YOUTILIZE® / PLC COMMUNICATIONS 7.7.5

Youtilize® / PLC Communications

Manufacturing Operations Software

This document has been designed and written to provide a thorough understanding of the Youtilize® / PLC communication protocol. The intended audience is both the Youtilize® developer (typically a C#.NET developer) and the PLC programmer. These two parties utilize different programming tools and languages however this document has been designed to try and convey all the material in a manner understandable to all. Furthermore, it is intended that non-programming personnel should be able to use this document to get a thorough and in-depth understanding of the communications protocol.

This document covers Youtilize® Software through version 7.7.5.0.

Contents

Version History

1.0	Initial draft	Brian Worrall	July 20, 2012
1.1	Added Status Code and Cmd Status Code Tags.	Joseph Kavchok III	August 8th 2012
1.2	Updated sequence diagrams and added timing diagrams. Also expanded the Workstation explanation considerably.	Brian Worrall	August 15th 2012
1.3	Added additional detail as requested by FlexLink personnel.	Brian Worrall	August 24th 2012
1.4	Added bit level detail as well as IPC state and information	Brian Worrall	August 27th 2012
1.5	Corrected Function Fault Code IPC States	Joseph Kavchok III	August 28th 2012
1.6	Corrected document to only reference information related to software version 7.7.5. Changed Document Format.	Joseph Kavchok III	September 28, 2012
1.14	WIP transaction table for a Function Motor error code table	Robert Henriksson	October 15, 2012

Background

Historically PLC (Programmable Logic Controller) programming has been used to control conveyor systems in real-time. A single PLC may control a whole line, or multiple may be employed. In the latter situation the PLC's may be working together in a limited form but typically they are deployed independently. The aim of the Youtilize® project was to merge the PLC and MES worlds so that the MES system could not only monitor product movement on the factory floor but also to interface with the PLC controllers to influence those product movements.

From the outset it has been imperative that the code be re-used from project to project, both to keep development & testing time low and to facilitate maintenance. Furthermore it was imperative to design a protocol that was not dependent upon a proprietary PLC architecture.

Note:

> Although a number of projects have been jointly undertaken to date the documentation herein is designed to describe the solution that grew out of the Grundfos project. That is, a generic solution that had re-usable code designed and implemented on both the Aegis and FlexLink sides of the communication protocol. On the PLC side this is aided by the use of re-usable 'modules' that are easily transposed from one PLC to another. The use of a generic re-usable solution is designed to greatly reduce the project implementation and deployment timelines as well as to ease maintenance.

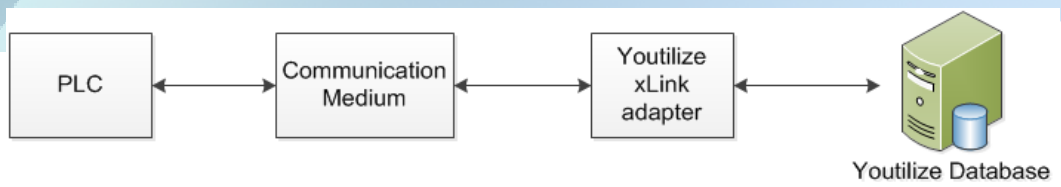
Architectural Overview

Introduction

In the Aegis MES world data collection from machines on the manufacturing floor is achieved with the use of an xLink adapter. Typically an xLink adapter operates a one-way communication, transforming third-party information into a standardized form which is then posted to the MES database. This architecture was extended for this project so as to allow for information from the MES solution to also be sent to the xLink adapter.

Communication between the PLC and Youtilize® is two-way and is implemented by the use of a segment of shared memory in the PLC which allows for messages to be sent in either direction.

As the 'flavor' of PLC could vary from project to project another key design feature was the use of a communication medium between Youtilize® and the PLC. This communication medium would ensure the connection and communication between Youtilize® and a PLC would be standardized irrespective of the PLC implementation.



So even though an Allen-Bradley PLC may perform the same functionality as a Siemens PLC their implementations of the OPC Standard differ. This could have resulted in different versions of the Youtilize®xLink adapter, one per 'flavor' of PLC. The impact of such a design would be to require significant testing every time a new PLC xLink adapter implementation was developed. The role of the communication medium is to provide a standard OPC interface to the xLink adapter irrespective of the 'flavor' of PLC employed. The communication medium chosen was the Matrikon OPC Server. Matrikon support many makes and models of PLC whilst providing a universal OPC implementation with which the xLink adapter can interact.

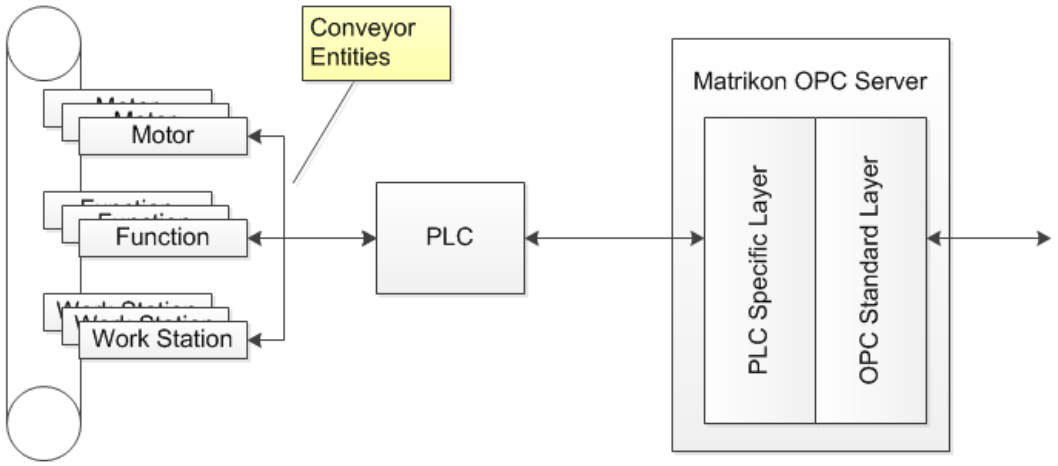
As a result, if a project introduces a make and/or model of PLC that has not been supported to date the customer would purchase the Matrikon OPC server for that PLC. FlexLink would need to port their code to that PLC but the changes required on the Aegis side of the communication medium would be limited to extending the xLink adapter to register each tag to the correct PLC memory location for the PLC implementation. This not only reduces the testing required but also the delivery time and cost.

