BGU IIOT DDS Assignment

System Overview

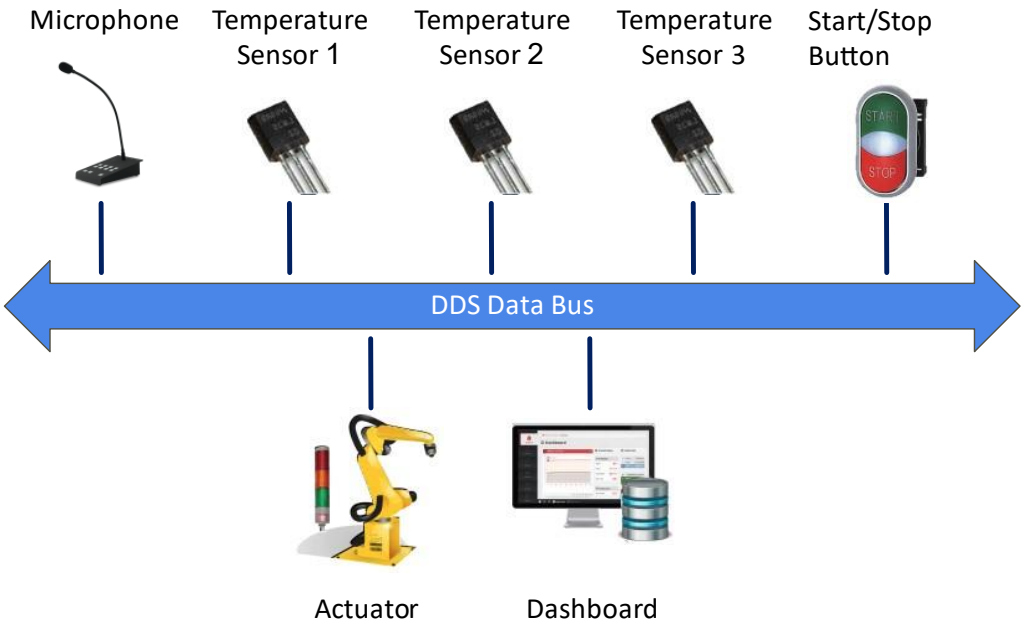


Figure 1: Connectivity Architecture

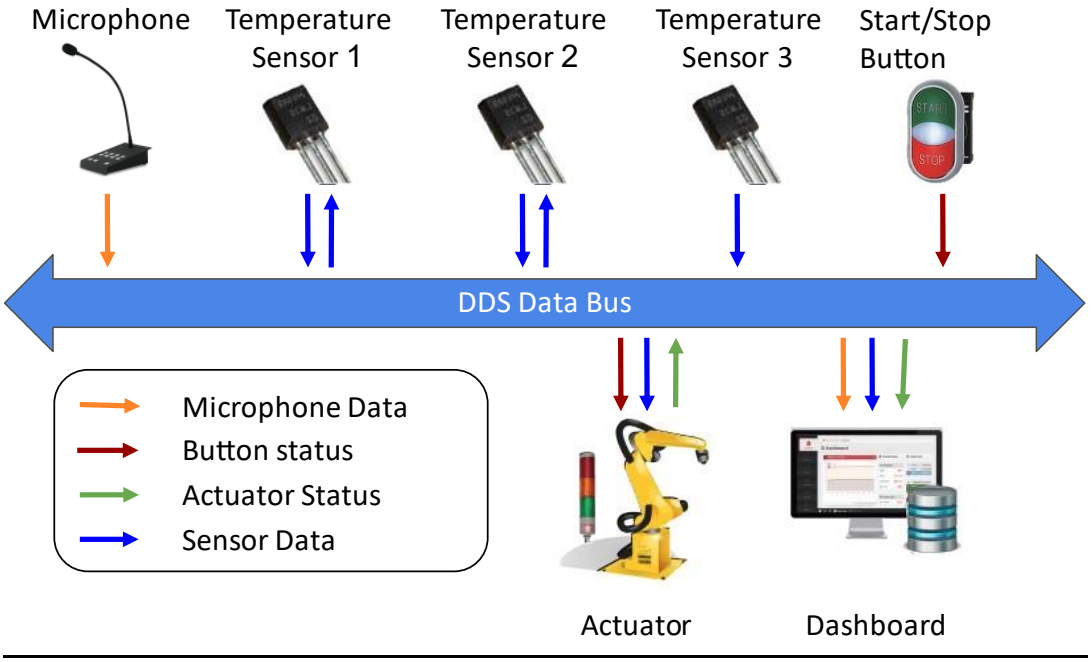


Figure 2: Connectivity Architecture – Data Flow View

Application’s Logic Description

Microphone (5)

- Sends the current state at 10 Hz (every 0.1 seconds).
- For the sake of simplicity, the message will be a string with the real time in it.
- It is only important to look at the current state, **there is no need to ensure the arrival of each message and no need for the previous messages here**.
- The microphone shall also print the message to a console.

Start/Stop Button (10)

- Sends a Starts/Stops command (default is Start).
- For the sake of simplicity:
    - A stop command shall be sent every 20 seconds.
    - 5 seconds after the stop command, a start command shall be sent.
- The Button shall also print the command to a console.

Temperature sensor 1/2 (10)

- The sensors shall publish a temperature reading at 10 Hz (every 0.1 seconds).
- For the sake of simplicity, the temperature shall be a random value 23±6 degrees (integer).
- The sensor shall also print the temperature to a console.
- While the actuator is in stopping status, the sensors shall not print.

