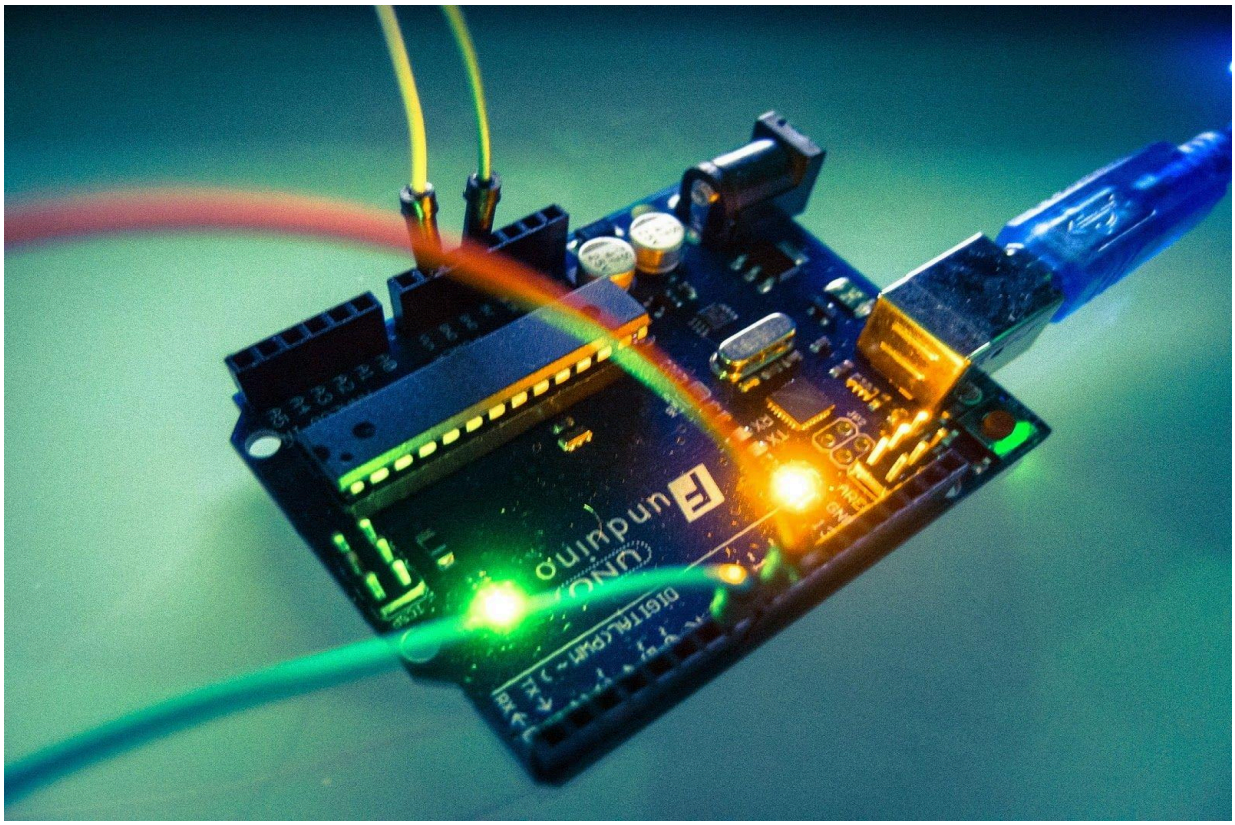
HydroCrafter: IoT-Integrated, Sensor-Driven Automated Hydroponics System



Content of the Project Process Description Report

1. Problem Statement
2. Solution
3. Hardware Components and Functionality
4. Software Components
5. How Each Component Works
6. Building the HydroCrafter System
7. Project Process

1.Problem Statement

Hydroponic systems require regular monitoring of water levels, nutrient solutions, and environmental conditions. This can be labor-intensive and complex, making it difficult for urban residents to maintain their systems.

2.Proposed Solution

HydroCrafter automates the monitoring and regulation of hydroponic systems using sensors for water levels, light, oxygen, and nutrient solutions. This system simplifies maintenance and encourages urban farming by reducing the need for frequent manual checks.

3.Hardware Components and Functionality

Arduino Microcontroller:

The Arduino board acts as the central control unit, receiving data from sensors and sending signals to actuators.

Sensors:

- Water Level Sensor: Measures the water level in the hydroponic system.

- Light Sensor (LDR): Measures light intensity to control the light exposure for plants.
- Oxygen Sensor: Monitors oxygen levels in the water.
- pH Sensor: Measures the acidity or alkalinity of the nutrient solution.
- DHT11 Sensor: Measures temperature and humidity.

Actuators:

- Water Pump: Regulates water flow in the system.
- Nutrient Dosing System: Adds nutrients to the water as needed.
- LED Lights: Provides adjustable light exposure.
- Servo Motor: Adjusts the position of light-blocking or light-enhancing mechanisms based on light sensor data.

4. Software Components

Arduino IDE

The Arduino Integrated Development Environment (IDE) is used to program the Arduino board.

Programming Languages

The Arduino is programmed using C++. Additional languages such as HTML, CSS, and JavaScript are used for web or mobile application development.

6. How Each Component Works

Motor Driver

The motor driver acts as an intermediary between the Arduino and the servo motor. It is necessary because the Arduino alone cannot provide sufficient power to drive the motor. The motor driver receives signals from the Arduino and draws power from an external battery to move the motor. The motor driver supplies enough power to the water pump, as the Arduino alone cannot provide the necessary voltage. The battery powers the motor driver, which in turn powers the Arduino and the pump.

