Arduino – water dispenser

Description

The pump will dispense water as long as the button is pressed.

Standby status: RED LED **ON**, GREEN LED **OFF**

Active pump status: RED LED **OFF**, GREEN LED **ON**

# Requirements

| Arduino board |  |
| --- | --- |
| R385 Pump |  |
| 5V Relay |  |
| 9V Battery |  |
| RGB Module |  |
| Other | -cables  -hose  -button  -breadboard |

# Code:

#define RED\_PIN 5

#define GREEN\_PIN 6

#define BLUE\_PIN 7

#define BUTTON\_PIN 2

#define RELAY\_PIN A0

int buttonStatus = HIGH; // Store the current button status

int pinValue; // Store the read value from the button

void setup() {

pinMode(RELAY\_PIN, OUTPUT);

pinMode(RED\_PIN, OUTPUT);

pinMode(GREEN\_PIN, OUTPUT);

pinMode(BLUE\_PIN, OUTPUT);

pinMode(BUTTON\_PIN, INPUT\_PULLUP); // internal pull-up resistor

Serial.begin(9600);

// Safely have everything turned off

digitalWrite(RELAY\_PIN, LOW);

digitalWrite(RED\_PIN, HIGH);

digitalWrite(GREEN\_PIN, LOW);

digitalWrite(BLUE\_PIN, LOW);

}

void loop() {

pinValue = digitalRead(BUTTON\_PIN);

delay(10); // Debounce delay

if (buttonStatus != pinValue) { // Button state has changed

buttonStatus = pinValue;

if (buttonStatus == LOW) { // Button was pressed

Serial.println("Button pressed!");

digitalWrite(RED\_PIN, LOW);

digitalWrite(GREEN\_PIN, HIGH);

digitalWrite(RELAY\_PIN, HIGH);

} else { // Button was released

Serial.println("Button released!");

digitalWrite(GREEN\_PIN, LOW);

digitalWrite(RED\_PIN, HIGH);

digitalWrite(RELAY\_PIN, LOW);

}

}

}