



B5

SSF 2023

Mochammad Dyenta D

Fahrezy H

Ivan Indrastata R

Fayza Nirwasita

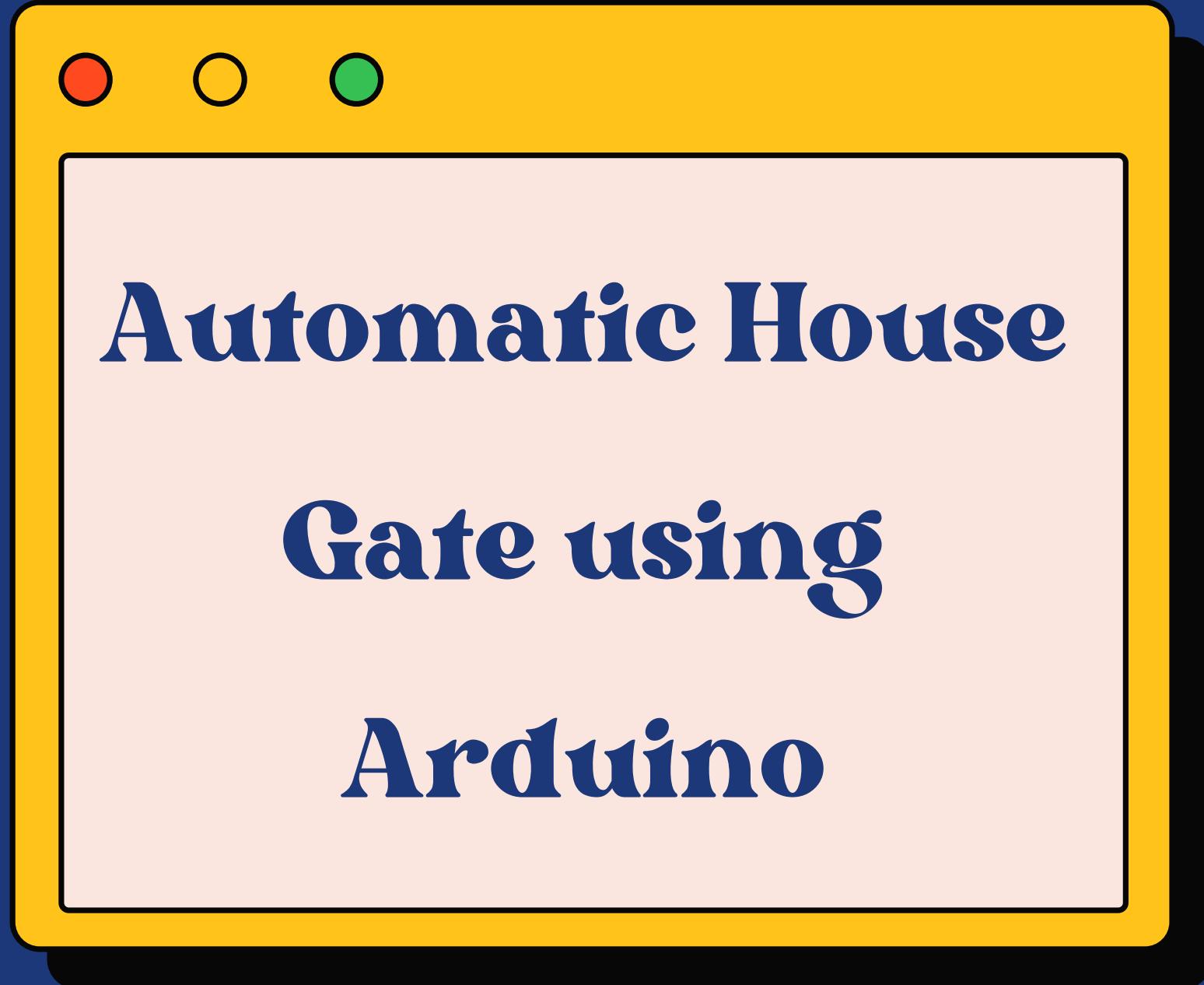
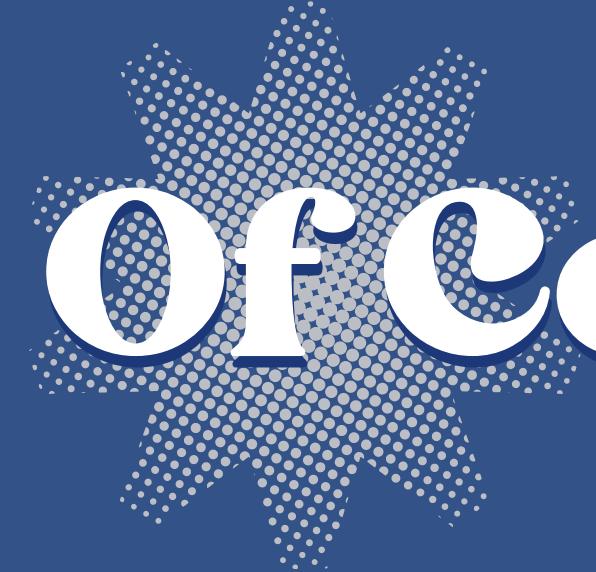


