

BGU IIOT DDS Assignment

System Overview

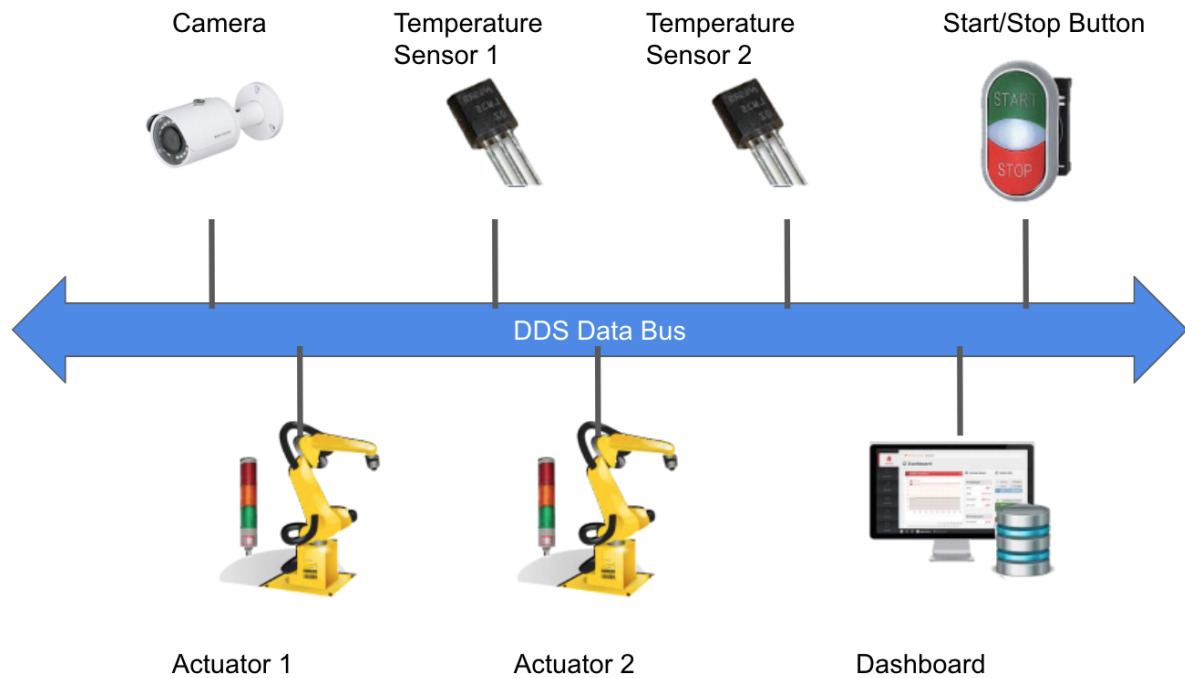


Figure 1: Connectivity Architecture

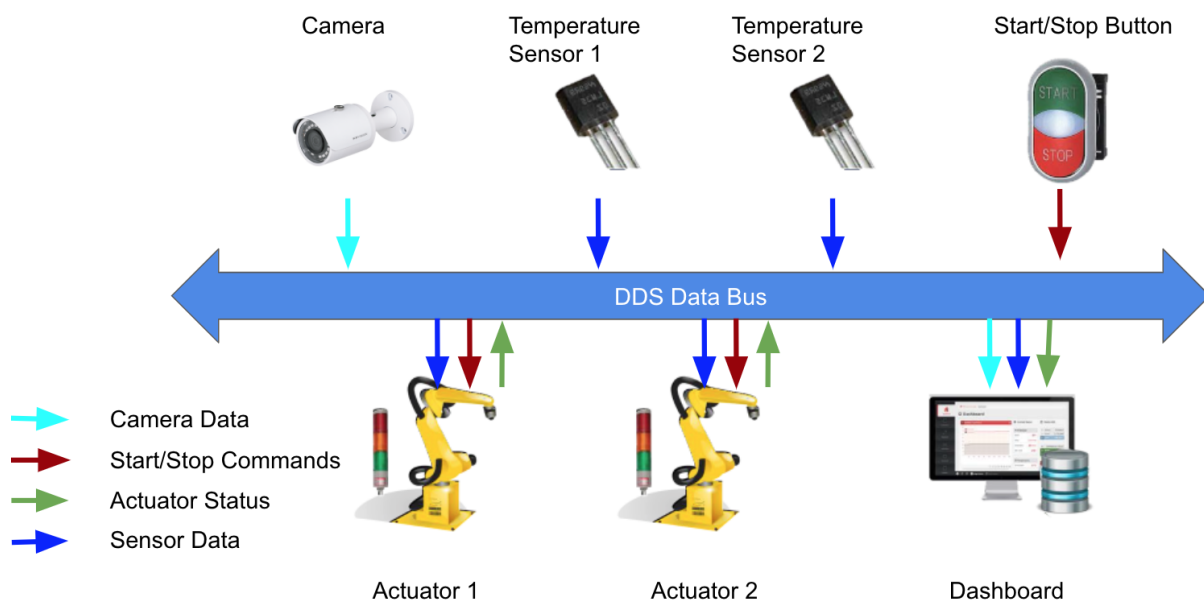


Figure 2: Connectivity Architecture – Data Flow View

Application's Logic Description

Camera (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 Camera 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 (10)

