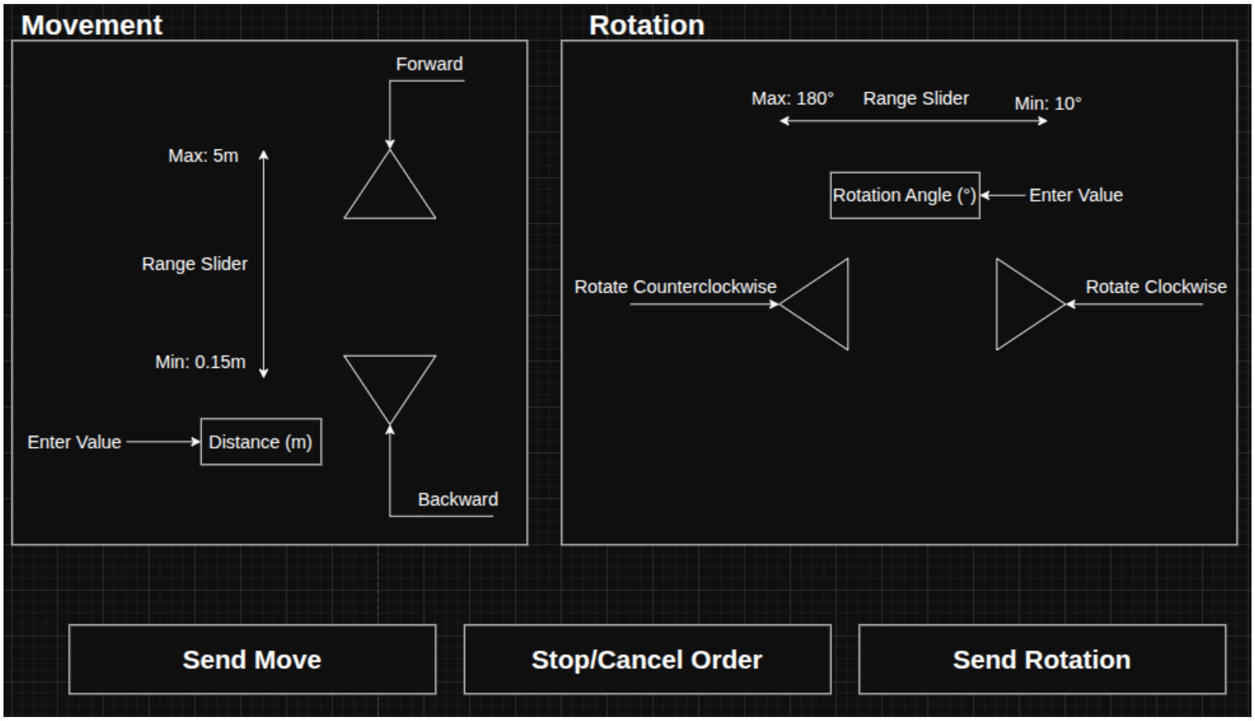
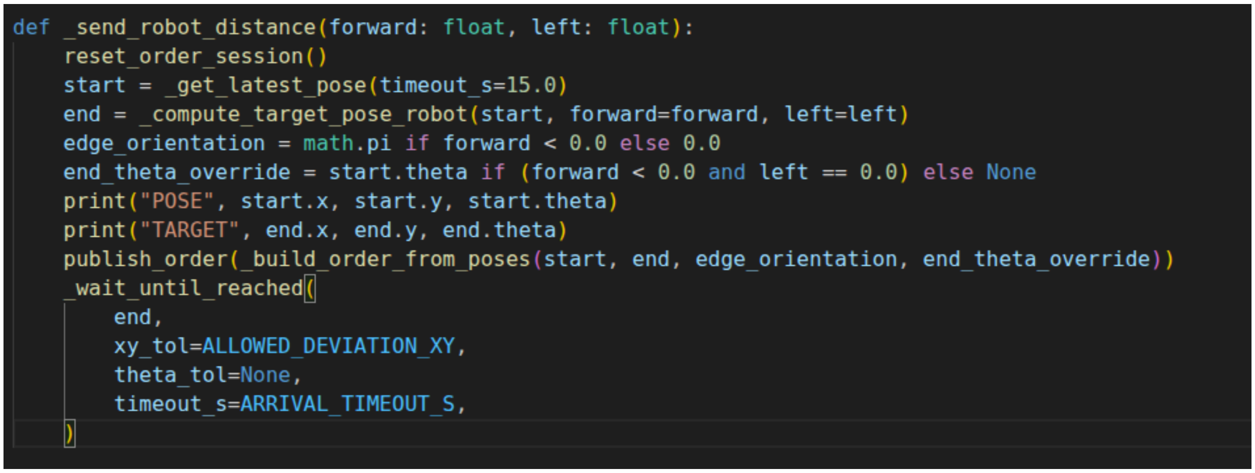
Giao diện UI

