



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

PANTAU.IN : SMART AQUATIC FIELD MONITORING SYSTEM

GROUP 21

Nahl Syareza Rahidra	2206830340
Annisa Ardelia Setiawan	2206059471
Mario Matthews Gunawan	2206810452
Muhamad Fauzan	2206819054

PREFACE

Dengan segala rasa dan puji syukur kepada Tuhan Yang Maha Esa atas karunia-Nya yang telah diberikan, kami membuat sebuah laporan atas alat dari proyek akhir kami yang berjudul “Pantau.in” tepat pada waktunya. Laporan ini kami susun untuk memberikan pemahaman tentang latar belakang, cara kerja, dan sistematika dari alat kami.

Proyek ini merupakan implementasi dari akumulasi pengetahuan yang telah kami dapatkan selama menjalani praktikum Sistem Siber Fisik. Kami juga ingin menyampaikan terima kasih yang sebesar-besarnya kepada assistant laboratorium pembimbing kami, Bang Ahmad Rifqi yang telah membantu kami dalam menghadapi permasalahan dan memberikan arahan selama pengerjaan proyek kami.

Meskipun kami telah berusaha untuk menyusun laporan ini dengan teliti dan seksama, kami menyadari bahwa masih terdapat beberapa kekurangan yang perlu diperbaiki. Oleh karena itu, kami berharap bahwa pembaca memberikan kritik dan saran yang konstruktif kepada kami agar kami bisa memperbaiki kesalahan-kesalahan yang ada pada laporan ini. Kami berharap laporan ini dapat memberikan manfaat dalam Pengembangan Ilmu Pengetahuan dan Teknologi. Terima kasih atas perhatian dan waktu yang telah diberikan kepada kami

Depok, Mei 27, 2024

Group 21

TABLE OF CONTENTS

CHAPTER 1.....	4
INTRODUCTION.....	4
1.1 PROBLEM STATEMENT.....	4
1.2 PROPOSED SOLUTION.....	4
1.3 ACCEPTANCE CRITERIA.....	5
1.4 ROLES AND RESPONSIBILITIES.....	6
1.5 TIMELINE AND MILESTONES.....	6
CHAPTER 2.....	7
IMPLEMENTATION.....	7
2.1 HARDWARE DESIGN AND SCHEMATIC.....	7
2.2 SOFTWARE DEVELOPMENT.....	9
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	13
CHAPTER 3.....	15
TESTING AND EVALUATION.....	15
3.1 TESTING.....	15
3.2 RESULT.....	17
3.3 EVALUATION.....	20
CHAPTER 4.....	22
CONCLUSION.....	22

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Pertanian merupakan salah satu sektor penting di Indonesia mengingat perannya sebagai sektor penghasil bahan pangan utama. Bahan pangan inilah yang nantinya akan diolah menjadi makanan sehari-hari untuk masyarakat. Terdapat berbagai macam hasil panen dari sektor pertanian, seperti kelapa sawit, padi, beras, tebu, jagung, dan singkong.

Selain menjadi penghasil pangan, bertani juga menjadi salah satu mata pencaharian umum bagi warga Indonesia di daerah pedesaan. Para petani mengolah sawah mereka dengan cara menebar benih, membajak sawah, dan menyalurkan irigasi. Setelah itu mereka akan melakukan masa panen, dimana mereka akan mengambil hasil bercocok tanam mereka dari lahan sawah. Dalam proses pertanian ini, terdapat beberapa masalah yang dapat dihadapi oleh petani, salah satunya terkait dengan irigasi.

Irigasi adalah pengaliran air ke suatu petak sawah dengan tujuan untuk memberikan tanaman panen kebutuhan air yang cukup sehingga tanaman dapat tumbuh dengan baik. Irigasi merupakan salah satu komponen penting dalam bertani. Namun sayangnya, para petani harus pergi ke petak sawah mereka untuk melihat tinggi air masing-masing petak sawah. Hal ini tentunya sangat menguras tenaga para petani tersebut

1.2 PROPOSED SOLUTION

Atas permasalahan tersebut, kelompok kami membuat sebuah inovasi berupa sebuah alat yang dapat secara otomatis mengukur tinggi air pada sebuah petak sawah dan kemudian memberikan sebuah bacaan tinggi kondisi air pada sawah sekarang. Atas solusi tersebut, kami menggagas sebuah alat bernama Pantau.in yang berfungsi untuk memberikan bacaan tinggi air sebuah petak sawah berdasarkan kondisi sawah tersebut secara *real-time* sehingga petani tidak perlu pergi ke masing-masing petak sawah untuk mengecek tinggi airnya.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. Sistem harus dapat mengukur tinggi air secara otomatis pada petak sawah.
2. Sistem harus dapat memberikan pembacaan tinggi air secara real-time.
3. Sistem harus dapat mengirimkan informasi tinggi air dari transmitter ke receiver dengan akurat.
4. Sistem harus mudah digunakan oleh petani tanpa memerlukan keahlian teknis khusus.
5. Sistem harus tahan terhadap kondisi lingkungan sawah yang keras, seperti hujan dan panas matahari.
6. Sistem harus memiliki daya tahan baterai yang cukup untuk penggunaan jangka panjang.
7. Sistem harus memiliki antarmuka yang ramah pengguna untuk memudahkan monitoring.
8. Sistem harus memiliki kemampuan untuk memberikan peringatan jika tinggi air berada di luar batas yang diinginkan.
9. Sistem harus dapat diintegrasikan dengan teknologi irigasi otomatis jika memungkinkan di masa depan.
10. Sistem harus memiliki biaya yang terjangkau bagi petani.

1.4 ROLES AND RESPONSIBILITIES

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

Roles	Responsibilities	Person
Leader	Manajemen berjalannya proyek dan programming Arduino UNO	Nahl Syareza Rahidra
Member	Merakit rangkaian fisik dan menyusun laporan.	Annisa Ardelia Setiawan
Member	Brainstorming ide dan perakitan rangkaian fisik	Mario Matthews Gunawan
Member	Merakit rangkaian fisik, menyusun laporan dan membantu programming Arduino UNO.	Muhamad Fauzan

