Department of Computer Science & Engineering
The University of Texas at Arlington

Detailed Design Specification
CSE 4317: Senior Design II
Spring 2025

LJCJ
UR20 Palletizer

Luis Contreras
Joshua Dominguez
Christopher Gonzalez
Jasper Gustafson

Revision History

Revision	Date	Author(s)	Description
0.1	2.02.2025	JG, JD, LC, CG	Document creation
0.2	2.28.2025	JG, JD, LC, CG	Complete draft

Contents

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The UR20 palletizing system is designed to automate the process of stacking uniform-sized boxes on pallets. The system consists of multiple layers that interact through the Programmable Logic Controller (PLC), which manages the UR20 robotic arm, safety sensors, conveyor belt, and laser based object detection.

This document is built upon the System Requirements Specification (SRS) and Architectural Design Specification (ADS) to define the functional and structural design decisions. For a comprehensive understanding of the system's high-level requirements and architectural framework refer to the SRS and the ADS, which serve as the foundation for the detailed design decisions presented in this specification.

2 System Overview

The overall structure of the system is designed to seamlessly support the palletizing task utilizing the UR20 robot arm and PLC. At the core of the system is the Programmable Logic Controller (3PE Teaching Pendant), which orchestrates all operations by receiving inputs from various sensors and executing predefined routines for box handling. The data flow begins with the conveyor belt layer, transporting boxes to be distributed on the pallet, which are indicated as ready by the Photo Eye sensor that will relay information to the PLC to move onto the next part of the routine, given it detects a box incoming. The Safety Sensor Layer regulates movement detects human presence to dynamically adjust or halt movement as needed, ensuring a safe operating environment. The UR20 Arm Layer is responsible for picking up and placing boxes using a vacuum gripper and movement control system, all based on commands from the PLC.

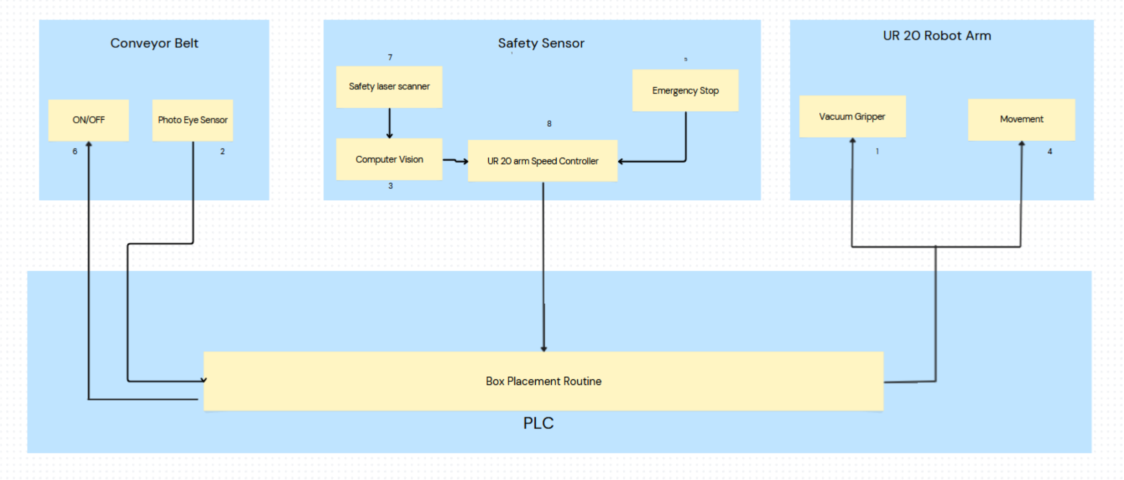

Figure 1: System architecture

3 UR 20 Robot Arm Layer Subsystems

The UR20 arm is the physical UR20 module. It can be controlled the a ’3PE Teach Pendant’ that allows touchscreen controls, or the Cobot URScript/Python URX library. This module covers the physical arm itself, the control mechanisms, and the gripper used for manipulating boxes.

