RTDE Connector Notes:

* When you send an input key to the robot, you need to send values for all the fields within that key. Why?
  + Would need to get the values of all the inputs beforehand and send them to RTDE on the backend
    - This becomes problematic. Would need to receive an output packet of every output in RTDE which means I would have to create a key in the XML that contains everything. I don’t want to do that in case of resource overload (dunno if that would happen)
    - If I don’t receive a packet with all outputs, I would need to associate every input with its output, IF it has one (do they all have one?) and then check the value. This could work, but would make the Connector very RTDE version-dependent (technically might be able to check RTDE version)
    - OR I could just assume and set the value of all the inputs to 0 (or their type-equivalent). This could cause problems for users because it would wipe out their stored values in case RTDE breaks connection and they restart. I don’t want to do that
* To create a “send all” command, I need to have a list of all the current fields (in order) for each key
  + When accessing the inputDict after initialization, each key’s value is a DataObject which does not have the fields or values readily available like the DataConfig does
  + Option 1: unpack the DataObject manually every time you want to send data, get the fields, and then send the updated values (not feasible and may take too long)
  + Option 2: Add an additional list to each key in inputDict, thus turning the value into a list itself
    - Doable, but will make the initial parsing and organizing of input keys and values messier than it already is
  + Option 3: Create a new dictionary that just contains the fields for each key
    - Works and is easy to use, but adds an additional dictionary to the code that has similar properties of inputDict
* RTDE doesn’t care about how requested outputs are split up in the XML
  + Calling “receive” will return a packet of data containing all outputs sent via the “send\_output\_setup” function
  + Only reason to have more than one key for controller outputs would be for your own organizational purposes
  + You can try splitting off an output you’re using into a completely different key and the code should work exactly the same