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

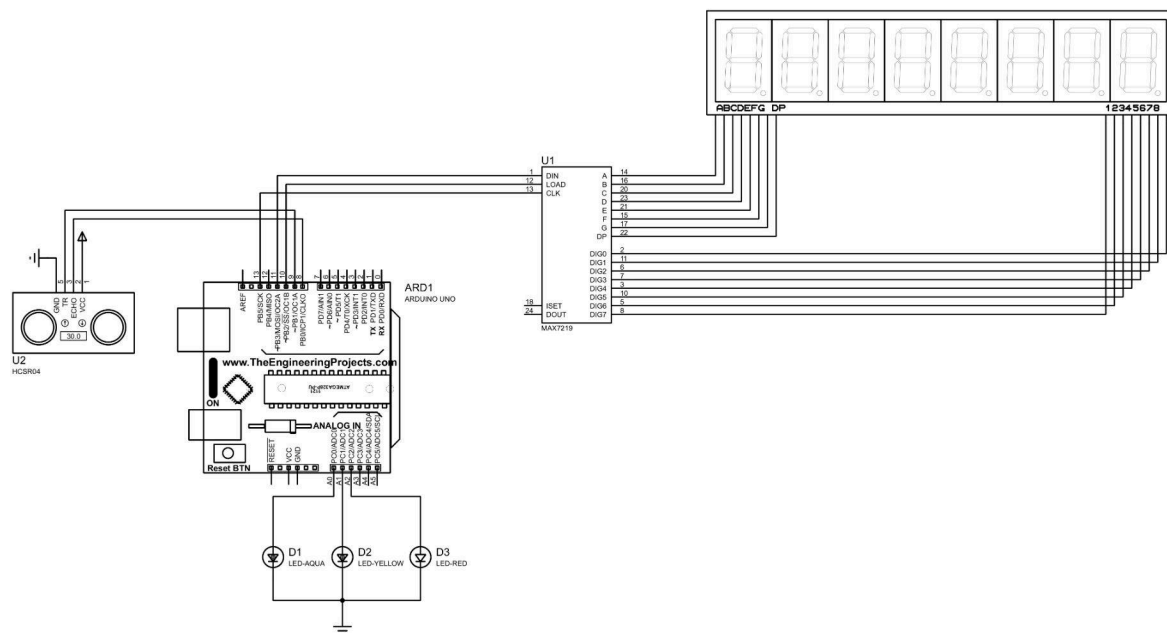
	May, 2024																																	
Task	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31											
Brainstroming idea																																		
Software Development																																		
Hardware Design																																		
Integration and Testing																																		
Final Product Assembly and Testing																																		
Making Project Report and PPT																																		
Presentation																																		

Table 2. Timeline and Milestones

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC



Dalam perancangan hardware, kami memakai beberapa komponen. Komponen-komponen tersebut berupa Arduino UNO, HC-SR04 Ultrasonic Sensor, MAX7219 Display, dan 3 buah LED. Alat akan bekerja dengan cara menerima input dari sensor HC-SR04 yang telah dihubungkan dengan pin PB0 untuk Echo dan PB1 untuk Trigger. Setelah HC-SR04 memberikan bacaan, maka Arduino UNO akan mengubah nilai bacaan sensor menjadi tinggi air dan kemudian meneruskan hasil mengonversi hasil bacaan tersebut ke dalam bentuk decimal. Selanjutnya, hasil konversi ini akan diteruskan ke MAX7219 Display pada digit 1 dan 0, dimana digit 1 akan menampilkan MSD dan digit 0 akan menampilkan LSD. Terdapat juga beberapa kondisi yang akan membuat lampu LED menyala.

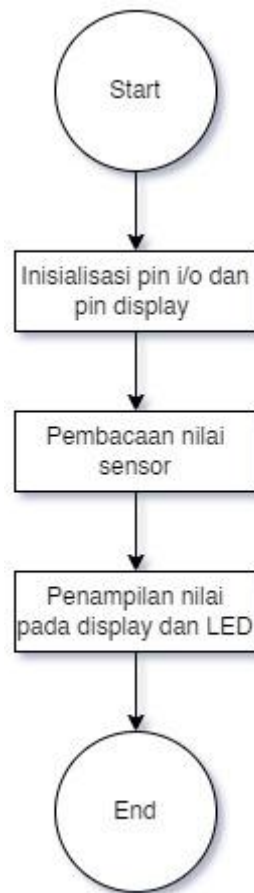
Komponen sensor yang kami pakai, yaitu HC-SR04 bekerja dengan cara memantulkan gelombang ultrasonik. Gelombang ini dipancarkan oleh Echo dan diterima oleh Trigger. Ketika trigger menerima pancaran gelombang tersebut, maka ia akan bernilai high untuk sementara. Deteksi jarak dapat dihitung dengan mencari selisih waktu ketika pin Echo memancarkan gelombang dan ketika pin Trigger menerima pancaran tersebut. Hasil dari selisih ini akan dimasukkan ke dalam sebuah rumus untuk mengubahnya menjadi jarak.

Display yang kita gunakan adalah MAX7219 Display. Display ini terdiri dari 8 7-segment Display yang telah dikelompokkan menjadi satu. Untuk menampilkan huruf atau angka dalam display ini, kita harus memilih digit mana yang ingin kita berikan data dan data yang ingin kita berikan. Data ini kemudian diterjemahkan ke dalam truth table 7-segment, dimana urutannya adalah dp, a, b, c, d, e, f, g.

Selain itu, terdapat juga 3 LED yang bisa menandakan bahwa level air dibawah nilai tertentu. Jika level air sama dengan atau dibawah 10 cm, maka LED merah akan menyala, jika sama dengan atau dibawah 20 maka lampu kuning akan menyala, jika sama dengan atau dibawah 30 maka lampu biru akan menyala.

2.2 SOFTWARE DEVELOPMENT

Secara garis besar, berikut adalah alur kerja dari alat kami



Program ini bekerja pertama kali dengan melakukan inisialisasi-inisialisasi dari pin untuk sensor dan display, kemudian microcontroller akan mendapatkan nilai hasil bacaan dari sensor yang kemudian akan ditampilkan pada display dan lampu LED tertentu akan menyala sesuai dengan nilai hasil bacaan

Berikut adalah rincian dari bagian ‘Inisialisasi pin i/o dan pin display’



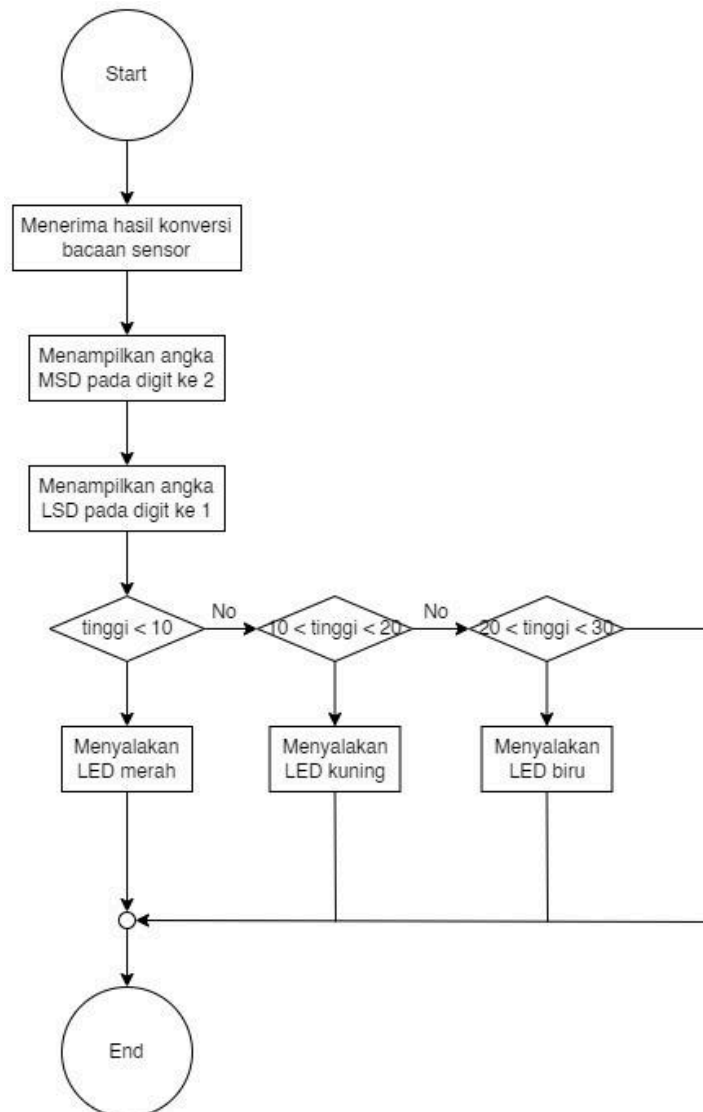
Dalam bagian ini kita akan menetapkan beberapa hal agar sensor dan display dapat berfungsi dengan baik. Kita perlu menetapkan beberapa hal dalam MAX7219 display seperti digit yang akan digunakan sebagai representasi angka, scan limit, segment intensity, dan menyalakan MAX7219 itu sendiri. Selain itu kita juga perlu mendefinisikan pin mana saja yang akan digunakan oleh sensor sebagai input/output

