



**Z. Kootbally/C. Schlenoff**  
University of Maryland  
College Park, MD

## Final Project

---

ENPM809B : Spring 2020  
Due **Wednesday, April 29, 2020**

### Contents

<b>Introduction</b>	<b>2</b>
<b>Config File</b>	<b>2</b>
<b>Instructions</b>	<b>3</b>
<b>Grading Rubric</b>	<b>3</b>

## Introduction

The final project consists of writing C++ code which can handle any challenges and situations discussed in class for ARIAC 2019.

Your code will be tested against an unseen config file which consists of a combination of:

- Drop part.
- Updated order.
- Faulty parts.
- Sensor blackout.
- Flipped part.
- Multiple shipments of the same order.

You will need to reuse and combine all your ROS assignments to handle the config file used in the final project.

## Config File

To help you get ready for the final project I have created a yaml file: [final\\_project\\_a.yaml](#) (uploaded on Canvas). However, your submitted code will be tested using [final\\_project\\_b.yaml](#), which you will not have access until after the final project is submitted. This is the same principle used in RWA-4.

Here is a summary of [final\\_project\\_a.yaml](#):

- Two shipments of the same order must be submitted.
  - **Opposite AGVs:** The shipments must be delivered on opposite AGVs.
- **Flipping Part:** The order consists of flipping one pulley. In total you will need to flip two pulleys (one per shipment).
- **Faulty Part:** There are faulty parts in the environment. They should not be part of the shipments. Reuse code from RWA-4 to deal with faulty parts.
- **Drop Part:** Parts will drop above the AGVs from the gripper when trying to place the parts in trays. One part will be dropped above each AGV. Dropped parts will still end up in the tray but in the wrong pose. You will need to reposition those parts.
- **Sensor Blackout:** During a certain period of time all sensors will stop reporting data. You can wait until the sensors are working again before continuing, which leads to downtime and must be avoided, or you can continue building kits during the blackout (recommended).
- **Updated Order:** There is no updated order in this file but [final\\_project\\_b.yaml](#) may contain one. You need to anticipate this challenge in the code you submit. See RWA-5 instructions on this challenge.
- **Time Limit:** You have to build both kits and submit them before a time limit.
- **Part availability:** Some parts are only available on the conveyor belt as they cannot be found in the bins. After reading an order, check if all the parts from the order can be found in bins. If some parts are not in the bins then they will be spawn on the conveyor belt.

## Instructions

There are no specific instructions for the final project. As mentioned previously, your code should be able to handle all the challenges thrown at you. If your code did well in the previous assignments and with [final\\_project\\_a.yaml](#) then you are ready for [final\\_project\\_b.yaml](#). Remember that challenges not occurring in [final\\_project\\_a.yaml](#) will probably occur in [final\\_project\\_b.yaml](#), so be prepared.

## Grading Rubric

Grading your code will be based on a combination of:

1. Your competition final score. We will use a threshold and award points based on this threshold.
2. The approach you took to handle some challenges. We would like your system to be agile based on the challenges and as such you must make smart decisions. We will watch videos generated from the simulation and we will judge your approaches.
  - I have included 2 video files from ARIAC 2019 finals. You can see how the participants approached some challenges to give you an idea.