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

Architectural Design Specification
CSE 4316: Senior Design I
Fall 2024

LJCJ
UR20

Luis Contreras
Joushua Dominguez
Christopher Gonzalez
Jasper Gustafson

Revision History

Revision	Date	Author(s)	Description
0.1	11.11.2024	LC, JD, CG, JG	Document creation

CONTENTS

List of Figures

List of Tables

1 Introduction

This project's goal is to program a collaborative palletizing robot intended for industrial use. The main component of the robot will be the PLC (programmable logic controller) designed for use with the UR20 cobot (collaborative robot) arm, and the secondary components will be the UR20 arm itself, a safety system, a conveyor belt. Each component will be an individual layer, described further later in this document. The intended product of the project will be a robot that can identify the location and position of boxes on a conveyor belt and stack them optimally, while being safe for humans to work around. The focus is the arm, but other requirements in the intended working environment (such as a conveyor belt and pallets) will be provided for testing.

The main component, the PLC, will be the control module for the entire system. All other components report back to and are controlled by the PLC. The second most important component is the UR20 arm, which is designed to work with the PLC controller. The arm will manipulate and palletize the boxes in response to the commands given by the PLC.

The PLC will make decisions based on the safety sensor system. The safety sensor system will detect human presence and inform the PLC whether it is safe to operate, ideally slowing down the movement of the UR20 arm when a human is nearby.

The conveyor belt is an external system. It constantly feed boxes to the UR20 arm, with the intention of mimicking an industrial environment. Ideally, it will be connected to the PLC's start/stop and safety system instead of having its own external controls, but the conveyor belt is only barely inside of the scope of this project, as it is likely that anyone installing this arm would have their own conveyor belt set up. Therefore, the conveyor belt is a low priority for this project.

2 System Overview

The overall structure of the software system is designed to seamlessly support the palletizing tasks of the UR20 robot arm system. This layered design enables modular functionality, enhancing system adapability and ensuring each layer performs a unique function while simultaneously providing crucial information to the main PLC layer which manages data distribution. As the central interface, the PLC efficiently processes the flow of sata from each subsystem layer, allowing for smooth system integration and effective coordination across all layers. This architecture promotes reliable, dynamic operation, optimizing the UR20's performance in real-time palletizing tasks.

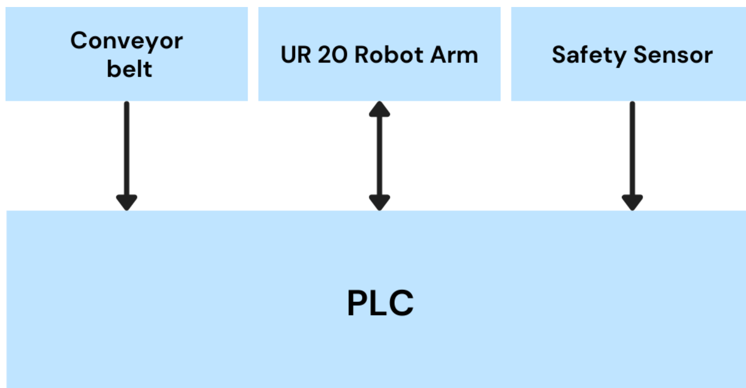

Figure 1: UR20 Architectural design diagram

2.1 Layer 1: PLC Description

The PLC will process most of the data through its peripherals, as seen in Figure 1. It is the primary interface between the different layers of the project. The built-in PLC will be configured with URScript, a Python-based scripting language in which vital functions will reside, such as the input function, the box offset algorithm, and finally, a position algorithm. The data flow will start with the input given by the Photo Eye scanner or the safety sensor, and the input functions will follow these two data paths. First being, an input given by the photo eye scanner will pass through the input function, which will then call the box offset algorithm to determine where the box is in 3D space given the location, which will go to the position algorithm and determine the motion necessary for the UR20 to satisfy this request. The safety algorithm will give the second path, which will trigger the input function and call the position algorithm to safely slow down the speed at which the UR20 is palletizing to create a safe work area for a cooperative application.

