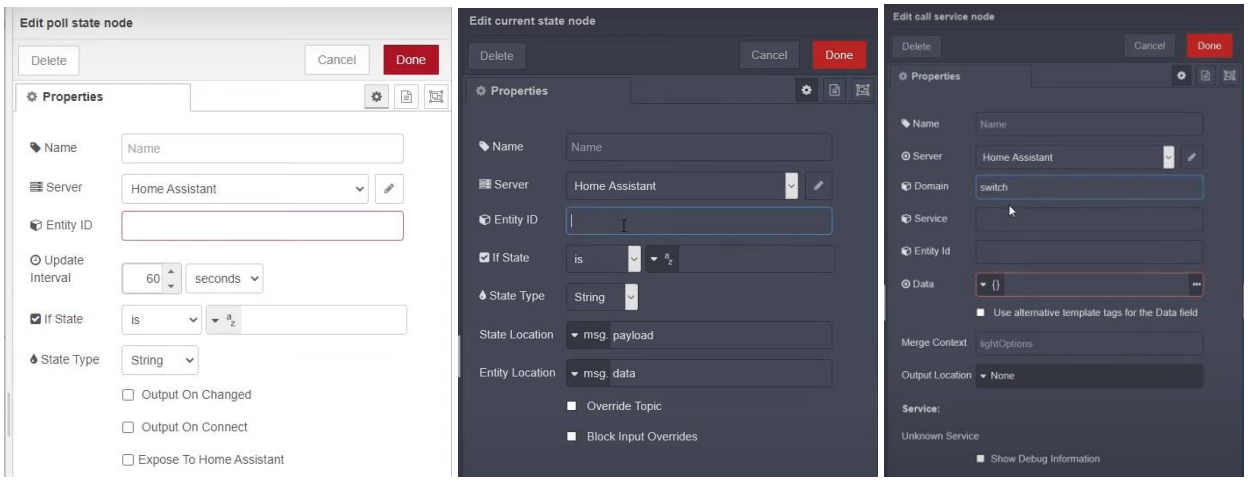
Documentation for Task 1

These are the 3 nodes you will need to add on your Home Assistant NodeRED flow along the already created ones.



Node configurations for ones to be added.

Poll State: This node will be used to poll the values of Wind Speed and Wind Direction at regular intervals, set the interval as per choice based upon how much error margin you can tolerate. The value can be anywhere between 1s to 10s for a 2-minute comparison.

- Add a unique Name to identify your node.
- Set the Home Assistant Server. Preferably check the Home Assistant addon option in server setup and it sets the port and addresses itself accordingly.
- Set the Entity ID according to you Ecowitt sensor wind speed instance i.e. “sensor.wind_speed_2” or similar.
- Set the Update interval as per choice (anywhere between 1-10s)
- Set the State type to numerical so it outputs a number representing wind speed and direction.
- Check the output on connect box to set the outputs on each call made and not just a change.

Leave the rest of options as default.

Current State: This node will be used in addition to turn on and off of the switch. Its basic purpose is to check that if the output is already ON, we don’t need to overflow the node with repetitive ON commands. Likewise, for OFF as well. It’s just a neat way to manage output node even though it won’t be an issue if you ever do end up overflowing the switch node.

- Add a unique Name to identify your node.
- Set the Home Assistant Server. (Pick the same server from dropdown which you created last time)
- Set the Entity ID according to your Switch instance i.e. “switch.tplink1” or similar.
- Set the If State to match the state of switch next to it (either on or off)

Leave the rest of options as default.

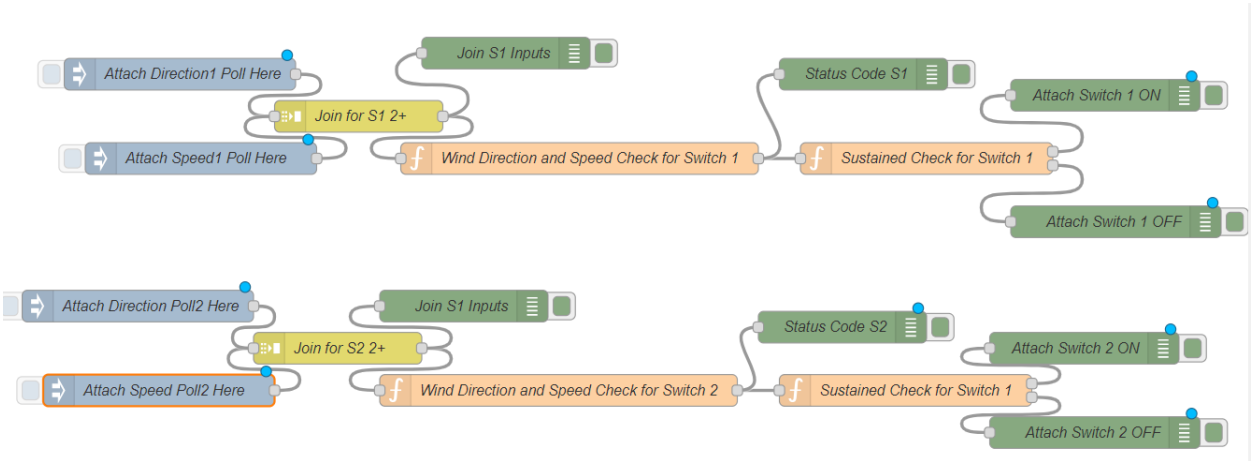
Call Service: This node will be used to turn on and off your switch instance. Place two of these next to each conditional check outputs. One for on and other for off switch action.

