Flood Monitoring and Early Warning

Phase3: Development part 1

ESP32

```
import machine
import time
# Define the LED pin
led_pin = machine.Pin(2, machine.Pin.OUT)
while True:
    led_pin.on()
    time.sleep(1)
    led_pin.off()
    time.sleep(1)
```

Temperature and humidity sensor

```
# main.py
import machine
import dht
import time
# Connect the DHT22 sensor to pin 4 (you can use a different pin)
dht_pin = machine.Pin(4)
dht_sensor = dht.DHT22(dht_pin)
```

```
while True:
    try:
       # Read temperature and humidity from the sensor
        dht_sensor.measure()
        temperature = dht_sensor.temperature()
        humidity = dht_sensor.humidity()
        # Print the values
        print("Temperature: {:.2f}°C".format(temperature))
        print("Humidity: {:.2f}%".format(humidity))
       except Exception as e:
        print("Error reading from the sensor:", e)
       # Wait for a moment before reading again
      time.sleep(2)
```

Ultrasonic sensor

```
import RPi.GPIO as GPIO
import time
# Set GPIO mode and define GPIO pins
GPIO.setmode(GPIO.BOARD)
trigger_pin = 11
echo_pin = 13
# Set up GPIO pins
GPIO.setup(trigger_pin, GPIO.OUT)
GPIO.setup(echo_pin, GPIO.IN)
```

```
  def measure_distance():
   # Trigger the ultrasonic sensor
   GPIO.output(trigger_pin, GPIO.HIGH)
   time.sleep(0.00001)
   GPIO.output(trigger_pin, GPIO.LOW)
    # Wait for the echo signal to be received
   while GPIO.input(echo_pin) == 0:
      pulse_start_time = time.time()
   while GPIO.input(echo_pin) == 1:
      pulse_end_time = time.time()
 # Calculate distance using the speed of sound (343 m/s)
   pulse_duration = pulse_end_time - pulse_start_time
   distance = pulse_duration * 17150  # 17150 is the constant for the speed of
sound
  distance = round(distance, 2)
   return distance
try:
   while True:
      distance = measure_distance()
      print(f"Distance: {distance} cm")
      time.sleep(1)
except KeyboardInterrupt:
   GPIO.cleanup()
```

Buzzer for alerting

```
import machine
import time
# Define the buzzer pin (replace 12 with your actual GPIO pin)
buzzer_pin = machine.Pin(12, machine.Pin.OUT)
def buzz(duration_ms=100):
    buzzer_pin.on()
    time.sleep_ms(duration_ms)
    buzzer_pin.off()
while True:
    try:
        # Buzz the buzzer for 100 milliseconds
        buzz(1000)
        time.sleep(1)  # Wait for 1 second before the next buzz
 except Exception as e:
        print("Error:", e)
```