Table Of Content



Latar
Belakang

Tujuan
Proyek

Implementasi

Testing

Evaluasi

Kesimpulan



Latar Belakang

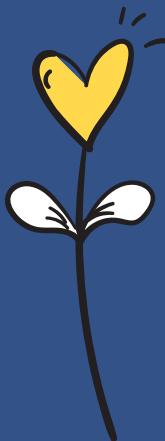
Internet of Things telah menjadi suatu tren dalam kehidupan kita sehari-hari, mengubah cara kita berinteraksi dengan teknologi dan dunia di sekitar kita. IoT telah membuka peluang untuk inovasi, otomatisasi, dan efisiensi di berbagai bidang, salah satunya adalah pagar otomatis untuk kendaraan. Kami membuat proyek automatic gate ini karena kami ingin sekali membantu penyandang disabilitas untuk mengurangi beban mereka seperti IoT yang telah kami buat ini

Terutama pada kota-kota besar di mana hampir semua orang memiliki kendaraan pribadi seperti mobil dan garasinya. Umumnya masing-masing kendaraan pribadi memiliki garasi, dan mobil adalah salah 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 Proyek



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.



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.

Deskripsi Proyek



Automatic Gate

Proyek ini dibuat dengan tujuan mempermudah manusia terutama penyandang disabilitas dan pada zaman sekarang manusia juga sangat membutuhkan alat seperti ini agar mempermudah pekerjaannya dan membuat pekerjaan nya lebih efisien

