

Proyek Akhir Sistem Embedded
Kelompok 9

SmartFLOW

Smart Flood-**L**evel **O**bservation and **W**arning

Alfonsus Tanara Gultom (2306267126)

Jonathan Matius Weni (2306161896)

Siti Amalia Nurfaidah (2306161851)

Wilman Saragih Sitio (2306161776)

Xavier Daniswara (2206030230)

Daftar Isi

01 Introduction

02 Hardware
Overview

03 Software
Implementation

04 Results and
Evaluation

05 Conclusion

- Banjir merupakan salah satu bencana alam yang sering
- terjadi dan dapat menyebabkan kerugian besar, terutama jika tidak terdeteksi sejak dini.
- Sistem pemantauan level air konvensional sering kali
- hanya memberikan informasi saat air sudah mencapai titik kritis.
- Hal ini menyebabkan keterlambatan dalam pengambilan
- tindakan dan meningkatkan risiko terhadap keselamatan dan kerugian materi.

02

Introduction: Solution

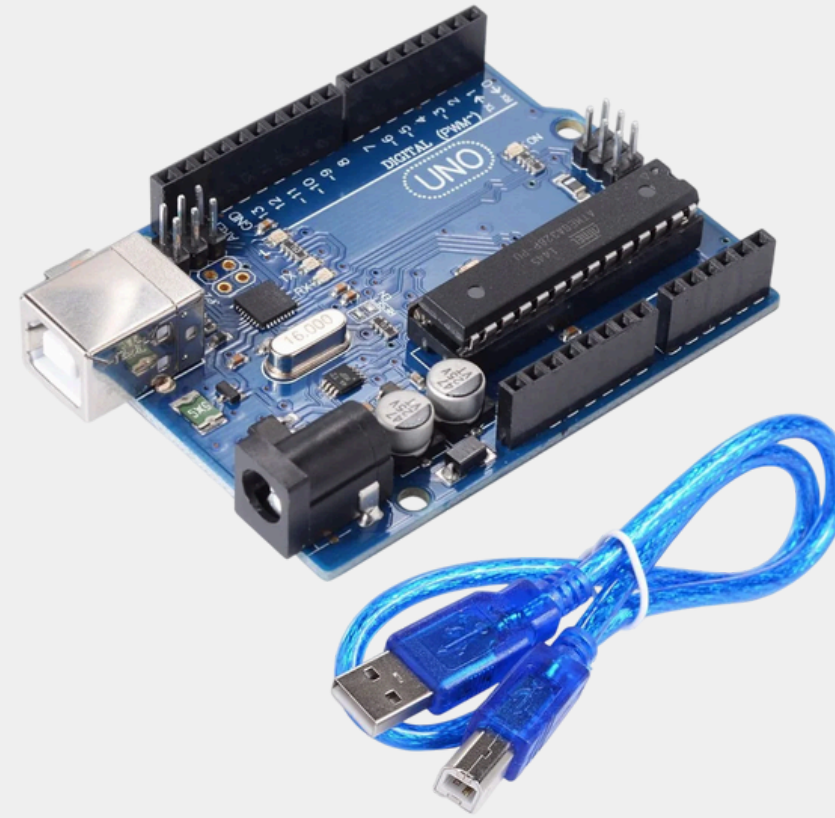
SmartFlow adalah sebuah sistem pemantauan level air pintar yang dirancang untuk memberikan peringatan dini terhadap potensi banjir. Sistem ini menggunakan sensor level air berbasis transistor untuk mendeteksi ketinggian air secara akurat.

Data yang diperoleh akan diproses melalui ADC dan dikirim melalui komunikasi SPI untuk ditampilkan pada serial monitor.

