MCMASTER UNIVERSITY
BACHELOR OF TECHNOLOGY
AUTOMATION ENGINEERING TECHNOLOGY

# LAB PROJECT EXPERIMENT

## **TITLE:** Industrial Network Communications

**Professor:** Dr. Tom Wanyama, P.Eng

**Lab Professor:** Mr. Alexander Barsan

**Submitted by:** **Hussein Al-Ansari** 400395232 **Adam Al-Jabari** 400399330 **Ahmed Mubarak** 400386327

**Course Name:** PROCTECH 4IC3 L02 (Industrial Networks and Controllers)

**Date of Lab:** Tuesday February 25th – Tuesday April 1st, 2025

**Date of Submission:** Tuesday April 8th, 2025

**Table of Contents:**

Title Page……………………………………………………………………………………….1

Table of Contents………………………………………………………………………………….2

Project Rationale & Description…………………………………………………………………..3

Communication between AB GuardLogix & Click C0-12ARE-1D PLC………………………...6

Slave CLICK PLC………………………………………………………………………………….10

Communication Between Eaton VFD & AB Compact GuardLogix PLC………………………13

Master Allen Bradley PLC………………………………………………………………………….14

Conclusion………………………………………………………………………………………….17

References…………………………………………………………………………………………..18

Appendices………………………………………………………………………………………….19

**Project Rationale**

The objective of this lab project was to design and implement a reliable and safe control system using a Master-Slave architecture between an Allen-Bradley GuardLogix PLC (Master) and a Click C0-12ARE-1D PLC (Slave). The project is coherent, consistent, and effectively reflects the intent of operator station interlocking, which is essential in industrial automation to ensure that machine operations occur only under appropriate and safe conditions.

The core concept of the project revolves around the centralized control philosophy — the Master AB PLC is responsible for all control decisions, while the Slave Click PLC acts as a remote operator station that gathers inputs from the operator and field devices and communicates this data to the Master PLC over EtherNet/IP using implicit messaging.

This communication setup enforces strict interlocking because the Master PLC does not perform any control action unless validated conditions from the Slave Click PLC are met. For example, system startup is only permitted when the Start pushbutton is pressed at the Click PLC, and the Stop pushbutton is inactive. Furthermore, the conveyor motor only operates if a product sensor detects an object, and the dispensing timer has finished, ensuring a safe and sequenced operation.

The project implementation supports this interlocking philosophy by using DS tags in the Click PLC to store the statuses of pushbuttons, switches, and sensors. These DS tags are communicated implicitly to the AB PLC after performing logic operations on the Click PLC. The AB PLC uses these values to control outputs such as the conveyor motor, indicator lights, and speed commands sent to the Eaton VFD.

In Auto Mode, the Click PLC also provides a scaled analog value representing motor speed based on a physical knob, further enhancing operator interaction while preserving centralized control. Local indicator lights at the Click PLC operator station provide real-time visual feedback to the user, controlled entirely by the Master PLC.

Additionally, the implementation of the VFD control and configuration through EtherNet/IP communication further aligns with the project objectives, reflecting real-world practices of integrating smart field devices into an industrial network.

Overall, the project demonstrates a clear and consistent approach to designing an industrial control system with proper operator station interlocking. The communication architecture, safety measures, control logic, and device integration effectively simulate a real-world manufacturing process, providing both technical accuracy and operational safety. This ensures that the system cannot operate without validated operator inputs and process conditions, aligning with industry-standard control strategies.