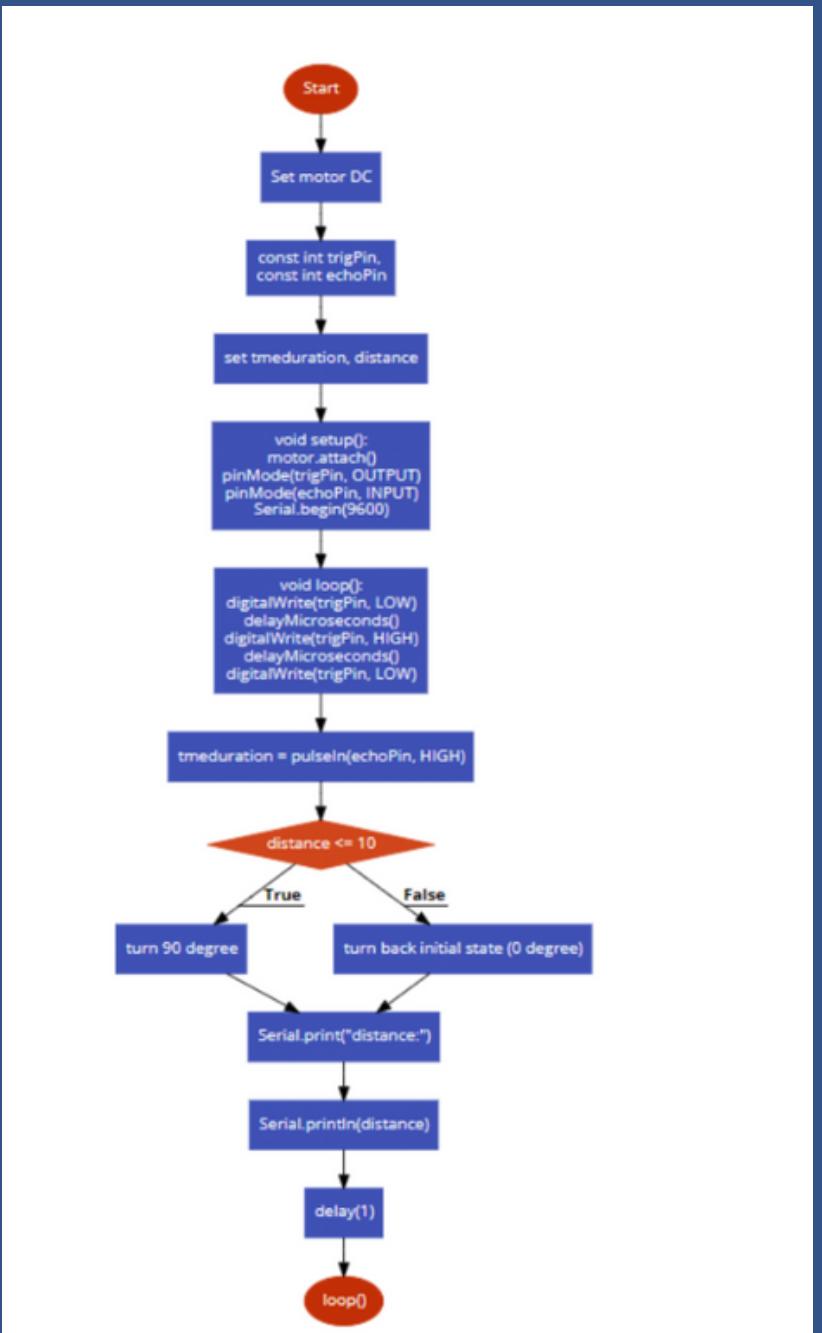
Deskripsi Proyek



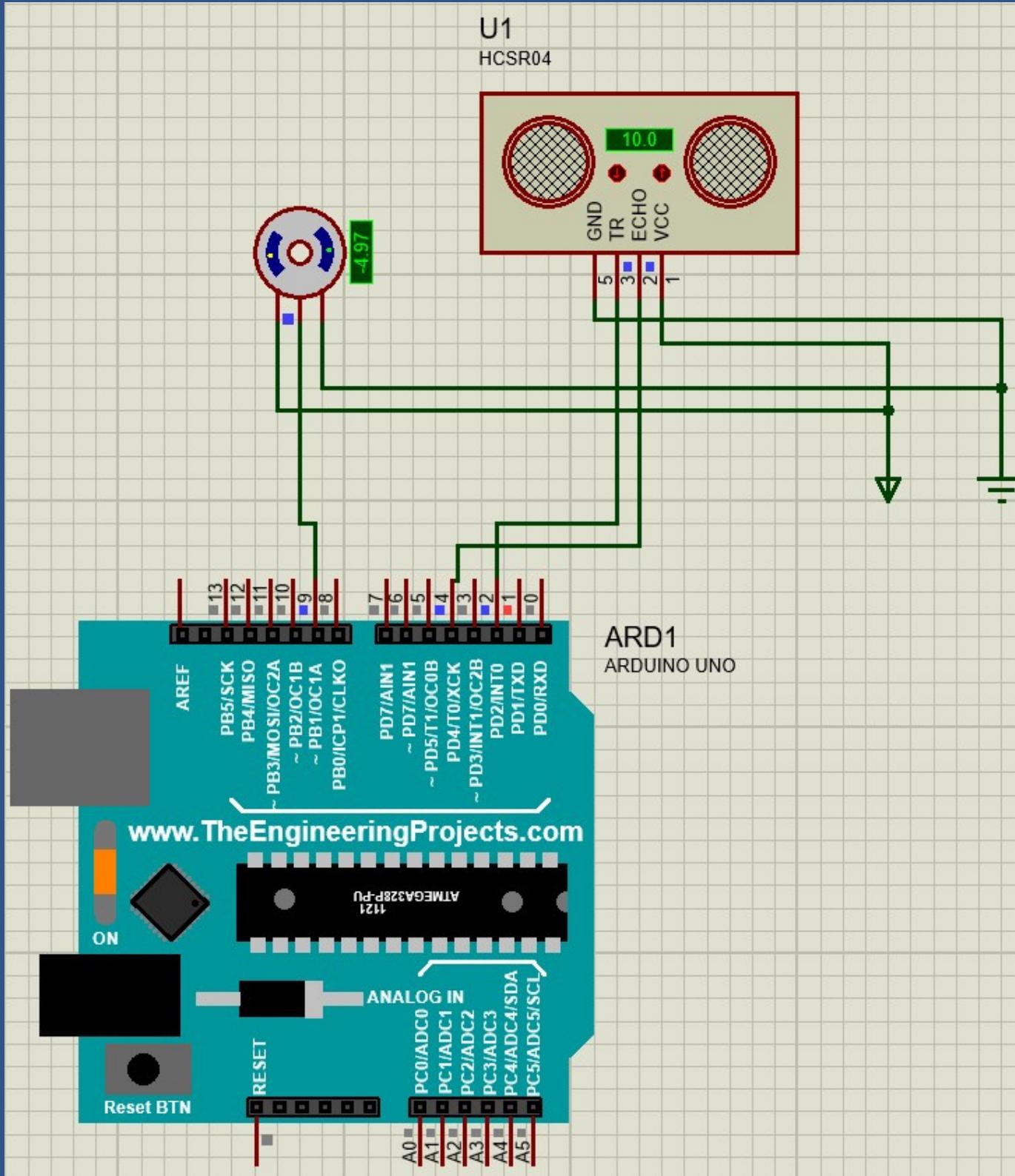
Automatic Gate

Automatic gate ini menggunakan Sensor servo motors dan juga menggunakan HCSR04

Software Development



Hardware Implementation



Arduino Uno : papan mikrokontroler yang berbasis mikrokontroler ATmega328P digunakan sebagai otak pemrograman