2.2 Layer 2: Safety Sensor

The Safety Sensor will determine if a human is in the area of the UR20 arm. If so, it will reduce the speed of the UR20 in order to create a safe work environment. This will be achieved with the use of a camera that will process the data in real-time with the use of computer vision, which will send a signal to the PLC which in turn will send a signal to the UR20 movement subsystem in order to maintain a safe speed for collaborative work.

2.3 Layer 3: UR20 Robot Arm

UR20 Arm consists of a vacuum gripper, the gripper controller, and the movement of the arm. The arm is the physical output of the software that resides in the PLC layer. Additionally, it contains a gripper grab/release controller (the controller for the air compressor), which will be turned on or off when needed to hold or drop a box. The commands given by the PLC will determine the position and orientation needed for the UR20 to place the box correctly.

2.4 Layer 5: Conveyor Belt

The conveyor belt system moves over pulleys driven by a motor. As the motor rotates, it propels the belt forward, allowing items placed on the belt to be conveyed along its length. The belt's speed can be adjusted to control the pace of movement. However, for this implementation, it will have two states: on or off. At the end of the conveyor belt, there will be a guide to reorient the boxes and a bracket in which the Photo Eye scanner will be placed.

3 Subsystem Definitions & Data Flow

- **Conveyor Belt Layer**
  - ON/OFF Controller
- **Safety Sensor Layer**
  - Computer Vision
  - UR20 Arm Speed Controller
- **UR20 Robot Arm Layer**
  - Vacuum Gripper
  - Grab Release Control
  - Movement
- **PLC Layer**
  - Input Function
  - Box Offset Algorithm
  - Position Algorithm

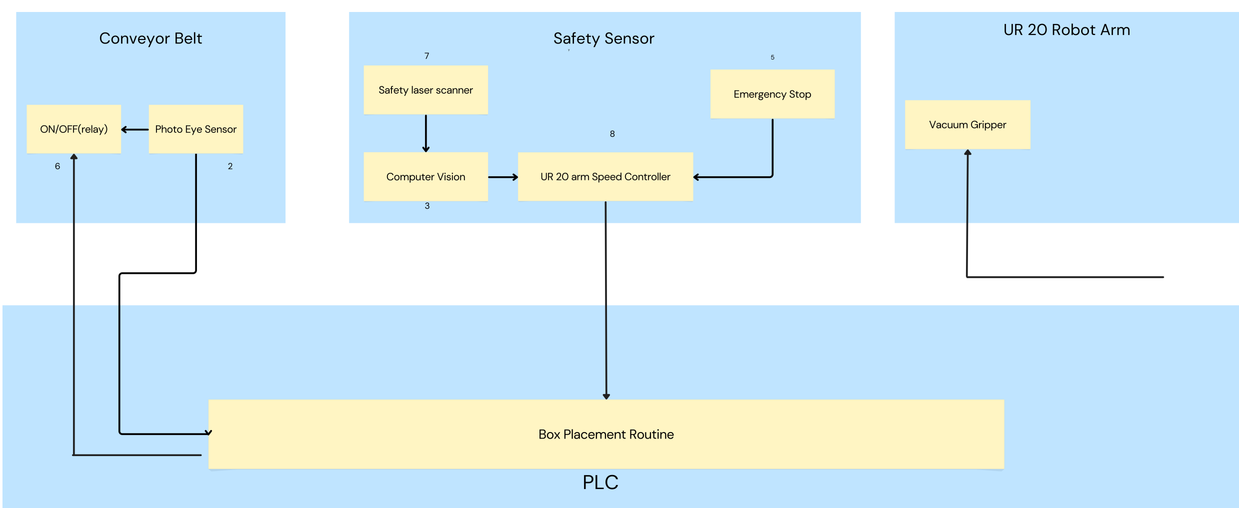

Figure 2: UR20 Data flow diagram

4 PLC Layer Subsystems

The UR20 has a system to practically create a single interface between the robot and PLC, in order to enhance the capabilities and functions of the cobot

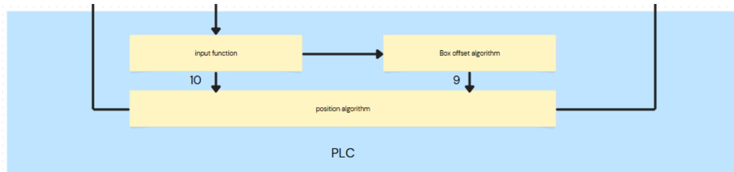

Figure 3: PLC

4.1 Input function

The input function would be the readings of the environment knowing the positions of the pallet and whehere there is a box to pickup based on the photo eye.