**Figure 1. Remote Station Cabinet**

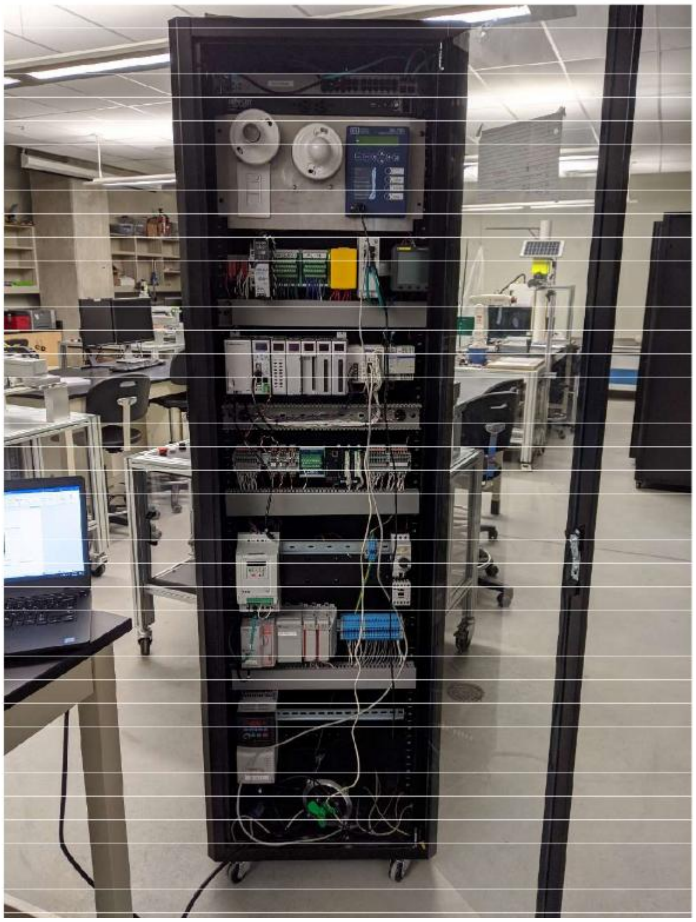

**Figure 2. Automation Cabinet**

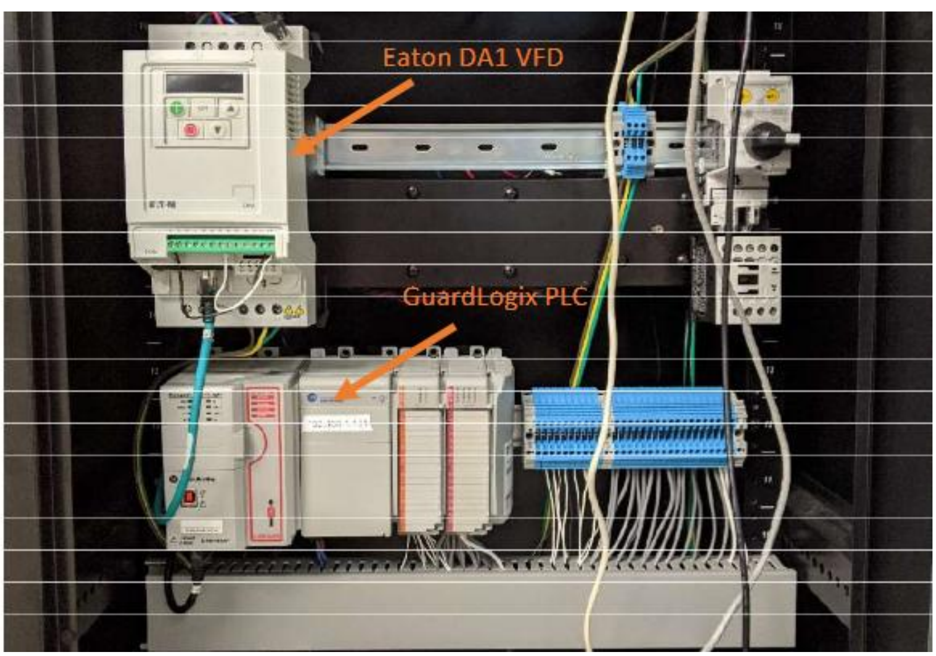


**Figure 3. Components used inside Automation Cabinet**

**Procedure to establish the Master-Slave communication between AB GuardLogix and Click C0-12ARE-1D PLC:**

1. **<u>Physically connect your devices to the same network:</u>**

In order for the Alleb-Bradley and Click PLC to communicate through EtherNet/IP, they must be physically wired to the same network. To establish that, connect both PLCs and your computer to the switch in the cabinet using ethernet cables. The job of the switch is to allow devices to see each other on a network. Also, make sure that both PLC AND your computer have the proper IP configuration with the same subnet mask, otherwise they won't be ale to communicate.

2. **<u>Launch PLC software:</u>**

Next you need to launch software for both PLCs and complete the PLC IO module configurations. These could include steps like manually setting up the CPU configuration, any IO modules, IP addresses, and subnet masks.

Click IP Address: 192.168.1.122
Click Subnet Mask 255.255.255.0

AB GuardLogix Address: 192.168.1.111
AB GuardLogix Subnet: 255.255.255.0

Group_1_Project

CPU Configuration(Page 1 of 3)

System Configuration
PLC Name :
Start-up I/O Config Check : No

| No | Module Type | Module Name | Module Setting |
|---|---|---|---|
| 1 | Power Supply | Unknown | |
| 2 | CPU | C0-11DD2E-D | |
| 3 | Slot0 | - | |
| 4 | Slot1 | - | |
| 5 | I/O 1 | C0-4AD2DA-1 | <IN ch1><br>Address=DF1, InputType=Current (0 to 20mA)<br>Input Range= 0.0mA to 20.0mA<br>Scale Range= 0.0 to 100.0 (Range Limitter:No)<br><IN ch2><br>Address=DF2, InputType=Current (0 to 20mA)<br>Input Range= 0.0mA to 20.0mA<br>Scale Range= 0.0 to 100.0 (Range Limitter:Yes)<br><IN ch3><br>Address=DF3, InputType=Current (0 to 20mA)<br>Input Range= 0.0mA to 20.0mA<br>Scale Range= 0.0 to 100.0 (Range Limitter:Yes)<br><IN ch4><br>Address=DF4, InputType=Current (0 to 20mA)<br>Input Range= 0.0mA to 20.0mA<br>Scale Range= 0.0 to 100.0 (Range Limitter:Yes)<br><OUT ch1><br>Address=DF5, InputType=Current (4 to 20mA)<br>Output Range= 4.0mA to 20.0mA<br>Scale Range= 0.0 to 100.0 (Range Limitter:Yes)<br><OUT ch2><br>Address=DF6, InputType=Current (4 to 20mA)<br>Output Range= 4.0mA to 20.0mA<br>Scale Range= 0.0 to 100.0 (Range Limitter:Yes) |
| 6 | I/O 2 | - | |
| 7 | I/O 3 | - | |
| 8 | I/O 4 | - | |
| 9 | I/O 5 | - | |
| 10 | I/O 6 | - | |
| 11 | I/O 7 | - | |
| 12 | I/O 8 | - | |