- Add a unique Name to identify your node.

- Set the Home Assistant Server. (Pick the same server from dropdown which you created last time)
- Set the Domain as switch which is what you will be controlling.
- Set the Service as either turn_on or turn_off as per requirement.
- Set the Entity ID according to your Switch instance i.e. “switch.tplink1” or similar (This should match the entity id from current state node as current state will be checking the state of this very switch)

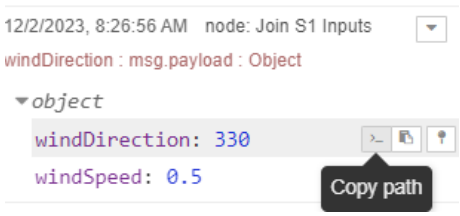
Leave the rest of options as default.

Add the relevant nodes and setup the below existing flow with the newly added nodes.



Once you add the Poll nodes you need to update the function “Wind Direction and Speed Check“ nodes with new attributes. Below are the required changes.

Deploy the changes and observe the join S1 inputs debug output. It should show a group of two elements, one being the windspeed and other the wind direction. Go over to the debug pane and you will see a small icon that says, “Copy Path”.



Copy the path for each and update it in the function node. If your path doesn’t contain “msg.#your_path” then keep the “msg.” part same and paste the rest of your path along it. Following is the complete function node code.

```
var windDirection = msg.payload.windDirection; // paste your own path here
var windSpeed = msg.payload.windSpeed; // paste your own path here
var currentTime = Date.now();
var statusCode = flow.get('statusCode_S1');
var startTimestamp = flow.get('startTimestamp_S1');
if (windSpeed > 5 && windDirection >= 310 && windDirection <= 360) {
    if (statusCode !== 200) {
        flow.set('statusCode_S1', 200);
        flow.set('startTimestamp_S1', currentTime);
    }
```

```
} else if (windSpeed < 1) {
    if (statusCode !== 400) {
        flow.set('statusCode_S1', 400);
        flow.set('startTimestamp_S1', currentTime);
    }
} else {
    flow.set('statusCode_S1', null);
    flow.set('startTimestamp_S1', null);
}
msg.payload = statusCode;
return msg;
```

Explanation:

The first function node code basically checks for the given conditions of wind speed and wind direction and if wind direction is between 310 and 360 along with wind speed above 5mph it sends out status code 200. And if the windspeed is below 1mph it sends out the code 400. The flow.get creates a variable that is global to this flow, meaning it can be shared across the nodes of this flow. If the condition of code 200 is met, it checks if the condition if this is a consecutive call or not, as once this condition is met, the new data would also correspond to same values, and we don't want to reset our timer based on that. So, if code 200 condition is met, it stores the current time stamp in the global flow variable `startTimestamp_S1` and set the status code as well using flow.set command. This ensures a start point for our timer to check if desired time has passed and should an action be taken or not.

The following Sustained Check Node has the code:

```
var statusCode = flow.get('statusCode_S1');
var startTimestamp = flow.get('startTimestamp_S1');
var currentTime = Date.now();

if (statusCode === 200 && currentTime - startTimestamp >= 20000) {
    // Code 200 sustained for 10 seconds, turn on the switch
    msg.payload = "Switch 1 ON";
    return [msg, null];
} else if (statusCode === 400 && currentTime - startTimestamp >= 20000) {
    // Code 400 sustained for 10 seconds, turn off the switch
    msg.payload = "Switch 1 OFF";
    return [null, msg];
} else {
    return [null, null]; // No action needed
}
```

This code checks the status code that has occurred, waits for the time specified in milliseconds such as above is 20000 or 20seconds and then sets the respective output high. This is used to implement the logic ON and OFF for switch. Add the State Check node followed by the service call to switch node consecutive to this node.