PLC Architecture

One PLC can control many conveyor entities. Although these physical entities may take many forms for the purpose of this communication protocol they can be classified as

- Motors
- Functions
- Work Stations

Each conveyor entity shall be controlled by a single PLC and a PLC may control many conveyor entities.

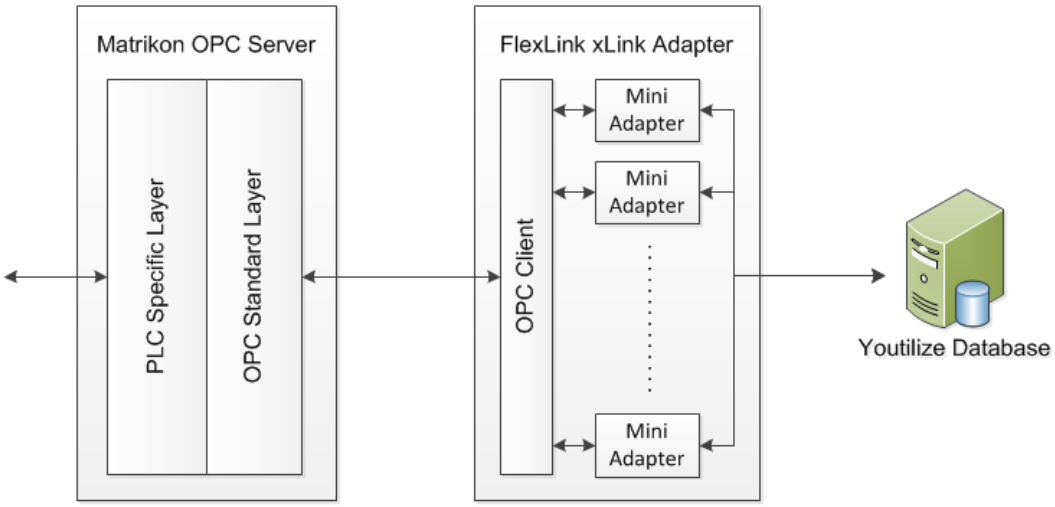


Within the PLC a segment of memory is used to convey messages to and from the MES solution via the Matrikon OPC Server. This segment of memory is sub-divided into a number of tags, each of which shall be used to convey a message in one direction or the other, but not both.

The ladder logic used to program a PLC shall be as standardized as possible and built in a modular manner so as to facilitate copying from one PLC to another; however 'flavors' of PLC code may be required depending upon the make and model of PLC being utilized.

MES Architecture

On the MES side of the communication medium the following architecture is employed:



The FlexLinkxLink adapter maintains a single OPC Client which connects to the Matrikon OPC Server. This singular OPC Client is then utilized by each mini adapter created within the FlexLink adapter allowing the mini adapter to communicate to a PLC via the Matrikon OPC Server. The number and type of mini adapters created within a FlexLink adapter is controlled by an XML file known as the FlexLink Configuration File which is identified with an ".FLX" file extension. The creation and content of this file is documented elsewhere and therefore shall not be covered in this document.

When a mini adapter is instantiated the FLX file details the type of conveyor entity the adapter is to support (either, a Motor, Function or Work Station). By default the behavior of the mini adapter is standardized however this standard behavior can be customized by the use of a mini adapter plug-in.

Communication

Messages can be passed between a PLC and a mini adapter by way of the OPC layer, where this OPC layer consists of the Matrikon OPC Server connected to the PLC and an OPC Client within a FlexLinkxLink adapter which connects to one or more mini adapters.

The messages themselves are communicated by the way of shared memory which has been sub-divided in a defined manner. Some memory locations are used to convey information from the PLC to the mini adapter, the others the other direction. A mini adapter, by way of the

OPC Client can register to receive an event when a particular memory location value changes thus allowing the mini adapter to be designed as an event-driven entity.

The sub-divisions in this shared memory are known as Tags and the tag structure has been standardized to reduce the programming required from project to project. The Tag structure can be broken down into four groups

- System Wide
- Motor
- Function
- Work Station

Special Considerations

So as to ensure every update written by the PLC is processed by the OPC Server to trigger the appropriate event to Youtilize® the minimum duration of a signal being high was tested on the Demo Rig. This minimum value was found to be 100ms on an Allen-Bradley PLC, so to be safe it is suggested a minimum duration of 200ms should be implemented.

Implementation

Communication

The full tag structure is grouped into four sections:

- System Wide Data
- Motor Data
- Function Data
- Work Station Data

System Wide Data

This section has yet to be implemented.

	To PLC	From PLC	Type	Data Tag in PLC
Overall		SYSTEM STATUS	DINT	HS_STATUS
		ANALOG DATA POINTS	REAL	HS_ANALOG[0-10]
		PROTOCOL REVISION	DINT	PROTOCOL_REVISION
	SYSTEM COMMANDS		DINT	HS_COMMANDS
Destination		UPDATE READY	BOOL*	HS_UPDATE_OUT[0].0
		UPDATE COMPLETE	BOOL*	HS_UPDATE_OUT[0].1
	UPDATE AVAILABLE		BOOL*	HS_UPDATE_IN[0].0
	ID TO UPDATE		DINT	HS_UPDATE_IN[1]
	UPDATE DESTINATION A		DINT	HS_UPDATE_IN[2]
	UPDATE DESTINATION B		DINT	HS_UPDATE_IN[3]
Verify	VERIFY COMMAND		BOOL*	HS_VERIFY_IN[0].0
	ID TO VERIFY		DINT	HS_VERIFY_IN[1]
		VERIFY DESTINATION A	DINT	HS_VERIFY_OUT[1]
		VERIFY DESTINATION B	DINT	HS_VERIFY_OUT[2]
		VERIFY COMPLETE	BOOL*	HS_VERIFY_OUT[0].0
Location	LOCATION COMMAND		BOOL*	HS_LOCATE_IN[0].0
	ID TO LOCATE		DINT	HS_LOCATE_IN[1]
		LOCATION COMPLETE	BOOL*	HS_LOCATE_OUT[0].0
		LOCATION VALUE	DINT	HS_LOCATE_OUT[1]
		FAULT CODE	DINT	HS_FAULTCODE[0-n]

Motor Data

The tags marked “From PLC” have been implemented however the “To PLC” tag (Motor Commands) has yet to be implemented.

