Building the IOT sensor system and Raspberry pi Integration using python code:

Certainly! Building an IoT sensor system with Raspberry Pi integration can be a multi-step process. Here's a high-level overview of Phase 3, Part 1:
1. *Components and Sensors Selection:* Choose the sensors you want to integrate into your IoT system, such as temperature, humidity, motion, or any other relevant sensors. Ensure they are compatible with the Raspberry Pi.
2. *Raspberry Pi Setup:* Set up your Raspberry Pi with the necessary software, including the operating system (e.g., Raspbian) and any libraries or frameworks required for sensor communication.
3. *Wiring and Connections:* Connect the sensors to the Raspberry Pi using the GPIO pins. Ensure you follow the datasheets and documentation for each sensor.
4. *Programming:* Write code to read data from the sensors. Python is commonly used for Raspberry Pi projects. You'll need to handle data acquisition, processing, and possibly data transmission.
5. *Data Storage and Analysis:* Decide how and where you'll store the data collected from the sensors. You can use a local database or cloud services for this purpose. Analyze the data if necessary.
6. *Communication:* Implement a communication protocol (e.g., MQTT, HTTP, or WebSocket) to send data to a central server or cloud platform for further processing or monitoring.
7. *Security:* Ensure data security by implementing encryption, access control, and other security measures to protect your IoT system.
8. *Testing:* Thoroughly test your system to ensure the sensors are working correctly, data is being transmitted, and the Raspberry Pi is functioning as expected.

Python Code:

```
#define ECHO_PIN1 15 //Pins for Sensor 1
#define TRIG_PIN1 2 //Pins for Sensor 1

#define ECHO_PIN2 5    //Pins for Sensor 2
#define TRIG_PIN2 18   //Pins for Sensor 2

#define ECHO_PIN3 26  //Pins for Sensor 3
#define TRIG_PIN3 27   //Pins for Sensor 3


int LEDPIN1 = 13;
int LEDPIN2 = 12;
int LEDPIN3 = 14;

void setup() {
```

```
  Serial.begin(115200);
  pinMode(LEDPIN1, OUTPUT);
  pinMode(TRIG_PIN1, OUTPUT);
  pinMode(ECHO_PIN1, INPUT);

   pinMode(LEDPIN2, OUTPUT);
  pinMode(TRIG_PIN2, OUTPUT);
  pinMode(ECHO_PIN2, INPUT);

   pinMode(LEDPIN3, OUTPUT);
  pinMode(TRIG_PIN3, OUTPUT);
  pinMode(ECHO_PIN3, INPUT);
}

float readDistance1CM() {
  digitalWrite(TRIG_PIN1, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN1, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN1, LOW);
  int duration = pulseIn(ECHO_PIN1, HIGH);
  return duration * 0.034 /2 ;
}

float readDistance2CM() {
  digitalWrite(TRIG_PIN2, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN2, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN2, LOW);
  int duration = pulseIn(ECHO_PIN2, HIGH);
  return duration * 0.034 / 2;
}

float readDistance3CM() {
  digitalWrite(TRIG_PIN3, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN3, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN3, LOW);
  int duration = pulseIn(ECHO_PIN3, HIGH);
  return duration * 0.034 / 2;
}
```

```
void loop() {
  float distance1 = readDistance1CM();
  float distance2 = readDistance2CM();
  float distance3 = readDistance3CM();

  bool isNearby1 = distance1 > 200;
  digitalWrite(LEDPIN1, isNearby1);


  bool isNearby2 = distance2 > 200;
  digitalWrite(LEDPIN2, isNearby2);


  bool isNearby3 = distance3 > 200;
  digitalWrite(LEDPIN3, isNearby3);

  Serial.print("Measured distance: ");
  Serial.println(readDistance1CM());
  Serial.println(readDistance2CM());
  Serial.println(readDistance3CM());
  delay(100);
}
```