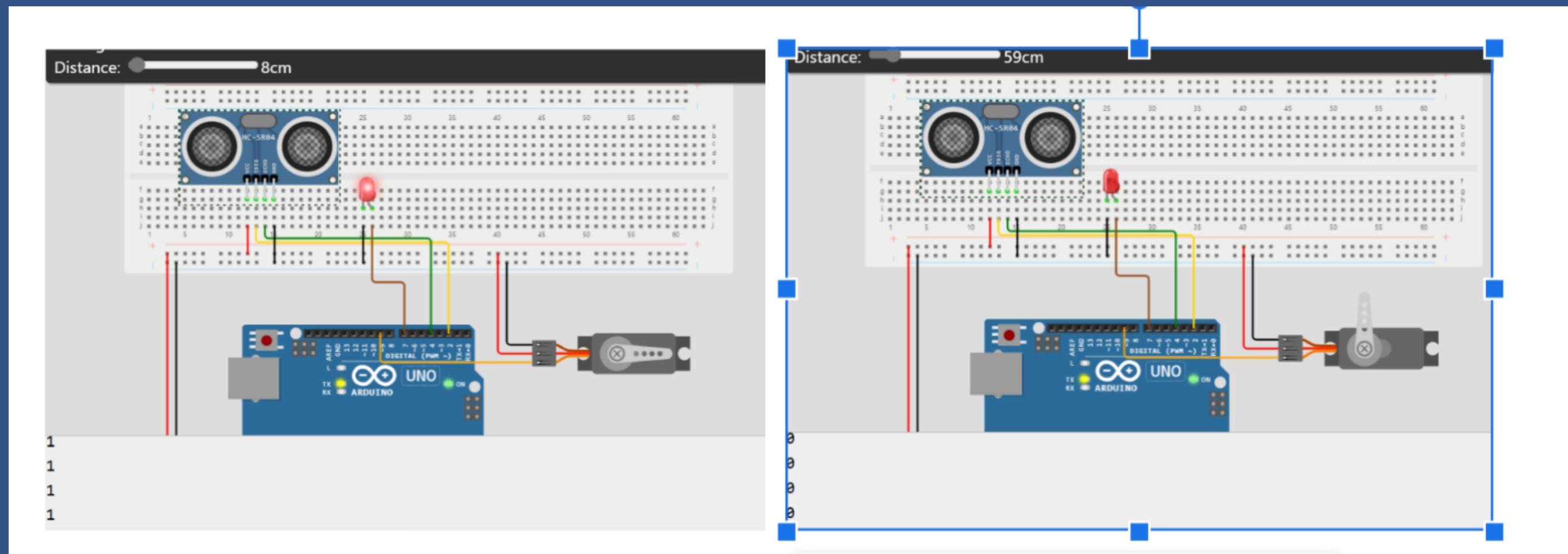
Mini Motor Servo:

Jenis motor yang digunakan untuk mengontrol posisi suatu objek. Digunakan untuk meneggerakan pagar otomatis

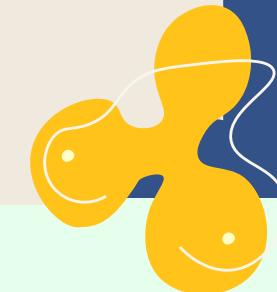
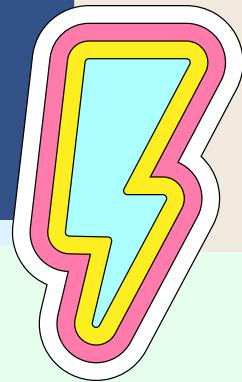
Sensor Ultrasonik HC-SR04

Sensor ultrasonik yang digunakan dalam proyek untuk mengukur jarak. Digunakan untuk mendeteksi benda yang mendekati pagar

Project Implementation

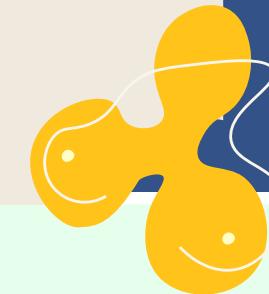
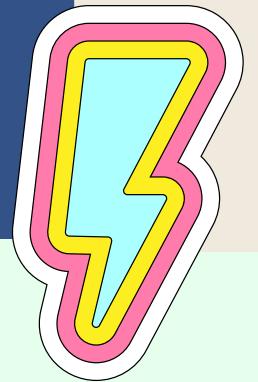


Analisis Kode



```
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
29 +
30 + init_serial:
31 +   CLR R24 ; Meng-clear R24 untuk meng-clear UCSR0A dan UBRR0H
32 +   STS UCSR0A, R24 ; Menyimpan isi dari R24 yang sudah di-clear ke UCSR04
33 +   STS UBRR0H, R24 ; Menyimpan isi dari R24 yang sudah di-clear ke UBRR0H
34 +   LDI R24, 103 ; Mengisi R24 untuk mengisi UBRR0L
35 +   STS UBRR0L, R24 ; Menetapkan baud rate dengan menyimpan isi dari R24
36 +   LDI R24, 1<<RXEN0 | 1<<TXEN0 ; Meng-enable RXB dan TXB
37 +   STS UCSR0B, R24 ; Menyimpan isi dari R24
38 +   LDI R24, 1<<UCSZ00 | 1<<UCSZ01 ; Async, no parity, 1 stop, 8 bits
39 +   STS UCSR0C, R24; Menyimpan isi dari R24
40 +   RET
41 +
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 + 11: IN   R21, TIFR1
50 +   SBRS R21, ICF1
51 +   RJMP 11             ;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 + -----
```

