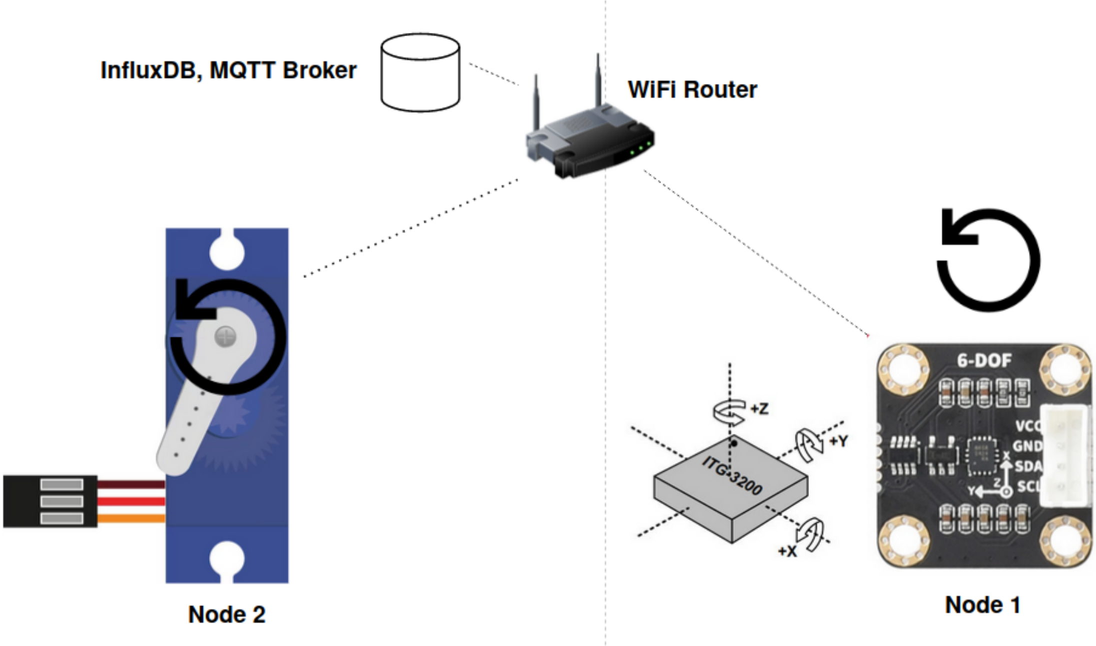
Assignment #1 : Control Servo Motor with Gyro Sensor

Description

- Create an embedded system software for Raspberry Pi Pico to control the position of the servo motor wirelessly with gyro sensor



Requirement

1.1 System consists of 2 Nodes. Node#1 is for gyro sensor. Node#2 is for Servo motor control.

1.2 Each node will be connected together with WIFI. MQTT will be used for communication protocol.

Node#1

1.3 Node#1 will read the gyroscope's angular velocity value (Pitch, Roll, Yaw) with period of 100 ms

1.4 Node#1 converts angular velocity to angle. Then it will publish the angle value to the InfluxDB via MQTT.

Node#2

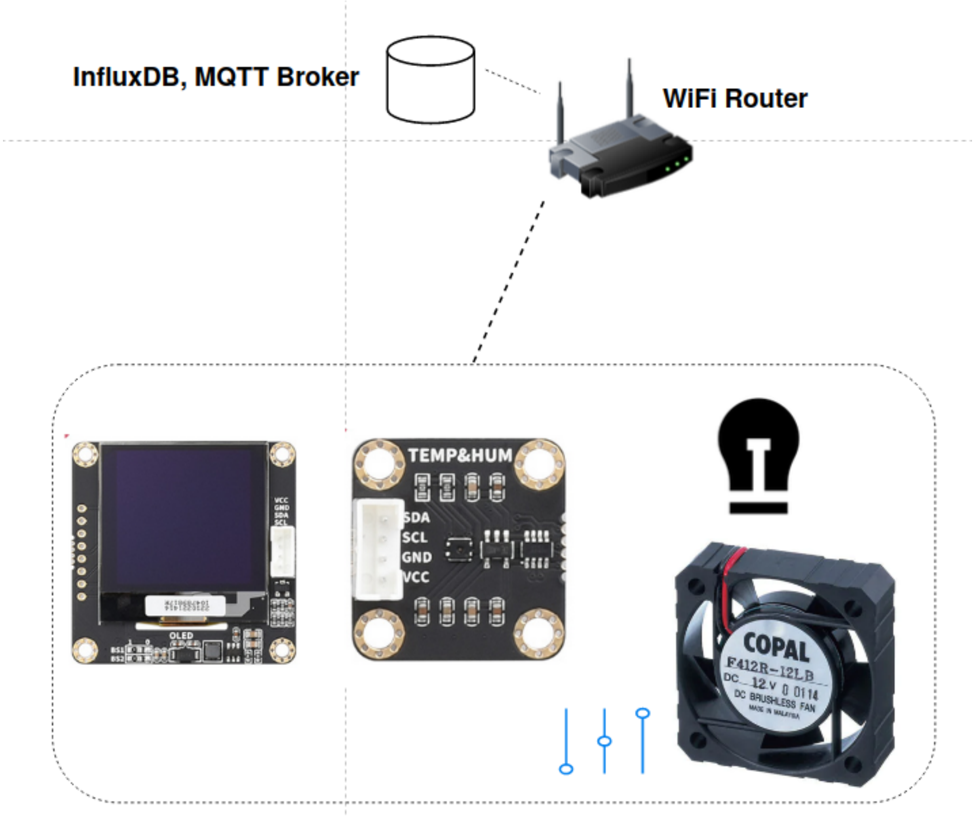
1.5 Node#2 will subscribe the angle value from the Node#1.