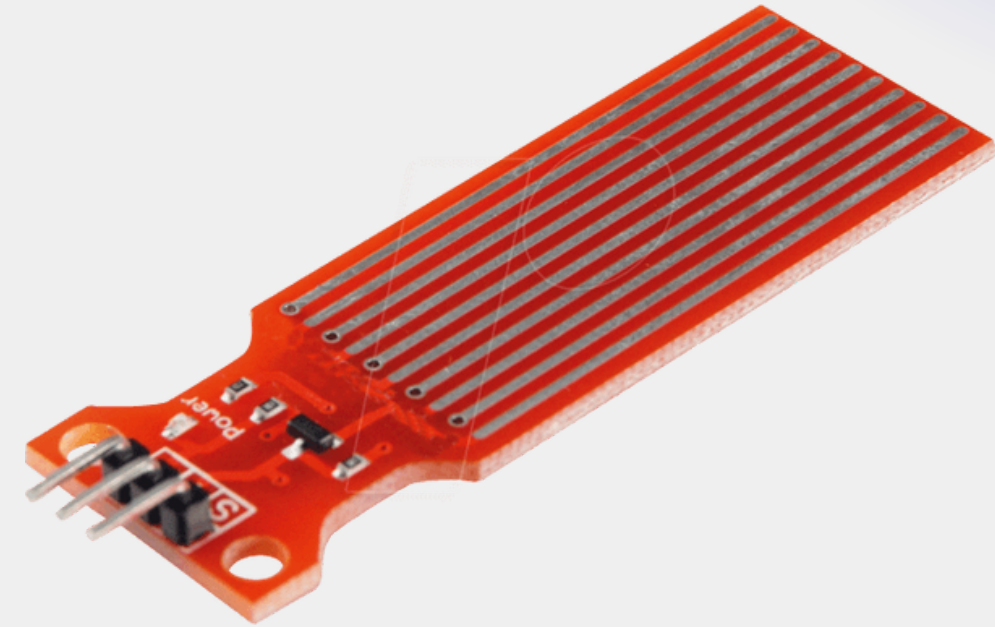
Terdapat LED indikator sebagai penanda visual status level air, serta menggunakan timer dan interrupt untuk pembaruan data secara berkala dan peringatan otomatis saat kondisi berbahaya terdeteksi.

- Real-time Water Level Monitoring: Mengukur dan membaca level air secara langsung.
- Analog Signal Processing menggunakan arus listrik melalui air pada water sensor untuk menghasilkan sinyal proporsional terhadap level air.
- Periodic Data Update menggunakan Timer untuk membaca data secara berkala.
- Early Flood Warning dengan Interrupt, warning akan aktif dan memberikan peringatan ketika level air mencapai ambang kritis.
- LED Indicator untuk memberikan informasi visual tentang status level air (Aman, Waspada, Bahaya).
- SPI Communication pada Arduino master untuk membaca data sensor dan slave untuk menampilkan data di serial monitor.

Komponen utama yang digunakan:



Arduino Uno (Master & Slave)



Liquid Level Sensor Module

05 Hardware Overview: Components

Komponen-komponen lainnya:

- Resistor & Supporting Components
- LED (Hijau, Kuning, Merah)
- Kabel jumper & Breadboard
- Power Supply (5V)

1. Pengaturan Awal dan Inisialisasi

- Pada bagian awal kode diawali dengan mendefinisikan beberapa konfigurasi pin I/O dan menginisialisasi komponen-komponen yang diperlukan.
- Vektor interrupt diatur untuk vektor Reset yang nanti akan mengarahkan eksekusi ke fungsi main, dan alamat untuk vektor interrupt Timer1 Compare Match A yang mengarahkan eksekusi ke TIMER1_COMPA_ISR.

2. fungsi Main

Fungsi main disini akan mengatur pin output dan akan memanggil beberapa subroutine inisialisasi:

- SPI_Init untuk mengatur komunikasi SPI sebagai Master
- ADC_Init untuk mengatur pembacaan sensor level air
- Timer1_Init untuk mengatur timer yang akan mengatur waktu pembacaan sensor
- Pengaktifan global interrupt dengan perintah SEI

3. Main Loop

Di dalam loop ini terjadi beberapa proses:

- Pembacaan nilai sensor level air menggunakan ADC dengan memanggil Read_ADC
- Penskalaan hasil ADC dari 10-bit menjadi 8-bit untuk transmisi dengan Scale_ADC_Result
- Pemeriksaan threshold alarm untuk level air dengan Check_Alarm_Threshold
- Pembaruan indikator LED berdasarkan level air melalui Update_LED_Indicators
- Pengiriman nilai ADC yang telah diskalakan ke slave melalui SPI dengan SPI_Transmit

