EVALUATION

Program untuk sistem proyek disusun menggunakan bahasa Assembly. Sebelumnya implementasi konsep program disimulasikan menggunakan bahasa C terlebih dahulu, kemudian kode program dijalankan pada Arduino UNO yang terhubung pada laptop menggunakan port cable USB. Simulasi dilakukan pada wokwi.com dan proteus. Kode kemudian dikompilasi dan memeriksa apakah terjadi *syntax error*. Setelah tidak ditemukan *syntax error*, program melakukan inisialisasi sebelum dijalankan.

Sebelum dijalankan, rangkaian dipastikan sudah sesuai desain seperti penempatan pin, sensor ultrasonik, dan mini servo motor. Sistem rangkaian yang sudah diprogram akan melakukan deteksi objek penghalang, dalam konsep ini adalah mobil yang berada pada jangkauan sensor HC-SR04 ultrasonic. Sensor tersebut kemudian mengukur jarak objek, jika di bawah 10 cm maka dianggap objek sedang mendekat ke gerbang. Jika masih di atas 10 cm, maka tidak melakukan apapun. Kemudian dari sensor HC-SR04 ultrasonik menyesuaikan range jarak, di mana membatas range jarak sebesar 10 cm. Setelah itu, kode akan menggerakkan rotasi servo motor untuk berputar sebesar 90 derajat. Servo motor akan menggerakkan gerbang ke atas, sehingga gerbang terbuka dan mobil dapat memasuki garasi atau lahan parkir. Setelah mobil tersebut melewati gerbang dan menjauh dari sensor, maka gerbang kembali ditutup dengan delay waktu sebesar 1 milisekon. Program akan terus berulang sesuai dengan mode yang dibuat, yaitu mode membuka gerbang dan menutup gerbang, dan mode menahan saat objek tetap berada dalam jangkauan yang sama atau tidak bergerak.

CHAPTER 4

CONCLUSION

Proyek *Automatic House Gate* memberikan salah satu solusi dari kebutuhan akan metode yang efisien dan nyaman untuk mengelola tempat parkir. Proyek ini memanfaatkan Internet of Things (IoT) untuk menciptakan sistem gerbang otomatis yang mendeteksi keberadaan kendaraan dan mengatur masuk dan keluarnya kendaraan. Solusi yang diusulkan menggunakan Arduino Uno, Arduino IDE, dan sirkuit elektronik, bersama dengan sensor ultrasonik HC-SR04, untuk mengukur jarak antara sensor dan kendaraan.

Dengan menerapkan sistem ini, operasi manual dan keterbatasan yang terkait dengan gerbang parkir tradisional dapat diatasi. Sistem gerbang otomatis memastikan kenyamanan dan keamanan pengguna, memungkinkan kendaraan melewati gerbang tanpa mengharuskan pengemudi untuk membuka atau menutupnya secara manual. Sistem ini menggabungkan mekanisme timing waktu dan delay untuk memastikan kelancaran operasi dan mencegah tabrakan.

