by using the UR-RTDE library developed by SDU [88]. It provides a real-time interface to exchange I/O data and control the robot arm from an external application or program. The library works by establishing a communication link between the computer and the UR robot arm using an Ethernet connection. It utilizes the Real-Time Data Exchange (RTDE) protocol, which is a proprietary protocol developed by Universal Robots for real-time communication [89]. The connection makes it possible to receive real-time feedback from the robot arm, such as joint positions, joint velocities, and other sensor data. Additionally, it enables sending commands and controlling the robot arm's motion in real-time.

The pneumatic clamps and the cylinder are controlled by using electrically actuated 5/2-way control valves. These are both connected to two digital output connectors of the UR control box. Each output can control the airflow in one output 6 mm tube of the valve, which is responsible for extending or compressing the clamps and the cylinder. The Robotiq Hand-E gripper is connected to the robot arm by a power and communication cable with a USB adapter. The microscope, camera, and rotary electric gripper are all connected by a USB cable (type 3.0 output) to the PC. While the camera and the gripper can be controlled by using specific Python libraries (pyrealsense2, minimalmodbus), the microscope needs the CytoSMART driver installed on the PC. The entire system is controlled by an application written in Python. This includes a simple human-machine interface to control the tasks and see the output of the microscope. The recorded data (images taken by microscope and camera) are stored locally on the PC. The functional interconnection of hard- and software is visualized in fig. 3.6. The UR control box, the rotary electric gripper, the incubator, and the PC are connected to standard 230V electrical sockets.

3.1.3 Process and workflow description

Similar to the manual process, the autonomous process is split into three high-level workflows.

- Workflow A: Analyzing cell growth.
- Workflow B: Changing media.
- Workflow C: Passaging.

The system should be able to execute the three workflows independently, depending on the user input or the scheduled plan. To enhance reusability and simplicity, the steps to achieve completion of the workflows are structured into workflow modules.

The following gives a mid-level overview of the purpose of each process module, including a summarized description of the substeps. A low-level description is given in THE APPENDIX???

- Module 1: Get a cell culture flask from the incubator (open clamps, open door, grip the flask, move the flask outside, close door and clamps)
- Module 2: Analyze cells (move flask to regripping station, regrip, place the flask on the microscope, take images at different positions, move back to start position)
- Module 3: Place a flask in a flask holder (move to the flask holder, place the flask inside, and move back to the start position)
- Module 4: Decap flask(s) (move to the flask, grip the lid, take off the lid, place the lid on a lid holder, and move back to the start position)
- Module 5: Remove liquid from a flask (take the open flask from a flask holder, move to the waste container, pour out all the liquid, and place the flask back into the flask holder)
- Module 6: Add liquid to a flask (heat up media or washing solution, take the bottle, decap the bottle, regrip the bottle, pour a specific amount into the flask, regrip the bottle, cap the