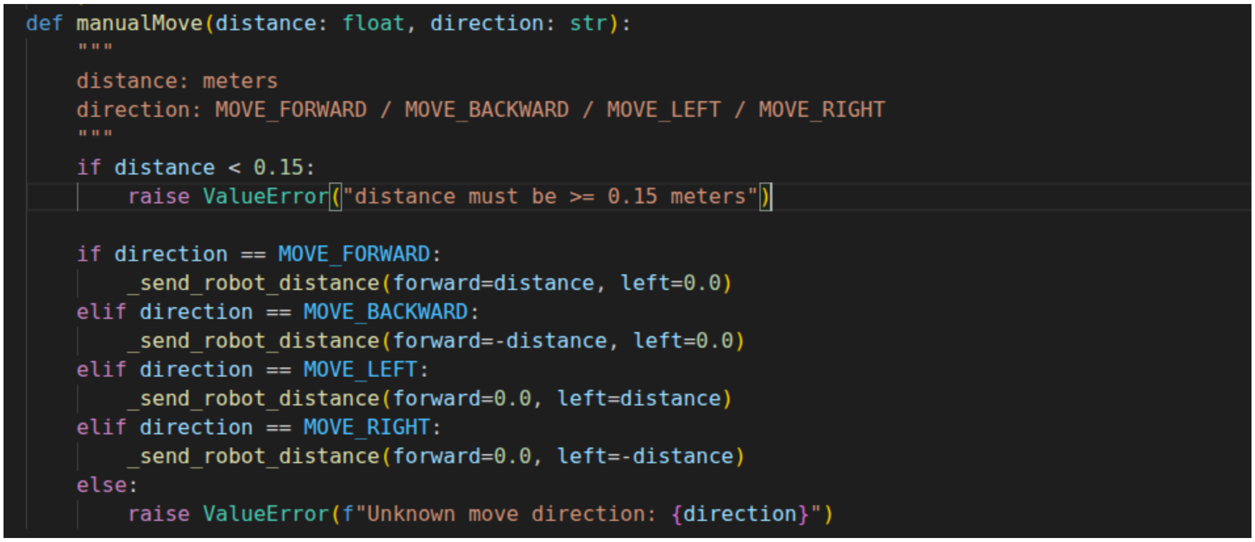


Function Move

Input: distance (m) and direction (MOVE_FORWARD, MOVE_BACKWARD)



