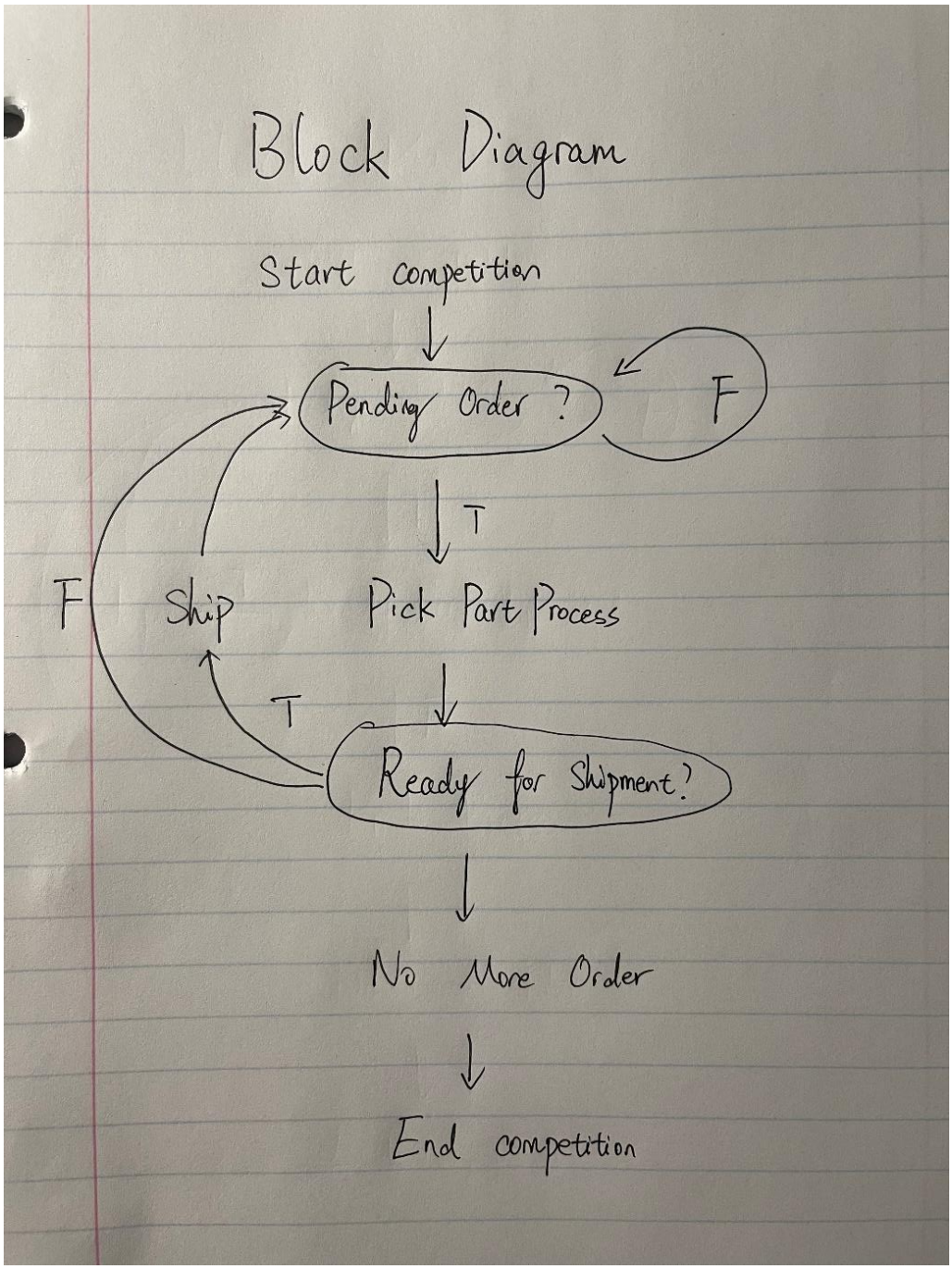
Theory of Operation

The purpose of this document is to briefly introduce the theory of the operation of ARIAC 2019 with the help of the UR10 Manipulator Robot. To begin with, the following two diagrams show the block design for this robot system.



The above block diagram shows how the system moves among states. After the gazebo is launched and the competition simulation starts, the system will continuously scan whether there is still a pending order by subscribing to */ariac/orders*, if a new order is found, the order information will guide the system on which bin to look for, and the camera (OCR) for appointed to that bin can help find a specific location of where the desired part can be located (by subscribing */ariac/material_locations*). This location, combined with the ur_kinetic library, can guide the robot to move to the correct location, pick up the part, move to the correct tray (argv), and then release the part. In the following paragraphs, this process will be discussed in more detail. Once the part is released onto the tray, the system decides whether it is ready for shipment (since sometimes an individual shipment can cover multiple different parts). After the shipment process, the system will scan and wail for new orders (yes in the current implementation the ur10 arm will not move until the shipment process is finished). For the current version, this waiting process will go on forever until the user kills the process.

The diagram below gives more details on how the UR10 robot arm is moved to the desired location and finishes the pick-up/put-down process. The new order information guides the system on which specific bin to look for the desired part, then the base is moved to the correct location first with an offset, the purpose of this offset is to avoid collision with the camera providing OCR above. Then, it is necessary to transform from the camera version to the real-world version (the camera can only provide visions where parts are relative to itself, it cannot tell its actual location in the real world). Through the kinetic_inverse function, it is possible to find solutions for all the angles of joints of the UR10. UR10 will be guided to move close enough to the part but stop right above the part, the vacuum gripper will be turned on and the part can be picked up. A while loop is designed since the vacuum gripper can sometimes fail (obtain gripper state by subscribing to /ariac/arm/gripper/state). Following the same logic, the UR10 then retreats its arm, moves the base, sets the angles of the joints, and releases the vacuum gripper so that the part can be properly put on the correct tray. This process continues until the parts specified are all ready for shipment or the user signals to stop. To avoid a collision, it is worth noticing that the UR10 will be set to move back to a preset default location before/after each

process, and the movement of the foundation/arm is separated apart, these designs can help the robot from crashing into the neighbor environment.