**Smart Rest Room On Automatic Room Lights using Arduino and PIR Sensor**

# Automatic Room Lights using Arduino and PIR Sensor

In this project, we will see the Automatic Room Lights using Arduino and PIR Sensor, where the lights in the room will automatically turn ON and OFF by detecting the presence of a human.

Such Automatic Room Lights can be implemented in your garages, staircases, bathrooms, etc. where we do not need continuous light but only when we are present.

Also, with the help of an automatic room light control system, you need not worry about electricity as the lights get automatically off when there is no person.

So, in this DIY project, we have implemented Automatic Room Lights using Arduino and PIR Sensor.

### Components Required for Automatic Room Lights using Arduino

* Arduino UNO
* PIR Sensor
* 5V Relay Module (Relay Board)
* LED
* 100Ω Resistor (1/4 Watt)
* Connecting Wires
* Breadboard
* Power Supply

**Steps to Create the IoT Water Overflow Control Project:**

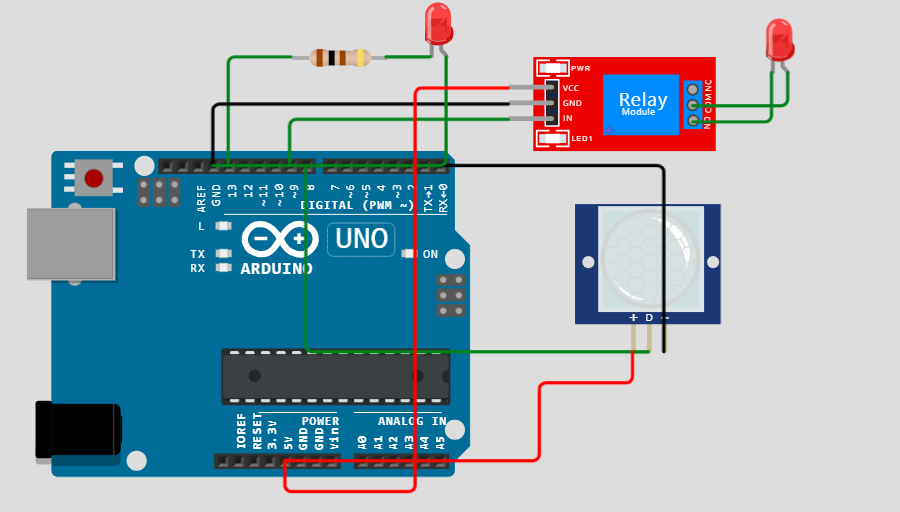
**Setup Hardware**

a. Connect the Relay module sensor to the microcontroller. Ensure you follow the sensor's datasheet and the microcontroller's pinout.

b. Connect the actuator to the microcontroller. The type of actuator will depend on your application

c. Connect any required power supplies to the microcontroller and actuator.

Refer below image

****

**Python program Here whe will use in raspberry pi**

You'll need a Python script to read data from the water level sensor, make decisions based on the water level, and control the actuator. Here's a simplified example using Raspberry Pi and RPi.GPIO:

Python

import time

import RPi.GPIO as GPIO

in1 = 9

sensor = 8

led = 13

t = 0

GPIO.setmode(GPIO.BCM)

GPIO.setup(in1, GPIO.OUT)

GPIO.setup(sensor, GPIO.IN)

GPIO.setup(led, GPIO.OUT)

GPIO.output(in1, GPIO.HIGH)

GPIO.output(led, GPIO.LOW)

while time.time() < 13:

GPIO.output(led, GPIO.HIGH)

time.sleep(0.05)

GPIO.output(led, GPIO.LOW)

time.sleep(0.05)

GPIO.output(led, GPIO.LOW)

while True:

GPIO.output(in1, GPIO.HIGH)

GPIO.output(led, GPIO.LOW)

if GPIO.input(sensor) == GPIO.HIGH:

t = int(time.time() \* 1000)

while time.time() \* 1000 < (t + 5000):

GPIO.output(in1, GPIO.LOW)

GPIO.output(led, GPIO.HIGH)

if time.time() \* 1000 > (t + 2300) and GPIO.input(sensor) == GPIO.HIGH:

t = int(time.time() \* 1000)

**4. Connect to the Internet (IoT):**

If you want to make it an IoT project, you'll need to add code for internet connectivity. This could be through Wi-Fi, Ethernet, or a cellular module, depending on your chosen microcontroller.

**5.Test and Debug:**

Test your setup with a controlled water source and verify that the system behaves as expected. Debug any issues that arise.

NOTE: Please note that this is a simplified example, and the actual implementation may vary based on your specific hardware and requirements. Additionally, consider safety measures, such as fail-safes and emergency shutdown procedures, when working with water control systems.