Berikut adalah rincian dari bagian ‘Pembacaan nilai sensor’



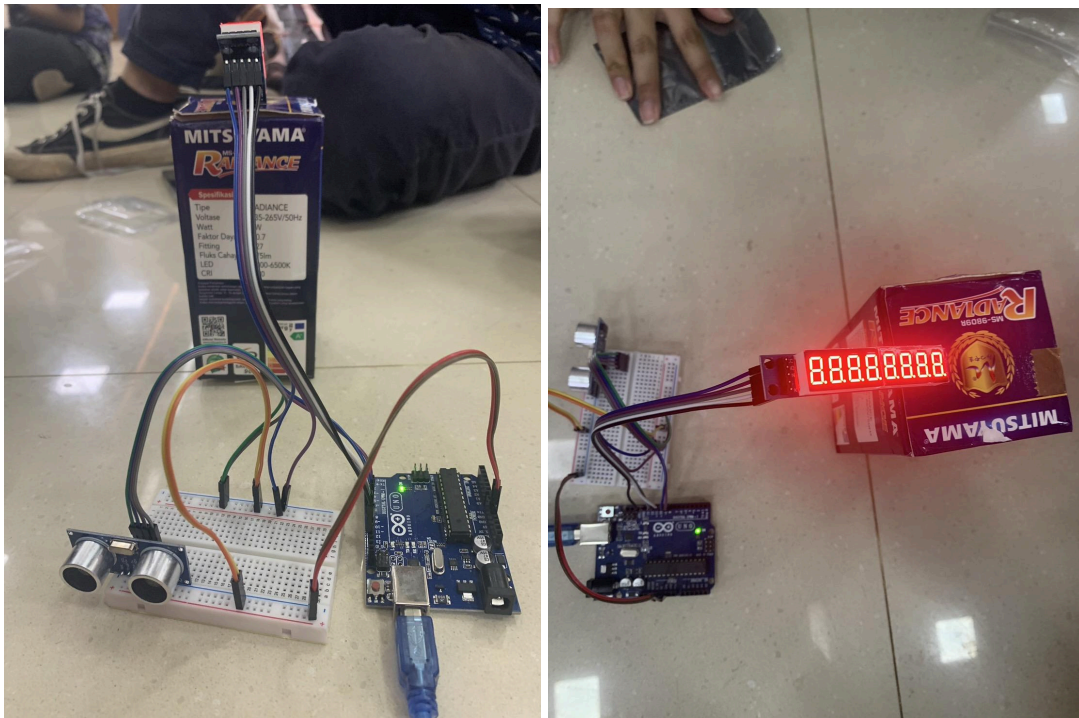
Sensor bekerja dengan cara memantulkan sebuah gelombang. Kemudian gelombang ini akan ditangkap dan kemudian dihitung selisih waktu dari pemancaran gelombang dan penangkapan gelombang. Setelah itu, akan dilalui sebuah proses untuk mengubah waktu menjadi jarak. Di sini kami menetapkan sebuah jarak maksimal yaitu 30 cm. Setelah bacaan jarak didapatkan, kita perlu menguranginya dengan 30, karena bacaan sensor hanyalah jarak antara sensor dengan permukaan air. Setelah mendapatkan nilai tersebut, akan ada tahap konversi agar data dapat ditampilkan dengan benar pada MAX7219 display

Berikut adalah rincian dari bagian ‘Penampilan nilai pada display dan LED’



Dalam bagian ini kita akan menampilkan hasil perhitungan pada sebelumnya sehingga ada representasi secara visual, baik dalam bentuk kalimat atau bentuk warna yaitu LED. Untuk penampilan pada display, hasil akan dipecah menjadi MSD dan LSD, dimana kedua angka tersebut akan ditampilkan pada digit ke 2 dan 1 secara berurutan. Setelah itu, akan ada pengecekan nilai oleh microcontroller untuk menentukan lampu LED yang akan menyala. Jika tinggi yang dibaca kurang dari 10, maka lampu LED merah akan menyala. Jika tinggi di antara 10 dan 20, maka lampu LED kuning akan menyala. Jika tinggi di antara 20 dan 30 maka lampu LED biru akan menyala

2.3 HARDWARE AND SOFTWARE INTEGRATION



Perancangan dan pengintegrasian hardware serta software dibuat secara paralel. Pada proyek yang kami buat, dibutuhkan komponen seperti sensor ultrasonik, MAX7219 Display yang didalamnya terdapat 8 buah seven-segmen, 3 led dengan warna yang berbeda serta arduino uno. Awalnya kita menginisialisasi pin untuk dapat berkomunikasi dengan display serta dapat menangani sensor ultrasonik HC-SR04. Sensor tersebut akan mendeteksi apabila ada objek yang menghalangi tembakan ultrasonik dari sensor, dalam kasus yang diterapkan maka sensor akan mendeteksi air yang naik di perkebunan maupun sawah.

Sensor nantinya akan mengirim pesan terkait berapa detik sensor tersebut menerima kembali gelombang ultrasonik yang dipancarkan. Pada program, akan di proses menjadi jarak asli dalam cm. Apabila tinggi air melebihi tinggi yang sudah di set untuk tinggi air yang dalam batas aman, tinggi air waspada dan tinggi air bahaya yang sudah ditentukan dengan ketentuan :

- $\text{tinggi} < 10 = \text{merah}$
- $10 < \text{tinggi} < 20 = \text{kuning}$
- $20 < \text{tinggi} < 30 = \text{biru}$