To PLC	From PLC	Type	Data Tag in PLC
	MOTOR STATUS	DINT	MTR_STATUS[U]
	MOTOR SPEED	DINT	MTR_SPEED[U]
	MOTOR POWER DRAW	DINT	MTR_POWER[U]
	MOTOR RUNTIME	DINT	MTR_RUNTIME[U]
	MOTOR TEMPERATURE	DINT	MTR_TEMPERATURE[U]
MOTOR COMMANDS		DINT	MTR_CMD[U]

A motor device allows FlexLink to send motor related data into the Aegis System. When an xLink mini adapter implements a Motor Data mini adapter it automatically subscribes to events for each of the “From PLC” tags with the OPC Server. When the PLC writes a value to one of the “From PLC” tags the OPC Server fires an event that the xLink mini adapter receives. Upon receiving an event notification the mini adapter sends an EquipmentChangeState (Motor Status) or a Measurement message to the Analyzers to record the information in the database.

The motor device has the following status information types:

Motor State: Unknown, Running, Stopped, Fault Stop

Motor Error: OverCurrent, HighUDC, LowUDC, PCBOverTemp, LockedRotor, InternalFault, TurnTimeLimit

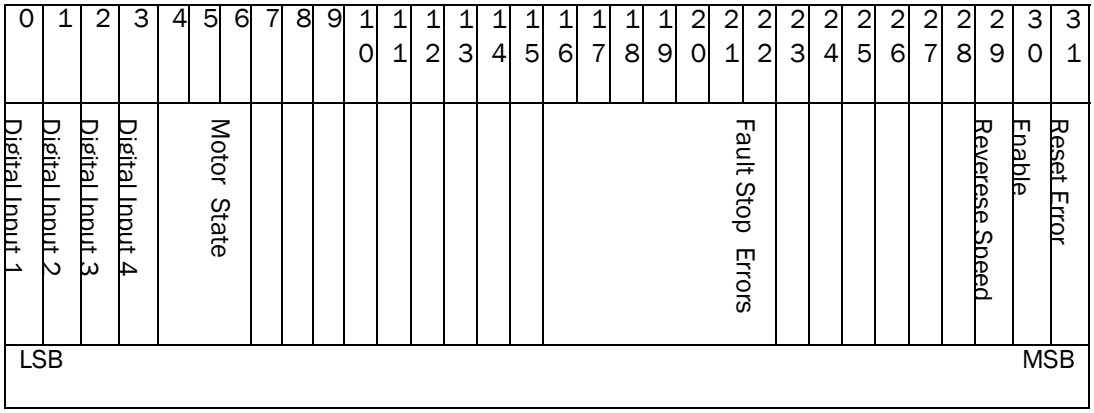
Speed (RPM)

Power (Watts)

Runtime (Hours)

Temperature (°C)

The Motor Status DWORD contains 32 bits, bits 4-6 represent the Motor State, bits 16-22 represent motor fault conditions.



The Digital Input bits are not implemented on the Aegis side.

The motor states are reported as follows:

Motor State	Hex Value	IPC State
Unknown	0x00	Unknown
Running	0x01	ReadyProcessingExecuting
Stop	0x02	Off
Fault_Stop	0x03	Down

When a Fault Stop occurs, Aegis will report a Down machine state and look to Fault Stop Errors to determine the type of down time alarm message to create.

Motor Error Code	Hex Value	IPC State
OverCurrent	0x01	Down
HighUDC	0x02	Down
LowUDC	0x04	Down
PCBOverTemp	0x08	Down
LockedRotor	0x10	Down
InternalFault	0x20	Down
TurnTimeLimit	0x40	Down

The Reverse Speed bit is not implemented on the Aegis side as the absolute of the reported motor speed value is reported to the Analyzers.

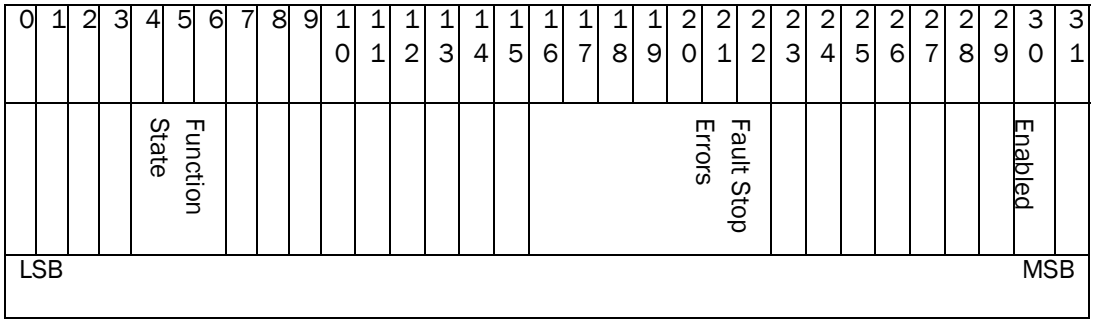
The Reset Error bit is not implemented on the Aegis side.

Function Data

The first two tags (Function Status and Extra Status Bits) have been implemented however the remainder has yet to be implemented.

