Project Title : Environmental Monitoring

Phase 3: Development Part 1

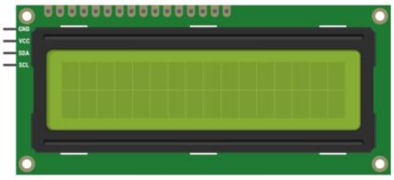
Abstract:

Smart parks represent a transformative approach to managing and enhancing green spaces in urban and rural settings. This abstract provides an overview of the concept, design, and benefits of smart parks. These intelligent ecosystems incorporate various technologies, such as IoT sensors, data analytics, and automation, to improve park management, sustainability, and visitor experiences. Smart parks offer features like optimized resource utilization, real-time monitoring of environmental conditions, and interactive applications for park-goers. By harnessing innovation and connectivity, these parks aim to create more eco-friendly, accessible, and enjoyable outdoor environments. This paper discusses the potential of smart parks to address modern urban challenges and foster a harmonious relationship between nature and technology.

Components:

***12C_LCD**

I2C_LCD is an easy-to-use display module, It can make display easier. Using it can reduce the difficulty of make, so that makers can focus on the core of the work.



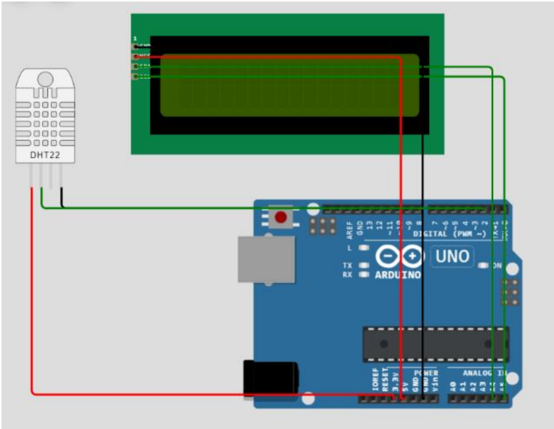
***Arduino UNO**

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.

***DHT22**

The DHT22 is the more expensive version which obviously has better specifications. Its temperature measuring range is from -40 to +125 degrees Celsius with +-0.5 degrees accuracy, while the DHT11 temperature range is from 0 to 50 degrees Celsius with +-2 degrees accuracy.

Circuit Diagram:

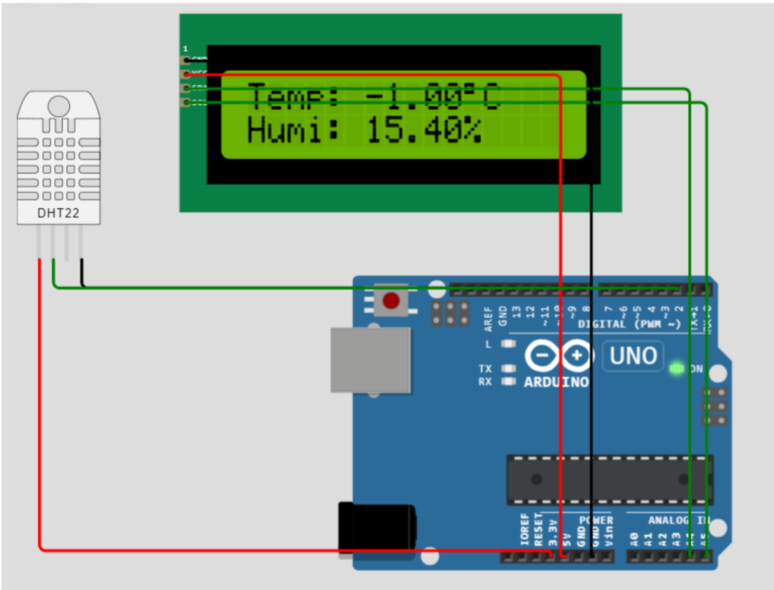


Coding:

```
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11

LiquidCrystal_I2C lcd(0x27, 16, 2);  // I2C address 0x3F, 16 column and 2 rows
DHT dht(DHTPIN, DHTTYPE);

void setup()
{
  dht.begin();     // initialize the sensor
  lcd.init();      // initialize the lcd
  lcd.backlight(); // open the backlight
}

void loop()
{
  delay(2000); // wait a few seconds between measurements

  float humi  = dht.readHumidity();    // read humidity
  float tempC = dht.readTemperature(); // read temperature

  lcd.clear();
  // check if any reads failed
  if (isnan(humi) || isnan(tempC)) {
```

```
    lcd.setCursor(0, 0);
    lcd.print("Failed");
  } else {
    lcd.setCursor(0, 0);  // start to print at the first row
    lcd.print("Temp: ");
    lcd.print(tempC);     // print the temperature
    lcd.print((char)223); // print ° character
    lcd.print("C");

    lcd.setCursor(0, 1);  // start to print at the second row
    lcd.print("Humi: ");
    lcd.print(humi);      // print the humidity
    lcd.print("%");
  }
}
```

Output:



Explaination:

This Arduino sketch is designed to read temperature and humidity data from a DHT11 sensor and display the results on a 2x16 character LCD screen using the LiquidCrystal_I2C library. In the setup, it initializes the sensor and LCD backlight, and in the loop, it repeatedly takes sensor readings with a 2-second delay. It displays the temperature and humidity on the LCD if the readings are valid, while showing "Failed" if the readings are unsuccessful, indicating a problem with the sensor. This code continuously monitors environmental data and provides visual feedback on the LCD screen.

This Arduino sketch integrates a DHT11 sensor for temperature and humidity measurement with a 2x16 character LCD display controlled through the LiquidCrystal_I2C library. The setup function initiates the sensor and backlight, while the loop function repeatedly reads data from the sensor at 2-second intervals. If the sensor provides valid readings, the code displays the temperature and humidity on the LCD screen, along with the degree symbol (°C) for temperature and the percentage symbol (%) for humidity. In the event of sensor failures, it promptly communicates "Failed" on the display. This code essentially creates a simple environmental monitoring system with a user-friendly visual interface.

Future Explaination:

This project could be extended to include features like data logging with an SD card module, wireless connectivity for remote monitoring, real-time clock integration for timestamping data, enhanced display options, an alerting system for threshold notifications, advanced data visualization, support for multiple sensors, portable operation with a battery source, machine learning for predictive analytics, and integration with home automation systems. These enhancements would make the project more versatile, enabling applications ranging from home automation to industrial environmental monitoring.