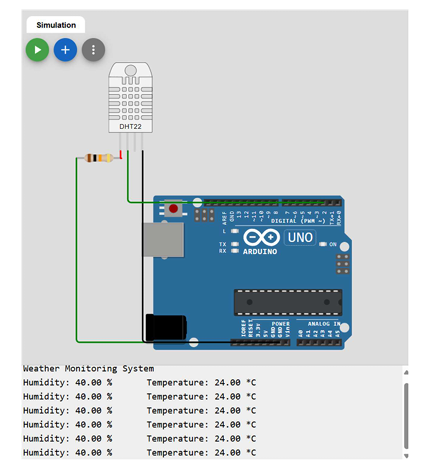
IOT – ASSIGNMENT

Pranav Rao Aiml-epsilon

2211cs020420

1. Write an Embedded C program to Create a Weather Reporting System that provides real-time environmental data to users.



#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#include <string.h>

#include "dht22.h"

#include "bmp280.h"

void initialize\_sensors(void);

void read\_temperature\_and\_humidity(float \*temperature, float \*humidity);

void read\_pressure(float \*pressure);

void display\_data(float temperature, float humidity, float pressure);

int main(void) {

float temperature = 0.0, humidity = 0.0, pressure = 0.0;

initialize\_sensors();

printf("Weather Reporting System Initialized.\n");

while (1) {

read\_temperature\_and\_humidity(&temperature, &humidity);

read\_pressure(&pressure);

display\_data(temperature, humidity, pressure);

\_delay\_ms(1000);

}

return 0;

}

void initialize\_sensors(void) {

if (!DHT22\_init()) {

printf("Error initializing DHT22 sensor!\n");

exit(1);

}

if (!BMP280\_init()) {

printf("Error initializing BMP280 sensor!\n");

exit(1);

}

}

void read\_temperature\_and\_humidity(float \*temperature, float \*humidity) {

if (!DHT22\_read(temperature, humidity)) {

printf("Error reading from DHT22 sensor!\n");

}

}

void read\_pressure(float \*pressure) {

if (!BMP280\_read\_pressure(pressure)) {

printf("Error reading from BMP280 sensor!\n");

}

}

void display\_data(float temperature, float humidity, float pressure) {

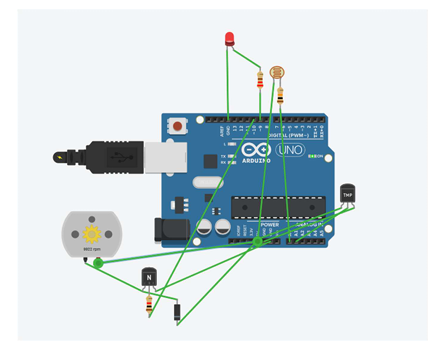
printf("Temperature: %.2f°C\n", temperature);

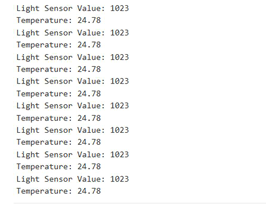
printf("Humidity: %.2f%%\n", humidity);

printf("Pressure: %.2f hPa\n", pressure);

}

2.Write a Embedded C Program to Create a Nome Automation System that simplifies daily routines (Any 2 devices) by controlling devices remotely.





#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#include <string.h>

#define LIGHT\_PIN 2

#define FAN\_PIN 3

void initialize\_devices(void);

void control\_light(uint8\_t state);

void control\_fan(uint8\_t state);

void process\_command(char \*command);

int main(void) {

char command[20];

initialize\_devices();

printf("Home Automation System Initialized.\n");

while (1) {

printf("Waiting for command...\n");

scanf("%s", command);

process\_command(command);

}

return 0;

}

void initialize\_devices(void) {

GPIO\_set\_mode(LIGHT\_PIN, OUTPUT);

GPIO\_set\_mode(FAN\_PIN, OUTPUT);

GPIO\_write(LIGHT\_PIN, 0);

GPIO\_write(FAN\_PIN, 0);

}

void control\_light(uint8\_t state) {

if (state) {

GPIO\_write(LIGHT\_PIN, 1);

printf("Light turned ON.\n");

} else {

GPIO\_write(LIGHT\_PIN, 0);

printf("Light turned OFF.\n");

}

}

void control\_fan(uint8\_t state) {

if (state) {

GPIO\_write(FAN\_PIN, 1);

printf("Fan turned ON.\n");

} else {

GPIO\_write(FAN\_PIN, 0);

printf("Fan turned OFF.\n");

}

}

void process\_command(char \*command) {

if (strcmp(command, "LIGHT\_ON") == 0) {

control\_light(1);

} else if (strcmp(command, "LIGHT\_OFF") == 0) {

control\_light(0);

} else if (strcmp(command, "FAN\_ON") == 0) {

control\_fan(1);

} else if (strcmp(command, "FAN\_OFF") == 0) {

control\_fan(0);