**Figure 4. CPU Configuration for Click PLC**

Group_1_Project

CPU Configuration(Page 2 of 3)

Com Port Setup

| No | Name | Port1 | Port2 | Port3 |
|---|---|---|---|---|
| 1 | Protocol | Modbus TCP | Modbus RTU | Modbus RTU |
| 2 | Node | - | 1 | 1 |
| 3 | Baud Rate | - | 38400 | 38400 |
| 4 | Parity Bit | - | ODD | ODD |
| 5 | Stop Bit | - | 1 | 1 |
| 6 | Data Bit | - | - | - |
| 7 | Time-out | - | 500ms | 500ms |
| 8 | Char Time-out | - | 2ms | 2ms |
| 9 | RTS ON Delay | - | 0ms | - |
| 10 | RTS OFF Delay | - | 0ms | - |
| 11 | Response Delay Time | - | 0ms | 0ms |
| 12 | IP Address | 192.168.1.122 | - | - |
| 13 | Subnet Mask | 255.255.255.0 | - | - |
| 14 | Default Gateway | 0.0.0.0 | - | - |
| 15 | Network Configuration | Manually | - | - |
| 16 | Client Timeout | 1000ms | - | - |
| 17 | Client Retries | 2 | - | - |
| 18 | Server Inactivity | 60sec | - | - |
| 19 | TCP Port Number | 502 | - | - |
| 20 | Concurrent Sessions | 3 | - | - |
| 21 | Client Inactivity Timeout | 60sec | - | - |
| 22 | DNS Server | Enable | - | - |
| 23 | DNS Server Address | DHCP | - | - |
| 24 | Preferred DNS Server | - | - | - |
| 25 | Alternate DNS Server | - | - | - |
| 26 | Enable Change IP | Yes | - | - |
| 27 | Enable Response Ping | Yes | - | - |
| 28 | Allow List(IP Address) | - | - | - |
| 29 | Allow List(Mac Address) | - | - | - |

**Figure 5. CPU Configuration for Click PLC (Continued)**

### 3. Establish communication path to both PLCs:

After you have physically connected your devices to the same network, now you need to set up the communication path through the backplate to both PLC from each software. This step is necessary for the computer to connect and download/upload programs to the PLCs.

### 4. Test communication:

Now you can use the ping command in command prompt to test if the computer can communicate to both PLCs. If you see no errors, this means that the computer can successfully send and receive messages from the PLC.

Example of ping command:

| Command | Notes |
|---|---|
| PING 192.168.1.1 | Simple ping of an IP address |

### 5. Setting up local addresses for Click PLC IO:

Now that you have successfully connected your PLCs and computer, its time to setup any inputs and outputs addresses to your PLC. We started by assigning local addresses for the inputs and outputs at the remote IO station to the click PLC. We assigned addresses to digital inputs from X001-X006, digital outputs from Y001-Y004, and analog inputs of knobs 1 &2 to addresses DF1 and DF2 relatively.

Group_1_Project

Address Picker

| Address | Modbus Address | Nickname | DataType | InitialValue | Retentive | Comment |
|---|---|---|---|---|---|---|
| X001 | 100001 (0000h) | Start_PB | Bit | Off | No | |
| X002 | 100002 (0001h) | Stop_PB | Bit | Off | No | |
| X003 | 100003 (0002h) | S2_Spring_Loaded | Bit | Off | No | |
| X004 | 100004 (0003h) | S1_Toggle | Bit | Off | No | |
| X005 | 100005 (0004h) | S3_Right_Position | Bit | Off | No | |
| X006 | 100006 (0005h) | S3_Left_Position | Bit | Off | No | |
| X101 | 100033 (0020h) | _IO1_Module_Error | Bit | Off | No | On when module is not functioning |
| X102 | 100034 (0021h) | _IO1_Missing_24V | Bit | Off | No | On when missing external 24VDC input |

Group_1_Project

Address Picker

| Address | Modbus Address | Nickname | DataType | InitialValue | Retentive | Comment |
|---|---|---|---|---|---|---|
| Y001 | 8193 (2000h) | Green_Light | Bit | Off | No | |
| Y002 | 8194 (2001h) | Red_Light | Bit | Off | No | |
| Y003 | 8195 (2002h) | White_Light | Bit | Off | No | |
| Y004 | 8196 (2003h) | Yellow_Light | Bit | Off | No | |

Group_1_Project

Address Picker(Page 3 of 3)

| Address | Modbus Address | Nickname | DataType | InitialValue | Retentive | Comment |
|---|---|---|---|---|---|---|
| DF1 | 428673 (7000h) | knob1 | Float | Disable | Yes | |
| DF2 | 428675 (7002h) | knob2 | Float | Disable | Yes | |

**Figure 6. Address Picker – Click PLC**

## 6. Establish implicit messaging between Click and AB GuardLogix:

To accomplish an implicit messaging between PLC, you have to extract an EDS file from the click PLC by going to the setup tab à EtherNet/IP adapter à then follow the instructions by searching "Click and AB PLC setup" on the help menu. This will show you instructions on configuring the implicit messaging. Part of that setup process is to select tags to be read from and written to in the Click PLC. These tags are shown in the figure below:

EtherNet/IP Setup
TCP Port Number = 44818
TCP Timeout = 30sec

EtherNet/IP Setup - Connection1
Data State of Originator = Hold

| Input | | | | |
|---|---|---|---|---|
| Block | Data Read From | | Data Block Offset | |
| No. | Start | End | Start | End |
| 1 | DS1 | DS100 | 1 | 200 |
| 2 | - | - | - | - |
| 3 | - | - | - | - |
| 4 | - | - | - | - |
| 5 | - | - | - | - |
| 6 | - | - | - | - |
| 7 | - | - | - | - |
| 8 | - | - | - | - |
| 9 | - | - | - | - |
| 10 | - | - | - | - |

WordSwap = Disable
ByteSwap = Disable

| Output | | | | |
|---|---|---|---|---|
| Block | Data Write To | | Data Block Offset | |
| No. | Start | End | Start | End |
| 1 | DS201 | DS300 | 1 | 200 |
| 2 | - | - | - | - |
| 3 | - | - | - | - |
| 4 | - | - | - | - |
| 5 | - | - | - | - |
| 6 | - | - | - | - |
| 7 | - | - | - | - |
| 8 | - | - | - | - |
| 9 | - | - | - | - |
| 10 | - | - | - | - |

WordSwap = Disable
ByteSwap = Disable

**Figure 7. Tags to be Read From and written To**

Then, export EDS file and save.

After you have saved your EDS file, you then have to add the EDS file to AB by going to the “tools” tab à device description installation tool à then add the EDS file you saved from the click PLC. Then right click on the EtherNet adapter in the bottom left of the screen à new module à attach the EDS file.

So far, you have established a communication and implicit messaging between the PLCs. By adding the EDS file to the Allen Bradley PLC, you are setting up a master-slave communication where tags are being implicitly communicated between PLC. You do not need messaging instructions in that case. The AB will implicitly read and write to tags you have configured in the EDS file. This means that you have setup the AB PLC as the Master and the Click PLC as the slave.

However, because DS tags are the one being communicated with AB PLC, we have to do some logic on the Click PLC to assign our local values to DS tags.

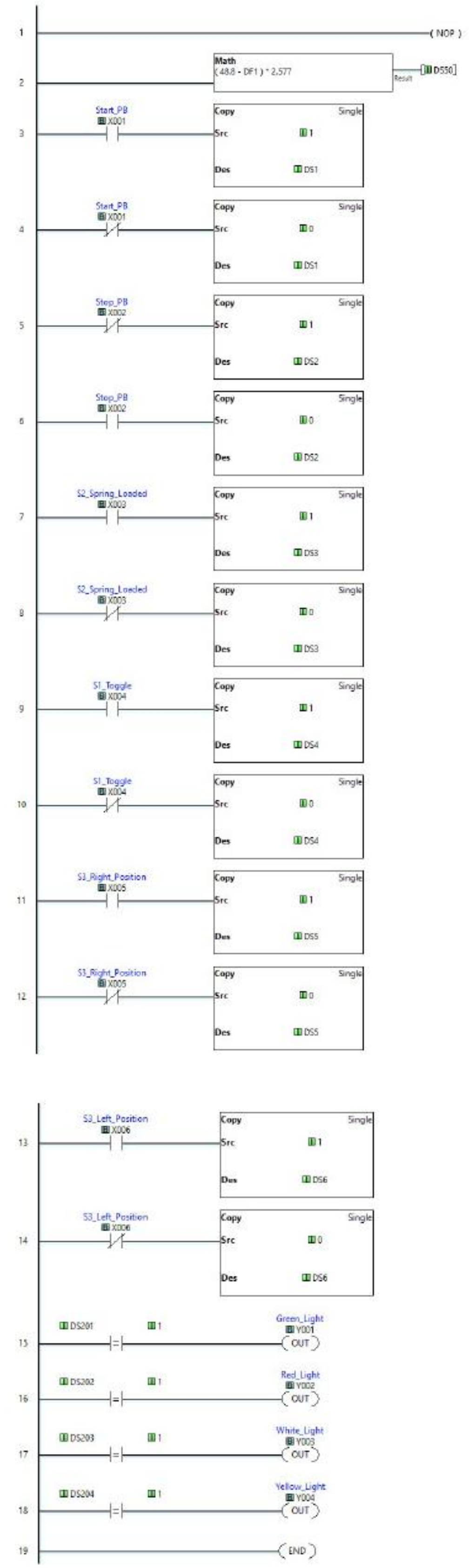


**Figure 8. Slave CLICK PLC Ladder Logic**

**Rung Comments:**

**Rung 1:**
Performs a math operation that scales the analog input from the knob (DF1) using the formula (4.88 - DF1) × 2.577 and stores the result in DS50. This scaled value represents motor speed and is sent to the AB PLC for use in Auto Mode.

**Rung 2 & 3:**
When the Start pushbutton (X001) is pressed, a value of 1 is copied into register DS1.
This signals the Master PLC that a start request has been made by the operator.

**Rung 4 & 5:**
When the Stop pushbutton (X002) is pressed, a value of 1 is copied into register DS2. This lets the Master PLC know that a stop command has been issued.

