MAKERTHON 2023

ACURA INTEGRA

CONNECTING IMAGINATION AND REALITY

Our Team

ACURA INTEGRA



Vinod Rao a/l Jayaprasadh



Muhammad Affan Nur iman bin Azman

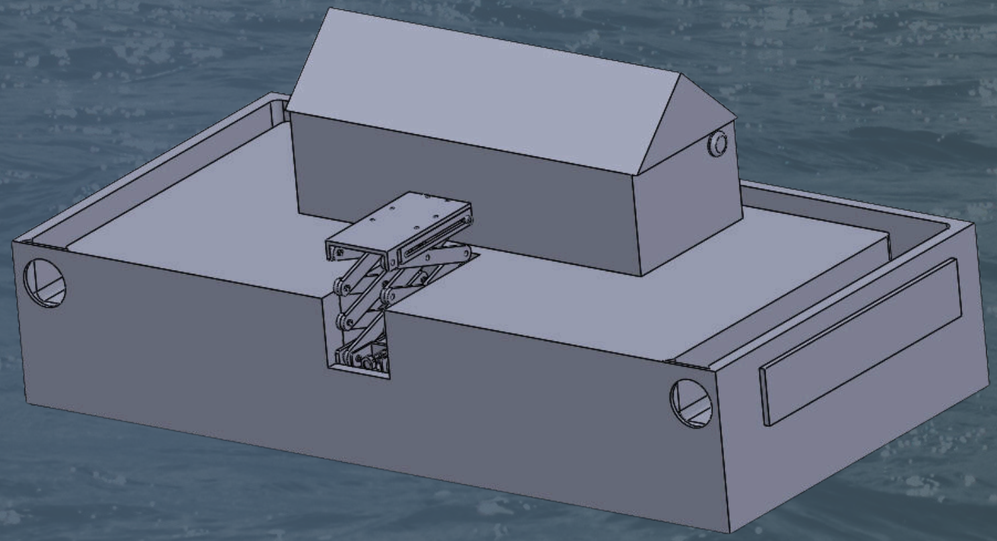


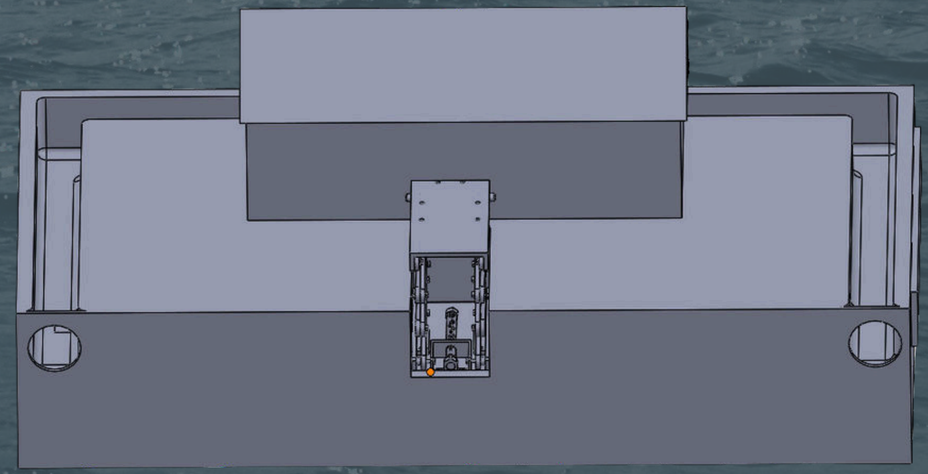
Nurul Ain Fazwin binti Mohamad Sakhmah

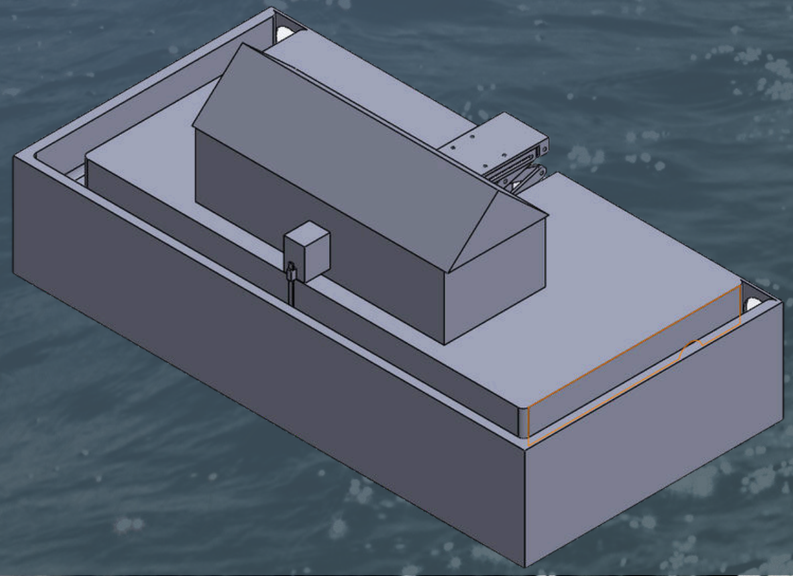


SITI AISYAH BINTI JAMIL

Back to Agenda Page







Early Flood Detection with Scissor Lift

Overview

The proposed IoT-based Early House Flood Detection System with a Scissor Lift aims to address this pressing issue on Malaysian flood crisis



OBJECTIVES

1. Flood Detection
2. Real-time Alerts & Remote Monitoring
3. Cost-Effective Project

GOALS

1. Build Mockup in Solidworks
2. Working Prototype



Methodology

NEWS SURVEY

comprehensively survey news reports from various sources, including TV broadcasts, over several years

