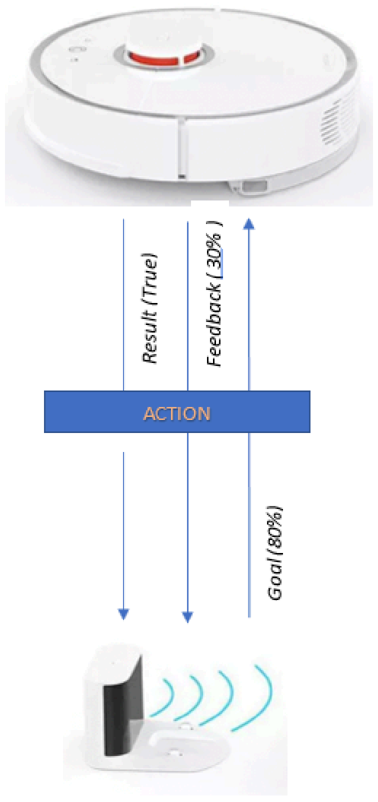
Exercise 2: ROS Actions

The scenario for this exercise is the same as the previous homework. We still have a robot vacuum cleaner that communicates with a charging station (CS) using ROS. This time, the robot and the charging station interact during the battery charging phase, in particular the scenario will be the following one:

- Suppose that the robot has already arrived and attached to the charging station.
- The robot has a low level of battery (b) (eg. 5%). Thus, it needs to recharge its battery.
- The communication between CS and robot is based on an Action.
- The CS (action client) sends a goal request that contains the charging power that it can provide (p <= 100%).
- The robot (action server) can accept or reject the request.
- If the action is accepted, The CS starts the charging procedure.
- However, the robot takes 1 minute to reach max(100%, b+p) (i.e., implement this logic by yourself). During this time, the robot keeps writing the current state of battery level in the feedback topic of the ROS action.
- Once the robot arrives at 100%, it communicates the result to the charging station.

An example of the proposed scenario is given below:



Given the previous scenario, we request you to implement the following structure:

1. Create a custom ROS action that contains the following:
    a. **Goal**
        i. **request** with a std_msgs/Header and an integer that represent the power (p) provided by the CS.
        ii. **response** (accept/reject).
    b. **Feedback** with a std_msgs/Header and the current level of charging. This feedback has to be updated every second (1Hz frequency).
    c. **Result**
        i. **response** with a std_msgs/Header and an integer with the new battery state (b+p).