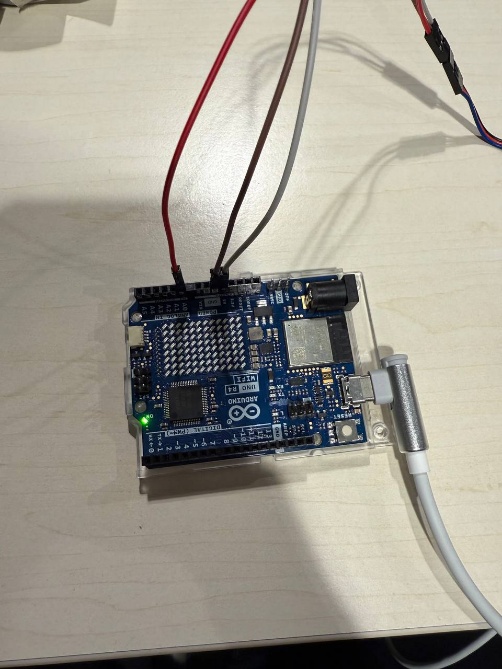
**Summary of the Project Solution**  
*(Portable Water Quality Monitoring Device – Turbidity and Temperature with Data Sent to Blynk.io)*

The Portable Water Quality Monitoring Device is an Arduino‐based system designed to provide an affordable and accessible means to assess water quality in real time. This solution aligns with the United Nations Sustainable Development Goal 6, which emphasizes clean water and sanitation for all. The device focuses on two primary water quality parameters: turbidity and temperature. By measuring the clarity of water with a turbidity sensor and its temperature using a Waterproof DS18B20 Digital Temperature Sensor, the system offers a simple, yet effective method for communities to monitor water safety.

**Hardware and Component Overview**

Initially, the project was attempted using an Arduino MKR WIFI 1010; however, this board was found to be incompatible with the SEN0189 Turbidity Sensor due to its 3.3V operating voltage. To resolve this issue, the project was reconfigured to use the Arduino R4 Uno and R2 Yun, both of which operate at 5V and are thus compatible with the sensor specifications. The hardware components include:

* **Arduino R4 Uno / R2 Yun:** These boards ensure proper voltage levels for the sensors.
* **Waterproof DS18B20 Digital Temperature Sensor:** Provides accurate temperature readings.
* **SEN0189 Turbidity Sensor (and a backup, Jopto Turbidity Sensor Detection):** Measures water clarity.
* **Breadboard, Jumper Wires, and 4.7K Resistors:** These components are used for prototyping and ensuring reliable connections.

****A computer chip connected to wires

AI-generated content may be incorrect.**

**Software and Data Transmission via Blynk.io**

The device’s sensor data is processed by the Arduino board and transmitted in real time to a mobile application via the Blynk.io platform. Blynk serves as an API that facilitates communication between the hardware and the user interface, allowing users to view live readings of water temperature and turbidity. The integration of Blynk.io not only simplifies the display of sensor data but also enables remote monitoring, which is critical in community health applications.

A brief overview of the code is as follows:

***Temperature Sensor Sample Code:***

#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE\_WIRE\_BUS 2

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

void setup() {

Serial.begin(9600);

sensors.begin();

}

void loop() {

sensors.requestTemperatures();

float tempC = sensors.getTempCByIndex(0);

Serial.print("Temperature: ");

Serial.print(tempC);

Serial.println(" °C");

delay(2000);

}

***Turibidity Sensor Sample Code:***

const int turbidityPin = A0; // Analog pin for turbidity sensor

void setup() {

Serial.begin(9600);

}

void loop() {

int sensorValue = analogRead(turbidityPin);

float voltage = sensorValue \* (5.0 / 1023.0);

Serial.print("Turbidity Voltage: ");

Serial.println(voltage);

delay(2000);

}

Additionally, Blynk.io integration is achieved through code similar to the following:

#include <BlynkSimpleEsp8266.h>

char auth[] = "YourBlynkAuthToken";

char ssid[] = "YourWiFiSSID";

char pass[] = "YourWiFiPassword";

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass);

}

void loop() {

Blynk.run();

float temperature = getTemperature(); // Function to read DS18B20 data

int turbidityValue = getTurbidity(); // Function to read SEN0189 data

Blynk.virtualWrite(V1, temperature);

Blynk.virtualWrite(V2, turbidityValue);

delay(2000);

}

**Implementation Workflow**

The overall workflow of the project is straightforward:

1. **User Interaction:** The user immerses the sensors into a water sample.
2. **Data Acquisition:** The DS18B20 and SEN0189 sensors measure the water temperature and turbidity, respectively.
3. **Data Processing:** The Arduino board reads sensor data, calculates necessary parameters (such as converting analog values to voltage), and determines whether the water is safe to drink based on predefined thresholds.
4. **Data Transmission:** Processed data is sent to the Blynk.io server, which then updates the mobile application interface, providing immediate feedback to the user.
5. **User Feedback:** The device communicates a simple status - such as “Safe,” “Moderate,” or “Unsafe” - to help the user make an informed decision about water consumption.

**Conclusion and Future Extensions**

In summary, the Arduino Water Quality System represents an effective, low-cost approach to water quality monitoring. By integrating reliable sensors with Arduino hardware and leveraging the Blynk.io platform for real-time data display, the project addresses a critical need for accessible water testing methods. Future enhancements could include adding additional sensors (such as a TDS sensor for measuring dissolved solids), further refining the data analysis algorithms, and implementing cloud connectivity for remote monitoring and data logging.

This project not only meets the immediate goal of providing a portable water quality assessment tool but also lays the groundwork for future scalability and integration in resource-limited environments. The design is both practical and adaptable, ensuring that communities can access vital information about their water quality, ultimately contributing to improved public health outcomes.

A circuit board with wires

AI-generated content may be incorrect.