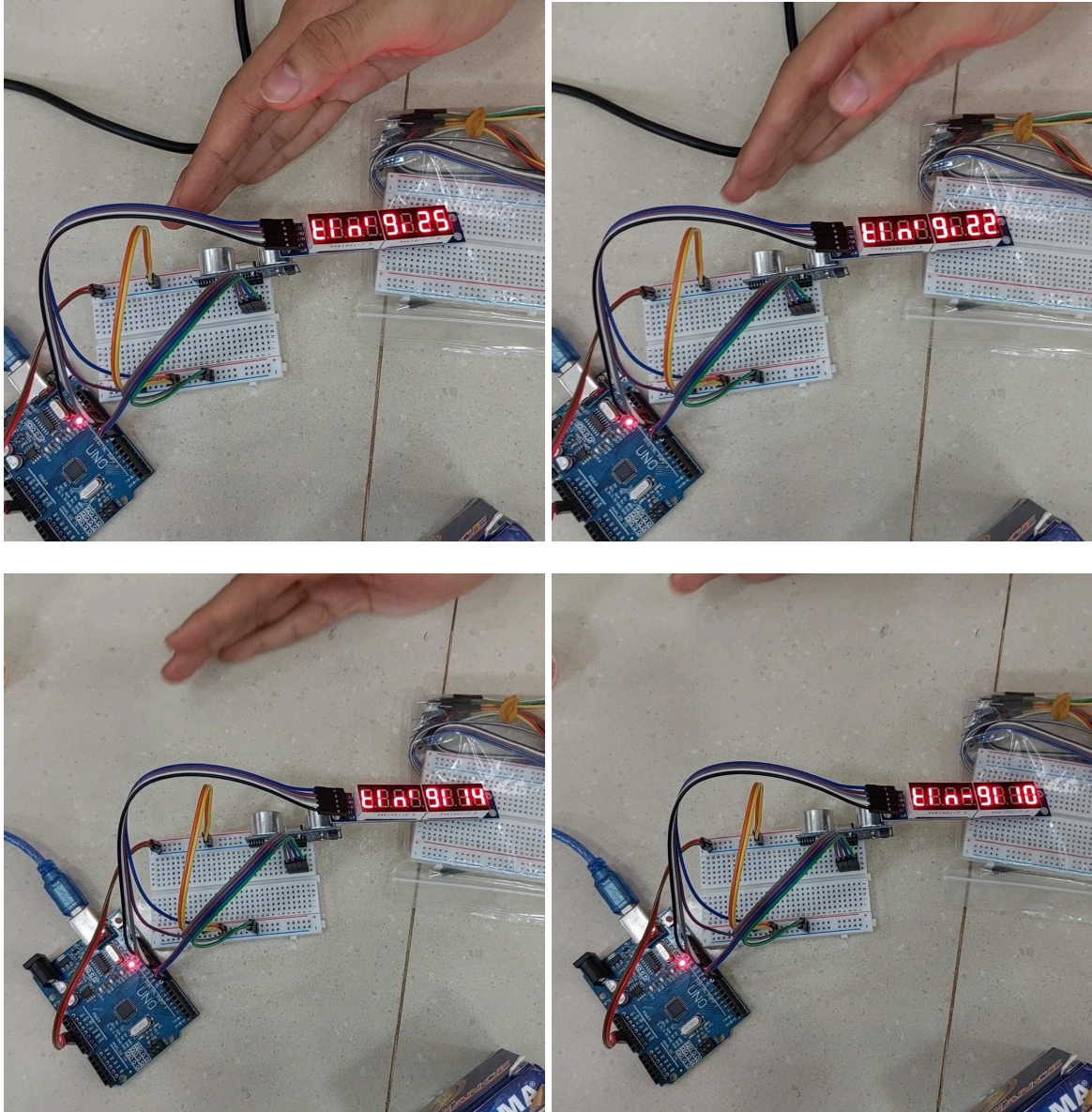
Maka akan terdapat LED yang menyala menyesuaikan tinggi air sekarang, ini diproses di software yang mengecek tinggi air setiap hitungan clocknya. Sederhananya, LED akan menjadi indikator dari status tinggi air. Semakin dekat objek dengan sensor, dalam hal ini semakin tinggi air dan dekat dengan sensor, maka semakin besar angka yang ditunjukkan.

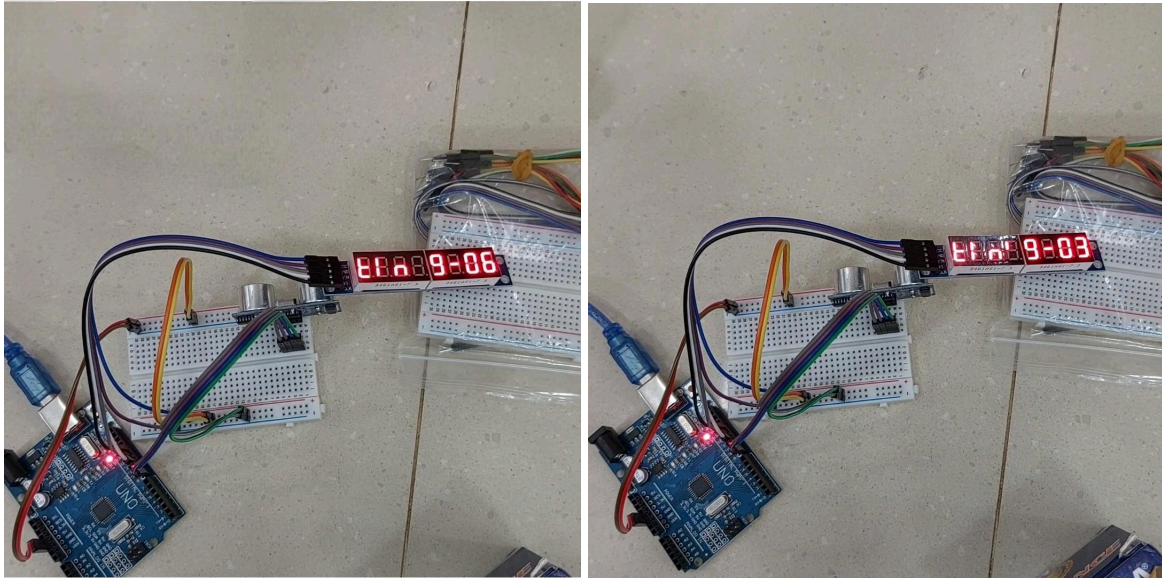
Pada rangkaian proteus terdapat IC MAX7219 yang terhubung dengan display, pada kenyataannya kita menggunakan MAX7219 Display yang sudah include IC dan Display sehingga tidak menyulitkan kita untuk menghubungkan pin pinnya.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