Analisis Kode



```
59 + 12: IN    R21, TIFR1
60 +  SBRS  R21, ICF1
61 +  RJMP  12          ;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 +
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 + 170: 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  170
79 +
80 + 180: 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  180
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
```

Analisis Kode



```
114 + agn: ;Subroutine untuk menampilkan karakter ke serial monitor
115 + LPM R29, Z+ ; Me-load satu char dari buffer yang disimpan di program memory Z
116 + CPI 29, 0 ; Meng-compare apakah R19 berasi 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 +
122 + ext:
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 + 13: 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 13 ;go back & repeat PWM signal
140 +     RCALL delay_ms ;0.5s delay
141 +     RET ;& return to main subroutine
142 +
```

```
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 + -----
```

```

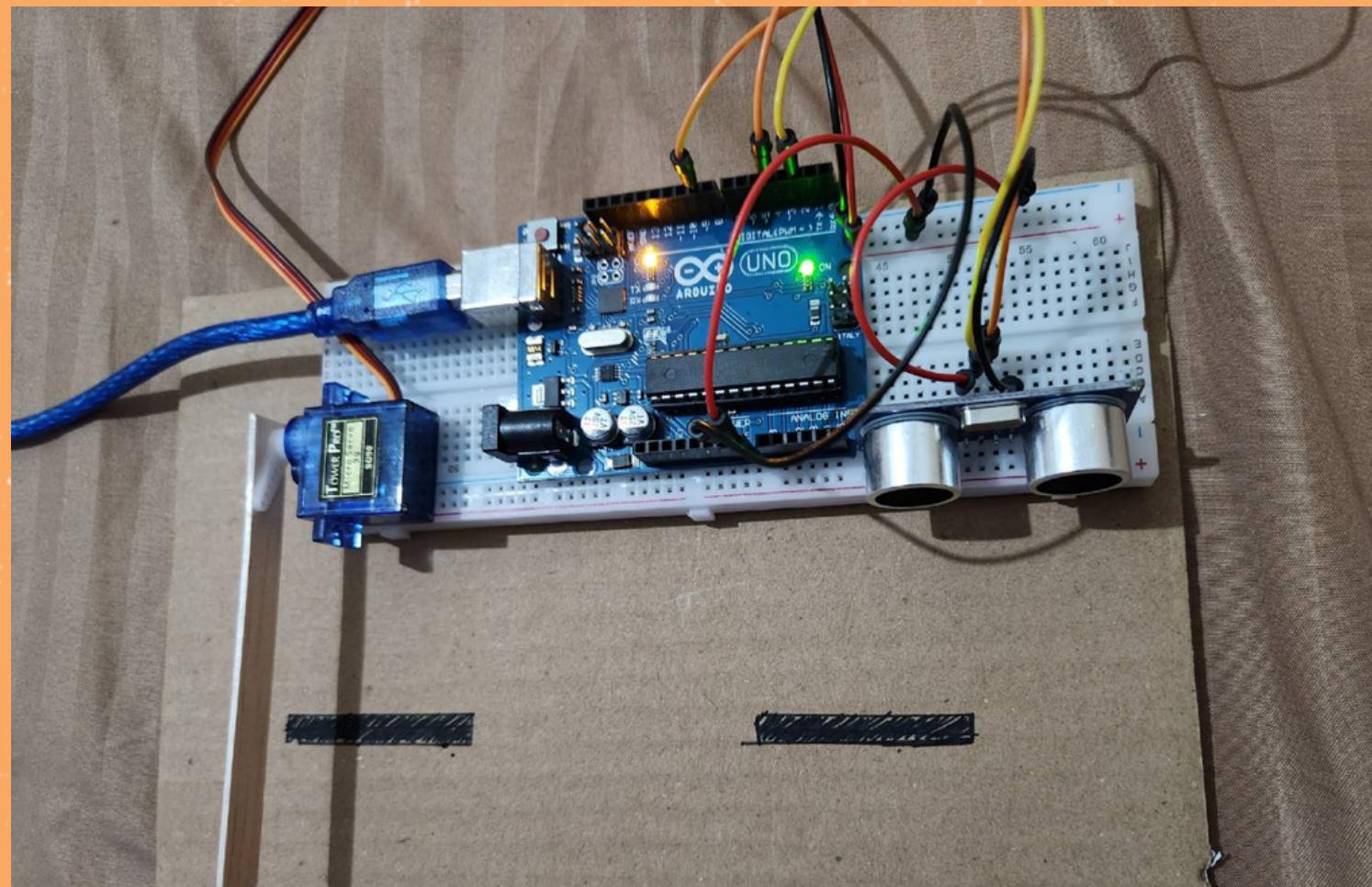
189 +
190 +
191 + 10: IN    R20, TIFR0      ;get TIFR0 byte & check
192 +     SBRS R20, OCF0A      ;if OCF0=1, skip next instruction
193 +     RJMP 10              ;else, loop back & check OCF0 flag
194 +
195 + ;-----
196 +     CLR  R20
197 +     OUT  TCCR0B, R20      ;stop timer0
198 + ;-----