4. Timer

- Timer1 diatur dalam mode CTC dengan prescaler 1024. Ketika penghitung mencapai nilai compare, akan terjadi interrupt TIMER1_COMPA_ISR.
- Interrupt TIMER1_COMPA_ISR terpicu saat nilai penghitung mencapai nilai compare
- Dalam ISR (Interrupt Service Routine):
 - Status register disimpan saat masuk ke ISR
 - Timer di-reset untuk memulai penghitungan baru
 - Status register dipulihkan sebelum keluar dari ISR
 - Program kembali dari interrupt menggunakan instruksi RETI
- Mekanisme ini memungkinkan main_loop memproses pembacaan ADC secara berkala

5. Pemrosesan ADC

Pembacaan sensor level air dilakukan melalui ADC dengan beberapa tahap:

- ADC diinisialisasi untuk membaca dari channel 0 atau PC0 dengan tegangan referensi AVCC
- Pada fungsi Read_ADC, konversi dimulai dan program menunggu hingga konversi selesai
- Hasil 10-bit diambil untuk diproses
- Hasil diskalakan menjadi 8-bit untuk memudahkan transmisi dan pemrosesan

6. Monitoring

- Subroutine membandingkan nilai ADC yang telah diskalakan dengan threshold maksimum
- Jika nilai melebihi batas:
- Buzzer diaktifkan pada pin PD7 sebagai feedback audio
- LED menyala sesuai kategori level air:
 - PC1 (LED level rendah): aktif jika nilai < 85
 - PC2 (LED level sedang): aktif jika nilai $85-169$
 - PC3 (LED level tinggi): aktif jika nilai ≥ 170
- Sistem memberikan feedback visual dan audio secara instan berdasarkan kondisi ketinggian air

7. SPI

- Sistem menggunakan protokol SPI untuk komunikasi dengan perangkat slave
- Inisialisasi SPI sebagai master dengan konfigurasi pin:
 - MOSI (PB3): output untuk mengirim data
 - SCK (PB5): output clock
 - SS (PB2): output chip select
- Proses transmisi data:
 - Aktivasi slave dengan mengatur SS ke level low
 - Penulisan nilai level air yang telah diskalakan ke register SPDR
 - Menunggu hingga bit SPIF pada register SPSR menandakan transmisi selesai
 - Menonaktifkan slave dengan mengembalikan SS ke level high

1. Pengaturan Awal dan Inisialisasi

- Tahapan inisialisasi sistem:
 - Pemanggilan UART_Init untuk konfigurasi komunikasi serial
 - Eksekusi Send_Welcome_Message untuk menampilkan pesan pembuka di terminal
 - Pengaturan SPI_Slave_Init untuk mengkonfigurasi pin SPI dalam mode slave
 - Aktivasi modul SPI untuk komunikasi dengan master
- Setelah inisialisasi selesai, program masuk ke slave_loop
- Slave_loop berjalan secara kontinyu untuk menerima dan memproses data dari master

2. Inisialisasi SPI sebagai Slave

- Proses inisialisasi SPI sebagai slave meliputi:
 - Konfigurasi pin PB4 (MISO) sebagai output untuk komunikasi balik ke master
 - Pengaturan bit SPE pada register SPCR untuk mengaktifkan modul SPI
 - Konfigurasi mode slave pada register SPCR
- Setelah inisialisasi selesai, slave siap menerima data dari master kapan pun

3. Loop Utama Slave

- Loop utama (slave_loop) menjalankan dua subroutine secara berulang:
 - SPI_Receive: mendeteksi dan menerima data dari master
 - Process_Data: memproses data yang telah diterima
- Proses penerimaan data SPI:
 - Memeriksa bit SPIF pada register SPSR untuk mendeteksi transmisi baru
 - Menyimpan data dari register SPDR ke register R18 saat data baru terdeteksi
 - Mengatur flag di register R20 untuk menandakan data baru tersedia