SOLIDWORKS FOR 3D MODEL MOCKUP

to create a detailed 3D model mockup of the proposed IoT device

SENSOR DEPLOYMENT

Water sensors will be strategically placed in areas prone to flooding, such as basements, near water fixtures, and low-lying regions.

ARDUINO IDE FOR PROGRAMMING

(IDE) will serve as our programming platform for the microcontroller



Methodology

BLYNK FOR USER INTERFACE DEVELOPMENT

The Blynk website to create a user interface that can be controlled through smartphones

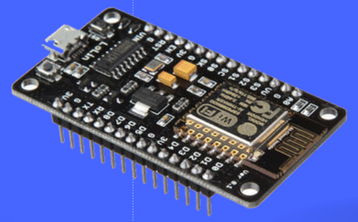
PROTOTYPE DEVELOPMENT

Assembling the scissor lift mechanism, connecting the water sensors, microcontroller, and communication module

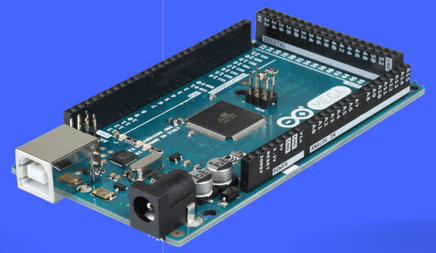
TESTING AND VALIDATION

Detect water levels, provide timely alerts, and accurately control the scissor lift mechanism.