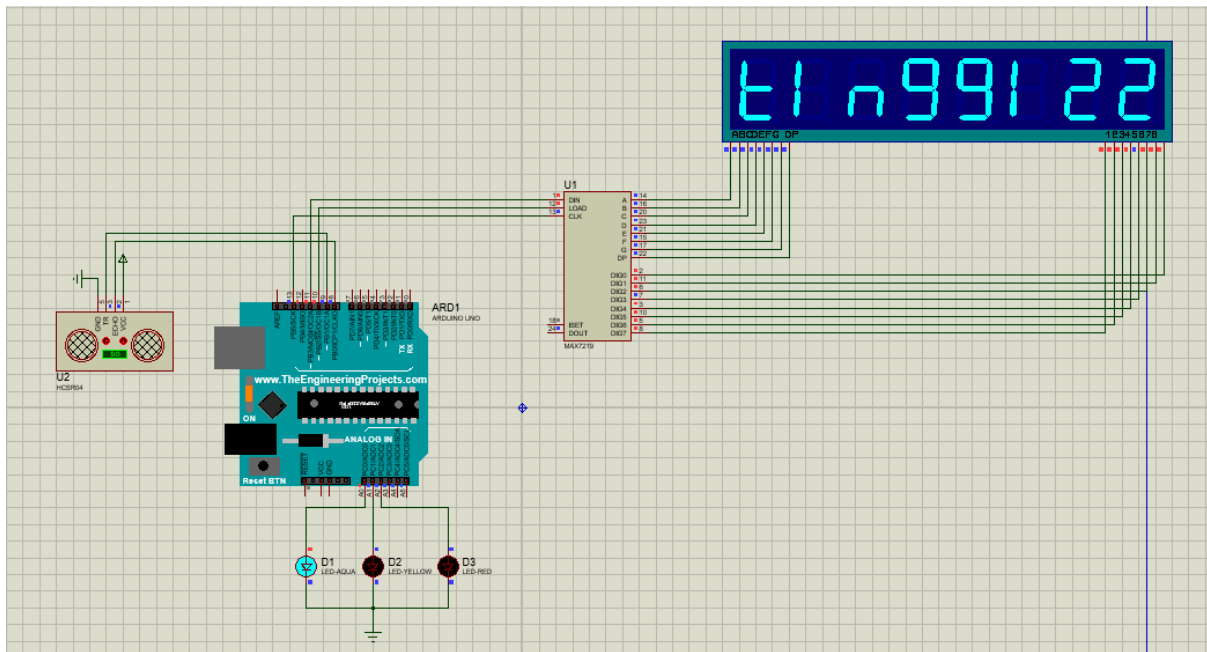
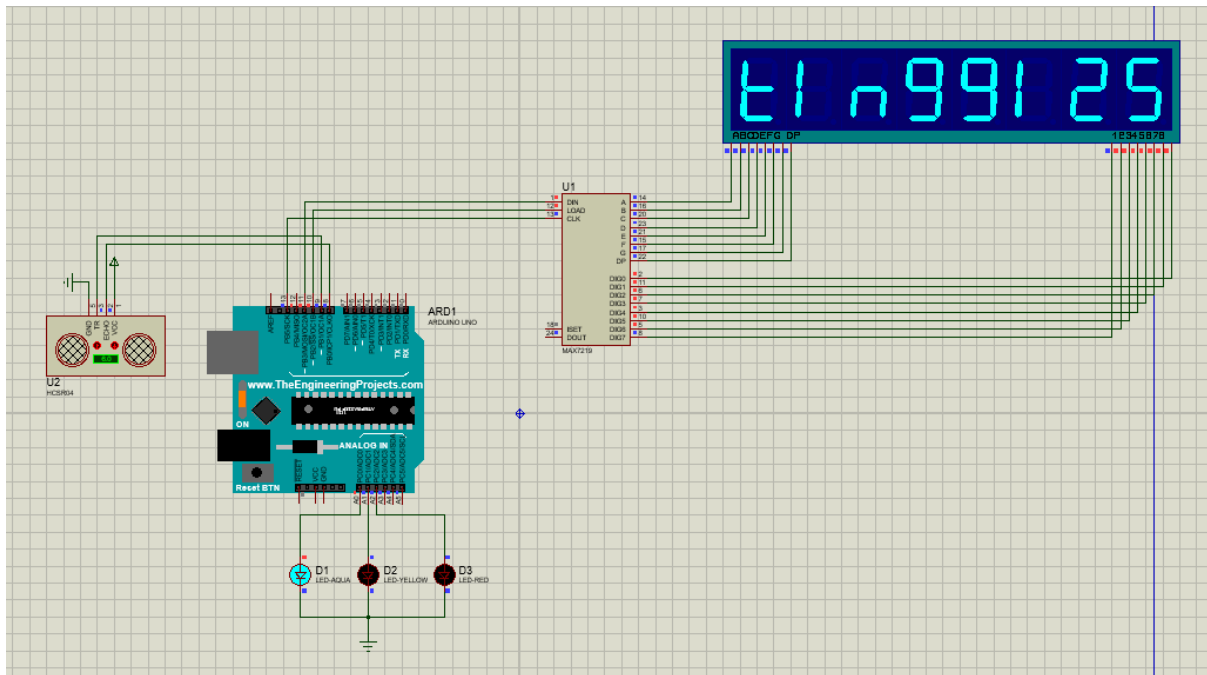


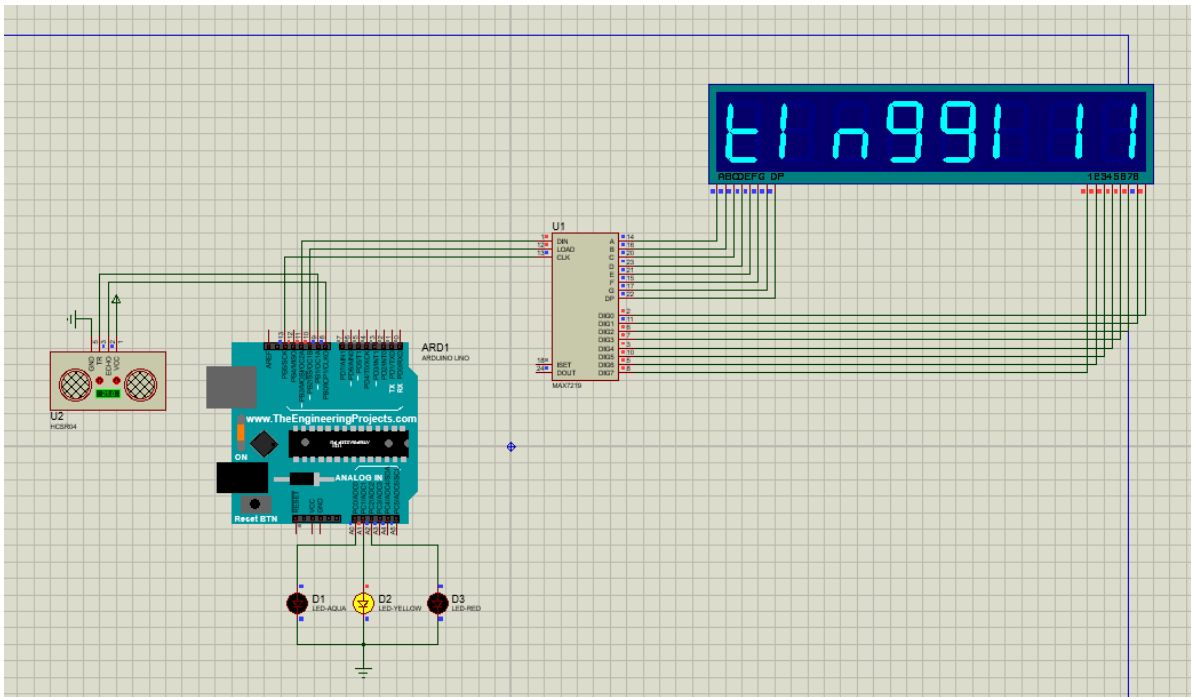
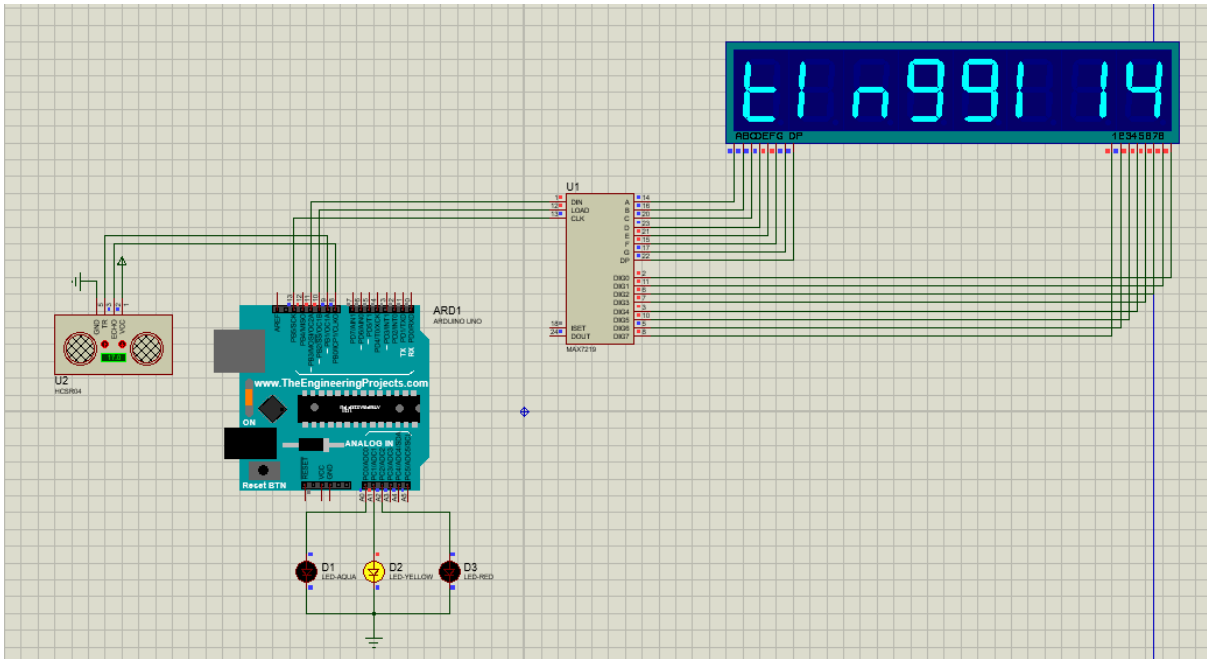


