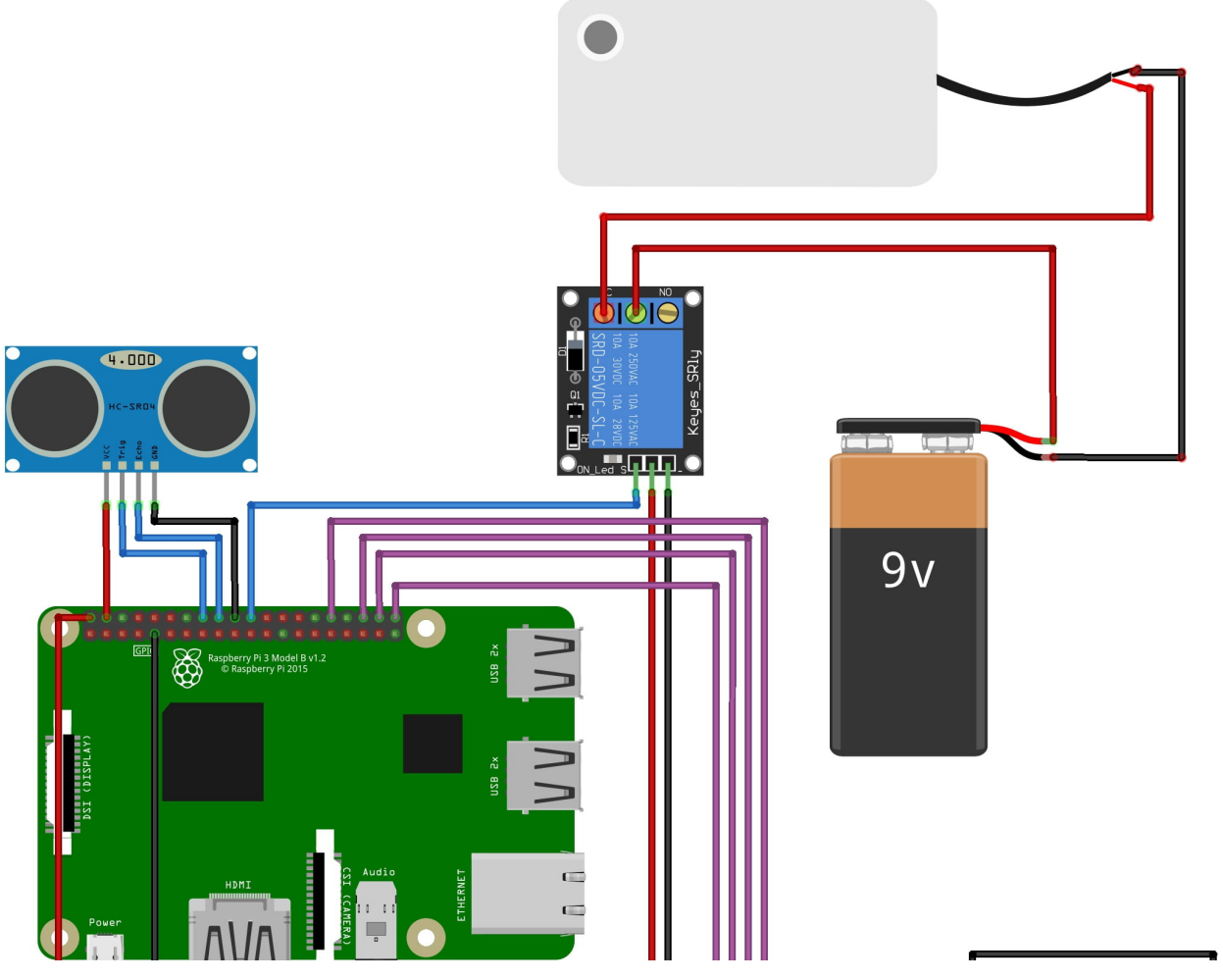
Phase 3: project report

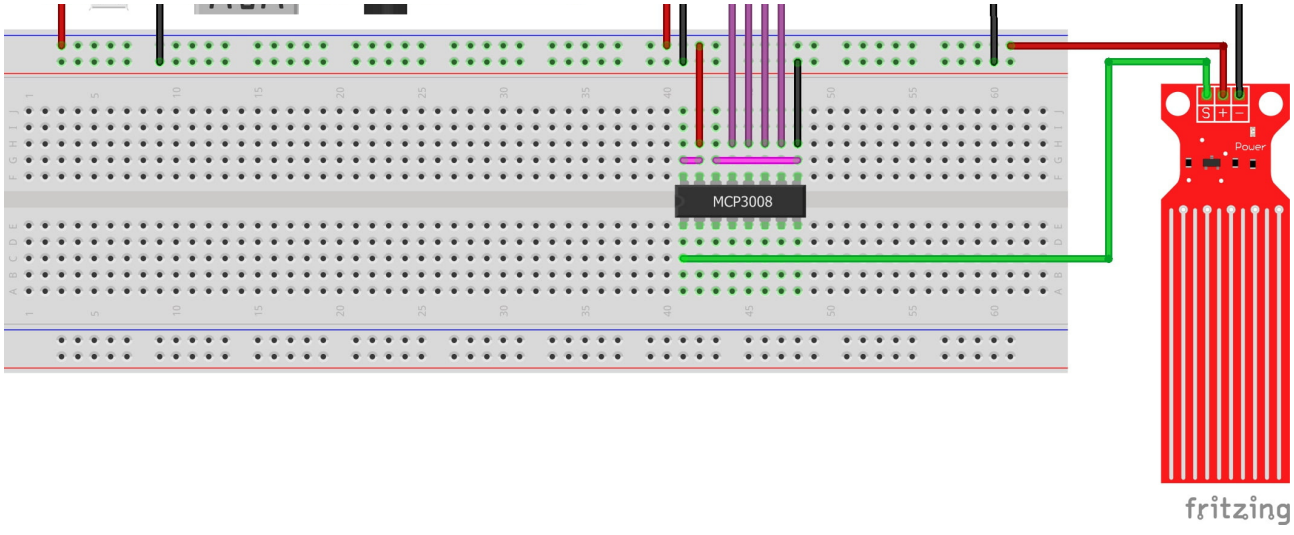
Flood monitoring system and early detection

Introduction:

Welcome to our innovative project focused on flood monitoring in dams, where we leverage the power of Raspberry Pi technology and advanced water level sensors to collect and analyze crucial data. As climate change intensifies the frequency and severity of extreme weather events, the need for reliable flood monitoring systems has never been greater. Our project seeks to address this pressing issue by combining the versatility of Raspberry Pi with the precision of water level sensors to provide real-time data on water levels in dams. This data is not only essential for early warning systems but also plays a pivotal role in making informed decisions for dam management and disaster mitigation. In the following sections, we will delve into the technical details and the significance of our approach, highlighting the potential for enhancing dam safety and environmental resilience.

Circuit diagram:





Circuit explanation:

The circuit you've described involves the use of a Raspberry Pi and a water level sensor. The primary purpose of this setup is to monitor the water level in a particular container or location. Here's a more detailed breakdown:

- Raspberry Pi: This is a credit-card-sized computer that can be used for a variety of purposes, including monitoring and controlling external devices.
- Water Level Sensor: This is a sensor designed to detect the level of water in a container or area. It typically works by measuring the conductivity or resistance of the water to determine its level.
- Simulation: This refers to the process of emulating or testing the system without physical components. In this case, you might be testing the Raspberry Pi's interaction with the water level sensor in a virtual environment before implementing it in the real world.
- Detecting Water Level Range: The primary function of the system is to determine the water level within a specified range. This range might represent the desired water level in a tank, a reservoir, or any other application where water level control is important.
- Alert Notification: When the system detects that the water level has gone beyond the defined range, it triggers an alert notification. This notification can be sent through various means, such as emails, text messages, or even visual/audible alarms.