3.1 Layer Hardware

The UR20 arm covers the UR20 arm, the control tablet, the control box, and gripper.

3.2 Layer Operating System

The Operating system used will be the UR PolycScope system that runs as the UR robots OS and programming interface.

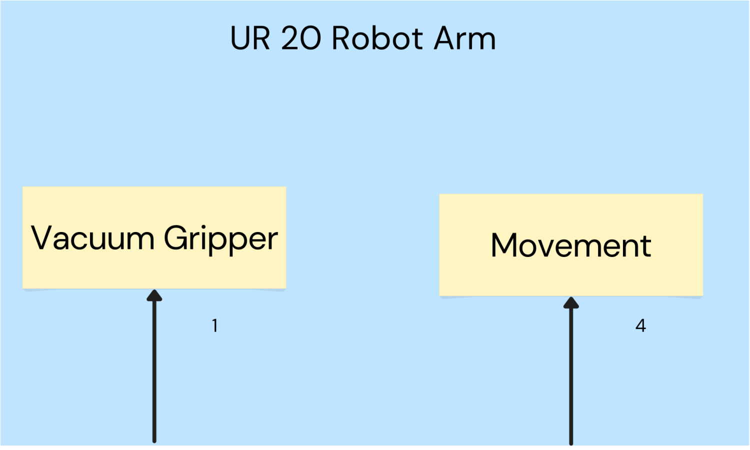

Figure 2: UR20 Arm subsystem

3.3 Vacuum Gripper

The Vacuum Gripper is tasked with the carrying the the incoming packages from the conveyor belt onto and releasing the package at the appropriate location on the pallet.

3.3.1 Gripper Software Dependencies

The Gripper will be dependent on the the voltage assignment of port (TBA) from the PLC to release and hold suction. Controlled through URScript/Python (URX library)

3.3.2 Vacuum Hardware

The hardware used in the gripper is the gripper itself as well as the vacuum. The gripper includes the use of an aluminum base plate, suction cups and air tube systems that connects to the vacuum. The vacuum will connect to a control box port for control.

3.3.3 Vacuum Programming Languages

The way to control the UR20 gripper will come from the URScript/Python

3.4 Arm Movement

The movement subsection of the Layer is input of the current arm location and movement of the arm to is next position whether to be in position place a box in its appropriate position or to return to the conveyor belt and await its next package.

3.4.1 Arm Hardware

As for the the hardware it would be the connection of the arm to its control box and control tablet access location and adjust arm position.

3.4.2 Arm Programming Languages

The way to control the UR20 gripper will come from the URScript/Python

4 Safety Sensor Layer Subsystems

The safety subsystem determines whether it is safe to continue operation (unsafe conditions are when a human is detected too close to the operation area or the emergency stop is triggered). All operations should halt when unsafe conditions are detected.

4.0.1 Safety Sensor Programming Languages

Python/URScript

4.0.2 Safety Sensor Software Dependencies

Python & URX Python library

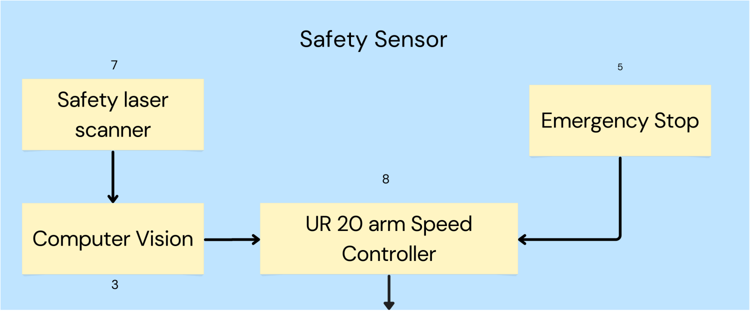

Figure 3: Safety sensor diagram

4.1 Proximity Sensor Subsystem

Detects whether a person has entered the operating area of the UR20.