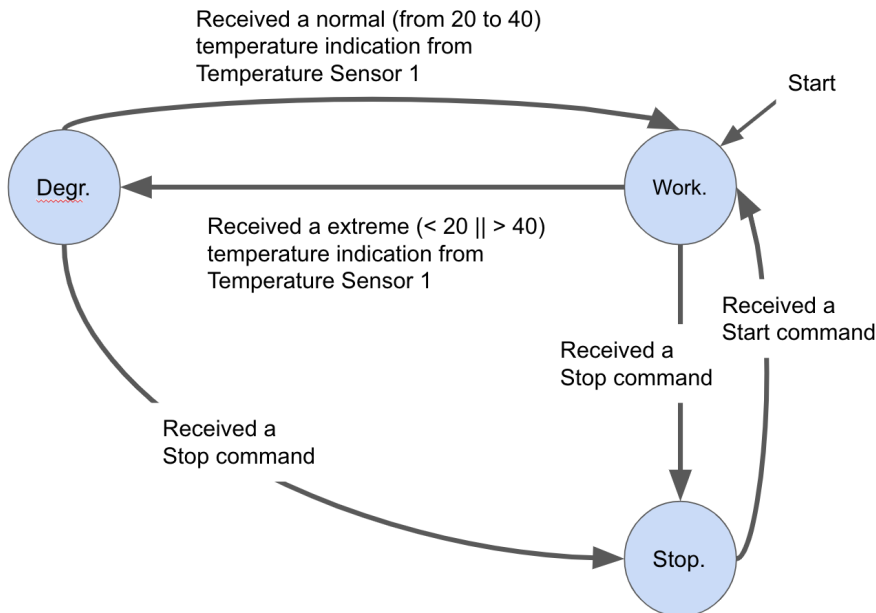
- The sensor shall publish a temperature reading at 1 Hz (every second).
- For the sake of simplicity, the temperature shall be a random value between 10-60 degrees (integer).
- The Sensor 1 shall also print the temperature to a console.

Temperature Sensor 2 (10)

- The sensor shall publish a temperature reading at 10 Hz (every 0.1 seconds).
- For the sake of simplicity, the temperature shall be a random value between 0-50 degrees (integer).
- The Sensor 2 shall also print the temperature to a console.

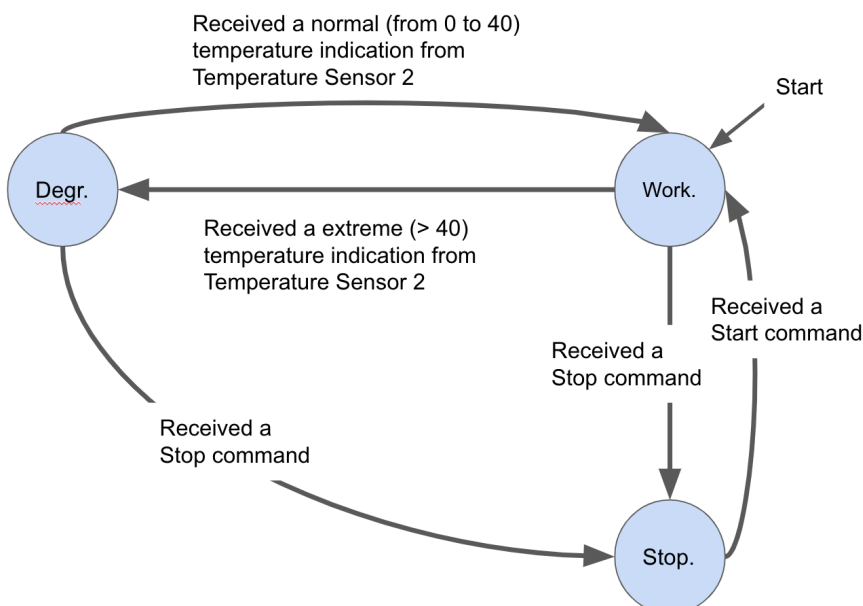
Actuator 1 (15)

- The actuator shall publish its current status **upon change** [Working, Degraded, Stopped].
- The actuator shall print its status to a console.
- The actuator shall change its status according to the following state machine:



Actuator 2 (15)

- The actuator shall publish its current status **upon change** [Working, Degraded, Stopped].
- The actuator shall print its status to a console.
- 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 from Temperature Sensor 1 and from Temperature Sensor 2 (there are different definitions of “extreme” for each sensor - see the graphs) **even if the dashboard was initialized after the sensors sent their last status messages.**
- The Dashboard shall display the last 10 statuses of each actuator **even if the dashboard was initialized after the actuators sent their last status messages.**
- The template of the Dashboard print to a console:
 - Camera: *<the last message received from the camera>*
 - Actuator 1: *[list of last 10 published statuses]*
 - Actuator 2: *[list of last 10 published statuses]*
 - Extreme Temp. 1: *[list of last 10 extreme temperatures]*
 - Extreme Temp. 2: *[list of last 10 extreme temperatures]*

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 Actuators' IDs should also correspond to the IDs of both students.

The grade

❖ Internal Counter	5
❖ Start/Stop Button	10
❖ Temperature Sensor 1	10
❖ Temperature Sensor 2	10
❖ Actuator 1	15
❖ Actuator 2	15
❖ Dashboard	20
❖ Readability and clarity of the code	15