

# Readme - Arduino Uno Irrigation System

This Arduino Uno irrigation system is designed to control watering based on temperature and light intensity. The system utilizes various components such as temperature and light sensors, a servo motor, and LEDs to provide an automated irrigation solution.

## Components Used

- Arduino Uno Board
- LiquidCrystal\_I2C library
- Wire library
- Servo library

## Functionality

The irrigation system operates as follows:

1. It reads the light intensity from the LDR (Light Dependent Resistor) sensor connected to analog pin A0.
2. It measures the temperature using the temperature sensor connected to analog pin A1.
3. The system checks the temperature and determines whether the red LED should be turned on based on predefined thresholds. If the temperature exceeds 27°C, the green LED is turned off and the red LED is turned on.
4. If the light intensity is low (night time) and the red LED is on, the system starts the irrigation process.
5. The LCD display (I2C) is used to provide status information.
6. When the irrigation system is activated, the yellow LED is turned on and the servo motor starts spinning gradually from 0 to 180 degrees and then back to 0 degrees. This simulates the opening and closing of a valve for irrigation.
7. After a predefined duration (10 seconds), the servo motor stops, and the green LED is turned on to indicate that the system is off.
8. The LCD display shows "OFF" to indicate the system status.

## Usage

1. Upload the provided code to the Arduino Uno board.
2. Make sure all the components are connected correctly as per the wiring instructions.
3. Power on the Arduino Uno.
4. The system will continuously monitor the temperature and light intensity.
5. The red LED will turn on if the temperature exceeds 27°C.
6. If it is nighttime and the red LED is on, the irrigation process will start.
7. The servo motor will rotate gradually, simulating the opening and closing of a valve.
8. After 10 seconds, the servo motor will stop, and the green LED will turn on, indicating that the system is off.

## Notes

- Ensure that the necessary libraries (Wire, LiquidCrystal\_I2C, and Servo) are installed in the Arduino IDE.
- Adjust the temperature thresholds and sensor readings in the code as per your specific requirements.
- Take precautions while handling electronic components and ensure proper connections to prevent damage to the Arduino board or other components.