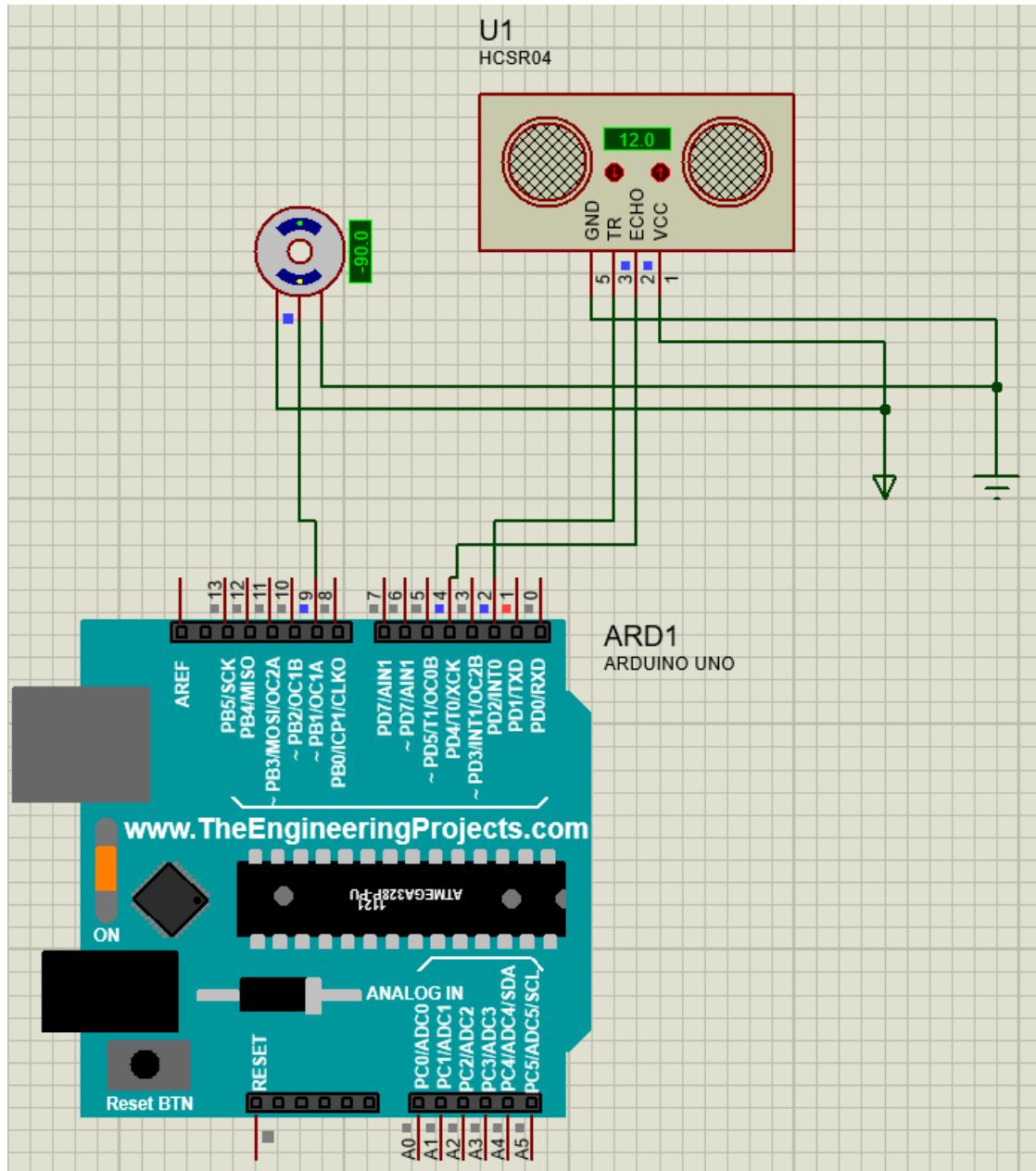
Kesimpulannya, proyek Automatic Turnstile for Car Parking menunjukkan aplikasi praktis teknologi IoT dalam meningkatkan manajemen tempat parkir. Dengan mengotomatiskan sistem gerbang dan mengintegrasikan kemampuan deteksi jarak, solusi ini menawarkan efisiensi, keamanan, dan pengalaman yang tidak merepotkan bagi pengemudi dan operator tempat parkir.

REFERENCES

- [1] Muhammad Ali Mazidi, Sarmad Naimi, and Sepehr Naimi, "The AVR Microcontroller and Embedded Systems: Using Assembly and C," Pearson New International Edition, Pearson, Essex, 2015.
- [2] Lee, Edward A., and Sanjit A. Seshia, "Introduction to Embedded Systems - A Cyber-Physical Systems Approach," 2nd edition, UC-Berkeley, 2015. [Online]. Available:
https://ptolemy.berkeley.edu/books/leeseshia/releases/Leeseshia_DigitalV2_2.pdf
- [3] "HC-SR04 Datasheet," Electroschematics. [Online]. Available:
<https://www.electroschematics.com/hc-sr04-datasheet/>. Accessed: May 7, 2023.
- [4] N. Vyas, "Arduino-Ultrasonic-Distance-Sensor-SC-SR04-With-Servo-Motor," GitHub. [Online]. Available:
<https://github.com/narayanvyas/Arduino-Ultrasonic-Distance-Sensor-SC-SR04-With-Servo-Motor>

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation

Photo by: Fayza Nirwasita & Ivan I.R.

15-16 Mei 2023 berkumpul secara offline.