4.1.1 Assumptions

Assumptions for the input function is that the orientation of the box is uniform such that it will be picked up the same way.

4.1.2 Responsibilities

The responsibitilies of the input fuction is soley to gather data that the PLC placement algorithm will use to preform its next action. Such as the current state of the pallet, the location of any other nearby objects or people for safety concerns.

4.1.3 Subsystem Interfaces

Table 2: Input fuction interfaces

ID	Description	Inputs	Outputs
#01	movement readings	destination position speeds safety sensor gripper power	formated data for PLC algorithm
#02	box detection	Photo Eye State position speeds safety sensor gripper power	formated data for box offset algorithm

4.2 Box offset algorithm

The box offset algorithm will be used to scan and determine the placement of the box

4.2.1 Assumptions

The boxes have uniform size and will fit on the pallet with a pre-defined pattern.

4.2.2 Responsibilities

The responsibitilies of the placement function will be used to determine location of the box in the x,y,z dimentional plane and send it to the placement algorithm

4.2.3 Subsystem Interfaces

Table 3: Box Offset algorithm interfaces

ID	Description	Inputs	Outputs
#03	movement	Photo Eye State position speeds safety sensor gripper power	formated data for PLC algorithm

4.3 Placement algorithm

The placement algorim is the central control of operations will tell the subsystems of the UR20 what actions to take given the output of the input function

4.3.1 Assumptions

The placement algorithm assumes that the gripper will have fixed strenght while moving a box and will be able to place without droping it. It expects belt and pallet location to be fixed and box orientation constant

4.3.2 Responsibilities

The responsibitilies of the placement function will be prefom send the appropriate values to the motors, in order to move arm to pick up the box then place it on pallet, and vacuum gripper to pick up and release the boxes. The algorithm will also be checking for safety based on the sensor to scan for people and stop or slow down movement.

4.3.3 Subsystem Interfaces

Table 4: Placement algorithm interfaces

ID	Description	Inputs	Outputs
#04	movement	Photo Eye State destination position speeds safety reading	adjusted motor spreeds
#05	gripper power	destination postion	ON/OFF

5 Safety Sensor Layer Subsystems

In this section, the layer is described in some detail in terms of its specific subsystems. Describe each of the layers and its subsystems in a separate chapter/major subsection of this document. The content of each subsystem description should be similar. Include in this section any special considerations and/or trade-offs considered for the approach you have chosen.

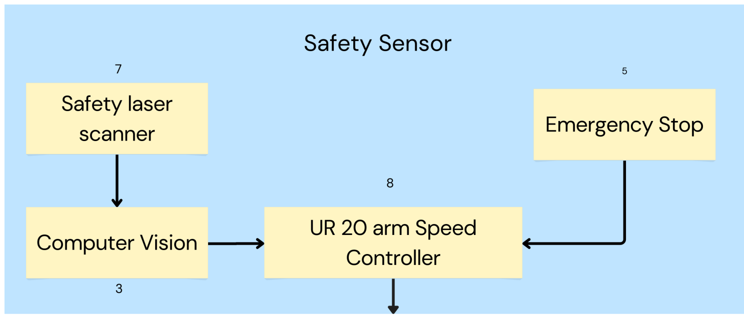

Figure 4: Example subsystem description diagram

5.1 UR 20 arm speed controller

The controller will determine whether there is a person within the working vicinity of the UR20 robot and signal the PLC to adjust for the case

5.1.1 Assumptions

The UR20 should have collision detection and should stop at impact

5.1.2 Responsibilities

To signal the PLC the presence of a person near the robot using a computer vision algorithm

5.1.3 Subsystem Interfaces

Pin 17 on the Pi is connected to CI17 on the Control Box, sending a halt signal when a person is detected.