4. Pemrosesan Data

- Proses pengolahan data dipicu oleh status flag R20:
 - Ketika flag menunjukkan data baru tersedia, Process_Data dijalankan
 - Data dari register R18 disalin ke register R19 untuk pemrosesan
- Subroutine Update_Terminal_Display dipanggil untuk visualisasi:
 - Menampilkan informasi level air ke terminal serial
 - Menunjukkan nilai ketinggian dalam format persentase
 - Memberikan status level dengan kategori: Rendah , Sedang , dan Tinggi

5. Tampilan Level Air di Terminal

- Fungsi `Update_Terminal_Display` menampilkan status air secara visual:
 - Dimulai dengan menampilkan label "Water Level"
 - Menggunakan subrutin `Send_Decimal_Value` untuk konversi nilai 8-bit ke format persentase
- Status level air ditampilkan melalui subrutin `Send_Status_Message` dengan kategorisasi:
 - Low Level: nilai < 33
 - Medium Level: nilai 33–65
 - High Level: nilai > 66
- Tampilan memberikan informasi lengkap tentang ketinggian air dalam bentuk yang mudah dipahami

6. Konversi Desimal ke Persentase

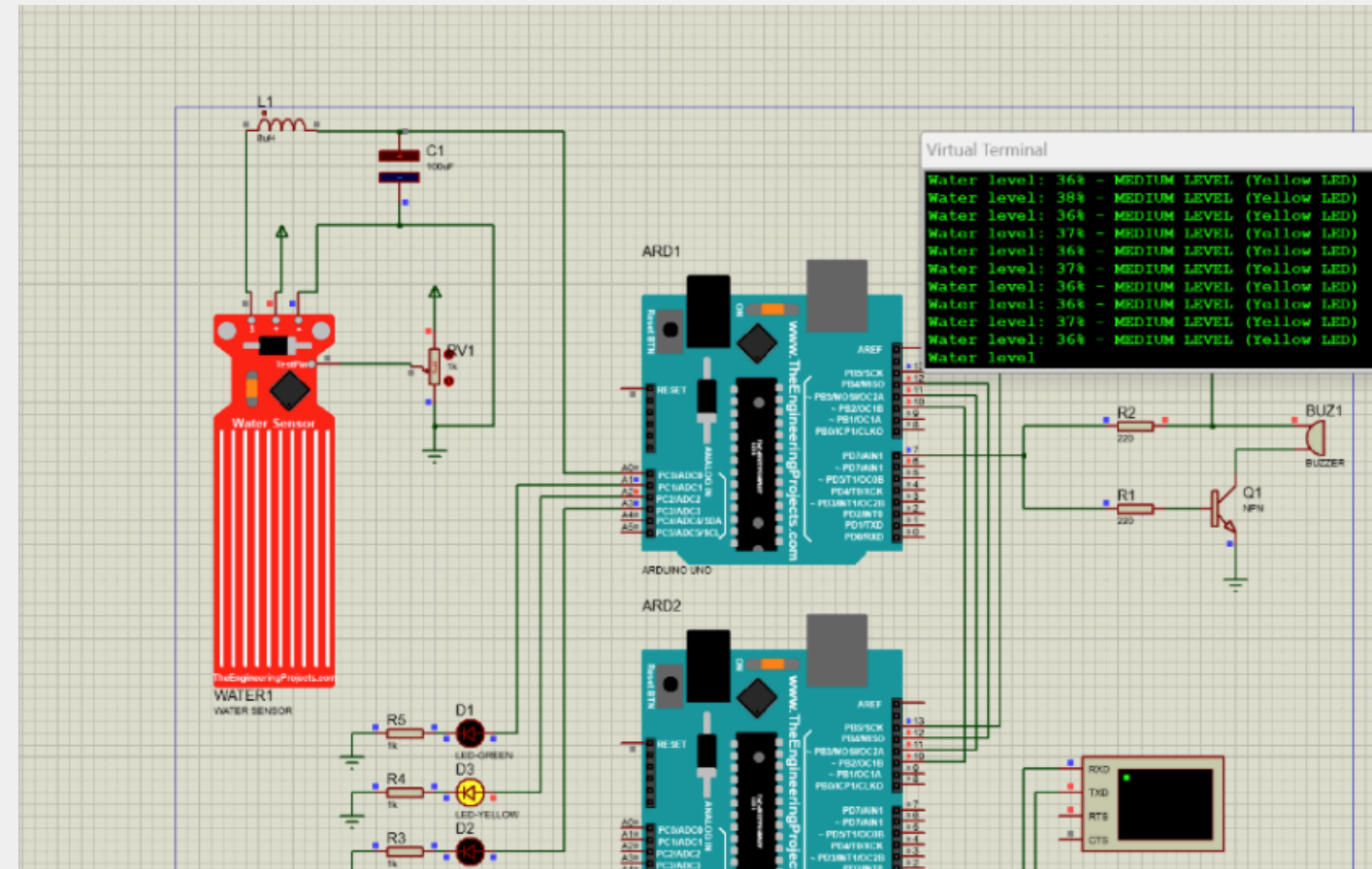
- Proses konversi nilai register R19 ke format persentase:
 - Nilai register dikalikan dengan 100
 - Hasil perkalian dibagi dengan skala maksimum (255)
 - Menghasilkan nilai persentase 0-100%
- Tampilan nilai ke terminal:
 - Digit ratusan, puluhan, dan satuan dikonversi ke karakter ASCII
 - Karakter dikirim secara berurutan ke terminal
 - Simbol persen (%) ditambahkan sebagai penanda satuan
 -