To PLC	From PLC	Type	Data Tag in PLC
	FUNCTION STATUS	DINT	FCT_STATUS[V]
	EXTRA STATUS BITS	DINT	FCT_STATUS_EXT[V]
	FUNCTION CYCLES	DINT	FCT_CYCLE[V]
	PICKUP AT LANE 1 COUNT	DINT	FCT_CNT_PA[V]
	PICKUP AT LANE 2 COUNT	DINT	FCT_CNT_PB[V]
	RELEASE AT LANE 1 COUNT	DINT	FCT_CNT_RA[V]
	RELEASE AT LANE 2 COUNT	DINT	FCT_CNT_RB[V]
FUNCTION COMMANDS		DINT	FCT_CMD[V]
Stop Function			FCT_CMD[V].0
Use Commands			FCT_CMD[V].8
Pickup at Lane 1			FCT_CMD[V].9
Pickup at Lane 2			FCT_CMD[V].10
Release at Lane 1			FCT_CMD[V].11
Release at Lane 2			FCT_CMD[V].12
Lane 1 Auto			FCT_CMD[V].16
Lane 1 Operator			FCT_CMD[V].17
Lane 1 Echo			FCT_CMD[V].18
Lane 1 PC Command			FCT_CMD[V].19
Lane 2 Auto			FCT_CMD[V].24
Lane 2 Operator			FCT_CMD[V].25
Lane 2 Echo			FCT_CMD[V].26
Lane 2 PC Command			FCT_CMD[V].27

The Function Status DINT contains 32 bits, bits 4-6 represent the Function State, bits 16-22 represent function fault conditions, and bit 30 represents the On/Off state.



Enabling bit 30, notifies Aegis that the function is on and then looks to the function state for more detail. Set bit 30 low to notify Aegis that the function is off. When bit 30 is low, the function state bits are ignored.

Function State	Hex Value	IPC State
Rescue	0x00	Unknown
Init	0x01	Setup
PreTurn	0x02	ReadyProcessingExecuting
DuringTurn	0x03	ReadyProcessingExecuting
PostTurn	0x04	ReadyProcessingExecuting
Calibration	0x05	Setup
Idle	0x06	ReadyIdle
Fault_Stop	0x07	Down

These status values are normalized to IPC State values (as shown) by the mini adapter prior to sending an EquipmentChangeState message to the Analyzers to record in the database.

When a Fault_Stop occurs, Aegis will report a Down machine state (as well as an EquipmentAlarm message) and look to the Fault Stop Errors to determine the type of down time alarm message to create.

Fault Stop Error	Hex Value	IPC State
OverCurrent	0x01	Down
HighUDC	0x02	Down
LowUDC	0x04	Down
PCBOverTemp	0x08	Down
LockedRotor	0x10	Down
InternalFault	0x20	Down
TurnTimeLimit	0x40	Down

If Collect Performance Metrics is enabled in the Function definition in the FLX file then the Function Status value is also used to start, finish or abort WIP transactions. These WIP transactions are anonymous – no unit barcode / serial number is conveyed, rather each WIP

transaction is assigned a unique GUID such that if the same unit were processed twice at the same station this could not be determined by the metrics.

If the Function Status is PreTurn or DuringTurn and a WIP transaction is not already in progress then a WIP Start is registered. If the Function Status is PostTurn or Idle and a WIP transaction is already in progress then a WIP Finish is registered. If a WIP transaction is already in progress when a Function Status of Fault_Stop is received then the WIP transaction is aborted.

The Extra Status Bits get precedence over the Function Status Bits. Further, Blocked takes precedence over Starved if both are set.

Extra Status Values	Hex Value	IPC State
Blocked	0x01	ReadyIdleBlocked
Starved	0x02	ReadyIdleBlocked

Communication Protocol

The following describes the standard communication protocol implemented for a Function entity.

WIP Transaction

The following steps will create a complete WIP transaction recording a PASSED status. A unique identifier (GUID) will be generated for every transaction.

This functionality will only work if the FlexLink System Configuration File for the specified function has the 'Collect Performance Metrics' box checked.

Step	Instruction	Status Tag Value (Decimal)	Comments
1	Turn On Function	1073741824	Set Bit 30 High
2	Set State to PreTurn or During Turn	1073741856 (PreTurn) 1073741872 (DuringTurn)	Starts a WIP transaction.
3	Set State to Idle or PostTurn	1073741920 (Idle) 1073741888 (PostTurn)	Finishes a WIP transaction with a status of PASSED.

Alarm Transaction

To tell Aegis the function has experienced an OverCurrent, the Function Status Value would need to have bit 30 (Enabled) high, bits 4-6 (Function State – Fault_Stop) high, and bit 16 (Fault_Stop Error) high resulting in a hex value of 0x40010070 or in decimal, 1,073,807,472.

Work Station Data

All tags in this group have been implemented.

The workstation device allows FlexLink to model any operator or machine driven station into the Aegis System. At an operator controlled workstation, a “Release” button with a backlight may be present. The workstation device has route control functionality.

Upon initialization of a Work Station mini adapter each of the “From PLC” tags is registered with the OPC Server so that any change to the value written by the PLC shall trigger an event handler within the mini adapter.

To PLC	From PLC	Type	Data Tag in PLC
	ID AT WORK STATION	DINT	HS_WS_ID[Z]
	CONTAINER AVAILABLE	BOOL*	HS_WS_STATUS[Z].0
	RELEASING CONTAINER	BOOL*	HS_WS_STATUS[Z].1
	PASS SIGNAL GIVEN	BOOL*	HS_WS_STATUS[Z].2
	FAIL SIGNAL GIVEN	BOOL*	HS_WS_STATUS[Z].3
	STATION TURNED ON	BOOL*	HS_WS_STATUS[Z].4
	RELEASE SIGNAL GIVEN	BOOL*	HS_WS_STATUS[Z].5
	RELEASE BUTTON REQUIRED	BOOL*	HS_WS_STATUS[Z].6
	RELEASE BUTTON ACTIVATED	BOOL*	HS_WS_STATUS[Z].7
	CONTAINER FULL	BOOL*	HS_WS_STATUS[Z].8
	CONTAINER EMPTY	BOOL*	HS_WS_STATUS[Z].9
DATA READ SUCCESS		BOOL*	HS_WS_CMD[Z].0
RELEASE CONTAINER		BOOL*	HS_WS_CMD[Z].1
PASS CONTAINER		BOOL*	HS_WS_CMD[Z].2
FAIL CONTAINER		BOOL*	HS_WS_CMD[Z].3
TURN STATION ON		BOOL*	HS_WS_CMD[Z].4
TURN STATION OFF		BOOL*	HS_WS_CMD[Z].5
UPDATE DESTINATION		BOOL*	HS_WS_CMD[Z].6
ACTIVATE RELEASE BUTTON		BOOL*	HS_WS_CMD[Z].7
DEACTIVATE RELEASE BUTTON		BOOL*	HS_WS_CMD[Z].8
RELEASE BUTTON REQUIRED		BOOL*	HS_WS_CMD[Z].9
RELEASE BUTTON NOT REQUIRED		BOOL*	HS_WS_CMD[Z].10
NEW DESTINATION A		DINT	HS_WS_NEW_DESTA[Z]
NEW DESTINATION B		DINT	HS_WS_NEW_DESTB[Z]

