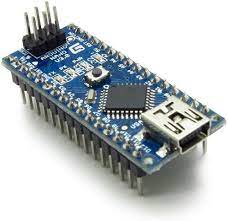
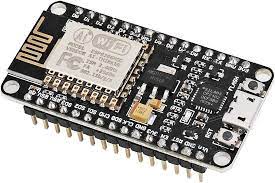


Naan Mudhalvan -IOT

# Problem Statement: SMART WATER MANAGEMENT

MENTOR NAME: EVALUATOR NAME:

Mr.MOHANKUMAR.M Ms.AKILANDESWARI.M

REQUIRED COMPONENTS :  
   
 1) Arduino Nano 35000  
  
   
 2) NodeMCU 50000  
  
   
 3) Adafruit OLED 70000  
  
   
  
   
  
  
  
  
  
4) Casing Box 6000  
   
  
 5) Kabel 1000  
   
 6) PCB 10000  
  
   
  
  
   
  
  
  
7) Arduino Nano 35000   
   
   
   
   
**SOURCE CODE:**

#define \_DISABLE\_TLS\_

#include <ThingerESP8266.h>

#include <ESP8266WiFi.h>

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define OLED\_RESET LED\_BUILTIN

#define USERNAME "test123"

#define DEVICE\_ID "SWM"

#define DEVICE\_CREDENTIAL "ABCDEFGHIJ"

#define SSID "test123"

#define SSID\_PASSWORD "test123"

Adafruit\_SSD1306 display(OLED\_RESET);

byte indikator = 13;

byte sensorInt = 0;

byte flowsensor = D3;

float konstanta = 4.5; //konstanta flow meter

volatile byte pulseCount;

float debit;

float harga;

unsigned int flowmlt;

Unsigned long totalmlt;

unsigned long old-time;

ThingerESP8266 thing(USERNAME, DEVICE\_ID, DEVICE\_CREDENTIAL);

void setup()

{

display.begin(SSD1306\_SWITCHCAPVCC, 0x3C);

display.clearDisplay();

display.display();

display.setTextSize(1);

display.setTextColor(WHITE);

display.setCursor(0, 0);

// Inisialisasi port serial

Serial.begin(9600);

pinMode(flowsensor, INPUT);

pulseCount = 0;

debit = 0.0;

flowmlt = 0;

totalmlt = 0;

old-time = 0;

harga = 0.0;

// digital pin control example (i.e. turning on/off a light, a relay, configuring aparameter, etc)

thing["sensor"] >> [](pson& out){

digitalWrite(flowsensor, HIGH);

attachInterrupt(digitalPinToInterrupt(D3), pulseCounter, FALLING);

out["debit"] = debit;

out["volume"] = totalmlt;

out["harga"] = harga;

};

}

void loop()

{

thing.handle();

display.clearDisplay();

if((millis() - old-time) > 1000)

{

detachInterrupt(sensorInt);

debit = ((1000.0 / (millis() - old-time)) \* pulseCount) / konstanta;

old-time = millis();

flowmlt = (debit / 60) \* 1000;

totalmlt += flowmlt;

harga = totalmlt\*0.002;

unsigned int frac;

Serial.print("Debit air: ");

Serial.print(int(debit));

Serial.print("L/min");

Serial.print("\t");

display.setCursor(0, 0);

display.print("Debit air: ");

display.setCursor(60, 0);

display.print(int(debit));

display.setCursor(85, 0);

display.print("L/min");

Serial.print("Volume: ");

Serial.print(totalmlt);

Serial.print("mL");

Serial.print ("\t");

display.setCursor(0, 12);

display.print("Volume: ");

display.setCursor(50, 12);

display.print (totalmlt);

display.setCursor(100, 12);

display.print("mL");

display.print("\t");

Serial.print("Harga: ");

Serial.print("Rp ");

Serial.println(harga);

display.setCursor(0, 24);

display.print("Harga: ");

display.setCursor(45, 24);

display.print("Rp ");

display.setCursor(70, 24);

display.println(harga);

display.display();

pulseCount = 0;

attachInterrupt(digitalPinToInterrupt(D3), pulseCounter, FALLING);

}

}

void pulseCounter()

{

// Increment the pulse counter

pulseCount++;

**Working Summary:**

The provided Arduino sketch monitors water consumption using a flow sensor and an ESP8266 microcontroller. Here's how it works:

**1. Initialization:**

- The program initializes various components, including the OLED display, WiFi connection, and interrupts for the flow sensor.

- WiFi credentials and ThingSpeak IoT platform details are set up.

**2. Flow Sensor:**

- The flow sensor is connected to pin D3 of the ESP8266. It measures the water flow rate in pulses.

- An interrupt service routine (`pulseCounter()`) is used to count pulses from the flow sensor, which indicates water flow.

**3. Data Calculation:**

- The program calculates water flow rate (`debit`) in L/min based on pulse count and a constant (`konstanta`).

- Total water volume (`totalmlt`) in mL is calculated using the flow rate, and the cost (`harga`) of the consumed water in Indonesian Rupiah (Rp) is calculated (assuming a cost of Rp 0.002 per mL).

**4. Data Transmission:**

- The measured data (flow rate, total volume, and cost) is sent to the ThingSpeak IoT platform through the `thing.handle()` function, making it accessible for further analysis and visualization.

**5. Display:**

- Real-time data, including flow rate, total volume, and cost, is displayed on an OLED screen and sent to the serial monitor for debugging.

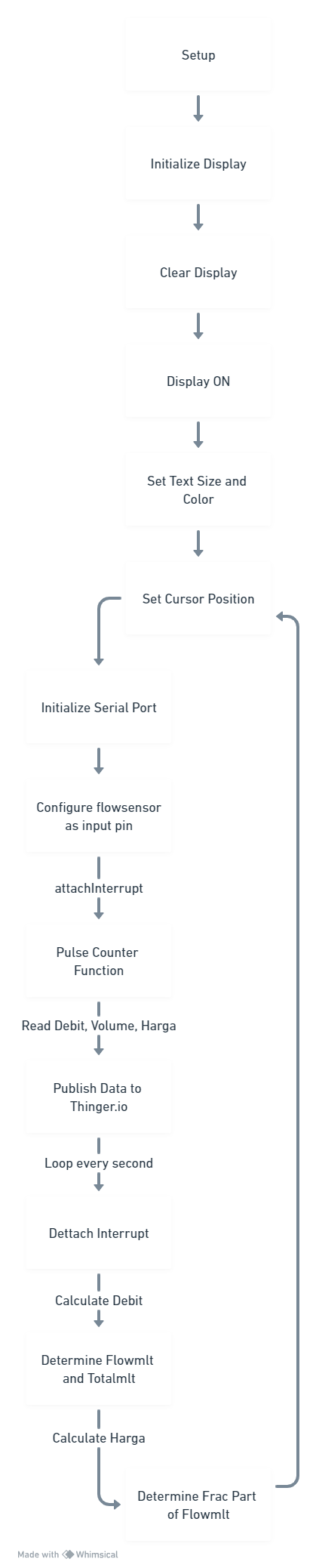
- The display updates every second with the latest water consumption information.

**6. Interrupt Handling:**

- Interrupts are used to accurately count pulses from the flow sensor, ensuring precise measurement of water flow.

**7. Continuous Monitoring:**

- The program continuously monitors water flow, calculates consumption metrics, and updates the display and IoT platform with real-time data.

**  
  
Conclusion:** The system offers real-time monitoring and alerts for maintaining a SMART WATER MANAGEMENT