Pada pengujian, kelompok kami berhasil menampilkan jarak objek ke sensor. Meskipun hasilnya belum sepenuhnya sesuai harapan karena sering terjadinya error akibat kesalahan hardware, namun tampilan pada display sudah sesuai dengan ekspektasi kami. Semakin dekat objek dengan sensor, dalam hal ini air, maka semakin tinggi angka yang ditampilkan pada layar. Hal ini menunjukkan bahwa sistem kami telah bekerja sesuai dengan konsep yang telah direncanakan.

Kami membagi hasil pembacaan menjadi tiga bagian dengan masing-masing ditandai oleh lampu LED dengan warna berbeda sesuai dengan tinggi air yang terdeteksi: tinggi air dalam batas aman ($20 < \text{tinggi} < 30$ cm) ditandai dengan LED biru, tinggi air waspada ($10 < \text{tinggi} < 20$ cm) ditandai dengan LED kuning, dan tinggi air bahaya ($\text{tinggi} < 10$ cm) ditandai dengan LED merah. Dengan melakukan testing ini dapat disimpulkan bahwa sistem kami memberikan solusi yang efektif bagi para petani dalam memantau tinggi air di sawah mereka secara real-time.

3.2 RESULT



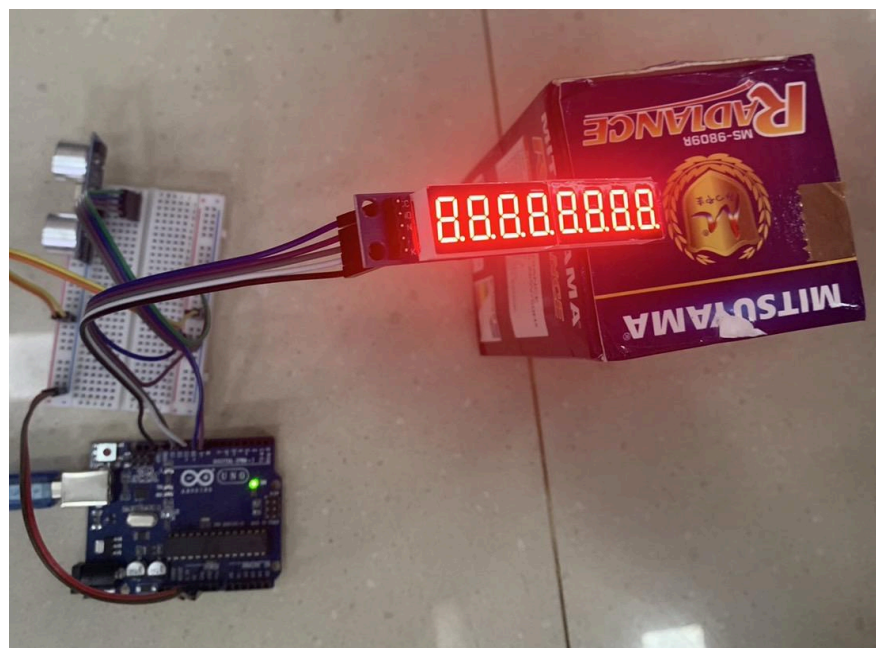


Sepanjang proses pengujian, perhatian khusus perlu diberikan pada keandalan dan konsistensi umpan balik visual yang disediakan oleh modul dalam menanggapi perubahan ketinggian air. Aspek penting ini dinilai untuk memastikan bahwa umpan balik yang diberikan tepat waktu, akurat, dan efektif dalam mengkomunikasikan status ketinggian air saat ini kepada pengguna. Selain itu, komunikasi dan kontrol yang tepat antara perangkat master dan slave diuji secara menyeluruh menggunakan modul I2C/SPI.

Evaluasi hasil dari testing ini memastikan keandalan dalam pertukaran data sehingga menghasilkan hasil yang stabil dan akurat, serta memfasilitasi koordinasi dan interaksi yang mulus antara komponen-komponen sistem pemantauan ketinggian air. Proses pengujian ini memastikan bahwa sistem pemantauan ketinggian air beroperasi dengan lancar, memberikan solusi yang dapat diandalkan bagi para petani untuk memantau dan mengelola irigasi di sawah mereka secara real-time.

3.3 EVALUATION

Penyambungan MAX7219 Display dengan kabel yang tertuju dengan breadboard seharusnya disambung menggunakan solder. Ini dikarenakan sering terjadi kondisi dimana tidak semua pin terhubung dan mengakibatkan display tidak menyala atau menyala namun tidak menunjukkan output sama sekali. Kondisi dimana display terhubung, namun tidak menampilkan output sama sekali :



Pada awalnya, kami mengira kesalahan utama display tidak menampilkan output karena tidak sempurnanya koneksi display dan kabel yang terhubung ke breadboard. Namun, ternyata faktor lainnya karena inisiasi pin dan display yang salah, akhirnya display dapat ditampilkan namun belum sempurna karena poin 1 dievaluasi ini.