7. Komunikasi UART

- Inisialisasi UART untuk komunikasi dengan terminal:
 - Pengaturan baud rate 9600 melalui register UBRR0L
 - Aktivasi transmitter dengan mengatur bit TXEN0 pada register UCSR0B
 - Konfigurasi format data 8-bit melalui pengaturan register UCSR0C
- Pengiriman data menggunakan fungsi Send_Char:
 - Memeriksa status buffer transmisi sebelum mengirim data
 - Menunggu hingga buffer siap menerima karakter baru
 - Mengirimkan karakter ke register UDR0 untuk transmisi

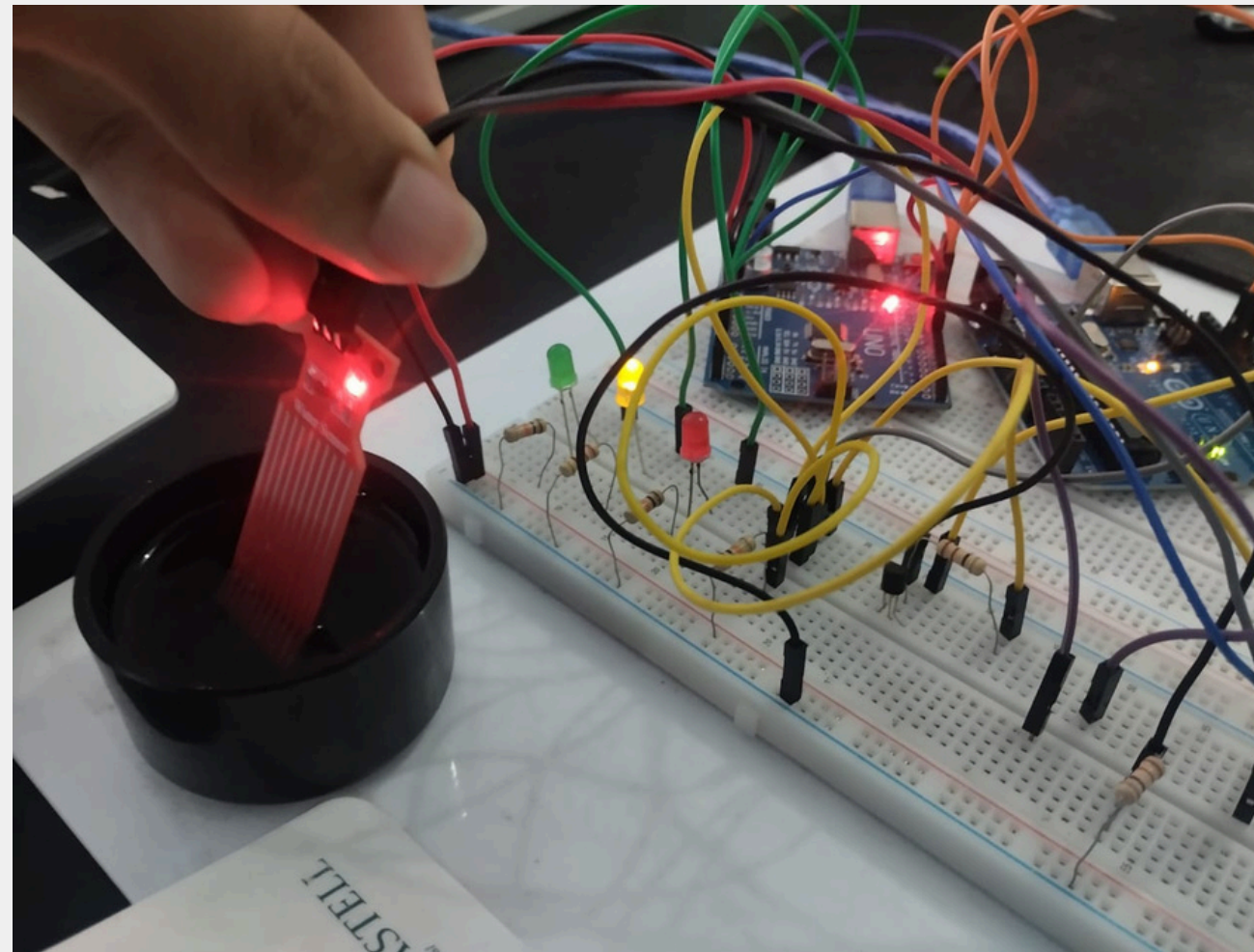
Results & Evaluation

6



Simulasi pada software Proteus menunjukkan bahwa sistem SmartFlow dapat berfungsi sesuai dengan batasan level air yang telah ditentukan. Potensiometer digunakan sebagai test pin untuk mensimulasikan nilai 0% (Low), 37% (Medium), dan 82% (High), dan LED indikator berhasil menyala sesuai kondisi. Selain itu, serial monitor pada Proteus juga menampilkan persentase level air dengan tepat.