Within this structure the HS_WS_STATUS[Z] status bits shall be defined as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Container Available	Releasing Container	Pass Signal Given	Fail Signal Given	Station Turned On	Release Signal Given	Release Button Required	Release Button Activated	Container Full	Container Empty																						
LSB																														MSB	

Explanation of tags

The tag structure is comprised two basic groups – Status tags and Command tags. On system startup Youtilize® registers with the Matrikon OPC Server so that each Status tag change fires an event within Youtilize®. Youtilize® then uses the Command tags to communicate with the PLC. When Youtilize® sets a Command tag the PLC is triggered to process that command, often updating a Status tag as a result. Once the PLC has processed a Command tag it should set the Command tag value back low / to zero. Youtilize® only ever sets command tags high.

ID AT WORKSTATION

This tag is used to record the unique id of the container as read by the RFID reader.

CONTAINER AVAILABLE

When set high this tag indicates that a container is fully within the workstation, the ID At Workstation has been written and the container is ready for processing.

RELEASING CONTAINER

When set high this tag informs Youtilize® that the container has been physically released from the workstation. When Container Available is set high this tag is set low, however once this tag is set high the PLC can either leave it high until Container Available goes high, or it can set it back low again.

PASS SIGNAL GIVEN

When set high this tag informs Youtilize® that the container at the workstation has completed its operations successfully.

Set High after receiving the PassContainer Command or a pass signal from a machine.
Set Low after receiving the FailContainer Command.

FAIL SIGNAL GIVEN

When set high this tag informs Youtilize® that the container at the workstation has completed its operations unsuccessfully.

Set High after receiving the FailContainer Command or a fail signal from a machine.
Set Low after receiving the PassContainer Command.

STATION TURNED ON

This tag is used to determine whether the workstation should be considered available for routing purposes when the PLC is routing containers. The PLC shall not route a container to a workstation whose Station Turned On tag is set low.

Set High after receiving the TurnStationOn Command.
Set Low after receiving the TurnStationOff Command.

RELEASE SIGNAL GIVEN

This tag is currently not implemented. This function was designed to give FlexLink to notify Aegis that the Release Button was pressed without setting the ReleaseContainer Status.

RELEASE BUTTON REQUIRED

This status tag is set high when Youtilize® sends the command tag Release Button Required and set low when Youtilize® sends the command tag Release Button Not Required. It is used to record the current state with regard to whether a physical button press is required or not.

If this command is received by the PLC at the same time as a Pass Container or Fail Container command the Release Button Required command shall be processed first.

– Forces FlexLink to wait for an operator to press a “Release” button. The default setting is not required, Low.

Set High after receiving the ReleaseButtonPressRequired Command.
Set Low after receiving the ReleaseButtonPressNotRequired Command.

RELEASE BUTTON ACTIVATED

This status tag is set high when Youtilize® sends the command tag Activate Release Button and set low when Youtilize® sends the command tag Deactivate Releases Button. It is used to turn on or off the button’s backlight. The default setting is not required, Low.

Set High after receiving the ReleaseButtonPressRequired Command.
Set Low after receiving the ReleaseButtonPressNotRequired Command.

CONTAINER FULL

When set high this indicates that the container at the workstation contains product.

CONTAINER EMPTY

When set high this indicates that the container at the workstation does not contain any product.

DATA READ SUCCESS

This command simply informs the PLC that the WIP Start has been fully registered with Youtilize®. If third party communication is required, this command should be used to inform machinery/software that a container is available.

RELEASE CONTAINER

This command allows Youtilize® to control when a container is to be released back into the line. The PLC, upon receiving this command, shall respond with Releasing Container.

PASS CONTAINER

This command allows Youtilize® to inform the PLC that the operations on the container at the workstation have been completed successfully.

FAIL CONTAINER

This command allows Youtilize® to inform the PLC that the operations on the container at the workstation have been completed unsuccessfully.

TURN STATION ON

This command is used to set the Station Turned On status value high.

TURN STATION OFF

This command is used to set the Station Turned On status value low.

UPDATE DESTINATION

When this command tag is set high it signals the PLC that a new route has been written to the New Destination registers. Upon receipt, the PLC should copy the New Destination values to local storage and set this command tag back to low. Also, the New Destination values must be reset to zero after copying to local storage.

ACTIVATE RELEASE BUTTON

This command allows Youtilize® to enable the physical release button.

DEACTIVATE RELEASE BUTTON

This command allows Youtilize® to disable the physical release button.

RELEASE BUTTON REQUIRED

This command is used to set the Release Button Required status value high.

RELEASE BUTTON NOT REQUIRED

This command is used to set the Release Button Required status value low.