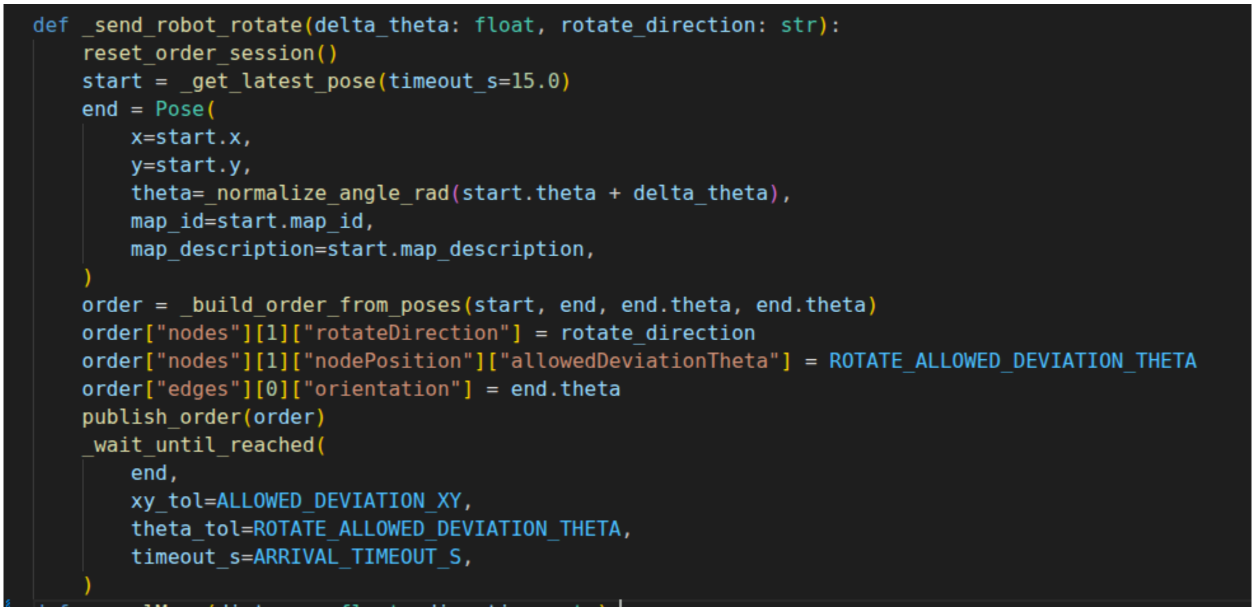
```python
def _send_robot_distance(forward: float, left: float):
    reset_order_session()
    start = _get_latest_pose(timeout_s=15.0)
    end = _compute_target_pose_robot(start, forward=forward, left=left)
    edge_orientation = math.pi if forward < 0.0 else 0.0
    end_theta_override = start.theta if (forward < 0.0 and left == 0.0) else None
    print("POSE", start.x, start.y, start.theta)
    print("TARGET", end.x, end.y, end.theta)
    publish_order(_build_order_from_poses(start, end, edge_orientation, end_theta_override))
    _wait_until_reached(
        end,
        xy_tol=ALLOWED_DEVIATION_XY,
        theta_tol=None,
        timeout_s=ARRIVAL_TIMEOUT_S,
    )
```



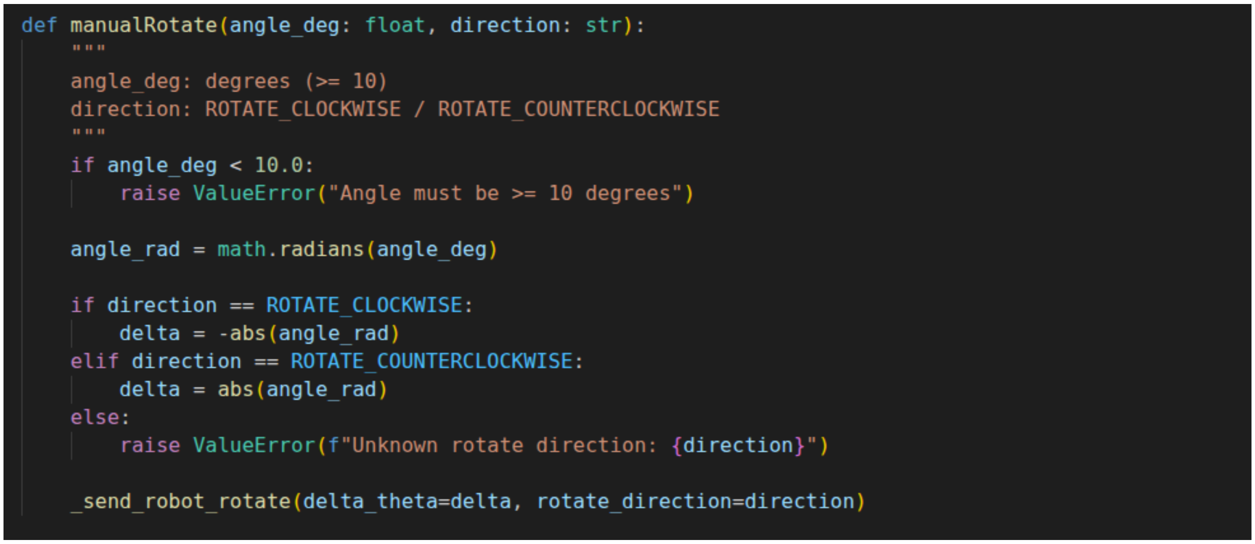
```python
def manualMove(distance: float, direction: str):
    """
    distance: meters
    direction: MOVE_FORWARD / MOVE_BACKWARD / MOVE_LEFT / MOVE_RIGHT
    """
    if distance < 0.15:
        raise ValueError("distance must be >= 0.15 meters")

    if direction == MOVE_FORWARD:
        _send_robot_distance(forward=distance, left=0.0)
    elif direction == MOVE_BACKWARD:
        _send_robot_distance(forward=-distance, left=0.0)
    elif direction == MOVE_LEFT:
        _send_robot_distance(forward=0.0, left=distance)
    elif direction == MOVE_RIGHT:
        _send_robot_distance(forward=0.0, left=-distance)
    else:
        raise ValueError(f"Unknown move direction: {direction}")
```