**Rung 6 & 7:**
If the S2 Spring Loaded switch (X003) is active, DS3 is set to 1; otherwise, it is reset to 0. This communicates the switch status to the AB PLC for logic decisions or interlocking.

**Rung 8 & 9:**
The S1 Toggle switch (X004) sets DS4 to 1 when ON and to 0 when OFF. This toggled state is shared with the AB PLC over Ethernet/IP.

**Rung 10 & 11:**
The S3 Right Position switch (X005) updates DS5 to 1 when active and 0 when inactive. This tells the AB PLC the current position status of a selector or physical element.

**Rung 12 & 13:**
The S3 Left Position switch (X006) sets DS6 to 1 when ON and 0 when OFF. This also provides positional information to the Master PLC for logic branching or control.

**Rung 14:**
Turns on the Green Light (Y001) when DS201 equals 1. This light is controlled by the Master PLC to indicate system go or readiness.

**Rung 15:**
Activates the Red Light (Y002) if DS202 equals 1. This serves as a fault or alert indicator at the operator station.

**Rung 16:**
When DS203 equals 1, the White Light (Y003) is turned ON. This is typically used for system active or neutral indication.

**Rung 17:**
The Yellow Light (Y004) is triggered ON when DS204 equals 1. This is used to show that the system is in a pause or dispensing state.

**Slave CLICK PLC Program Summary:**

The Slave Click PLC was programmed to act as the remote operator station for the control system. Its primary role was to read local operator inputs (pushbuttons, switches, and analog knob) and communicate this data to the Master Allen Bradley (AB) PLC over Ethernet/IP using implicit messaging. Additionally, the Click PLC controlled local indicator lights based on commands received from the Master PLC, providing visual feedback to the operator.

The Click PLC program handled several key functions. The Start and Stop pushbuttons were configured to update specific data registers (DS1 and DS2) to signal the AB PLC when the operator requested to start or stop the system. Various position and toggle switches (X003 to X006) were monitored, and their statuses were written to data registers (DS3 to DS6) for communication to the Master PLC.

An important feature of the program was scaling the analog input from a knob connected to DF1. The analog value was processed using a mathematical formula to convert it into a suitable range for motor speed control in Auto Mode, with the scaled value stored in DS50 and sent to the Master PLC.

Furthermore, the Click PLC managed the local output indicator lights (Green, Red, White, and Yellow) based on data received from the Master PLC (DS201 to DS204). This allowed the operator to receive real-time visual feedback on the system status, such as when the system was running, in pause mode, or experiencing a fault.

Overall, the Slave Click PLC program ensured effective communication and interlocking between the operator station and the Master PLC. It provided a reliable and safe interface for the operator while maintaining centralized control of system operations through the AB PLC.

| Address | Nickname | DataType | ProgramName | Rung | Row | Column | Instruction |
|---|---|---|---|---|---|---|---|
| X001 | Start_PB | Bit | Main Program | 3 | 1 | A | Contact (NO) |
| X001 | Start_PB | Bit | Main Program | 4 | 1 | A | Contact (NC) |
| X002 | Stop_PB | Bit | Main Program | 5 | 1 | A | Contact (NC) |
| X002 | Stop_PB | Bit | Main Program | 6 | 1 | A | Contact (NO) |
| X003 | S2_Spring_Loaded | Bit | Main Program | 7 | 1 | A | Contact (NO) |
| X003 | S2_Spring_Loaded | Bit | Main Program | 8 | 1 | A | Contact (NC) |
| X004 | S1_Toggle | Bit | Main Program | 9 | 1 | A | Contact (NO) |
| X004 | S1_Toggle | Bit | Main Program | 10 | 1 | A | Contact (NC) |
| X005 | S3_Right_Position | Bit | Main Program | 11 | 1 | A | Contact (NO) |
| X005 | S3_Right_Position | Bit | Main Program | 12 | 1 | A | Contact (NC) |
| X006 | S3_Left_Position | Bit | Main Program | 13 | 1 | A | Contact (NO) |
| X006 | S3_Left_Position | Bit | Main Program | 14 | 1 | A | Contact (NC) |
| X101 | _IO1_Module_Error | Bit | I/O 1 | | | | Analog Input && Output |
| X102 | _IO1_Missing_24V | Bit | I/O 1 | | | | Analog Input && Output |
| Y001 | Green_Light | Bit | Main Program | 15 | 1 | AF | Out |
| Y002 | Red_Light | Bit | Main Program | 16 | 1 | AF | Out |
| Y003 | White_Light | Bit | Main Program | 17 | 1 | AF | Out |
| Y004 | Yellow_Light | Bit | Main Program | 18 | 1 | AF | Out |
| DF1 | knob1 | Float | Main Program | 2 | 1 | AF | Math |
| DF1 | knob1 | Float | I/O 1 | | | | Channel 1 (Input) |
| DF2 | knob2 | Float | I/O 1 | | | | Channel 2 (Input) |

**Figure 9. Slave CLICK PLC Program Tags**

7. **Setting Up Communication Between the Eaton VFD and AB Compact GuardLogix PLC**

To enable communication between the Eaton Variable Frequency Drive (VFD) and the Allen-Bradley Compact GuardLogix PLC, the following setup steps were performed:

A. **Download Required Files:** The Electronic Data Sheet (EDS) file and the user manual for the Eaton VFD were downloaded from Avenue. The EDS file is essential for integrating the VFD as a recognizable device in Studio 5000.