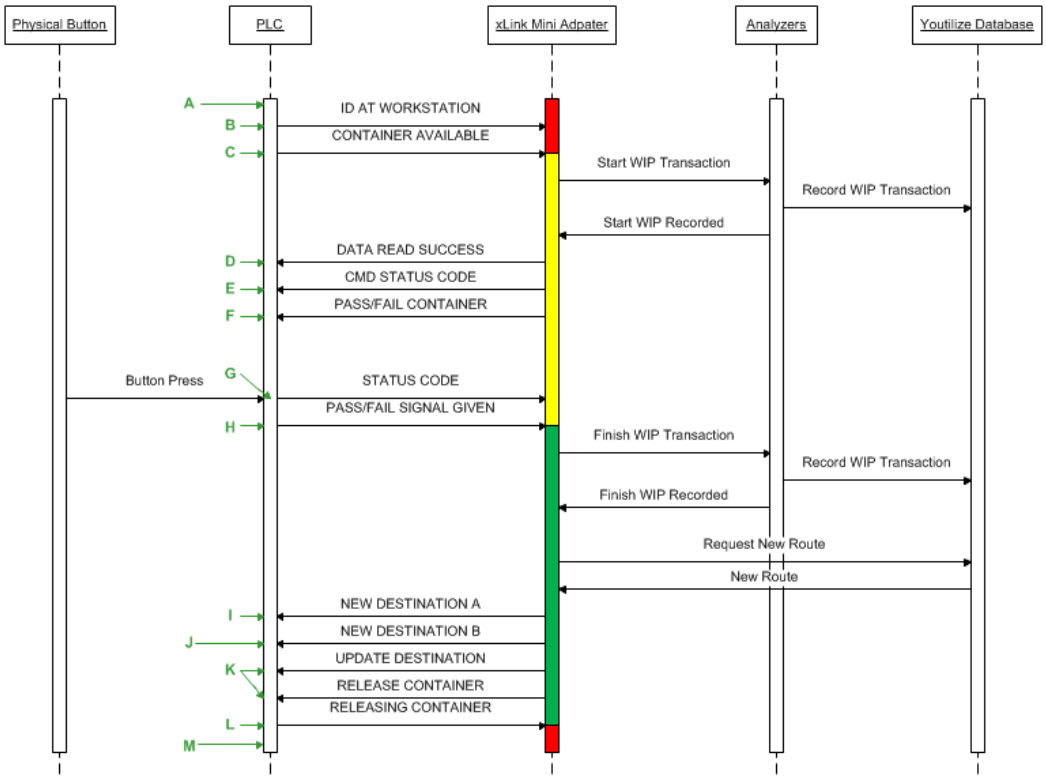
NEW DESTINATIONA \ NEW DESTINATION B

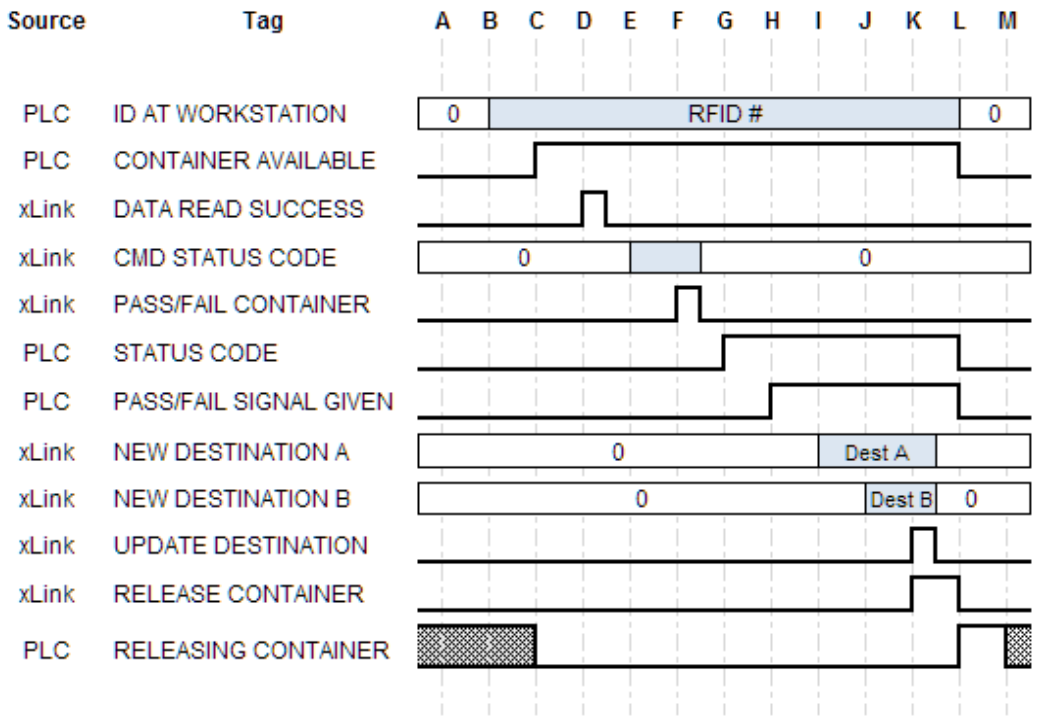
These tag values are used to convey the route for the container at the workstation. Each DINT block shall be assigned an identifying letter starting with “A” or ‘B’.

Each bit in the destination DINT registers identify a physical route point and are identified by their Route Index. The first DINT is identified by the letter “A” and identifies up to 32 route points – A.1 through A.32. Thus the maximum number of physical route points is 64; from A.1 through B.32.

Communication Protocol

The following describes the standard communication protocol implemented for a Work Station entity.





Please note that the hashed area for the Releasing Container tag (before C and after M) denotes that the value may be either high or low.

Basic operation is as follows:

