Code:

```
from machine import Pin, PWM
import dht
import time
# Define the GPIO pins for the PIR sensor, DHT22 sensor, and servo motor
pir_pin = Pin(6, Pin.IN, Pin.PULL_DOWN)  # PIR sensor connected to GPIO15
dht_pin = Pin(5, Pin.IN)  # DHT22 sensor connected to GPIO16
servo_pin = Pin(3)  # Servo motor connected to GPIO17

# Initialize the DHT22 sensor
dht_sensor = dht.DHT22(dht_pin)

# Initialize the PWM for the servo motor
pwm = PWM(servo_pin)
pwm.freq(50)  # Standard frequency for servo motors

def set_servo_angle(angle):
    """Set the servo motor to a specific angle."""
    duty = int(3277 + (angle / 180) * (6553 - 3277))  # Convert angle to duty cycle
    pwm.duty_u16(duty)

def read_dht22():
    """Read temperature and humidity from the DHT22 sensor."""
    try:
        dht_sensor.measure()
        temp = dht_sensor.temperature()
        humidity = dht_sensor.humidity()
        return temp, humidity
    except Exception as e:
        print(f"Error reading DHT22 sensor: {e}")
```

```
        return None, None

def check_pir():
    """Check the PIR sensor."""
    return pir_pin.value() == 1

# Main loop
try:
    while True:
        # Check PIR sensor for motion detection
        if check_pir():
            print("Motion detected!")

            # Read DHT22 sensor
            temperature, humidity = read_dht22()

            if temperature is not None:
                print(f"Temperature: {temperature:.2f} °C, Humidity: {humidity:.2f} %")

                # Rotate the servo motor to 90 degrees when motion is detected
                set_servo_angle(90)
                time.sleep(1)  # Hold the position for 1 second

                # Optionally, rotate the servo back to 0 degrees
                set_servo_angle(0)
            else:
                print("Temperature and humidity data not available.")
        else:
            print("No motion detected.")
            # Optionally, you can keep the servo in a neutral position or off
            set_servo_angle(0)
```

```
        time.sleep(2)  # Wait for 2 seconds before checking again

except KeyboardInterrupt:
    print("Program interrupted")

# Cleanup PWM signal
pwm.deinit()
```