Dari evaluasi yang telah dipaparkan di atas, serta kekurangan-kekurangan lain yang ditemukan dalam proyek ini, berikut ini merupakan beberapa perbaikan dan langkah lebih lanjut untuk memperbaiki sistem kami menjadi lebih baik:

- Algoritma dan parameter modul pemantauan harus disesuaikan lebih lanjut untuk meningkatkan kemampuan pengambilan keputusan yang lebih akurat dalam berbagai skenario. Hal ini akan meningkatkan kemampuan modul untuk memberikan pembacaan ketinggian air yang lebih tepat.
- Modul harus diintegrasikan dengan sistem irigasi yang ada untuk memastikan kompatibilitas dan operasi yang efisien. Integrasi ini memungkinkan modul bekerja secara harmonis dengan fitur lain dan meningkatkan keseluruhan keamanan dan efisiensi pengelolaan air di sawah.
- Pengujian lebih lanjut perlu dilakukan dalam berbagai kondisi lingkungan, seperti perubahan cuaca dan kondisi lahan yang berbeda, untuk memastikan efektivitas dan keandalan modul dalam situasi nyata.

CHAPTER 4

CONCLUSION

Dalam dunia pertanian, irigasi merupakan komponen yang sangat penting bagi petani dalam bertani. Agar memudahkan pengukuran tinggi air dalam petak sawah, kelompok kami membuat sebuah alat pengukur tinggi air secara real time bernama “Pantau.in”. Pada praktikum yang kami buat, alat ini dapat bekerja dengan baik mengukur tinggi air menggunakan sensor HC-SR04 serta dapat menampilkan ketinggian air pada MAX7219 Display.

Pada alat yang kami buat, petani dapat terbantu dengan lampu LED yang menandakan batas aman, batas waspada, dan batas bahaya pada ketinggian air di sawah mereka. Batas aman adalah kondisi saat ketinggian air berada pada ketinggian 20-30 cm dimana air dirasa cukup untuk irigasi sawah. Kemudian batas waspada berada di ketinggian 10-20 cm dimana air sudah agak berkurang dan berpotensi habis. Sedangkan batas bahaya berada pada ketinggian 0-10 cm dimana sawah kekurangan air untuk irigasi.

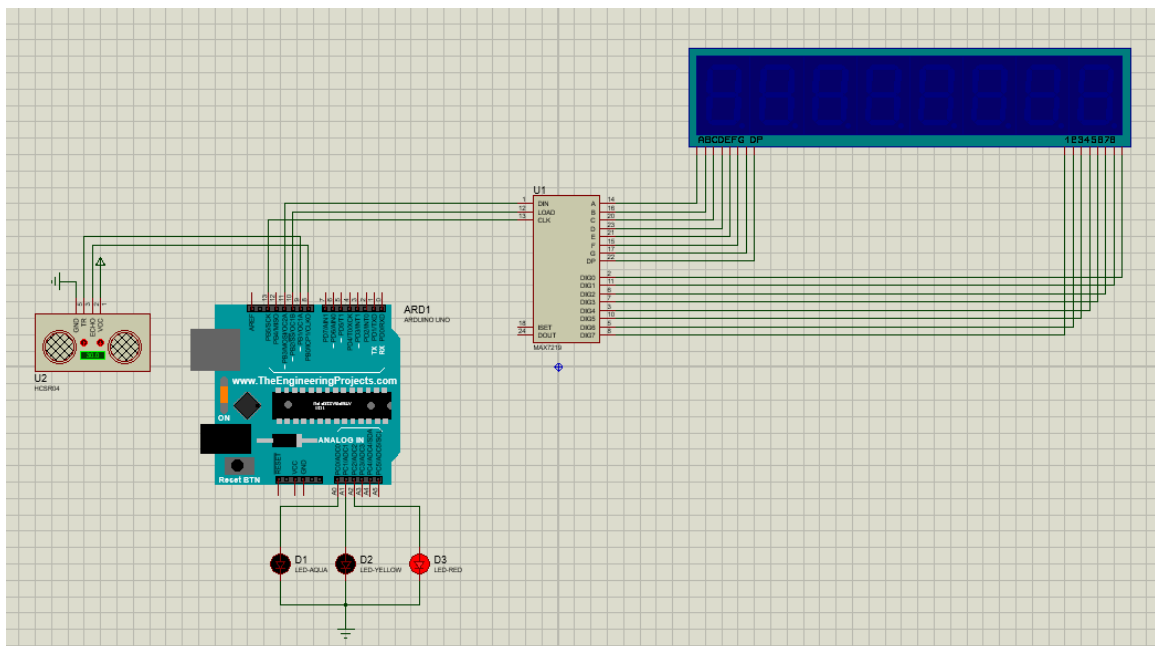
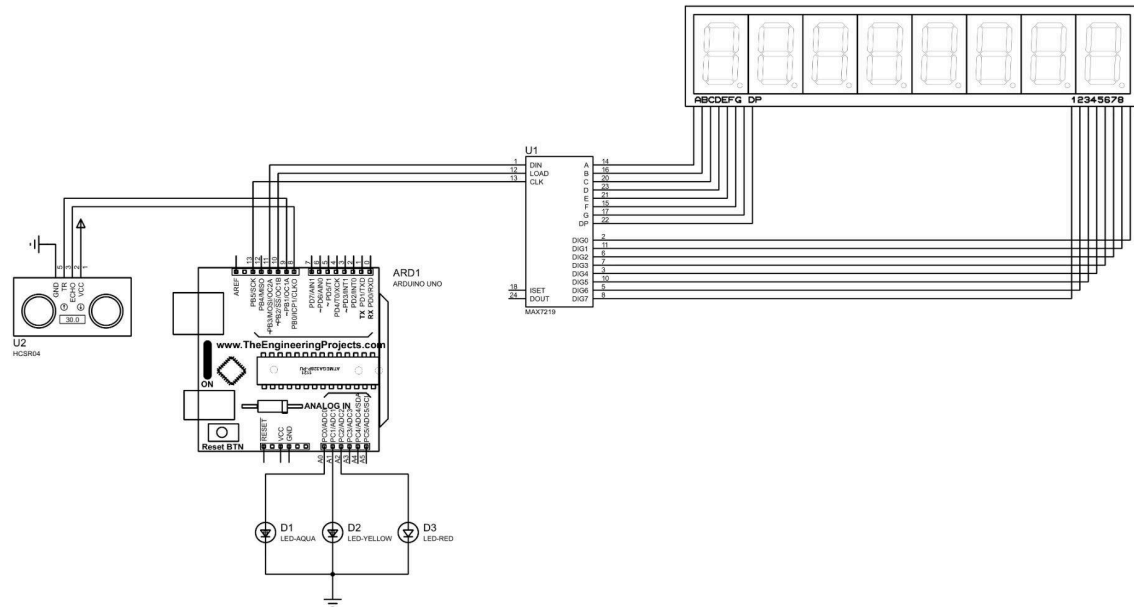
Melalui pengujian yang telah kami lakukan, kami menyimpulkan bahwa alat ini dapat berfungsi dengan baik dalam kondisi nyata. Sensor ketinggian dapat bekerja dengan baik dimana display dapat menunjukkan ketinggian air secara real time dalam cm dan lampu LED dapat menyala sesuai batas mana ketinggian air berada. Kedepannya, kami dapat melakukan pengembangan lebih lanjut dengan memastikan bahwa alat ini dapat dengan baik pada daerah pertanian agar tahan lama. Harapannya, “Pantau.in” dapat menjadi inovasi yang efektif bagi para petani dalam mengelola irigasi sawah mereka.