Function Rotation

Input: angle (degree) and direction (ROTATION_CLOCKWISE, ROTATION_COUNTERCLOCKWISE)

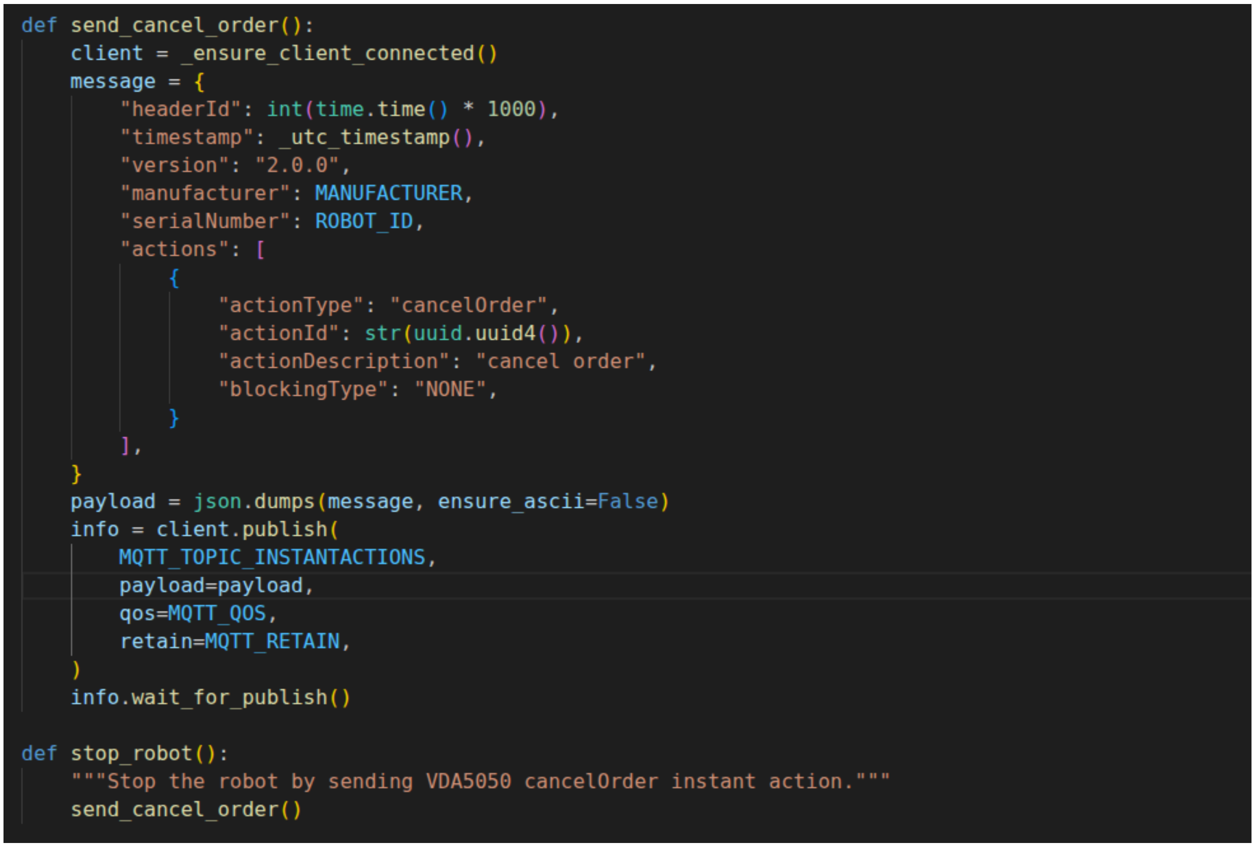


```python
def _send_robot_rotate(delta_theta: float, rotate_direction: str):
    reset_order_session()
    start = _get_latest_pose(timeout_s=15.0)
    end = Pose(
        x=start.x,
        y=start.y,
        theta=_normalize_angle_rad(start.theta + delta_theta),
        map_id=start.map_id,
        map_description=start.map_description,
    )
    order = _build_order_from_poses(start, end, end.theta, end.theta)
    order["nodes"][1]["rotateDirection"] = rotate_direction
    order["nodes"][1]["nodePosition"]["allowedDeviationTheta"] = ROTATE_ALLOWED_DEVIATION_THETA
    order["edges"][0]["orientation"] = end.theta
    publish_order(order)
    _wait_until_reached(
        end,
        xy_tol=ALLOWED_DEVIATION_XY,
        theta_tol=ROTATE_ALLOWED_DEVIATION_THETA,
        timeout_s=ARRIVAL_TIMEOUT_S,
    )
```



```python
def manualRotate(angle_deg: float, direction: str):
    """
    angle_deg: degrees (>= 10)
    direction: ROTATE_CLOCKWISE / ROTATE_COUNTERCLOCKWISE
    """
    if angle_deg < 10.0:
        raise ValueError("Angle must be >= 10 degrees")

    angle_rad = math.radians(angle_deg)

    if direction == ROTATE_CLOCKWISE:
        delta = -abs(angle_rad)
    elif direction == ROTATE_COUNTERCLOCKWISE:
        delta = abs(angle_rad)
    else:
        raise ValueError(f"Unknown rotate direction: {direction}")

    _send_robot_rotate(delta_theta=delta, rotate_direction=direction)
```

Function Cancel Order



```python
def send_cancel_order():
    client = _ensure_client_connected()
    message = {
        "headerId": int(time.time() * 1000),
        "timestamp": _utc_timestamp(),
        "version": "2.0.0",
        "manufacturer": MANUFACTURER,
        "serialNumber": ROBOT_ID,
        "actions": [
            {
                "actionType": "cancelOrder",
                "actionId": str(uuid.uuid4()),
                "actionDescription": "cancel order",
                "blockingType": "NONE",
            }
        ],
    }
    payload = json.dumps(message, ensure_ascii=False)
    info = client.publish(
        MQTT_TOPIC_INSTANTACTIONS,
        payload=payload,
        qos=MQTT_QOS,
        retain=MQTT_RETAIN,
    )
    info.wait_for_publish()

def stop_robot():
    """Stop the robot by sending VDA5050 cancelOrder instant action."""
    send_cancel_order()
```

- **Nhập giá trị distance (m) hoặc kéo thanh trượt và nhấn nút chọn direction di chuyển sau đó nhấn nút Send Move để di chuyển tiến/lùi.**

- **Nhập giá trị angle (độ) hoặc kéo thanh trượt và nhấn nút chọn direction xoay sau đó nhấn nút Send Rotation để xoay.**