Table 5: Speed Controller Subsystem Interface

ID	Description	Inputs	Outputs
#06	send signal	presence detection	presence signal

5.2 Computer vision algorithm

A computer vision algorithm to scan for the presence of the shape of a person being near the UR20 robot

5.2.1 Assumptions

The camera will have a vantage point as to be able to see all around the robot

5.2.2 Responsibilities

Inform controller of detected pressence near the UR20

5.2.3 SUBSYSTEM INTERFACES

Table 6: Computer Vision Subsystem Interface

ID	Description	Inputs	Outputs
#07	presence detection	camera reading	presence detection

6 UR 20 Arm Layer Subsystems

The UR20 arm is a layer controlled by the PLC used exclusively for manipulation. It is a collaborative robot arm that will be used to grab the boxes and position them upon the pallet. It has three subsystems: the movement interface, the vacuum gripper, and the vacuum gripper controller.

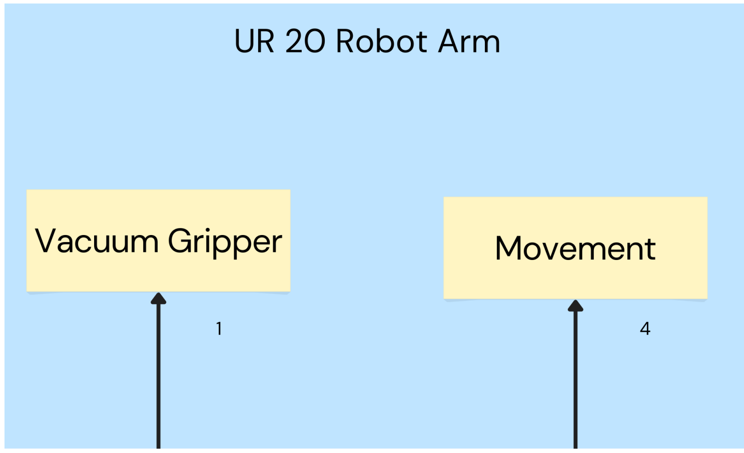

Figure 5: UR20 Robot Arm Layer

6.1 Movement

This subsystem of the UR20 arm is technically an interface within the PLC, but as the PLC and UR20 are married so closely together, and this is more closely related to the arm itself, it is listed as a subsystem of the arm. The interface will allow easy control of the arm's movements. Its input is data from the PLC's decision making algorithm and its output is to directly control the robot's movements.

6.1.1 Assumptions

The assumption being made is that the UR20 arm is going to exclusively be a tool of the PLC, and thus will only interface with the PLC. The movement interface may even be built into the PLC for this robot, as it seems to be with some other UR cobots, but as of right now it is being assumed that a control system will have to be programmed for it on the PLC.

6.1.2 Responsibilities

The responsibility of the movement subsystem is to translate the box location, current arm location, and current desired speed into a coherent set of movements for the UR20 arm. These factors will be combined to create a set of movements and movement speed that will move the arm to the desired location.Ã

6.1.3 Movement Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here.

Table 7: Movement interface

ID	Description	Inputs	Outputs
#08	Change in X,Y,Z Coordinates	X Y Z	NO OUTPUT
#09	Movement Speed	Speed	NO OUTPUT
#10	Rotation Angles	Perpendicular Longitudinal	NO OUTPUT

6.2 Vacuum Gripper

The vacuum gripper is physical hardware and its software inputs and outputs are covered in the Vacuum Gripper Controller subsystem. The vacuum gripper is connected to a vacuum generator that will be adjusted to provide a static amount of force capable of picking the boxes and not dropping them until the controller determines that it should do so.

6.2.1 Assumptions

The vacuum gripper will be a hardware-only system with a constant vacuum that can be blocked in order to drop a box. It will interface exclusively with the vacuum gripper controller and have no outputs.