199 +     LDI  R20, (1<<OCF0A)
200 +     OUT  TIFR0, R20      ;clear OCF0 flag
201 +     RET
202 +
203 + ;=====
204 +     delay_20ms:           ;delay 20ms
205 +         LDI  R21, 255
206 +         17: LDI  R22, 210
207 +         15: LDI  R23, 2
208 +         16: DEC  R23
209 +         BRNE R22, 16
210 +         DEC   R22
211 +         BRNE R21, 15
212 +         DEC   R21
213 +         BRNE R20, 17
214 +         RET
215 +
216 + ;=====
217 +     delay_ms:
218 + ;-----
219 +     ldi r25, hi8(200) ; Memasukkan nilai delay dalam satuan ms ke r25
220 +     ldi r24, lo8(200) ; Memasukkan nilai delay dalam satuan ms ke r24
221 +     outerLoop:
222 + ; karena innerLoop berjalan selama 4 cycle, maka di-loop 4000x sehingga menghasilkan
223 + ; 1 millisecond (4 x 4000 : 16Mhz = 10^-3)
224 +     ldi r31, hi8(4000) ; mengisi r30 dan r31 dengan 4000
225 +     ldi r30, lo8(4000)
226 +     innerLoop:
227 +     sbiw r30, 1 ; Mengurangi isi r30 dan r31 sebanyak 1
228 +     brne innerLoop ; Jika hasil pengurangan baris sebelumnya tidak menghasilkan nol,
229 + ; innerLoop akan terus diulang
230 + ; -----
231 +     sbiw r24, 1 ; Mengurangi r24 dan r25 sebanyak 1
232 +     brne outerLoop; Jika hasil pengurangan baris sebelumnya tidak menghasilkan nol,
233 + ; maka outerLoop akan terus diulang dan innerLoop akan dijalankan kembali
234 +     RET
235 + ;=====