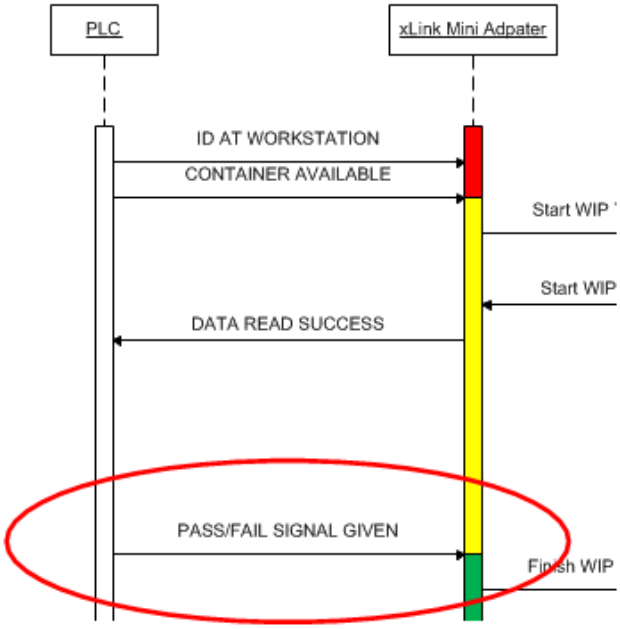
1. Asa pallet moves into a work station an RFID reader allows the PLC to record the RFID value within a temporary storage location within the PLC.
2. When the pallet is fully located in the work station the PLC writes the RFID value from the temporary storage location to the ID AT WORK STATION register (HS_WS_ID[Z]).
3. The PLC immediately sets the bit HS_WS_STATUS[z].0 high and resets the RELEASING CONTAINER (HS_WS_STATUS[Z].1) bit low.
4. The mini adapter receives the ID AT WORK STATION event and reads the value into its own in-memory storage.
5. The mini adapter receives the CONTAINER AVAILABLE event and changes its internal status from Unknown to WIP Start and records a WIP Start.
6. When operations have completed at the work station the PLC informs the mini adapter of the operation status by either setting the HS_WS_STATUS[Z].2 (for PASS) or the HS_WS_STATUS[Z].3 (for FAIL) bit high.
7. The mini adapter, upon receiving a PASS or FAIL SIGNAL GIVEN event when in the WIP Start state, completes the in-process WIP transaction by recording a WIP Finish whose status matches that reported by the PLC.
8. When the PLC releases the pallet from the work station it sets the
   RELEASING CONTAINER (HS_WS_STATUS[Z].1) bit high.
   CONTAINER AVAILABLE (HS_WS_STATUS[Z].0) bit low.
   PASS SIGNAL GIVEN (HS_WS_STATUS[Z].2) bit low.
   FAIL SIGNAL GIVEN(HS_WS_STATUS[Z].3) bit low.
9. Upon receiving the RELEASING CONTAINER event the mini adapter sets its internal state to Unknown.

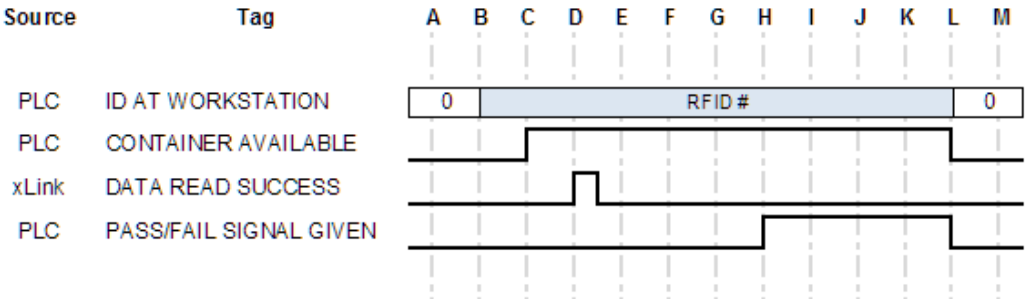
It can be seen from the above scenario that a number of additional tags exist. These allow a workstation to be configured in one of a number of configurations that reflect differing real world situations. The work station configurations supported are as follows:

Configuration number	Pass/Fail set by:	Release button required?	Routing set by:
1	PLC	No	PLC
2	PLC	No	Mini Adapter
3	PLC	Yes	PLC
4	PLC	Yes	Mini Adapter
5	Mini Adapter	No	PLC
6	Mini Adapter	No	Mini Adapter
7	Mini Adapter	Yes	PLC
8	Mini Adapter	Yes	Mini Adapter

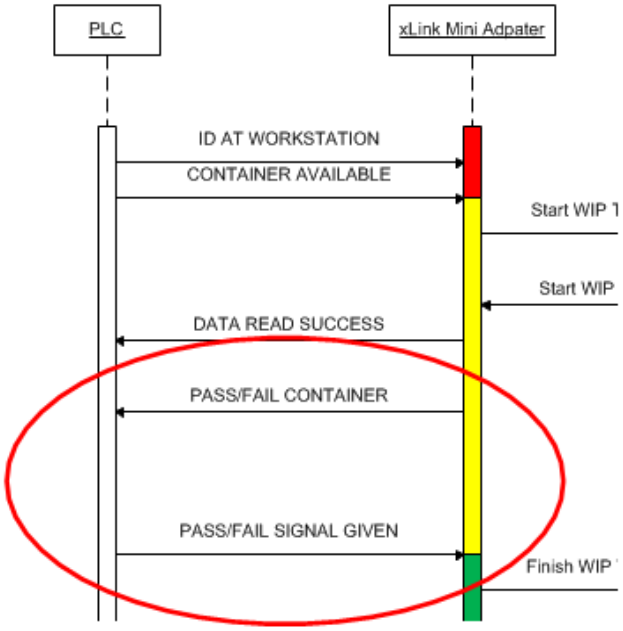
Pass/Fail

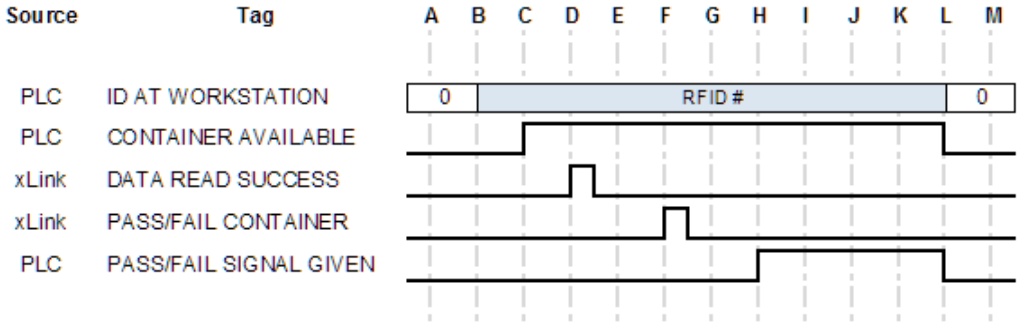
In those configurations where the Pass/Fail is set by the PLC either the Pass Signal Given or Fail Signal Given tag is set by the PLC. This signal change triggers a corresponding event in the mini adapter that performs the WIP Finish operation.