Light Sensor (LDR)

The Light Dependent Resistor (LDR) measures light intensity. In this project, the LDR is connected to the Arduino, which reads its value to determine the light intensity. The LDR operates efficiently at 5V, making it compatible with the Arduino's power supply. When the light level drops, the resistance increases, and vice versa. Depending on the light level, the servo motor adjusts to either allow more light in or block excess light.

Code integration:

```
int light = analogRead(A0);

if (light >= 40) { myServo.write(0);}

else if (light >= 30 && light < 40) {  myServo.write(30);}

else if (light >= 20 && light < 30) { myServo.write(60);}

else if (light >= 10 && light < 20) { myServo.write(70);}

else { myServo.write(90); }
```

Sonar Sensor

The sonar sensor measures water depth by emitting sound waves and calculating the time taken for the echo to return. This distance data helps in controlling the water pump to maintain optimal water levels.

Code integration:

```
#include <NewPing.h>

#define TRIG_PIN 10

#define ECHO_PIN 11

NewPing sonar(TRIG_PIN, ECHO_PIN);

unsigned int distance = sonar.ping_cm();

Serial.print("Distance: ");  Serial.print(distance); Serial.println(" cm");


if (distance >= 12) {

  analogWrite(ENA, 255);

  digitalWrite(IN1, HIGH);

  digitalWrite(IN2, LOW);}

else if (distance <= 11) {

  analogWrite(ENA, 0);

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, LOW);}
```

DHT11 Sensor

The DHT11 sensor is a digital sensor that provides precise temperature and humidity readings. It sends the data to the Arduino, which can then use these values to control other components, such as activating a cooling system if the temperature exceeds a certain threshold or adjusting ventilation based on humidity levels.

Code integration:

```
#include "DHT.h"

#define DHTPIN 12

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);



float h = dht.readHumidity();

float t = dht.readTemperature();



if (isnan(h) || isnan(t)) { Serial.println(F("Failed to read from DHT sensor!")); }

else {

  Serial.print("Humidity: ");  Serial.print(h); Serial.println("%");

  Serial.print("Temperature: ");Serial.print(t); Serial.println("C");

}
```

Water Pump

Controls water flow in the hydroponic system. The motor driver receives signals from the Arduino to turn the pump on or off, ensuring the plants receive adequate water.

Code integration:

```
#define ENA 3

#define IN1 4

#define IN2 5

void setup() {

  pinMode(ENA, OUTPUT);

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT); }

void loop() {

  unsigned int distance = sonar.ping_cm();

  if (distance >= 12) {

    analogWrite(ENA, 255);

    digitalWrite(IN1, HIGH);

    digitalWrite(IN2, LOW);}

 else if (distance <= 11) {

    analogWrite(ENA, 0);

    digitalWrite(IN1, LOW);

    digitalWrite(IN2, LOW); } }
```

Servo Motor

The servo motor is used to adjust the position of a cover or blinds over a window or light source. Depending on the light intensity read by the LDR, the Arduino sends a signal to the servo motor to either open or close the cover. This helps in maintaining optimal lighting conditions in the room. Adjusts the position of light-blocking or light-enhancing mechanisms.The servo motor receives signals from the Arduino to rotate to specific angles, controlling light exposure for the plants.

Code integration:

```
#include <Servo.h>

Servo myServo;


void setup() {

  myServo.attach(9); }


void loop() {

  int light = analogRead(A0);

  if (light >= 40) {  myServo.write(0); }

  else if (light >= 30 && light < 40) { myServo.write(30);}

  else if (light >= 20 && light < 30) { myServo.write(60);}

  else if (light >= 10 && light < 20) { myServo.write(70);}

  else { myServo.write(90); } }
```

Building the HydroCrafter System

1. Connect the Sensors:
    - Connect the water level sensor, LDR, oxygen sensor, pH sensor, and DHT11 sensor to the Arduino board.

2. Set Up the Actuators:
    - Connect the water pump, nutrient dosing system, and adjustable LED lights to the Arduino through the motor driver.

3. Program the Arduino:
    - Write and upload code to the Arduino using the Arduino IDE to control the sensors and actuators.

4. Test and Calibrate:
    - Test the system components and calibrate the sensors for accurate readings.

Project Process

1. The first step in building the HydroCrafter system was selecting the appropriate hardware components. We needed sensors to monitor water levels, light, oxygen, and nutrient conditions, as well as actuators to control the water pump, nutrient dosing system, and LED lights.

2. We connected the sensors and actuators to the Arduino board.
3. We wrote the Arduino code to read data from the sensors and control the actuators based on the sensor readings. The code included logic for: Reading and interpreting sensor data, Controlling the water pump and servo motor. And Displaying data on the serial monitor for debugging and monitoring.
4. We tested each component individually to ensure they were working correctly. Calibration was performed to ensure accurate readings from the sensors.
5. Finally, we assembled all components into a cohesive system. We installed the sensors in the hydroponic setup, connected the actuators, and integrated the Arduino board.

This step-by-step process provides a comprehensive and beginner-friendly explanation of how we made the HydroCrafter from scratch, detailing each stage of the project and the role of each component.

Authors:
1. Sanjoy Das [2020331023]
2. Nayem Ahmed [2020331048]
3. Manisa Rahman [2020331058]
4. Tajul Islam Tarek [2020331067]
5. Mustakim Billah Nafees [2020331097]