FINAL PROJECT – TASK BREAKDOWN:

There are three distinct parts to this project;

A) Design of the **Electrical System**, and
B) Creation of the **Ladder-Logic Program** for the PLC to automate the system,
C) Writing the **Final Report**.

Task A – Electrical Design

Two electrical diagrams are required for the project; a schematic diagram and a wiring diagram.

Both diagrams must show all of the electrical (power and control) devices utilized by the system including both the primary devices specified in this document and any auxiliary devices (contactors, overload relays, power supplies, etc.) that are required to build an operational system. Additionally, both diagrams must be electronically-generated, logically organized, and completely labeled.

Access to the software/PLCs in the lab is not required for the design of the electrical system.

1) Create the Schematic Diagram – Due Tuesday by July 9th

The Schematic Diagram is used to depict the electrical system from an operational standpoint. Components should be logically laid out and grouped together based upon their function. For example, the "power" components in the system, such as the electric motors and any other directly-connected components (overload relay heaters, main contacts, etc.) should be grouped together in the same area of the diagram.

Additionally, each component should be clearly labeled with a unique identifier, such as "IM-1" for the induction motor that drives conveyor #1, or "OL-3" for the components (heaters, contact) that are contained in the overload relay that protects induction motor #3.

Although the final schematic diagram must be electronically-generated, it is generally easier to begin the design process with a neat, hand-drawn diagram. Initially sketch out the system, being sure to include all of the required components. Then re-organize the drawing so that the components are logically grouped and optimally arranged within the diagram. Once you are satisfied with the schematic diagram, then begin creating the electronic version.

ELECTRIC SYSTEMS REVIEW – The instructor will be available during the lab sessions (after lecture) on July 2nd in order to briefly review and critique the schematic diagrams.

Note that a complete, hand-drawn diagram is sufficient for the "Electrical Systems Review".

2) Create the Equipment Schedule – Also Due by July 9th

This can either be completed simultaneously with the schematic diagram or after the schematic diagram has been completed.

The equipment schedule is basically an organized and tabulated listing of all of the components utilized within the electrical system. A different row of the table should be utilized for each unique component, beginning with the component's unique identifier(s) followed by the required number of that component. Additionally, the name of the component and a description of the component (ratings, etc.) should be provided, along with any other important information.

The equipment schedule should have all of the information that would be required in order to purchase all of the system components.

The electronically-generated "finalized" versions of both the **Schematic Diagram** and the **Equipment Schedule** are due by **5:00pm** on the **9th** of **July** (Tuesday).

3) Create the Wiring Diagram – This should be started after the schematic diagram is complete.

The Wiring Diagram is used to depict the electrical system from an installation standpoint. Thus, the wiring diagram should contain all of the information that would be required by an electrician in order to install and wire all of the system components.

Wiring diagrams are fundamentally different from schematic diagrams, and can be more difficult to create due to the required grouping of the system's components and the requirement that all wires must begin an end at either one of the terminals of a system device or at a terminal strip contained within a panel or junction box.

All of the components contained within a single device should be grouped together within a block designated for that device, and each terminal of the device should be assigned a unique identifier. For example, a block should be utilized to depict the contactor that is used to (de)energize induction motor #2, and all of the contactors components (field coil, NO contacts) should be contained within that block.

Additionally, if multiple devices would be housed in the same panel, then they should be similarly grouped within the wiring diagram.

Wires that interconnect panels and/or external devices should be also be grouped together if they would be contained in a shared raceway.

The "finalized" **Wiring Diagram** is a required component of the **Final Report** that is due by **8pm** on the **24th** of **July** (Thursday).

Task B – Programming the PLC

You must create the Ladder-Logic Program that will be executed by the PLC in order to automate the system as specified within the project handouts.

The program should be well-organized, and rung comments should be placed through-out the diagram to help clarify the program's operation. Additionally, logical tag names must be chosen for the various ladder instructions based upon their function or purpose and tag descriptions should be included where necessary to help clarify the function or purpose of the instructions.

The final version of the **Ladder-Logic Program** that is required for proper operation of the system must be completed and have its operation verified, on the scale-model system within the Q-215 lab by the instructor, by no later than **5pm** on the **23rd** of **July** (Tuesday).

1) Startup – Begin by creating all of the logic for **System Startup**.

Note that access to the software/PLCs in the lab is not required for the initial stage of the ladder-logic program's creation. Although having access to the software/PLCs in the laboratory becomes important when you want to begin testing and troubleshooting your code (ladder), all that is needed to begin programming is a pencil and some paper.

To be honest, sitting in front of a computer and creating your ladder from scratch can actually be more difficult than trying to create your initial ladder while in the comfort of your home. Why? Because the initial logic of the task is what's most important, and it's easy to lose track of that while simultaneously having to manage the intricacies of the RSLogix software.

First, sketch on the proposed ladder on paper, being sure to choose logical Tag names for the instructions. Then, organize and logically group the required rungs for "System Startup". Once you are satisfied with the initial ladder, then begin entering the program into the RSLogix 5000 software.

Verify proper operation of all of the system components during system startup.

Make sure that all "status bits" are properly initialized such that the system functions properly both immediately after the program is downloaded into the controller and whenever the controlled is switched from run-mode to program-mode and then back to run-mode.

Once system startup is functioning properly, review both the layout of the ladder and the tag conventions initially chosen, and make any modifications required to help facilitate the completion of the rest of the program.

2) Normal Operation – Once System Startup has been verified, then begin laying out the logic for **Normal Operation**.

Completing the logic required for Normal Operation will be the most time-intensive part of the programming phase of this project.

Although the "Startup" portion of the code can easily be completed in real-time which sitting at a workstation, careful planning will be required in order to complete Normal Operation.

You may find it a lot easier to create a flowchart that describes the system's operation before you begin programming.

Determine the manner in which you will complete each of the required tasks, and identify and status-bits or other variables that will be necessary to perform and monitor those tasks. For example – how will you specify the next crate type that is being passed to the sorting gate?

Do not worry about the rest of the logic that will be required to control the system during the other phases of the system's operation until both Startup and Normal Operation have been verified.

3) System Shutdown – Begin working on **System Shutdown** after System Startup and Normal Operation are both functioning properly.

4) Once **Startup**, **Normal Operation**, and **System Shutdown** are complete and their operation is verified, then begin on the remaining task; **Emergency Stop**.

5) When verifying the operation of your program, check to make sure that improper button presses will not cause the system to malfunction. For example, pressing the "Startup" button should have no effect once the system is already in "Normal Operation".