**Техническое задание (ТЗ) на разработку "Умной системы вертикального озеленения" (Версия 2.0)**

**1. Общее описание проекта**  
Разработать и собрать прототип автоматизированной системы для вертикального озеленения на базе микроконтроллера **Arduino ESP32**. Система должна обеспечивать **автоматический полив с точной дозировкой** и **удаленный мониторинг** ключевых параметров среды через **IoT-платформу (MQTT)** с отображением данных в **веб-интерфейсе (GyverHub)**. В перспективе система должна быть масштабируема для управления несколькими стендами в Биоквантуме.

**2. Цели и задачи**

* **Цель:** Повышение эффективности и автономности вертикального озеленения за счет автоматизации и удаленного контроля с точным дозированием воды.
* **Задачи:**
  1. Собрать аппаратную часть системы (сенсоры, исполнительные механизмы, управляющий контроллер) на основе печатной платы.
  2. Разработать и прошить firmware для Arduino ESP32.
  3. Реализовать передачу данных по протоколу MQTT.
  4. Настроить визуализацию и управление в веб-интерфейсе GyverHub.
  5. Протестировать систему на действующей стенке в Биоквантуме.
  6. Подготовить документацию для тиражирования системы.

**3. Этапы выполнения работ**

**ЭТАП 1: ПРОЕКТИРОВАНИЕ И ПОДБОР ОБОРУДОВАНИЯ**

**Задача 1.1: Формирование требований к датчикам и исполнительным устройствам.**

* Определить перечень контролируемых параметров:
  + Влажность почвы (основной параметр для полива).
  + Температура и влажность воздуха.
  + Уровень освещенности (опционально, для мониторинга).
  + Уровень воды в баке для полива.
  + **Расход воды (дозировка) с помощью датчика потока.**
* Определить перечень исполнительных устройств:
  + **Перистальтический насос для точной и дозированной подачи воды.**
  + Система трубок и капельниц для распределения воды.

**Задача 1.2: Подбор конкретных компонентов и составление спецификации.**

* **Микроконтроллер:** Arduino ESP32 (например, ESP32 DevKit C) – для Wi-Fi подключения и множества пинов.
* **Датчики:**
  + **Влажность почвы:** Аналоговые датчики типа FC-28 или **емкостные датчики** (более надежные, например, Capacitive Soil Moisture Sensor v1.2).
  + **Температура и влажность воздуха:** Цифровой датчик DHT22 (AM2302) или более точный SHT31.
  + **Освещенность:** Фоторезистор или цифровой датчик BH1750.
  + **Уровень воды в баке:** Ультразраковой датчик HC-SR04 или датчик уровня воды (поплавковый).
  + **Датчик потока воды:** YF-S201 или аналогичный.
* **Исполнительные устройства:**
  + **Помпа:** **Перистальтический насос 12V DC** с подходящей производительностью (например, 100 мл/мин).
  + **Драйвер мотора:** Драйвер для управления насосом с помощью ШИМ (PWM) для контроля скорости, например, **TB6612FNG** или **DRV8833**.
* **Сопутствующие компоненты:**
  + **Проектирование печатной платы:** Компоненты для пайки на PCB (разработанной в EasyEDA).
  + **Блок питания:** Импульсный источник питания 12V с достаточной мощностью (например, 60W) для насоса и **модули преобразователей напряжения** (Step-Down) на 5V и 3.3V для контроллера и датчиков.
  + Водопроводные трубки (совместимые с перистальтическим насосом), тройники, капельницы.
  + Водонепроницаемая емкость для бака.
  + Корпус для электроники.

**Результат этапа:** Утвержденный список компонентов (спецификация) для заказа.

**ЭТАП 2: ПРОТОТИПИРОВАНИЕ И СБОРКА АППАРАТНОЙ ЧАСТИ**

**Задача 2.1: Сборка и отладка сенсорного модуля на макетной плате.**

* Подключить датчики (влажности почвы, DHT22, BH1750, **датчик потока**) к ESP32.
* Написать тестовые скетчи для чтения данных с каждого датчика.
* Провести калибровку датчиков (особенно датчика влажности почвы для сухой и мокрой среды, **датчика потока - импульсов на литр**).

**Задача 2.2: Сборка и тестирование модуля полива.**

* Подключить **перистальтический насос через драйвер моторов** к ESP32.
* Написать тестовый скетч для управления насосом с помощью ШИМ (изменение скорости) и **контроля объема воды с помощью датчика потока**.
* Проверить герметичность и работоспособность системы подачи воды (трубки, капельницы) на тестовом стенде.