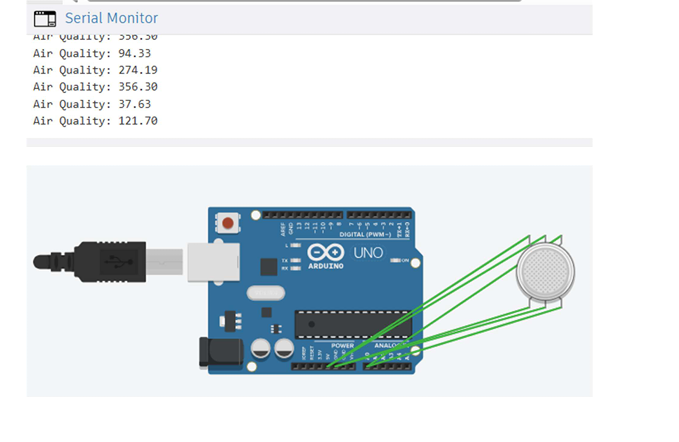
} else {

printf("Invalid command.\n");

}

}

3.Write an Embedded C program to create an Air Pollution Monitoring System that tracks air quality levels in real-time to ensure a healthier environment.



#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#include "mq135.h"

#define AQI\_THRESHOLD 100

void initialize\_system(void);

int read\_air\_quality(void);

void display\_air\_quality(int aqi);

void check\_air\_quality(int aqi);

int main(void) {

int air\_quality;

initialize\_system();

printf("Air Pollution Monitoring System Initialized.\n");

while (1) {

air\_quality = read\_air\_quality();

display\_air\_quality(air\_quality);

check\_air\_quality(air\_quality);

\_delay\_ms(1000);

}

return 0;

}

void initialize\_system(void) {

if (!MQ135\_init()) {

printf("Error initializing MQ135 sensor!\n");

exit(1);

}

}

int read\_air\_quality(void) {

int aqi = MQ135\_read();

return aqi;

}

void display\_air\_quality(int aqi) {

printf("Air Quality Index (AQI): %d\n", aqi);

}

void check\_air\_quality(int aqi) {

if (aqi > AQI\_THRESHOLD) {

printf("Warning: Poor air quality detected! AQI: %d\n", aqi);

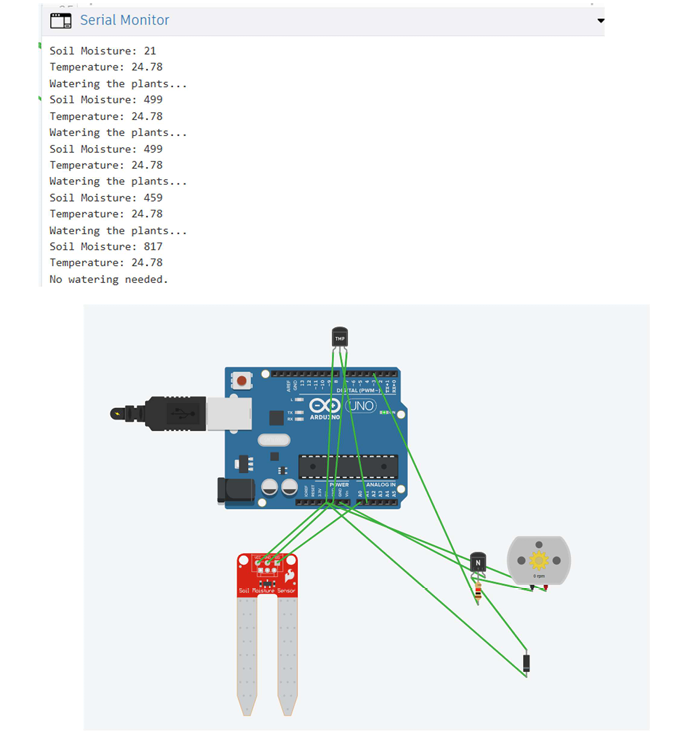
} else {

printf("Air quality is good. AQI: %d\n", aqi);

}

}

4.Write an Embedded C program to create an IoT based smart Irrigation system for Agriculture that automates watering based on weather and soil conditions



#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#include "soil\_moisture.h"

#include "dht22.h"

#include "wifi\_module.h"

#define SOIL\_MOISTURE\_THRESHOLD 40

#define TEMPERATURE\_THRESHOLD 35

#define HUMIDITY\_THRESHOLD 60

#define PUMP\_PIN 3

void initialize\_system(void);

void control\_irrigation(int soil\_moisture, int temperature, int humidity);

void fetch\_weather\_data(int \*temperature, int \*humidity);

int main(void) {

int soil\_moisture = 0;

int temperature = 0;

int humidity = 0;

initialize\_system();

printf("IoT-Based Smart Irrigation System Initialized.\n");

while (1) {

soil\_moisture = SoilMoisture\_read();

fetch\_weather\_data(&temperature, &humidity);

control\_irrigation(soil\_moisture, temperature, humidity);

\_delay\_ms(5000);