Results & Evaluation



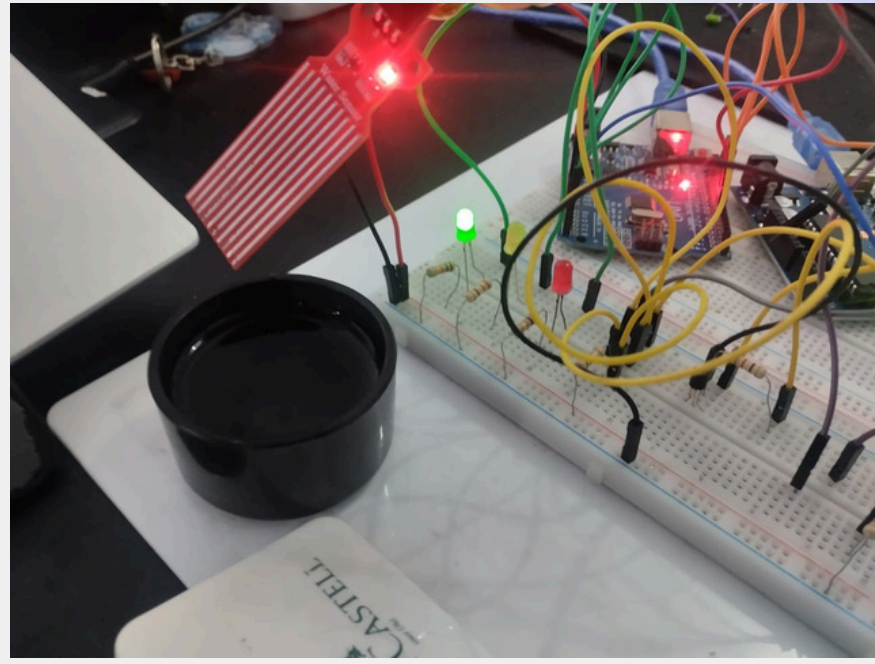
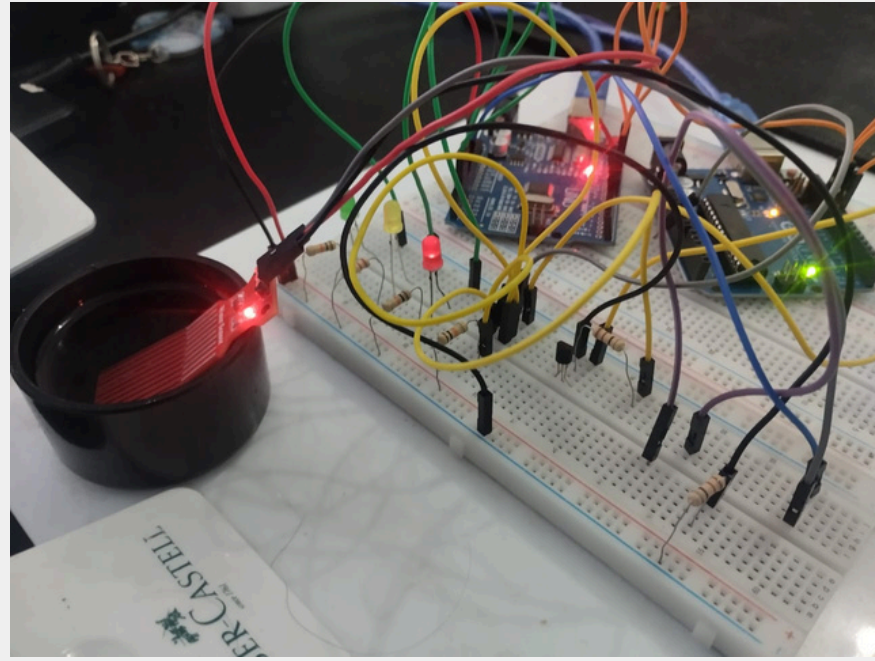
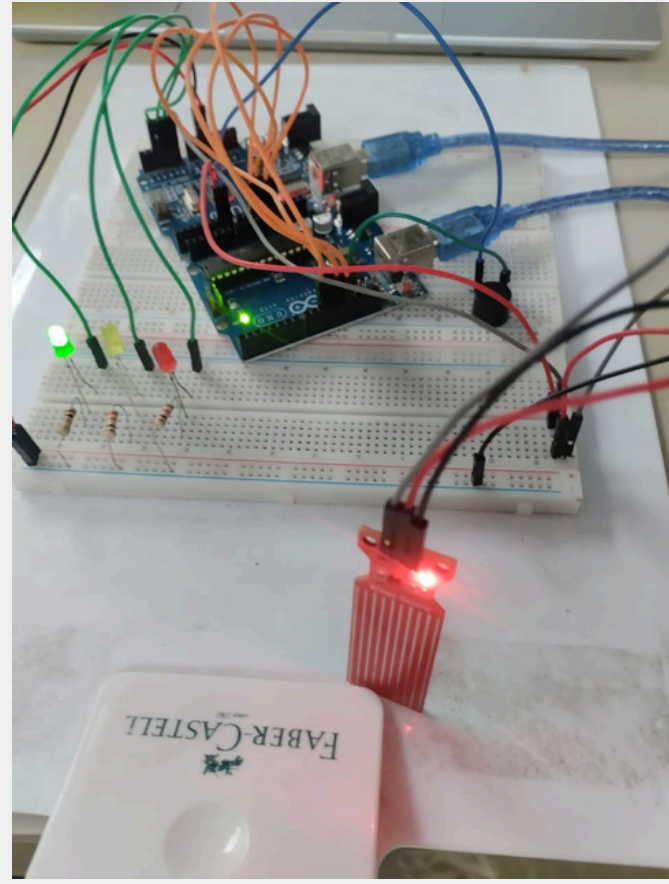
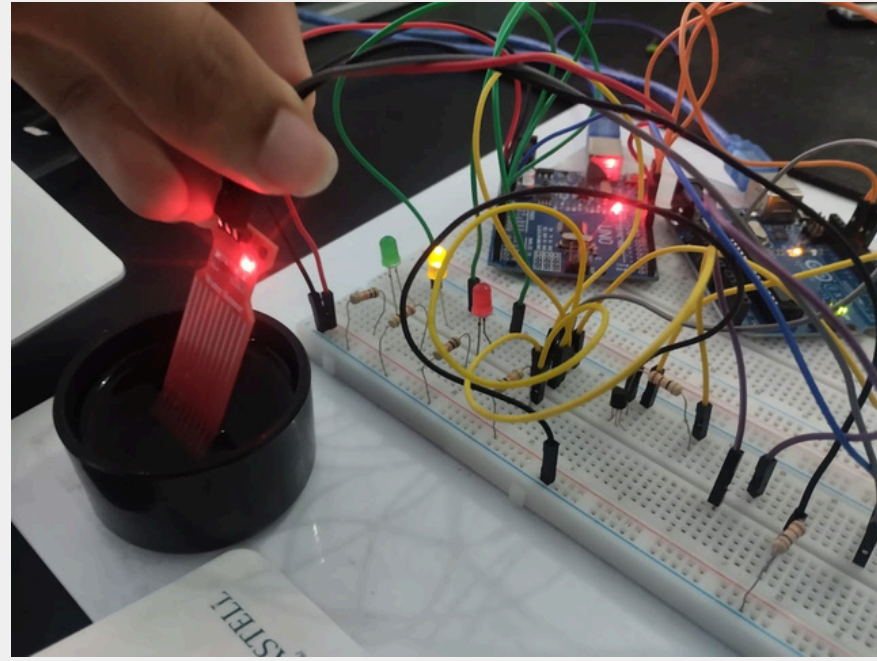
Pada pengujian fisik, sensor air mampu memicu LED indikator (hijau, kuning, merah) sesuai kedalaman sensor dalam air. Namun, persentase pada serial monitor Arduino IDE belum mencerminkan kondisi sebenarnya karena nilai persentasenya tidak sesuai dengan kedalaman sensor.