**Задача 2.3: Проектирование и изготовление печатной платы.**

* Разработать принципиальную схему и развести печатную плату в **EasyEDA**, объединяющую все компоненты (ESP32, драйвер мотора, разъемы для датчиков, клеммы для питания).
* Заказать изготовление PCB и произвести **пайку всех компонентов на плату**.
* Протестировать работоспособность собранной платы.

**Задача 2.4: Компоновка и монтаж системы.**

* Установить собранную плату в корпус.
* Разместить датчики в горшках на вертикальной стенке.
* Закрепить трубки и капельницы, установив **перистальтический насос и датчик потока в разрыв трубки**.
* Установить бак для воды в удобное место.
* Подключить **блок питания и модули преобразователей напряжения**.

**Результат этапа:** Физически собранный и работоспособный прототип системы на стенде, собранный на собственной печатной плате.

**ЭТАП 3: РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (Firmware)**

**Задача 3.1: Базовая логика работы.**

* Написать скетч, который в цикле опрашивает все датчики.
* Реализовать алгоритм автоматического полива с точной дозировкой:
  + Если влажность почвы ниже заданного порога X -> включить насос на заданной скорости и **подавать заданный объем воды (контролируя по датчику потока)**.
  + Добавить "защиту от дурака": таймаут на полив (например, не более 500 мл за один цикл) и проверку уровня воды в баке.

**Задача 3.2: Интеграция с IoT (MQTT).**

* Настроить подключение ESP32 к Wi-Fi сети Биоквантума.
* Реализовать подключение к MQTT-брокеру (например, Mosquitto, работающий на будущем сервере с Home Assistant или на отдельной VPS).
* Настроить публикацию данных (telemetry) с датчиков в MQTT-топики (например, biokvantum/vertical\_garden/temperature, biokvantum/vertical\_garden/soil\_moisture, biokvantum/vertical\_garden/water\_flow).
* Реализовать подписку на MQTT-топики для управления (например, biokvantum/vertical\_garden/pump/set для принудительного включения помпы, biokvantum/vertical\_garden/pump/speed для установки скорости).

**Задача 3.3: Реализация веб-интерфейса на GyverHub.**

* Используя библиотеку **GyverHub**, создать интерфейс управления для системы.
* В интерфейсе должны отображаться:
  + Текущие показания всех датчиков.
  + Графики изменения показаний во времени.
  + Кнопка для ручного включения/выключения помпы и **поле для задания объема воды**.
  + Поля для настройки порогов автоматического полива (X и Y), **скорости насоса, объема воды для полива**.
  + Индикатор статуса подключения (Online/Offline).

**Результат этапа:** Прошивка для ESP32, обеспечивающая автономную работу, передачу данных по MQTT и веб-интерфейс для управления.

**ЭТАП 4: ИНТЕГРАЦИЯ, ТЕСТИРОВАНИЕ И ДОКУМЕНТИРОВАНИЕ**

**Задача 4.1: Комплексные испытания.**

* Разместить систему на целевой вертикальной стенке.
* Провести длительный тест (не менее 7 дней) в реальных условиях.
* Проверить корректность работы автоматического полива, **точность дозировки**.
* Проверить стабильность подключения к Wi-Fi и MQTT-брокеру.
* Протестировать веб-интерфейс с разных устройств (ПК, смартфон).

**Задача 4.2: Создание серверной инфраструктуры (перспектива).**

* **Вариант А (GyverHub):** Настроить постоянный хост для GyverHub (например, на одноплатном компьютере Raspberry Pi).
* **Вариант Б (Home Assistant):** Установить и настроить Home Assistant на сервере.
  + Настроить MQTT-брокер как интеграцию.
  + Добавить устройства в Home Assistant через MQTT Discovery.
  + Создать Lovelace-дашборд с показаниями датчиков и элементами управления.

**Задача 4.3: Документирование.**

* Составить инструкцию по сборке системы (схемы подключения, список деталей, **схема в EasyEDA**).
* Составить инструкцию по настройке ПО (прошивка, настройка MQTT, настройка GyverHub/Home Assistant).
* Написать руководство пользователя.
* Выложить весь код и **проект платы** в репозиторий (например, GitHub) с подробным README.

**Результат этапа:** Полностью функционирующая, протестированная система и комплект документации для ее повторения.

**4. Критерии приемки готового проекта:**

1. Система самостоятельно, без вмешательства человека, поддерживает влажность почвы в заданном диапазоне, **точно дозируя воду**.
2. Данные со всех датчиков (включая **расход воды**) стабильно передаются по MQTT и отображаются в веб-интерфейсе GyverHub.
3. Через веб-интерфейс можно в реальном времени управлять насосом (включать/выключать, **задавать скорость и объем**) и изменять настройки системы.
4. Система стабильно работает не менее 72 часов без сбоев (потеря связи, зависание).
5. Предоставлена вся необходимая техническая документация, исходный код и **проект печатной платы**.