Dataset:

```
timestamp,water_level
2023-10-16 12:00:00,40
2023-10-16 12:00:01,42
2023-10-16 12:00:02,45
2023-10-16 12:00:03,48
...
```

Code:

```
import RPi.GPIO as GPIO
import time

# Set up GPIO pins
GPIO.setmode(GPIO.BCM)
water_level_pin = 17
GPIO.setup(water_level_pin, GPIO.IN)

# Define the water level range
minimum_level = 10
maximum_level = 90

# Main monitoring loop
try:
    while True:
        # Read water level sensor data
        water_level =
GPIO.input(water_level_pin)

        # Check if water level is outside the
defined range
        if water_level < minimum_level or
water_level > maximum_level:
            # Trigger an alert notification
(e.g., send an email)
            # You would need to implement
this part based on your chosen notification
method.

        time.sleep(1)  # Wait for 1 second
before the next reading

except KeyboardInterrupt:
```

```
    GPIO.cleanup()  # Clean up GPIO on
program exit
```

Code explanation:

```
import RPi.GPIO as GPIO
import time
```

- These lines import the necessary Python libraries:
  - RPi.GPIO: This library allows you to control the GPIO (General Purpose Input/Output) pins on a Raspberry
  - Pi.time: It provides time-related functions.

```
# Set up GPIO pins
GPIO.setmode(GPIO.BCM)
water_level_pin = 17
GPIO.setup(water_level_pin, GPIO.IN)
```

- These lines configure the GPIO pins on the Raspberry Pi. The setmode function sets the numbering mode to Broadcom SOC channel numbers. It's followed by defining a variable water_level_pin which is set to GPIO pin 17. Then, GPIO.setup configures the pin as an input pin.

```
# Define the water level range
minimum_level = 10
maximum_level = 90
```

- Here, you define the desired water level range. minimum_level and maximum_level are set to 10 and 90, respectively. These values can be adjusted to match the specific range you want to monitor.

```
# Main monitoring loop
try:
    while True:
        # Read water level sensor data
        water_level =
GPIO.input(water_level_pin)
```

- This part initiates the main monitoring loop using a while loop that runs indefinitely. It continuously captures data:
  - water_level: Reads the digital input from the water level sensor connected to the GPIO pin.

```
# Check if water level is outside the defined
range
        if water_level < minimum_level or
water_level > maximum_level:
            # Trigger an alert notification
(e.g., send an email)
            # You would need to implement this
part based on your chosen notification method.
```

- Here, it checks if the water level reading is outside the defined range. If it is, you can add code to trigger an alert notification. However, in the provided code, the alert notification part is left as a comment and should be implemented based on your chosen notification method.

```
time.sleep(1)  # Wait for 1 second before the
next reading
```

- It adds a 1-second delay before taking the next water level reading to avoid overwhelming the system with data.

```
except KeyboardInterrupt:
    GPIO.cleanup()  # Clean up GPIO on program
exit
```

- Finally, this block ensures that if the program is interrupted by a KeyboardInterrupt, like when you manually stop the program, it will clean up the GPIO configuration using GPIO.cleanup(). This is essential for the proper handling of GPIO pins and prevents any potential issues.

Conclusion:

This project focuses on the development of a water level monitoring system using a Raspberry Pi and a water level sensor. The system comprises a Python script and a dataset collected during its operation. The Python script interacts with the Raspberry Pi's GPIO pins to read data from the water level sensor, defining a desired range for water level monitoring. The script continuously checks the water level readings and triggers alert notifications if the water level falls outside the specified range. The dataset generated is timestamped, making it suitable for time-series analysis and monitoring trends or anomalies in water levels. This system provides a foundation for effective water level management and the prevention of issues related to water level fluctuations.