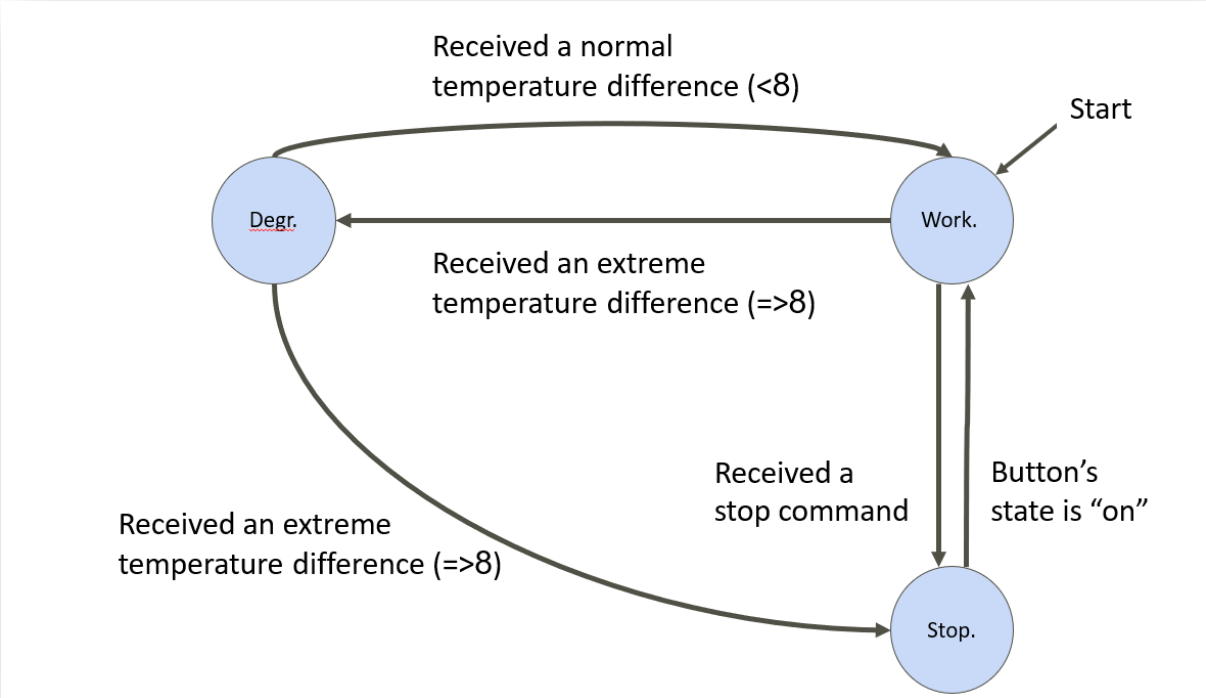
Temperature sensor 3 (5)

- The sensor shall publish a temperature reading at 1 Hz (every second).
- For the sake of simplicity, the temperature shall be a random value between 23±1 degrees (integer).
- The sensor shall also print the temperature to a console.

Actuator (20)

- The actuator shall publish its current status **upon change** [Working, degraded, Stopped].
- The actuator shall print its status to a console at 1 Hz (every second).
- After changing its status to stop, the actuator shell calibrate itself before returning to "working" status. For the sake of simplicity, that means waiting for 0.5 seconds. Before waiting, the actuator shall print to a console the reason for stopping ("received a stop command"/"calibration is needed".)
- If the reason for stopping is extreme temperature, the actuator shell also print to a console a message in the format: "Difference measured: <temperature difference>, calibration thermometer measurement: <Temperature sensor 3 reading>.

- The actuator shall change its status according to the following state machine:



Dashboard (20)

- The Dashboard shall print the status of the system to a console every 5 seconds.
- The Dashboard shall display the last 10 measurements **only** of an extreme temperature difference from Temperature sensor 1 and from Temperature sensor 2 (**even if the dashboard was initialized after the sensors sent their last status messages**.
- The Dashboard shall display the last 10 statuses of the actuator **even if the dashboard was initialized after the actuators sent their last status messages**.
- The template of the dashboard print to a console:
    - `Microphone: <the last microphone's message>`
    - `Actuator: [list of last 10 published statuses]`
    - `Thermometer 1: [extreme reading list (1)]`
    - `Thermometer 2: [extreme reading list (1)]`
    - `The last calibration's time: <last calibration's time>`

(1) Extreme reading list is a list containing the last temperature measured before calibration is needed (the last 10 readings only)

What to do?

- Implement the application logic described above and use DDS capabilities as you see it right. Our recommendations:

    - Understand fully the assignment and what needs to be done
    - Understand which types of messages, topics, participants and QoS will be used in the system
    - Make a small publisher - subscriber example for yourself to play with different types of QoS
    - Implement each element one at a time and check the system (by running) after each step
    - Run the full system, see if everything works as expected
    - Submit
- Serve 7 python files (one for each component) + DDS XML configuration file (named DDS.xml) in a zip over Moodle. The name of the file shall include the IDs of both students.

The grade

❖ Microphone	5
❖ Start/Stop Button	10
❖ Temperature sensor 1	5
❖ Temperature sensor 2	5
❖ Temperature sensor 3	5
❖ Actuator	20
❖ Dashboard	20
❖ Readability and clarity of the code	30