4.1.1 Proximity Sensor Hardware

An IR proximity sensor mounted at the base of the robot arm, opposite the conveyor belt, detects the position and distance of any objects.

4.2 Computer Vision Subsystem

The computer vision system detects if a human enters the operational area of the UR20 using RaspberryPi with a camera module and machine learning interface.

4.2.1 Computer Vision Data Structures

Input: consists of Raw image frames captured by a Raspberry Pi camera module. Output: Classification confidence score indicating the probability of a human in dead zone of UR20 proximity sensor.

4.2.2 Computer Vision Data Processing

Captured frames are passed to a TensorFlow Lite model to detect humans in the frame. If the model outputs a score higher than a select threshold (0.63 in our case found through fine tuning), additional bounding box checking ensures that the detected human is within a critical section of the camera's view. If more than one person is in the dead zone, a GPIO pin is set to high, activating an emergency stop condition on the UR20.

4.2.3 Computer Vision Hardware

This diagram is of the circuit that sends the halt signal from the Raspberry Pi to the UR20. When the input from the Pi is low, the output of the circuit is 24V, meaning the logic is inverted.

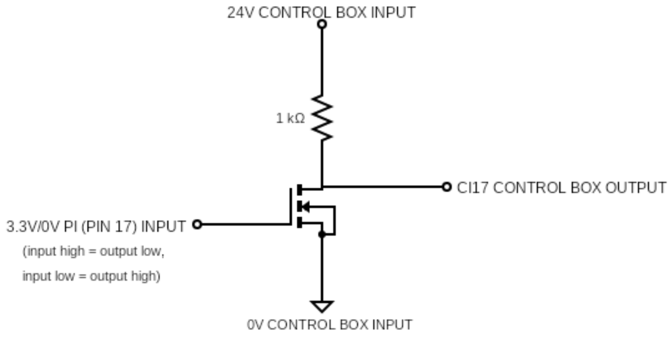

Figure 4: Pi to Control Box wiring

4.3 Speed Control Subsystem

Determines the safest operational speed to perform at.

4.3.1 Speed Control Data Structures

Receives estimated distance of the nearest person from the Computer Vision subsystem.

4.3.2 Speed Control Data Processing

If a person is close but not within operational range, slow movement. If a person is within operational range, stop movement. Otherwise, operate at normal speed UNLESS the emergency stop is triggered, at which case ALL operation will be stopped, and if possible, manual repositioning mode will be enabled (where the arm can be moved by applying pressure).

4.4 Emergency Stop Subsystem

Triggers a complete stop of all operation.

4.4.1 Proximity Sensor Hardware

This is a physical button located on the 3PE Teach Pendant.

4.4.2 Proximity Sensor Data Structures

This is integrated fully with the UR20's basic control system. A single signal will be sent to all modules that indicates whether the UR20 is controllable or in stop mode. In stop mode, no operation will be possible in any module.

5 Conveyor Belt Layer Subsystems

The conveyor belt layer has two tasks: feed a consistent stream of boxes to the arm, and comply with safety standards. This layer has two subsystems: The ON/OFF controller, which receives signals from the PLC to start or stop the conveyor belt, and the 'Box Ready' Photo Eyes signaling system, which uses photo eyes to detect when another box is ready to be picked up.

5.1 Conveyor Belt Hardware

A conveyor belt provided by the Senior Design labs.