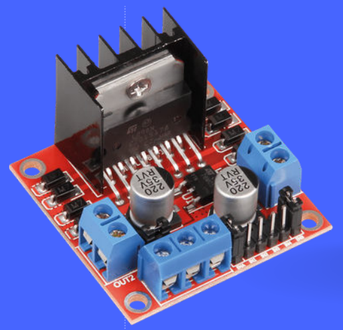
Component List



NodeMCU ESP 8266



Arduino Mega

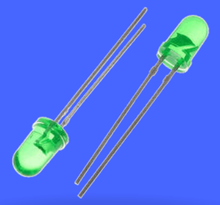


Motor Driver LM298N

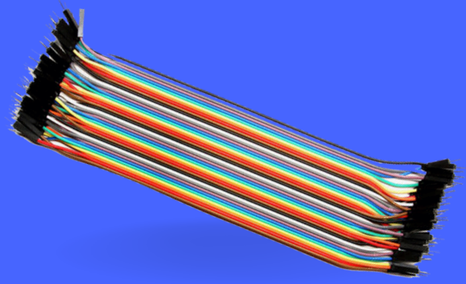
Component List



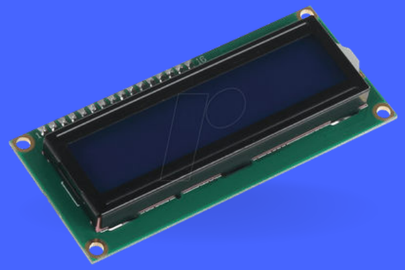
Buzzer



LED 5mm



Jumper Cables

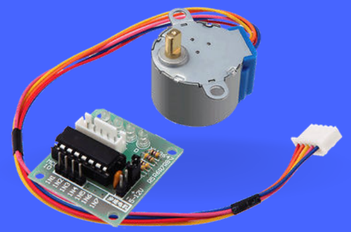


16 x 2 LCD

Component List



USB-Type B Cable

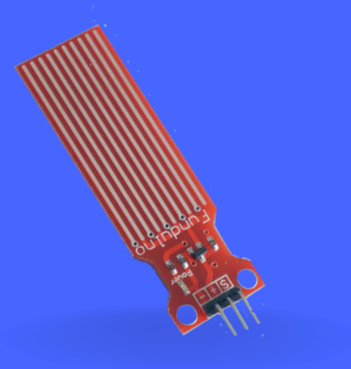


5V DC Motor

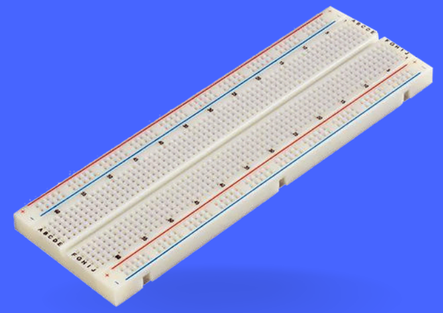


5V DC Power Supply

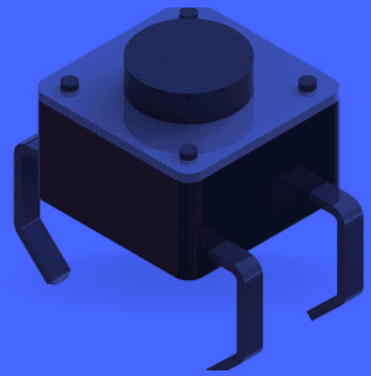
Component List



Water Sensor

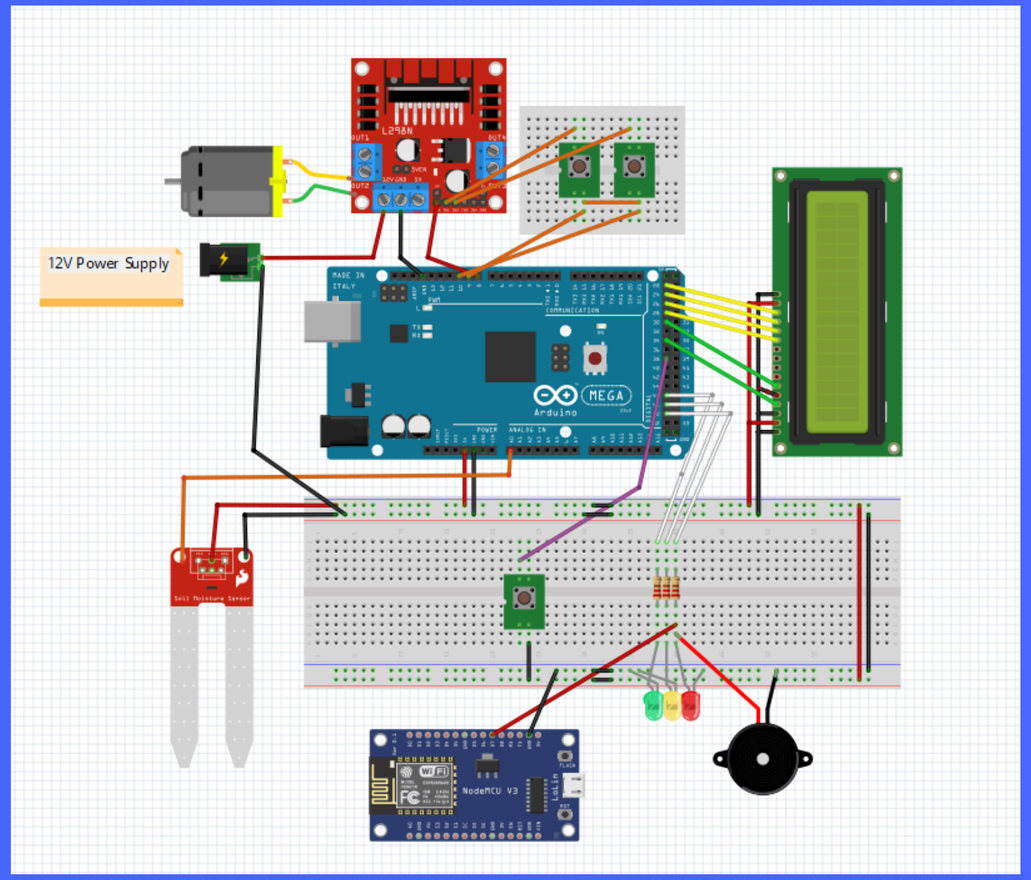


Breadboard

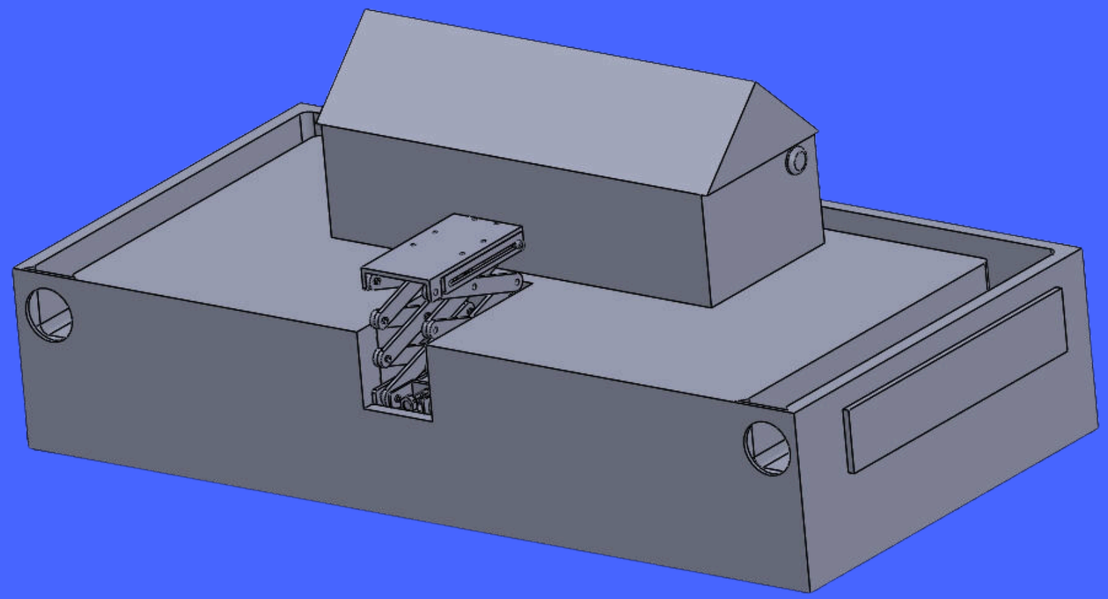


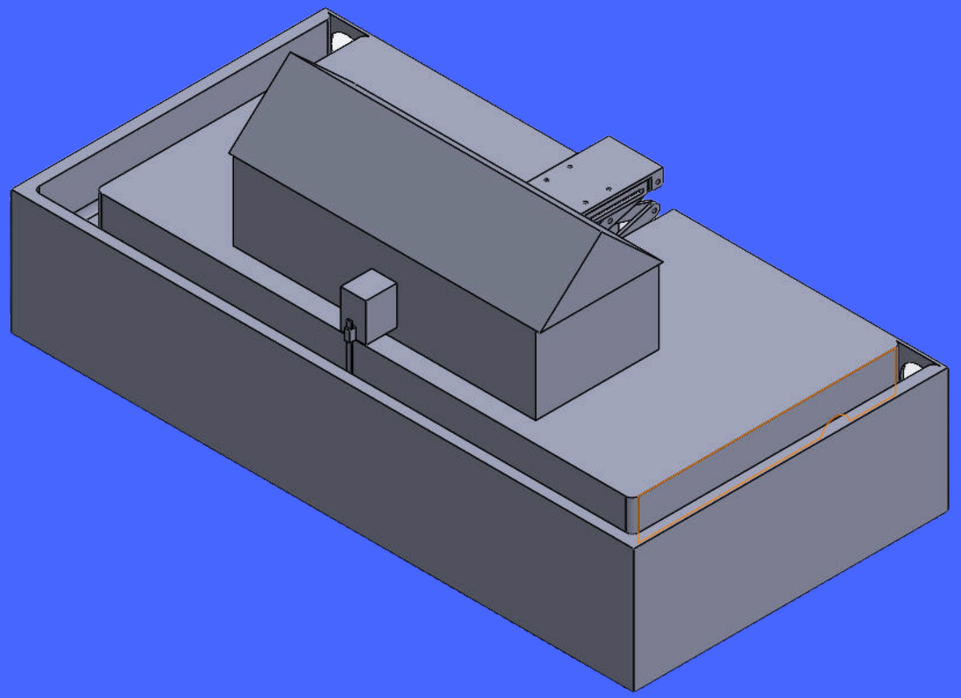
Switch

Schematic Diagram



3D Model





Code Block

```
#include <BlynkSimpleEsp8266.h>  // Include the Blynk library
#include <LiquidCrystal.h>       // Include the LiquidCrystal library


// Pin assignments for Arduino Mega
const int waterSensorPin = A0;          // Analog pin for water sensor
const int relayPin = 7;                 // Digital pin for relay
const int motorEnablePin = 8;           // Digital pin for motor enable
const int motorInput1Pin = 9;           // Digital pin for motor input