B. **Configure VFD Parameters:** Several key parameters on the Eaton VFD were configured to enable control via the PLC:
   - **P1-12 → Set to 4:** This parameter configures the drive for Fieldbus communication. By setting it to 4, we are telling the VFD that it will be controlled through an external device, in this case, the PLC.
   - **P1-01:** Displays the maximum RPMs the motor is allowed to run.
   - **P1-02:** Displays the minimum RPMs the motor can operate at.

C. **Add EDS File to Studio 5000:** To install the EDS file:
   - Navigate to **Tools > EDS Hardware Installation Tool** in Studio 5000.
   - Select **Register an EDS file**, then browse and install the Eaton VFD EDS file downloaded earlier.
   - This step should be done while offline in Studio 5000.

D. **Add the VFD as a Module:** Once the EDS file is registered:
   - Add the VFD as a new module under the **Ethernet adapter** in the I/O Configuration.
   - During setup, change the **input and output data types** to **DINT (integer)**.
   - Assign the correct **IP address** of the VFD to establish Ethernet/IP communication.

E. **Verify Controller Tags:** After successfully adding the VFD module, navigate to the **Controller Tags** section. You should see eight tags created automatically:
   - **4 Input Tags** (from VFD to PLC)
   - **4 Output Tags** (from PLC to VFD)

F. **Tag Mapping (Refer to VFD Manual - Page 36):** Each tag corresponds to a specific control or monitoring function. According to the Eaton VFD manual:
   - **Output Tag 1:** Control Command (e.g., Start/Stop)
   - **Output Tag 2:** Frequency Reference (Speed command)
   - **Input Tag 1:** Drive Status
   - **Input Tag 2:** Actual Frequency
   - (Refer to the manual for full tag mapping details and data word structures.)

This configuration enables the Compact GuardLogix PLC to command and monitor the Eaton VFD using standard Ethernet/IP protocol.

Eaton VFD Manual:
https://avenue.cllmcmaster.ca/d2l/le/content/677553/viewContent/4980733/View