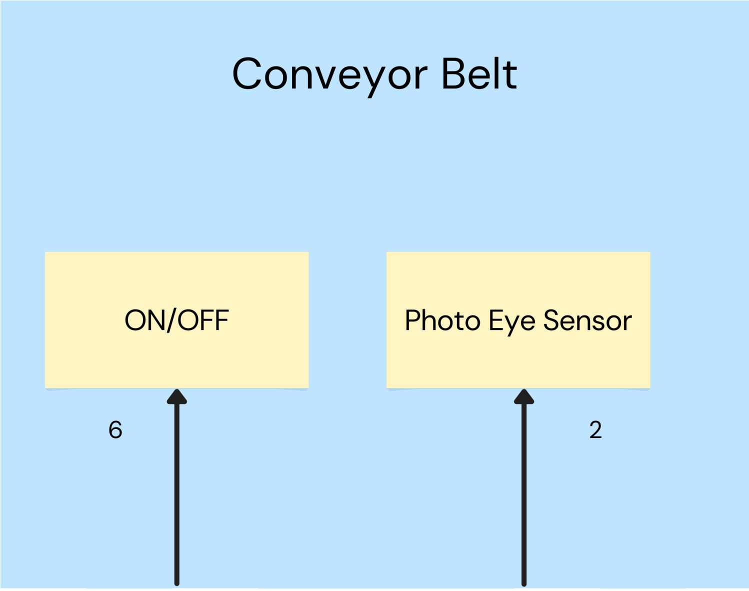

Figure 5: Conveyor belt diagram

5.2 Power Controller Subsystem

The power controller hardware subsystem in the conveyor belt layer is responsible for controlling the state of the conveyor belt. The subsystem receives a command from the PLC to switch the belt "ON" or "OFF" when a box moves from or into the correct position, OR when the safety system indicates an emergency stop.

5.2.1 Power Controller Hardware

A power supply toggle for shut-off, wired to the robot arm to be controlled by the PLA.

5.2.2 Power Controller Data Structures

This is a binary on/off toggle of the conveyor belt, so a single binary bit should be all that is necessary.

5.3 Photo Eye Subsystem

The photo eye is a simple hardware module positioned at the end of the conveyor belt to indicate that a box has passed and is ready to pick up.

5.3.1 Photo Eye Hardware

Wiring between the Photo Eye and robot arm inputs.

5.3.2 Photo Eye Data Structures

The most common photo eye design signals use a single bit, active when presence is initially detected, and otherwise the bit signals as inactive.

6 PLC Layer Subsystems

The 3PE Teaching Pendants act as the PLC (Programmable Logic Controller) for this system. The PLC is the central control unit responsible for executing the palletizing process in a predefined sequence. It receives inputs from various sensors and controls the UR20 arm, conveyor belt, and gripper systems. The PLC is programmed to execute a routine to efficiently manage the automation cycle.

6.1 Layer Hardware

The PLC layer utilizes the same hardware as the subsystem listed below.

6.2 Layer Operating System

The PLC utilizes a Real-Time Operating System to ensure timely and deterministic control of hardware components.

6.3 Box Placement Routine

The Box Placement Routine governs the entire palletizing sequence, ensuring efficient box handling. It takes input from the safety and photo eye sensor, processes the data, and outputs commands to control the conveyor belt, vacuum gripper, and UR20 robot arm.

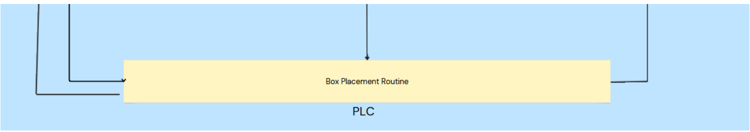

Figure 6: Box Placement Routine diagram

6.3.1 Subsystem Hardware

The Box Placement Routine subsystem utilizes the hardware from each layer as follows:

- Photo Eye Sensor: Provides input to detect boxes on the conveyor belt
- Safety Laser Scanner: Provides input to ensure safe operation
- Conveyor Belt Motor: Controlled by the PLC to turn ON/OFF
- Vacuum Gripper: Controlled by the PLC to pick and release boxes.
- UR20 Robot Arm: Moves to predefined positions for picking and placing boxes

6.3.2 Subsystem Operating System

PolyScope is the software included on the 3PE Teach Pendant, utilized in setting the predefined arm movements.

6.3.3 Subsystem Software Dependencies

URX Python library

6.3.4 Subsystem Programming Languages

URSCript and Python are used to control the PLC.