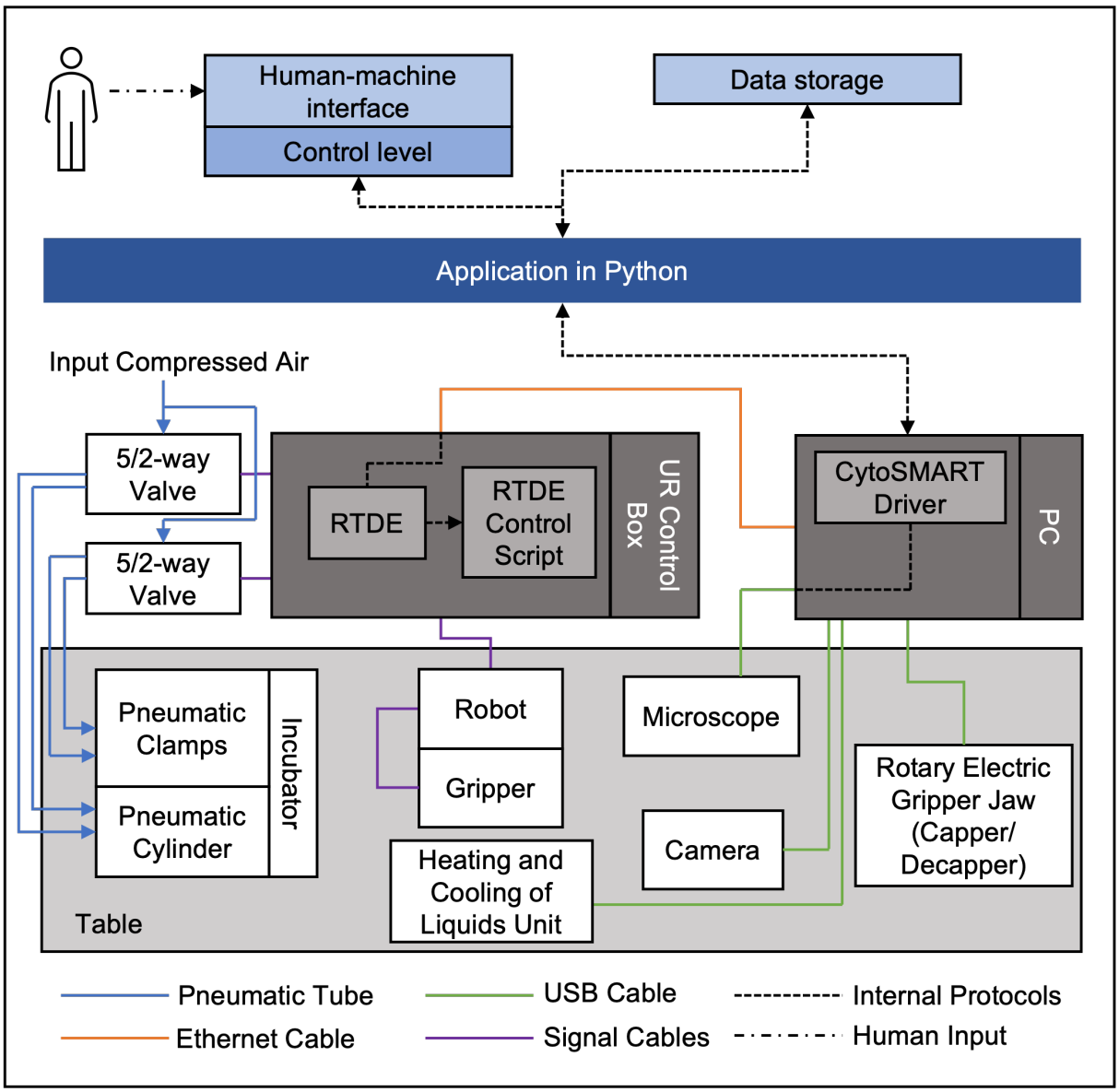

Figure 3.6: Simplified overview of the functional interconnection of hard- and software.

bottle, and place it back into the bottle holder)

- Module 7: Add trypsin to a flask (move to the trypsin unit, move up the bottle dispenser, push it down, and move back to the start position)
- Module 8: Get three empty flasks and place them into the flask holders (move to flask storage, grip an empty flask, move to the flask holder, place the flask inside, and move back to the start position (3 times))
- Module 9: Split cells into empty flasks (take the full flask from the flask holder, move to the empty flasks, pour liquid three times, and place the empty flask into the flask storage)
- Module 10: Cap flask(s) (get a lid from the lid holder, place the lid on a flask, cap the flask, and move back to the start position)
- Module 11: Place a flask in the incubator (open clamps, open door, move the flask inside, place the flask on the flask storage, close door, and close clamps)

Throughout the implementation of the modules, careful consideration was given to the selection of start and end positions, ensuring the flexibility to arrange them in any order, thus enabling diverse workflows to be executed. The process diagram illustrating the three primary workflows can be observed in fig. 3.7. This design approach allows for seamless customization and adaptation of the system to various operational requirements.