1.6 Node#2 will control the angle of servo motor corresponds to the change of Yaw angle (Z-axis)

Assignment #2 : Temperature & Humidity Control System

Description

- Create an embedded system software for Raspberry Pi Pico to monitor and control the temperature and humidity value with a fan motor.



Requirement

1.1 Read the temperature and humidity from TEMP&HUM sensor with the period of 0.5 seconds

1.2 Display the temperature and humidity on OLED display with refresh rate of 3 seconds

1.3 If temp > Threshold1 or If humid > Threshold2.
    Then, Turn on the fan motor (with constant speed) and RED_LED
    Else, turn off the motor and RED_LED

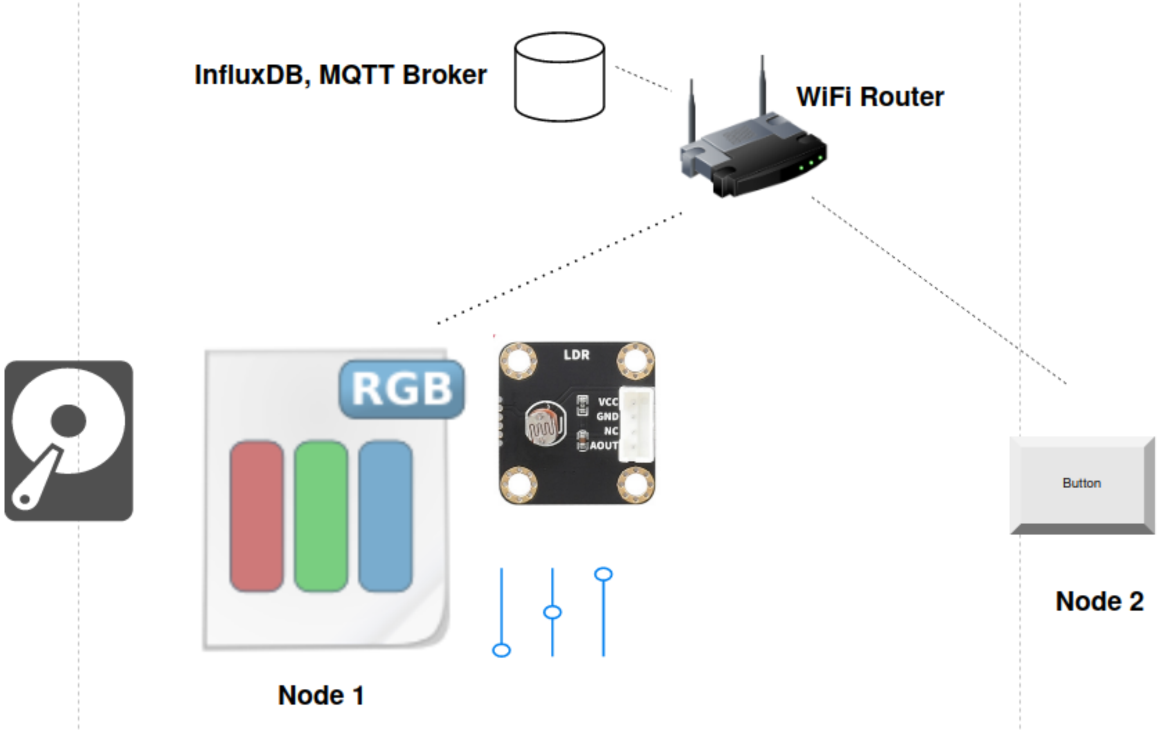
1.4 The speed of the fan motor can be adjusted by the VOL_ADJ module (0 - 100% Speed). Update rate is within 100 ms.

1.5 Send the temperature and humidity data to the InfluxDB with MQTT Protocol. Use WIFI as the wireless connection to the server.

Assignment #3 : Smart Light System

Description

- Create an embedded system software for Raspberry Pi Pico to control the RGB_LED with sensors



Requirement

1.1 System consists of 2 Nodes. Node#1 is for RGB_LED control. Node#2 is for the remote button.

1.2 Each node will be connected together with WIFI. MQTT will be used for communication protocol.

Node#1

1.3 Read LDR light sensor with period of 0.5 seconds,

1.4 Turn off RGB_LED when there is a light.
Turn on RGB_LED when the light is getting dark with (R, G, B) value.

1.5 Brightness adjust mode : brightness of each R, G, B value can be adjusted after the button (KEY) is pressed. Then, use VOL_ADJ to adjust. Press the button again to rotate to a different colour.

1.6 Display the R, G, B value to the OLED display for every 3 second

1.7 Backup the current RGB value into the EEPROM for every 1 second. Everytime the system is rebooted, the previous RGB value from EEPROM will be loaded as the current value.

1.8 Remote control : Toggle the RGB_LED when button from the Node#2 is pressed. Button status of Node#2 can be checked by subscribing the button state of Node#2.

Node#2

1.9 Publish the button state when the button is pressed.