}

return 0;

}

void initialize\_system(void) {

if (!SoilMoisture\_init()) {

printf("Error initializing Soil Moisture sensor!\n");

exit(1);

}

if (!DHT22\_init()) {

printf("Error initializing DHT22 sensor!\n");

exit(1);

}

if (!WiFiModule\_init()) {

printf("Error initializing WiFi Module!\n");

exit(1);

}

GPIO\_set\_mode(PUMP\_PIN, OUTPUT);

GPIO\_write(PUMP\_PIN, 0);

}

void control\_irrigation(int soil\_moisture, int temperature, int humidity) {

if (soil\_moisture < SOIL\_MOISTURE\_THRESHOLD && temperature > TEMPERATURE\_THRESHOLD && humidity < HUMIDITY\_THRESHOLD) {

GPIO\_write(PUMP\_PIN, 1);

printf("Pump turned ON. Soil Moisture: %d, Temperature: %d°C, Humidity: %d%%\n", soil\_moisture, temperature, humidity);

} else {

GPIO\_write(PUMP\_PIN, 0);

printf("Pump turned OFF. Soil Moisture: %d, Temperature: %d°C, Humidity: %d%%\n", soil\_moisture, temperature, humidity);

}

}

void fetch\_weather\_data(int \*temperature, int \*humidity) {

char weather\_data[100];

if (WiFiModule\_fetch("api.weather.com/data", weather\_data)) {

sscanf(weather\_data, "TEMP:%d,HUM:%d", temperature, humidity);

} else {

printf("Error fetching weather data.\n");

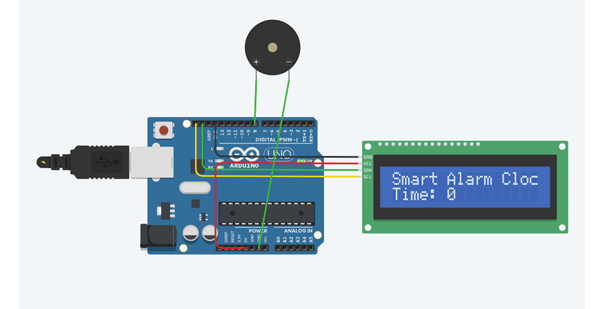
\*temperature = 0;

\*humidity = 0;

}

}

5.Write an Embedded C program to create a Smart Alarm Clock that adjust to your schedule and environment, waking you up intelligently.



#include <stdio.h>

#include <stdlib.h>

#include <stdint.h>

#include "soil\_moisture.h"

#include "dht22.h"

#include "wifi\_module.h"

#define SOIL\_MOISTURE\_THRESHOLD 40

#define TEMPERATURE\_THRESHOLD 35

#define HUMIDITY\_THRESHOLD 60

#define PUMP\_PIN 3

void initialize\_system(void);

void control\_irrigation(int soil\_moisture, int temperature, int humidity);

void fetch\_weather\_data(int \*temperature, int \*humidity);

int main(void) {

int soil\_moisture = 0;

int temperature = 0;

int humidity = 0;

initialize\_system();

printf("IoT-Based Smart Irrigation System Initialized.\n");

while (1) {

soil\_moisture = SoilMoisture\_read();

fetch\_weather\_data(&temperature, &humidity);

control\_irrigation(soil\_moisture, temperature, humidity);

\_delay\_ms(5000);

}

return 0;

}

void initialize\_system(void) {

if (!SoilMoisture\_init()) {

printf("Error initializing Soil Moisture sensor!\n");

exit(1);

}

if (!DHT22\_init()) {

printf("Error initializing DHT22 sensor!\n");

exit(1);

}

if (!WiFiModule\_init()) {

printf("Error initializing WiFi Module!\n");

exit(1);

}

GPIO\_set\_mode(PUMP\_PIN, OUTPUT);

GPIO\_write(PUMP\_PIN, 0);

}

void control\_irrigation(int soil\_moisture, int temperature, int humidity) {

if (soil\_moisture < SOIL\_MOISTURE\_THRESHOLD && temperature > TEMPERATURE\_THRESHOLD && humidity < HUMIDITY\_THRESHOLD) {

GPIO\_write(PUMP\_PIN, 1);

printf("Pump turned ON. Soil Moisture: %d, Temperature: %d°C, Humidity: %d%%\n", soil\_moisture, temperature, humidity);

} else {

GPIO\_write(PUMP\_PIN, 0);

printf("Pump turned OFF. Soil Moisture: %d, Temperature: %d°C, Humidity: %d%%\n", soil\_moisture, temperature, humidity);

}

}

void fetch\_weather\_data(int \*temperature, int \*humidity) {

char weather\_data[100];

if (WiFiModule\_fetch("api.weather.com/data", weather\_data)) {

sscanf(weather\_data, "TEMP:%d,HUM:%d", temperature, humidity);

} else {

printf("Error fetching weather data.\n");

\*temperature = 0;

\*humidity = 0;

}

}