SmartFlow berhasil diimplementasikan dengan dua Arduino sebagai Master dan Slave yang terhubung melalui SPI. Sistem mampu membaca data sensor, menyalakan LED indikator, dan mengaktifkan buzzer ketika level air melebihi batas. Slave menampilkan data secara deskriptif melalui UART ke serial monitor.

Penggunaan interrupt dan timer mendukung pemrosesan data secara real-time. Meski ada kendala dalam konversi nilai sensor, fungsi utama sistem telah berhasil dijalankan. Masalah utama terdapat pada konversi nilai sensor di Arduino Slave. Nilai maksimum sensor air bisa mencapai 440, sementara nilai digital yang dikirim Master hanya sampai 255. Ini menyebabkan proses pemetaan dan konversi ke persen di kode Slave menjadi tidak akurat. Hal ini menyulitkan proses pemetaan nilai menjadi 0–100% secara akurat. Meskipun demikian, kesalahan ini terbatas hanya pada tampilan persentase, tanpa mempengaruhi sistem indikator LED maupun alarm yang bekerja sesuai batasan.

Kesimpulan:

- SmartFlow berhasil diimplementasikan menggunakan dua Arduino (Master dan Slave) yang bekerja sama melalui komunikasi SPI, dan seluruh fitur seperti sensor air, LED indikator, buzzer, dan tampilan serial monitor dapat berjalan.
- Sistem berhasil mendeteksi level air secara real-time, mengaktifkan alarm jika pada high level, dan menyampaikan status melalui indikator visual dan teks.
- Kendala utama terdapat pada konversi nilai sensor ke persentase di Slave, namun dengan sensor yang lebih baik dan skala lebih besar, sistem ini dapat digunakan di daerah rawan bencana sebagai alat monitoring level air yang efektif.



Thank You

Proyek Akhir Sistem Embedded
Kelompok 9