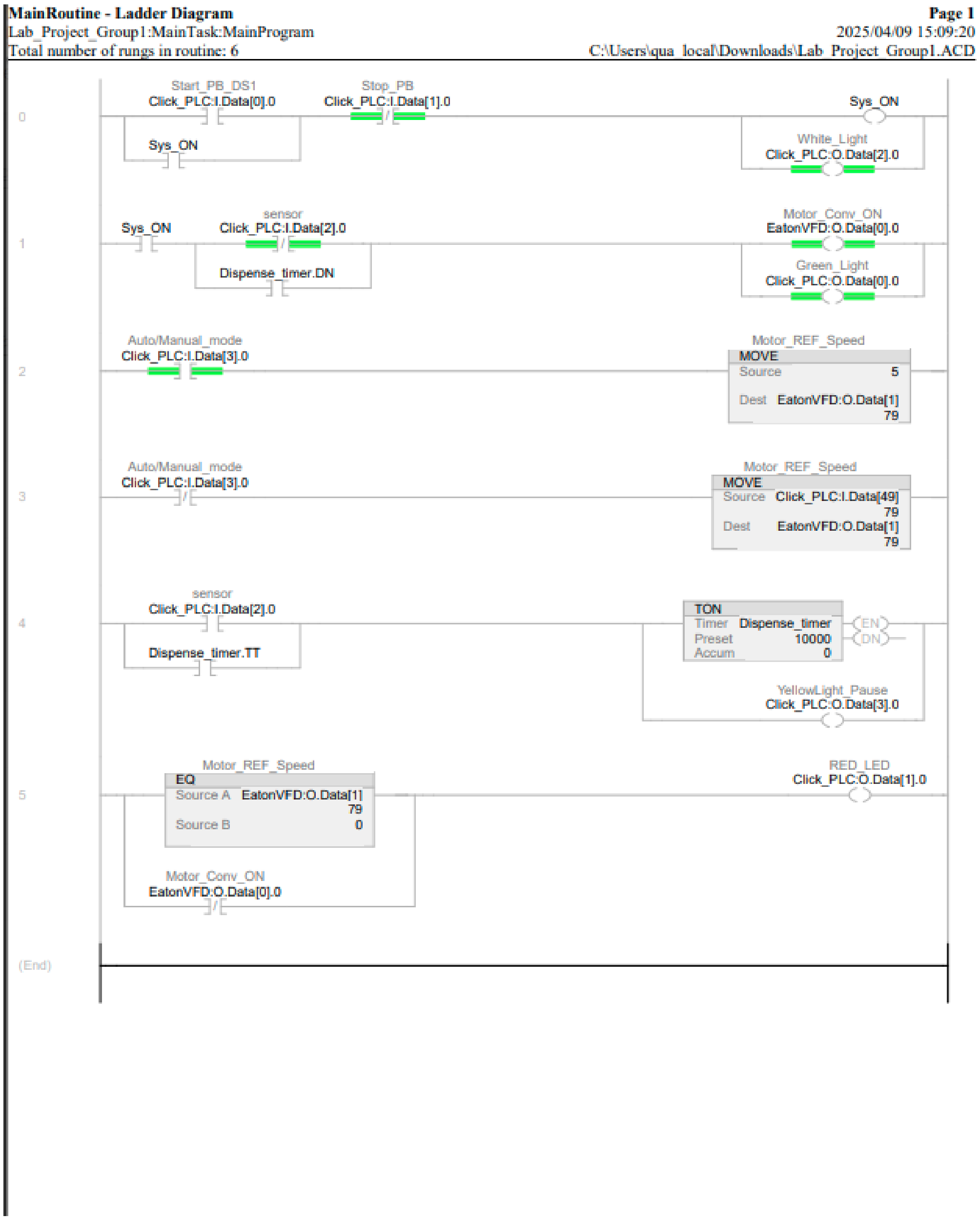


**Figure 10. Master Allen Bradley PLC Program**

**Rung comments:**

**Rung 0:**
Start/Stop logic for system power. When the Start PB (Click_PLC1:Data[1].0) is pressed and Stop PB (Click_PLC1:Data[1].1) is not active, the Sys_ON latch is energized to power up the system.

**Rung 1:**
System status and indicator lights. If Sys_ON is active and the sensor (Click_PLC1:Data[2].0) is triggered, and the Dispense_timer.DN is done, then the white and green indicator lights turn ON, and the motor conveyor is enabled.

**Rung 2:**
Manual speed control. If in manual mode (Click_PLC1:Data[3].0 is OFF), set motor speed to a fixed value of 5 using the MOVE instruction.

**Rung 3:**
Auto mode motor speed control. If in auto mode (Click_PLC1:Data[3].0 is ON), the motor speed is controlled by a variable source (Click_PLC1:Data[4].0) and moved to the drive.

**Rung 4:**
Dispensing process timer. When the sensor detects a product and the timer is running (Dispense_timer.TT), a 10-second timer is activated. After timing, the yellow light (pause indication) turns ON.

**Rung 5:**
Conveyor motor control logic. If motor speed equals 0 (EatonVFD:O.Data[1] == 0), the red LED turns ON as an alert. If the motor reference speed is not zero, the conveyor motor is turned ON.

**Master PLC program summary:**

This program is written for the Master PLC, which oversees the operation of an automated dispensing and conveyor belt system. It communicates with a Click PLC acting as the Slave located at the operator station.

The system started using a physical Start push button wired to the Click PLC at the operator station. When the Start button is pressed and the Stop button is not active, the Click PLC sends a signal to the Master PLC to enable the system by setting the Sys_ON bit. Once the system is on, a product detection sensor triggers a 10-second timer in the Master PLC to initiate the dispensing process. During this time, the conveyor motor is stopped to allow for accurate and safe dispensing.

A yellow indicator light turns on to show that dispensing is active. Once the timer finishes, the conveyor motor restarts automatically, and the green and white indicator lights turn on, indicating that normal operation has resumed. The system includes both manual and automatic motor speed control modes. In manual mode, the motor runs at a fixed speed, while in automatic mode, speed values are sent from the operator station (Click PLC) to the Master PLC. Additionally, a red LED is activated if the motor reference speed is zero, signaling an alert condition.

This Master PLC program ensures synchronized control of the conveyor and dispensing system, with input signals and commands originating from the Slave Click PLC at the operator station.