6.2.2 Responsibilities

The responsibility of the vacuum gripper is to securely hold and transport boxes. It will be able to drop the boxes on command and in the process it will shift the boxes as little as possible in order to produce the expected placement.

6.2.3 Movement Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here.

Table 8: Movement interface

ID	Description	Inputs	Outputs
#11	Vacuum Control Signal	ONOFF	NO OUTPUT

6.3 Vacuum Gripper Controller

The vacuum gripper controller is the software interface between the PLC and the vacuum gripper. It signals the vacuum hardware to kill the vacuum and drop the box being held.

6.3.1 Assumptions

The vacuum gripper controller will be on at all times EXCEPT when it is told that a box needs to be dropped.

6.3.2 Responsibilities

The vacuum gripper controller must turn off the vacuum long enough for the cobot to move away, which should occur soon after. If it turns on again too soon, the box will be pulled out of the desired position.

6.3.3 MOVEMENT SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here.

Table 9: Movement interface

ID	Description	Inputs	Outputs
#12	Drop Signal	Trigger	ONOFF
#13	Current Speed	Speed	NO OUTPUT

7 Conveyor Belt Layer Subsystems

The conveyor belt layer is managed by the PLC and is solely responsible for controlling the conveyor belts movement. This layer has one subsystem: The ON/OFF controller, which recieves signals from the PLC to start or stop the conveyor belt based on when the UR20 robot arm is ready to receive another box. The control of the conveyor belt is simplified to an ON/OFF mechanism, rather than dyanmically adjusting speed based on palletizing rate.

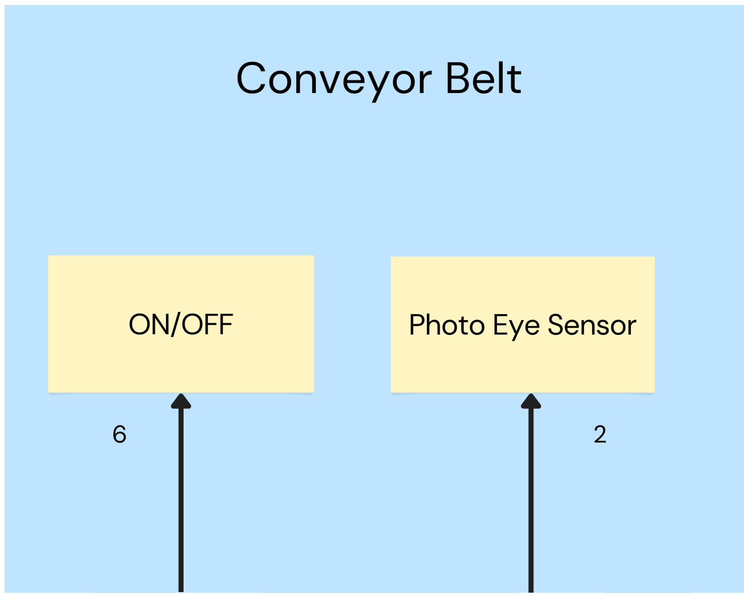

Figure 6: Conveyor Belt Power Controller

7.1 Power Controller Subsystem

The power contoller 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" when the robot is prepared to recieve a new box and "OFF" once the box has been processed, ensuring smooth and controlled movement of boxes.

7.1.1 Assumptions

The assumption of a required automatic speed control is not required. The belts state change is assumed to be sufficient for the palletizing application, without the need to detect the exact positions of boxes on the belt. Only ensuring that the boxed move in a controlled manner when ready.

7.1.2 Responsibilities

The power controllers main responsibility is to maintain the stability of the conveyor, preventing any from advancing too far or falling off during the palletizing process. By managing the conveyor's ON/OFF state, it ensures that boxes are presented to the robot in a timely and safe manner.

7.1.3 Subsystem Interfaces

The ON/OFF signal from the PLC serves as the sole input for the power controller. When the PLC sends an "ON" signal, the conveyor belt starts moving and will stop movement when the "OFF" signal is received.

Table 10: Controller Interface

ID	Description	Inputs	Outputs
#16	Power Controller signal	ON/OFF	NO OUTPUT

REFERENCES