TRAFFIC MANAGEMENT

CODE 1:

```
import paho.mqtt.client as mqtt


def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe("traffic_data")


def on_message(client, userdata, msg):
    # Process the received data
    print(msg.topic + " " + str(msg.payload))


client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message


client.connect("mqtt.eclipse.org", 1883, 60)


client.loop_forever()
```

CODE 2:

```
import paho.mqtt.client as mqtt
import your_traffic_analysis_module
```

```
def on_message(client, userdata, msg):
    # Process the received data using your analysis module
    data = msg.payload.decode('utf-8')
    analysis_result = your_traffic_analysis_module.analyze(data)
    # Implement traffic management decisions based on the analysis
    # (e.g., adjusting traffic signals, notifying authorities)

# Rest of the MQTT code here
```

CONCEPT OF CODE:

IoT Sensors and Devices: Deploy IoT sensors such as cameras, traffic flow sensors, and environmental sensors at key points in the traffic management system. These devices collect data and communicate it to a central server. The code for these devices will depend on the hardware and sensors you use, but typically involve microcontrollers or single-board computers like Raspberry Pi or Arduino.

Data Communication: You'll need to implement a communication protocol to transmit data from IoT devices to the central server. MQTT, HTTP, or WebSockets are commonly used. Below is a simple Python example for an IoT device using MQTT

Central Server: Develop the central server that receives, processes, and manages the data from IoT devices. Use languages like Python, Node.js, or Java for this purpose.

Data Processing: Code on the central server processes incoming data, analyzes traffic patterns, and makes decisions based on the data. For instance, you might use Python for this part.

Traffic Control Mechanisms: Implement control mechanisms to manage traffic based on the data analysis. This may involve adjusting traffic signals, providing real-time information to drivers, or even communicating with autonomous vehicles. Control mechanisms can vary greatly depending on the specific goals of your traffic management system.

Security: Ensure that your code and data transmission are secure, as traffic management systems can be a critical part of urban infrastructure.