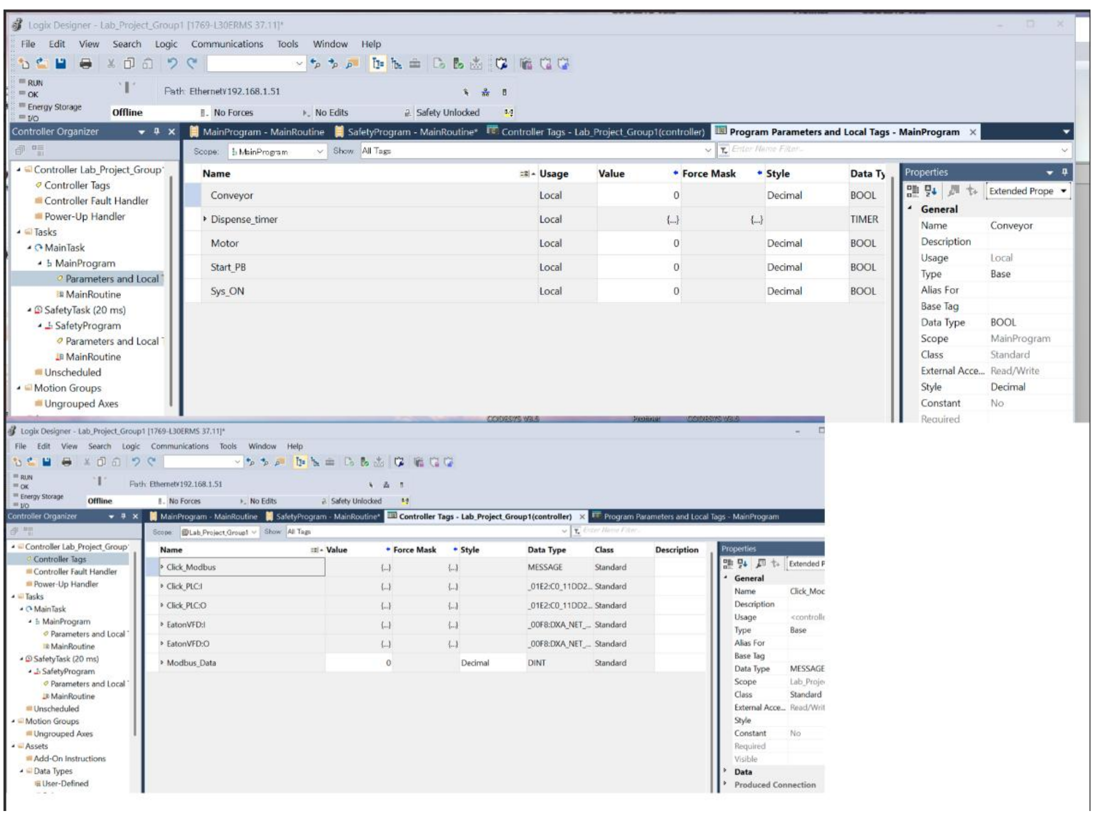


**Figure 11. Master Allen Bradley PLC Tag**

## Conclusion

This project successfully demonstrated the implementation of an industrial network communication system using a Master-Slave architecture between an Allen-Bradley GuardLogix PLC (Master) and a Click PLC (Slave) at the operator station. Through proper IP configuration, EDS file integration, and implicit messaging, the two PLCs communicated effectively over Ethernet/IP. Additionally, integration with an Eaton VFD allowed real-time motor speed control, supporting both manual and automatic operation modes. The system logic ensured the conveyor motor stopped during dispensing and restarted afterward, improving process safety and precision. Visual indicators provided real-time feedback to operators, enhancing usability and system awareness. Overall, the project fulfilled its objective of developing a reliable, networked PLC-based dispensing and conveyor control system.

## Improvement Recommendations

To further enhance the system, several improvements can be considered. First, integrating an HMI panel would provide more intuitive control and real-time monitoring, especially for speed adjustment in automatic mode or system status alerts. Second, implementing fail-safes or watchdog timers would improve fault detection and recovery in case of communication loss between the Master and Slave PLCs. Third, adding more sensors (e.g., level sensors for product bins or downstream position sensors) could help expand automation capabilities. Finally, logging system data—such as number of products dispensed, motor runtime, or error events—would support performance tracking and predictive maintenance in a real industrial environment.

**References:**


1. Rockwell Automation. (2021). *Studio 5000 Logix Designer Application Version 32.00 User Manual*. Retrieved from https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1756-pm021_-en-p.pdf

2. Rockwell Automation. (2020). *EtherNet/IP Network Configuration User Manual*. Retrieved from https://literature.rockwellautomation.com/idc/groups/literature/documents/um/enet-um001_-en-p.pdf

3. Automation Direct. (2021). *CLICK PLC User Manual*. Retrieved from https://cdn.automationdirect.com/static/manuals/clickplcm/clickplcm.html

4. Eaton Corporation. (2019). *DG1 VFD User Manual*. Retrieved from https://www.eaton.com/content/dam/eaton/products/industrialcontrols-drives-automation-sensors/drives/dg1-series/drives-dg1-user-manual-en.pdf

5. McMaster University. (2025). *PROCTECH 4IC3 Lab Project - Local Control and Remote Interlocking* Faculty of Engineering Technology, Automation Engineering Technology Program.

6. AutomationDirect. (n.d.). *CLICK example for AB CompactLogix*. AutomationDirect. Retrieved April 10, 2025, from https://www.automationdirect.com/microsites/clickplcs/click-help/Content/235.htm

7. AutomationDirect. (n.d.). *CLICK EtherNet/IP adapter setup*. AutomationDirect. Retrieved April 10, 2025, from https://www.automationdirect.com/microsites/clickplcs/click-help/Content/233.htm

8. Rockwell Automation. (2021). *Logix 5000 controllers messages* (Publication 1756-PM012A-EN-P). Retrieved from https://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm012_-en-p.pdf

9. Rockwell Automation. (2021). *Allen-Bradley Ethernet reference manual* (Publication ENET-RM002-EN-P). Retrieved from https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/enet-rm002_-en-p.pdf

10. Rockwell Automation. (2021). *Logix 5000 controllers general instructions* (Publication 1756-RM003-EN-P). Retrieved from https://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm003_-en-p.pdf

**Appendices:**


- Appendix A: Figure 1. Remote Station Cabinet
- Appendix B: Figure 2. Automation Cabinet
- Appendix C: Figure 3. Components used inside Automation Cabinet
- Appendix D: Figure 4. CPU Configuration for Click PLC
- Appendix E: Figure 5. CPU Configuration for Click PLC (Continued)
- Appendix F: Figure 6. Address Picker – Click PLC
- Appendix G: Figure 7. Tags to be Read From and written To
- Appendix H: Figure 8. Slave CLICK PLC Ladder Logic
- Appendix I: Figure 9. Slave CLICK PLC Program Tags
- Appendix J: Figure 10. Master Allen Bradley PLC Program
- Appendix K: Figure 11. Master Allen Bradley PLC Tag