const int motorInput2Pin = 10;          // Digital pin for motor input
const int cwSwitchPin = 2;              // Digital pin for CW rotation
switch
const int ccwSwitchPin = 3;             // Digital pin for CCW rotation
switch
const int pumpPin = 11;                 // Digital pin for water pump
const int emergencyButtonPin = 12;      // Digital pin for emergency bu
const int redLedPin = 4;                // Digital pin for red LED
const int greenLedPin = 5;              // Digital pin for green LED
const int yellowLedPin = 13;            // Digital pin for yellow LED


// Pin assignments for NodeMCU ESP8266
const char* ssid = "your_wifi_ssid";
const char* password = "your_wifi_password";
char auth[] = "your_blynk_auth_token";


// Thresholds for water levels
const int floodThreshold = 500;         // Adjust this value based on
sensor readings
const int residenceThreshold = 200;     // Adjust this value based on
sensor readings


// LCD screen object
LiquidCrystal lcd(32, 30, 28, 26, 24, 22);  // Modify the pin numbers
accordingly
```

```
void setup() {
  Serial.begin(9600);  // Initialize serial communication for debugging

  // Initialize LCD screen
  lcd.begin(16, 2);
  lcd.print("Water Level:");


  // Connect to WiFi using NodeMCU
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Blynk.begin(auth, WiFi.localIP());

  pinMode(relayPin, OUTPUT);            // Set relay pin as output
  pinMode(motorEnablePin, OUTPUT);      // Set motor enable pin as output
  pinMode(motorInput1Pin, OUTPUT);      // Set motor input 1 pin as output
  pinMode(motorInput2Pin, OUTPUT);      // Set motor input 2 pin as output
  pinMode(cwSwitchPin, INPUT_PULLUP);   // Set CW rotation switch pin as
input with internal pull-up resistor
  pinMode(ccwSwitchPin, INPUT_PULLUP);  // Set CCW rotation switch pin as
input with internal pull-up resistor
  pinMode(pumpPin, OUTPUT);             // Set water pump pin as output
  pinMode(emergencyButtonPin, INPUT_PULLUP);  // Set emergency button pin
as input with internal pull-up resistor
  pinMode(redLedPin, OUTPUT);           // Set red LED pin as output
  pinMode(greenLedPin, OUTPUT);         // Set green LED pin as output
  pinMode(yellowLedPin, OUTPUT);        // Set yellow LED pin as output
}


void loop() {
  Blynk.run();  // Run the Blynk library


  // Read water sensor value
  int waterLevel = analogRead(waterSensorPin);
```

