Flood monitoring typically involves collecting and analyzing data from various sources, such as sensors, weather forecasts, and satellite imagery. Here's a simplified Python code outline for a basic flood monitoring system:

```
import requests
import json

def get_weather_data():
    # Use a weather API to fetch current weather conditions for the flood-prone area
    api_key = "your_weather_api_key"
    location = "your_location"
    url = f"https://api.weather.com/data/current/{location}?apikey={api_key}"

    response = requests.get(url)
    data = json.loads(response.text)
    return data

def check_water_level():
    # Use sensors or public data sources to check the water level in the area
    # You can simulate this with random data for testing purposes
    water_level = 10  # Replace with real water level data

    return water_level

def main():
    weather_data = get_weather_data()
    water_level = check_water_level()

    if weather_data['precipitation'] > 0.1 or water_level > 15:
        # Take action when flood conditions are detected
        # Send alerts, activate pumps, or trigger other flood mitigation measures
        print("Flood conditions detected. Take action!")

if __name__ == "__main__":
    main()
```

In this code, we have functions to get weather data from a weather API and check the water level using sensors or public data sources. The main function is where you can define the logic to take action when flood conditions are detected, such as sending alerts or activating flood mitigation systems. You should replace the placeholders with real data sources and actions as needed for your specific flood monitoring system.