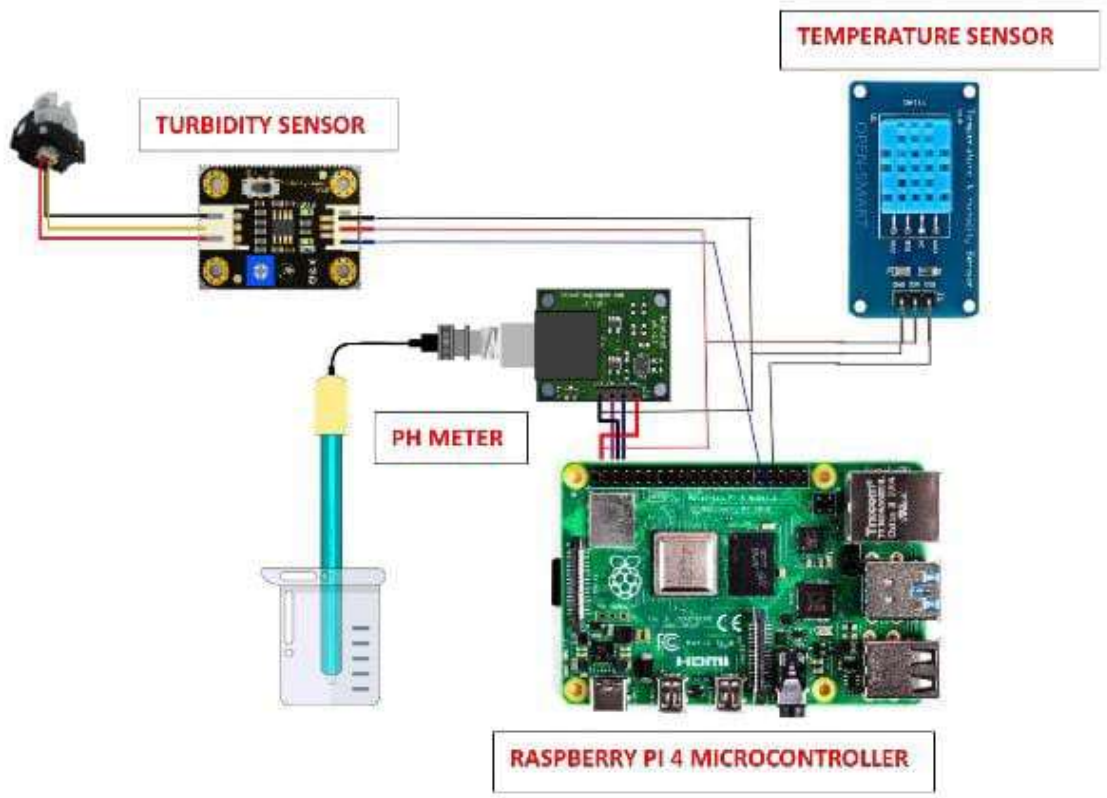
REAL-TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

CIRCUIT DIAGRAM



PROGRAMMING:

```
import ibmiotf.application
import ibmiotf.device
import time import
random
import sys
from twilio.rest import Client
import keys
Client = Client(keys.account_sid, keys.auth_token)
```

0rganization ID
pnco2k
Device Type
watermonitoringsystem
Device ID
watermonitoringsystemid
Authentication Method
use-token-auth
Authentication Token
y1KKoQTKx?i@jA&q9R

```
pH = random.randint(1, 14)
turbidity = random.randint(1, 1000)
temperature = random.randint(0, 100)

def myCommandCallback(cmd):
    print("Command Received: %s" % cmd.data['command'])
    print(cmd)
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod,
                     "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()

while True:
    pH = random.randint(1, 14)
    turbidity = random.randint(1, 1000)
    temperature = random.randint(0, 100)

    data = {'pH': pH, 'turbid': turbidity, 'temp': temperature}

    def myOnPublishCallback():
        print("Published pH= %s" % pH, "Turbidity:%s" % turbidity, "Temperature:%s" %
temperature)

   success = deviceCli.publishEvent("demo", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not Connected to ibmiot")
    time.sleep(1)
    deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()
```