REFERENCES

- [1] “AVR Microcontrollers AVR Instruction Set Manual OTHER Instruction Set Nomenclature.” Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-0856-AVR-Instruction-Set-Manual.pdf> (accessed May 28, 2024).
- [2] “Assembly via Arduino - Programming HC-SR04 Sensor.” <https://akuzechie.blogspot.com/2021/12/assembly-via-arduino-programming-hc.html> (accessed May 28, 2024).
- [3] Dejan, “Ultrasonic Sensor HC-SR04 and Arduino Tutorial,” *HowToMechatronics*, Aug. 04, 2018. <https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/> (accessed May 28, 2024).

APPENDICES

Appendix A: Project Schematic



Appendix B: Github Documentation

Github: <https://github.com/NahlSyareza/Pantau.in-SSF.git>

Appendix C: Testing Documentation

Hasil pembacaan dan testing rangkaian fisik

https://drive.google.com/file/d/15Bwlp_sDhifi6h0Pg0xmJpmyfAhlZcU9/view?usp=sharing

Appendix D: Source Code

```
;-----  
; Assembly Code  
;-----  
#define __SFR_OFFSET 0x00  
#include "avr/io.h"  
;-----  
.global main  
.global SPI_MAX7219_init  
.global MAX7219_disp_text  
;=====  
main:  
;-----  
  
    RCALL SPI_MAX7219_init  
    RCALL MAX7219_disp_text  
  
    LDI    R25, 0xFF  
    OUT    DDRC, R25  
    SBI    DDRB, 1          ;pin PB1 as o/p (Trigger)  
    CBI    DDRB, 0          ;pin PB0 as i/p (Echo)  
    ;-----  
agn: SBI    PORTB, 1  
    RCALL delay_timer0  
    CBI    PORTB, 1          ;send 10us high pulse to sensor  
    ;-----  
    RCALL echo_PW           ;compute Echo pulse width count  
    ;-----  
    RCALL byte2decimal      ;covert & display on MAX7219  
    ;-----  
    RCALL delay_ms  
    RJMP   agn  
;=====  
echo_PW:  
;-----
```

```

    LDI    R20, 0b00000000
    STS    TCCR1A, R20      ;Timer 1 normal mode
    LDI    R20, 0b11000101 ;set for rising edge detection &
    STS    TCCR1B, R20      ;prescaler=1024, noise cancellation ON
    ;-----
11: IN     R21, TIFR1
    SBRS   R21, ICF1
    RJMP   11               ;loop until rising edge is detected
    ;-----
    LDS    R16, ICR1L       ;store count value at rising edge
    ;-----
    OUT    TIFR1, R21       ;clear flag for falling edge detection
    LDI    R20, 0b10000101
    STS    TCCR1B, R20      ;set for falling edge detection
    ;-----
12: IN     R21, TIFR1
    SBRS   R21, ICF1
    RJMP   12               ;loop until falling edge is detected
    ;-----
    LDS    R28, ICR1L       ;store count value at falling edge
    ;-----
    SUB    R28, R16         ;count diff R22 = R22 - R16
    OUT    TIFR1, R21       ;clear flag for next sensor reading
    RET

;=====
;MAX7219 subroutines
;=====
SPI_MAX7219_init:
;-----
.equ    SCK, 5
.equ    MOSI, 3
.equ    SS, 2
;-----
    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, 0b00000011 ;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

;=====
MAX7219_disp_text:
;-----
    LDI    R17, 0x08        ;select digit 7
    LDI    R18, 0x0F        ;data = t
    RCALL  send_bytes       ;send command & data to MAX7219
    ;-----
    LDI    R17, 0x07        ;select digit 6
    LDI    R18, 0x06        ;data = i
    RCALL  send_bytes       ;send command & data to MAX7219
    ;-----
    LDI    R17, 0x06        ;select digit 5
    LDI    R18, 0x15        ;data = n
    RCALL  send_bytes       ;send command & data to MAX7219
    ;-----
    LDI    R17, 0x05        ;select digit 4
    LDI    R18, 0x7B        ;data = g
    RCALL  send_bytes       ;send command & data to MAX7219
    ;-----
    LDI    R17, 0x04        ;select digit 3
    LDI    R18, 0x7B        ;data = g
    RCALL  send_bytes       ;send command & data to MAX7219
    ;-----
    LDI    R17, 0x03        ;select digit 2
    LDI    R18, 0x06        ;data = i
    RCALL  send_bytes       ;send command & data to MAX7219
    ;-----
    RET

;=====
send_bytes:

```

```

        CBI    PORTB, SS        ;enable slave device MAX7219
        OUT    SPDR, R17        ;transmit command
        ;-----
112:    IN      R19, SPSR
        SBRS   R19, SPIF        ;wait for byte transmission
        RJMP   112              ;to complete
        ;-----
        OUT    SPDR, R18        ;transmit data
        ;-----
113:    IN      R19, SPSR
        SBRS   R19, SPIF        ;wait for byte transmission
        RJMP   113              ;to complete
        ;-----
        SBI    PORTB, SS        ;disable slave device MAX7219
        RET
        ;-----
min10LED:
        LDI    R25, 0x04
        OUT    PORTC, R25
        RJMP   180

min20LED:
        LDI    R25, 0x02
        OUT    PORTC, R25
        RJMP   180

min30LED:
        LDI    R25, 0x01
        OUT    PORTC, R25
        RJMP   180

;=====
byte2decimal:
;-----
        CLR    R26              ;set counter1, initial value 0
        CLR    R27              ;set counter2, initial value 0
        CLR    R30              ;set counter3, initial value 0
        MOV    R24, R28
        LDI    R28, 30
        SUB    R28, R24
        CPI    R28, 11

```

```

BRMI min10LED
CPI R28, 21
BRMI min20LED
CPI R28, 31
BRMI min30LED
LDI R24, 5
MOV R29, R28
;-----
170: CPI R24, 0
BREQ 175
ADD R28, R29
SUBI R24, 1
RJMP 170
;-----
175: CPI R28, 5
BRMI 180
SUBI R30, 1
SUBI R28, 5
RJMP 175
;-----
180: CPI R28, 10 ;compare R28 with 10
BRMI dsp ;jump when R28 < 10
INC R27 ;increment counter2 by 1
SUBI R28, 10 ;R28 = R28 - 10
RJMP 180
;-----
dsp: MOV R18, R27
LDI R17, 0x02 ;select digit 1 for MSD
RCALL send_bytes ;send command & data to MAX7219
;-----
MOV R18, R28
LDI R17, 0x01 ;select digit 0 for LSD
RCALL send_bytes ;send command & data to MAX7219
;-----
RET
;=====
;delay subroutines
;=====
delay_timer0: ;10 usec delay via Timer 0
;-----
CLR R20

```

```

    OUT    TCNT0, R20        ;initialize timer0 with count=0
    LDI    R20, 20
    OUT    OCR0A, R20        ;OCR0 = 20
    LDI    R20, 0b00001010
    OUT    TCCR0B, R20        ;timer0: CTC mode, prescaler 8
    ;-----
10: IN     R20, TIFR0        ;get TIFR0 byte & check
    SBRS   R20, OCF0A        ;if OCF0=1, skip next instruction
    RJMP   10                ;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_ms:
;-----
    LDI    R21, 255
16: LDI    R22, 255
17: LDI    R23, 50
18: DEC    R23
    BRNE   18
    DEC    R22
    BRNE   17
    DEC    R21
    BRNE   16
    RET

```