Code Block

```
// Update LCD screen with water level
lcd.setCursor(0, 1);
lcd.print("W: ");
lcd.print(waterLevel);


// Check if water level is above the flood threshold
if (waterLevel > floodThreshold) {
  digitalWrite(redLedPin, HIGH);       // Turn on red LED
  digitalWrite(greenLedPin, LOW);      // Turn off green LED
  digitalWrite(yellowLedPin, LOW);     // Turn off yellow LED
  digitalWrite(relayPin, HIGH);        // Activate relay
  digitalWrite(pumpPin, HIGH);         // Turn on water pump


  // Check the CW rotation switch state
  if (digitalRead(cwSwitchPin) == LOW) {
    digitalWrite(motorInput1Pin, HIGH);   // Set motor input 1 to HIGH
    digitalWrite(motorInput2Pin, LOW);    // Set motor input 2 to LOW
    digitalWrite(motorEnablePin, HIGH);   // Enable the motor
  }
  // Check the CCW rotation switch state
  else if (digitalRead(ccwSwitchPin) == LOW) {
    digitalWrite(motorInput1Pin, LOW);    // Set motor input 1 to LOW
    digitalWrite(motorInput2Pin, HIGH);   // Set motor input 2 to HIGH
    digitalWrite(motorEnablePin, HIGH);   // Enable the motor
  }
  else {
    digitalWrite(motorEnablePin, LOW);    // Disable the motor
  }


  // Send a warning message if water level is at the red level
  if (waterLevel > residenceThreshold) {
    Blynk.notify("Flood detected! Take immediate action!");
  }
 }
// Check if water level is between residence threshold and flood
```

