TEAM ID	NM2023TMID12392
TEAM LEAD	Theerthammal K
TITLE	SMART CITY WASTE MANAGEMENT IN PYTHON
NAME	Selvapriya S

`Smart Bin Monitoring System with SMS Alert

The goal of this project is to create a smart bin monitoring system that sends an SMS alert when the bin is full.

Requirements:

* Raspberry Pi

* Ultrasonic sensor

* Breadboard and jumper wires

* Python IDE

* Twilio account for sending SMS

Step 1: Hardware setup

* Connect the ultrasonic sensor to the Raspberry Pi using the breadboard and jumper wires.

* Follow the wiring diagram for the ultrasonic sensor to connect the VCC, GND, TRIG, and ECHO pins to the appropriate GPIO pins on the Raspberry Pi.

Step 2: Software setup

* Install the necessary libraries for the ultrasonic sensor in Python.

One popular library isthe RPi.GPIO library.

* Install the Twilio library for sending SMS.
* Write a Python script to read the distance measurements from the ultrasonic sensor and send an SMS alert when the bin is full.

CODE:

```
import RPi.GPIO as GPIO from
twilio.rest import Client import time

# Twilio account SID and auth token

account_sid = 'YOUR_ACCOUNT_SID' auth_token
= 'YOUR_AUTH_TOKEN'

# Phone numbers

from_number = '+1415XXXXXXX'

# your Twilio phone number
```

```
to_number = '+1415XXXXXXX' #
recipient's phone number
# Create a Twilio client

client = Client(account_sid, auth_token)

# Ultrasonic sensor pins

TRIG = 23
ECHO = 24

# Set up GPIO pins

GPIO.setmode(GPIO.BCM)
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN) def
get_distance():

# Get distance from ultrasonic sensor

GPIO.output(TRIG, True)
time.sleep(0.00001)
GPIO.output(TRIG, False) while
GPIO.input(ECHO)==0: pulse_start =
time.time() while
```

```
GPIO.input(ECHO)==1: pulse_end =
time.time()
pulse_duration = pulse_end - pulse_start distance
= pulse_duration * 17150 distance =
round(distance, 2) return distance while True:

# Read distance measurement

distance = get_distance() print("Distance:",
distance, "cm") # Send SMS alert if bin is
full if distance < 10:

# adjust threshold as needed

message = "The bin is full!"
client.messages.create(body=message,from_=from_number,
to=to_number)
print("SMS sent:", message) time.sleep(1)
```

Summary:

This code sets up the GPIO pins for the ultrasonic sensor and defines a function **get_distance()** that returns the distance measurement in centimeters.

The while loop continuously reads the distance measurements and sends an SMS alert when the distance is less than a certain threshold (indicating the bin is full).

Note:

You may need to adjust the threshold value based on the specific bin and sensor being used.

Conclusion:

Overall, this mini project is a simple but effective way to monitor and manage waste in a smart city using Python and a Raspberry Pi.