```

```
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
```

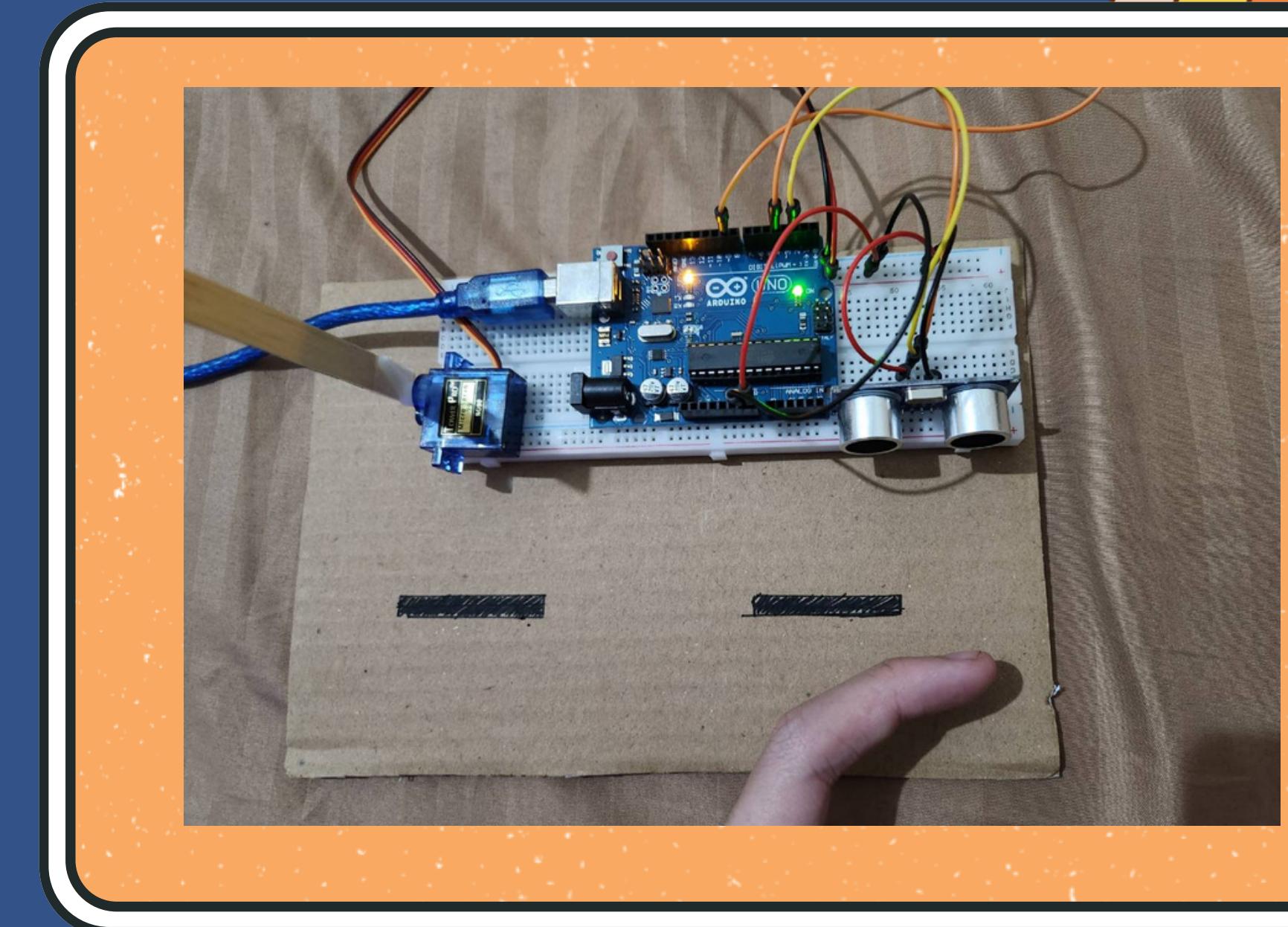
Rangkaian



Initial State

Rangkaian bekerja dengan semestinya dimana saat keadaan initial stat maka pagar akan tertutup karena sensor tidak membaca ada gerakan, sedangkan dalam keadaan mendekripsi objek pagar akan terbuka karena sensor membaca ada benda yang mendekat

Deteksi Objek



Kesimpulan

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 mengotomatiskan sistem gerbang dan mengintegrasikan kemampuan deteksi jarak, solusi ini menawarkan efisiensi, keamanan, dan pengalaman yang tidak merepotkan bagi pengemudi dan operator tempat parkir.



Terima Kasih