6.3.5 Subsystem Data Structures

- Sensor Data Packets:
  - Photo Eye Sensor: Binary Bit (box detected or not)
  - Safety Scanner: Binary Bit (object in range or not)
- Robot Position Data: Predefined coordinates and angles for picking and placing boxes.
- Conveyor Belt Status: Binary Bit (ON/OFF)
- Vacuum Gripper Status: Binary Bit (Activated/Deactivated)

6.3.6 Subsystem Data Processing

The Box Placement Routine follows a state-driven algorithm, executing the following logic:

1. IF Phot Eye Sensor detects a box –> STOP conveyor belt.
2. IF safety laser scanner detects object in range of robot –> Adjust UR20 movement speed/ Halt all movement
3. Activate vacuum gripper and move UR20 to box position
4. Move to predefined pallet position
5. Release vacuum gripper and return UR20 to standby position
6. Turn conveyor belt on
7. Repeat until palletizing sequence is complete

7 Appendix A

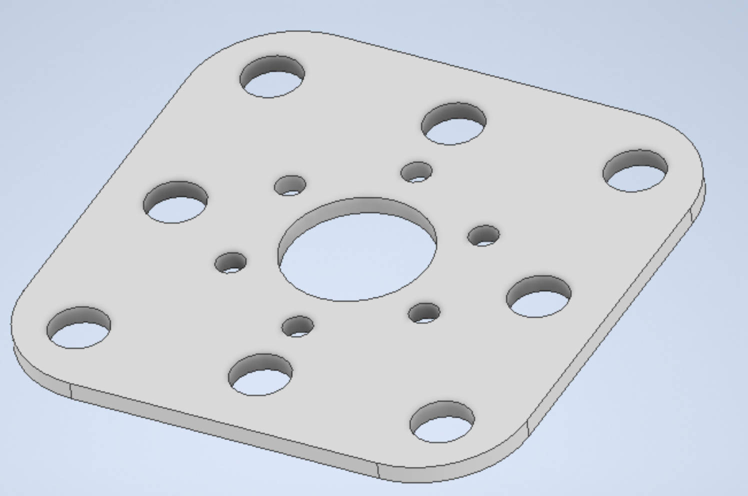

Figure 7: UR20 Gripper Base Plate Design

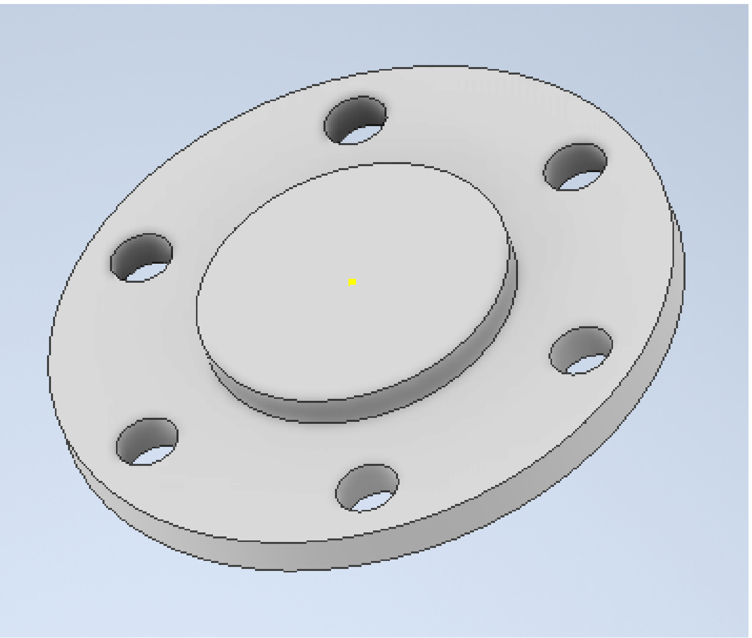

Figure 8: UR20 Gripper Base Plate Adapter Design