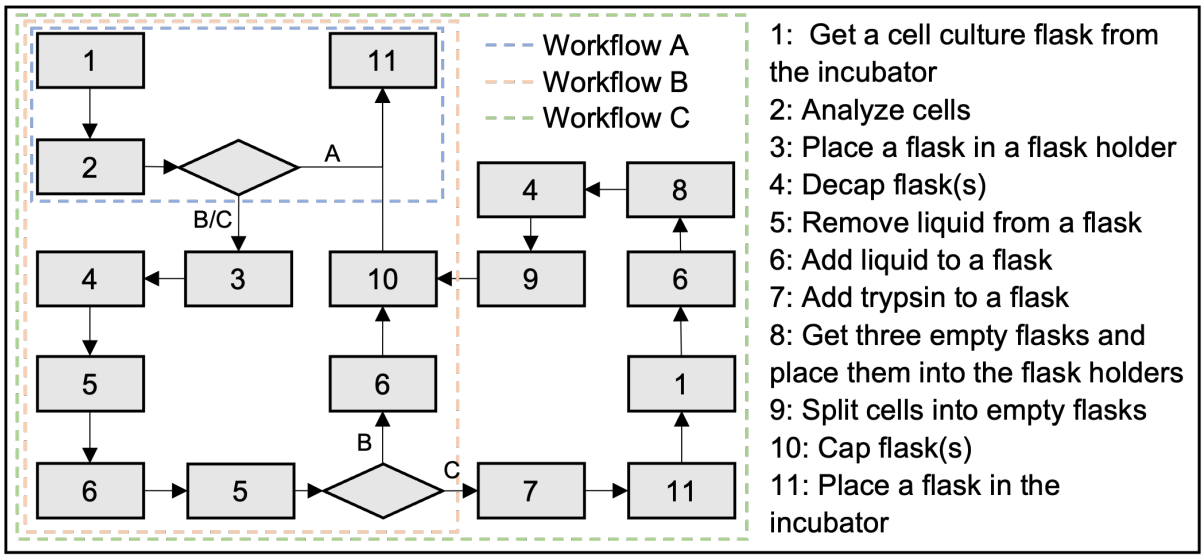

Figure 3.7: Process diagram for the three main autonomous workflows.

3.2 Vision-based Detection and Volume Estimation of Liquids

This chapter explains the methodology used for vision-based detection and volume estimation of liquids. As seen in subsection 2.2.2, very little research has been done in the field of liquid detection in research laboratories. Additionally, to the author's knowledge, there is currently no established approach for estimating volumes based on computer vision in research laboratory environments. This chapter starts by explaining the idea and approach of the vision-based system (subsection 3.2.1). It continues by providing details about the process for segmentation and depth estimation of transparent vessels and liquids inside of them (subsection 3.2.2). After, the methodology and procedures for the vision-based volume estimation of liquids will be explained (subsection 3.2.3).

3.2.1 Idea and Approach

The proposed approach consists of a two-step process for liquid volume estimation. First, train a convolutional neural network (CNN) on the TransProteus dataset for segmentation and monocular depth estimation of liquids and transparent vessels. The resulting model can be used for liquid and transparent detection for process monitoring (e.g., is a flask present or not). By using the camera intrinsics and combining the segmentation and depth maps, segmented vessel and liquid XYZ maps are created. Subsequently, a custom dataset is generated, containing images of transparent vessels commonly used in research laboratories, together with the liquid volumes stored as information. The initial custom dataset is then converted into a new dataset containing segmented depth maps of liquids and vessels using the trained segmentation and depth estimation model. Finally, a new CNN is trained on the converted dataset for liquid volume estimation. The approach is based on the idea that a model can learn from the shape and distance of the vessel and liquid as well as from the relationship between liquid and vessel maps (e.g., the percentage filled) to infer a better estimation of the volume than just from providing an image. The approach is visualized in fig. 3.8.