High-Level Design (HLD)

NanoRiego

1. Purpose

This High-Level Architecture (HLA) document provides a top-down structural and functional overview of the NanoRiego system. It describes the primary architectural components, their roles, interfaces, and the overall interaction model that supports the system's functionality as defined in the System Requirements Specification (SysRS).

2. Scope

The architecture spans hardware, embedded firmware, and mobile application layers. It highlights component boundaries, inter-process communication, and data flow between subsystems. The HLA serves as a blueprint for detailed design and integration efforts.

3. System Architecture Overview

NanoRiego is structured as a three-tier system comprising the following subsystems:

3.1 Hardware Control Layer (NanoRiegoPCB)

- Central PCB hosting all hardware modules
- Interfaces: digital outputs to relays, UART for Bluetooth, I2C for RTC
- Microcontroller: Arduino Nano
- Power inputs: USB-C and/or auxiliary DC source

3.2 Embedded Logic Layer (NanoRiegoPIO)

- Executed on Arduino Nano
- Responsibilities:
    - Interprets Bluetooth serial commands
    - Manages RTC-driven scheduling logic
    - Controls valves and pump with timing and safety constraints
    - Handles persistent configuration storage

3.3 User Interface Layer (NanoRiegoAPP)

- Android-based mobile application
- Communicates with controller via HC-05 over Bluetooth
- Responsibilities:
    - Provides control and monitoring interface
    - Syncs time and date
    - Manages irrigation program settings and manual overrides

3. Component Descriptions

3.1 Arduino Nano (Controller Core)

- Hosts the main firmware
- Interfaces with Bluetooth (via SoftwareSerial)
- Polls RTC over I2C
- Controls relays via digital GPIO

3.2 Relay Module

- 4-channel module
- Channels 1–3: valves 1–3
- Channel 4: pump

3.3 Bluetooth Module (HC-05)

- Serial Bluetooth link to Android device
- Operates at 3.3V logic
- Auto-reconnect enabled after initial pairing

3.4 RTC Module (DS1307)

- Real-time timekeeping
- Powered via 5V with coin-cell backup

3.5 EEPROM Storage

- Part of the Arduino microcontroller
- Stores configuration (schedules, system flags)

3.6 Android App

- Two primary views: Dashboard and Settings
- Widgets include toggle buttons, sliders, date/time inputs, and Bluetooth manager
- Sends/receives ASCII-encoded commands via BT

4. Component Interactions

Component A	Component B	Protocol / Interface	Data Exchanged
NanoRiegoAPP	HC-05	Bluetooth (Serial)	Commands, schedule, time sync
HC-05	Arduino Nano	UART	Serial command transmission
Arduino Nano	Relay Module	GPIO	Valve and pump actuation signals
Arduino Nano	DS1307 RTC	I2C	Time synchronization and polling

5. Deployment Model

Component	Location	Power Source
Arduino Controller	Outdoor/Greenhouse Box	USB-C / 12V Adapter
Mobile App	Android Device	Internal Battery
Relay Module	Inside Controller Enclosure	Supplied by PCB

6. Architectural Requirement Allocation

Requirement ID	Description	Component Responsible
SYS-FR1	Control up to three solenoid valves	Arduino Nano, Relay Module
SYS-FR2	Operate pump with active irrigation zone	Arduino Firmware, Relay Logic
SYS-FR3	Accept scheduling instructions from app	Android App, Bluetooth Module
SYS-FR4	Execute scheduled events based on RTC	Arduino Firmware, RTC Module
SYS-FR5	Allow manual zone activation via app	Android App, Firmware Logic
SYS-FR6	Persist configuration and schedule data	Arduino EEPROM
SYS-FR7	Time/date sync and pairing from app	Android App, Bluetooth Module
SYS-FR8	Return current system state	Arduino Firmware

SYS-IR4	Use structured ASCII command messaging	Android App, Firmware Parser
SYS-SR1	Implement control logic state machine	Arduino Firmware
SYS-RS1	Prevent multiple simultaneous valve activations	Firmware Scheduling Logic
SYS-RS4	Indicate system and connection status via UI	Android App

7. Design Principles

- **Simplicity**: Minimal component interdependence for easy testing
- **Modularity**: Components can be replaced or upgraded with minimal disruption
- **Robustness**: EEPROM persistence and fallback behaviors
- **Responsiveness**: Fast command execution and visual feedback

8. Future Considerations

- Cloud integration for remote access
- Sensor modules for soil moisture or temperature
- Extended scheduling features with dynamic durations

This HLD will be refined during subsequent detailed design phases and updated accordingly as the system architecture evolves.