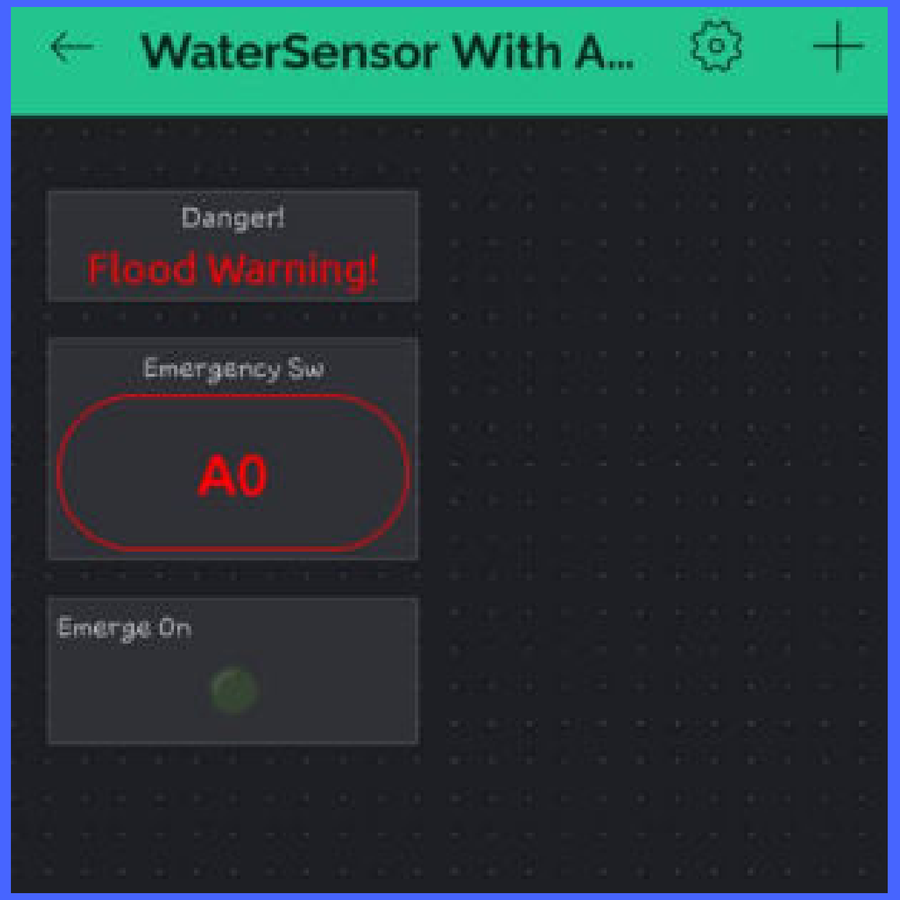
```
 else if (waterLevel > residenceThreshold) {
   digitalWrite(redLedPin, LOW);        // Turn off red LED
   digitalWrite(greenLedPin, LOW);      // Turn off green LED
   digitalWrite(yellowLedPin, HIGH);    // Turn on yellow LED
   digitalWrite(relayPin, LOW);         // Deactivate relay
   digitalWrite(pumpPin, LOW);          // Turn off water pump
   digitalWrite(motorEnablePin, LOW);   // Disable the motor
 }
 // Water level is below residence threshold
 else {
   digitalWrite(redLedPin, LOW);        // Turn off red LED
   digitalWrite(greenLedPin, HIGH);     // Turn on green LED
   digitalWrite(yellowLedPin, LOW);     // Turn off yellow LED
   digitalWrite(relayPin, LOW);         // Deactivate relay
   digitalWrite(pumpPin, LOW);          // Turn off water pump
   digitalWrite(motorEnablePin, LOW);   // Disable the motor
 }


 // Check if the emergency button is pressed
 if (digitalRead(emergencyButtonPin) == LOW) {
   // Simulate water sensor trigger
   waterLevel = 1000;
 }


 delay(1000);  // Delay for stability
}
```

Blynk

User Interface



Component Cost



Expense	Amount (RM)
Arduino Mega	70.00
NodeMCU ESP 8266	14.90
Motor DriverLM298N	4.90
Step Up	

Component Cost



Expense	Amount (RM)
Buzzer	0.70
LED 5mm - Red	0.10
LED 5mm - Green	0.10
LED 5mm - Yellow	0.10
Jumper Cables (Male to Male)	4.50
Jumper Cables (Male to Female)	2.00
Jumper Cables (Female to Female)	2.00

Component Cost



Expense	Amount (RM)
16 x 2 LCD	8.00
USB Type B Cable	3.20
5V DC Motor	5.00

Component Cost



Expense	Amount (RM)
Water Sensor	4.90
Breadboard	7.80
Switch	0.50
Total Altogether	165.00

MAKERTHON 2023



IOT PROTOTYPE Demonstration

Early House Flood Detection System with a Scissor Lift


End of Presentation
ANY QUESTIONS?
memecreator.org