In those configurations where the Pass/Fail is set by the Mini Adapter the sequence of events is modified to include one extra communication so the mini adapter tells the PLC the status to report back to the mini adapter:





This ensures the same WIP Finish event is triggered no matter which entity determines the Pass/Fail status of the WIP operation.

Button Press Required

A work station may be configured with a physical button that must be pressed before a pallet may leave the station.

On station start-up the mini adapter sends the PLC the station configuration.

If a physical button is present at the work station and the operator must press this button to trigger a pallet to leave then on station start-up the mini adapter sends the PLC:

Activate Release Button

Release Button Required

Otherwise, if no physical button is present then:

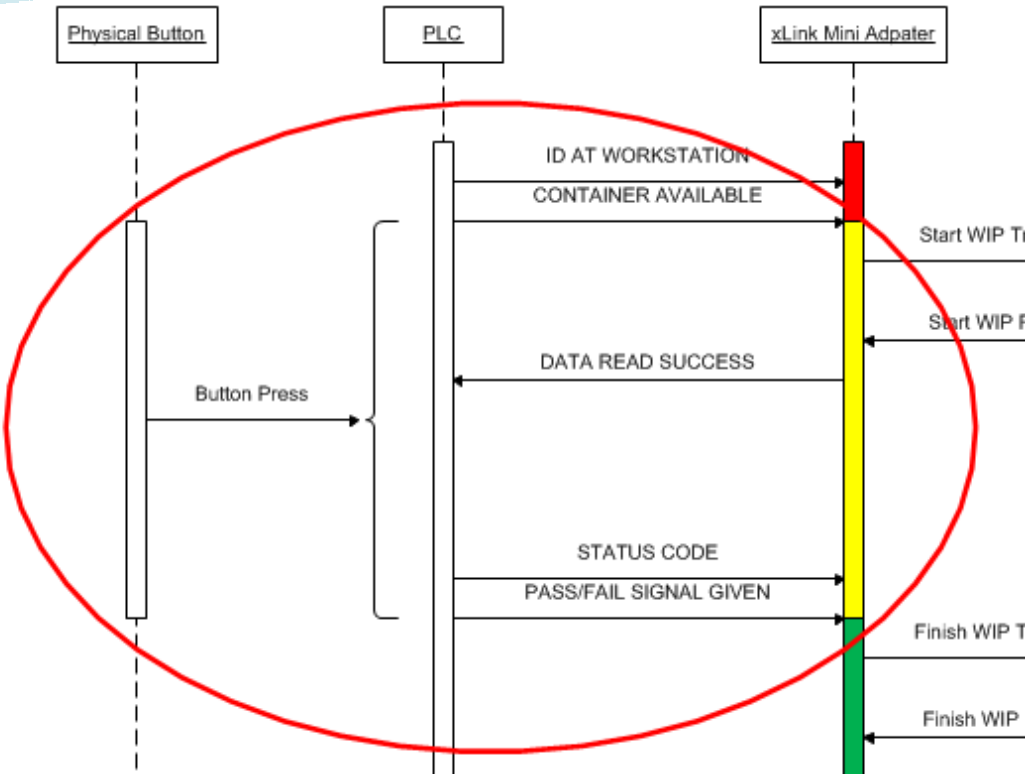
Activate Release Button

Release Button Not Required

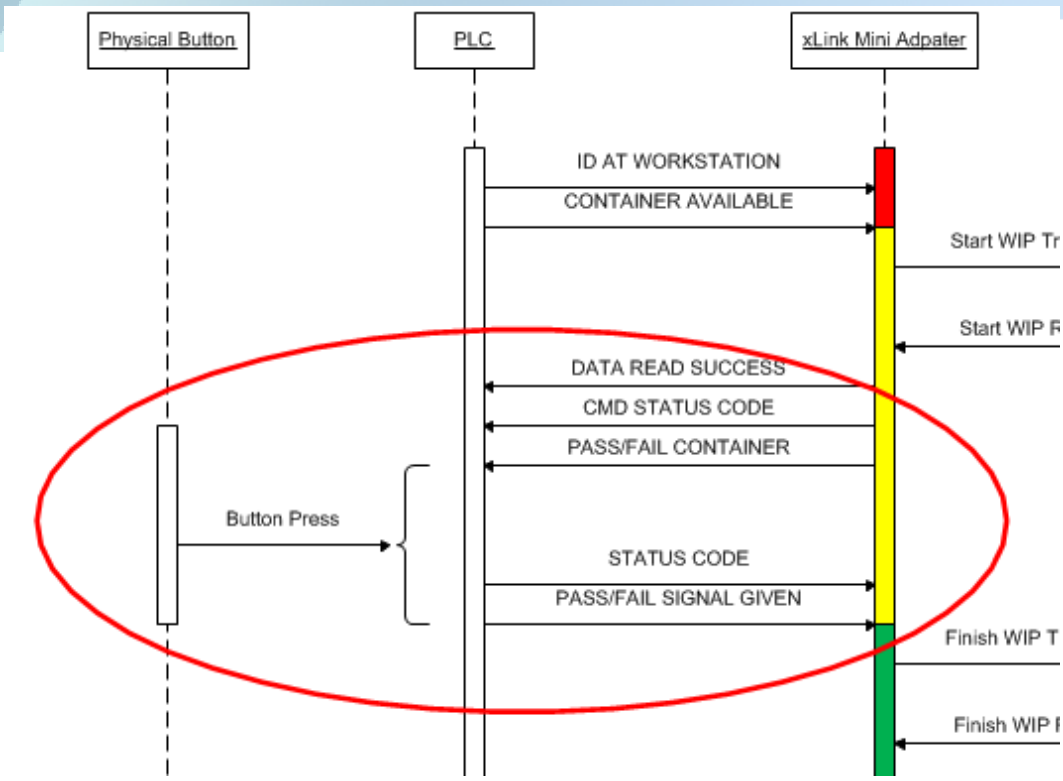
The PLC then uses the Release Button Required (HS_WS_CMD[Z].9)/ Release Button Not Required (HS_WS_CMD[Z].10) value to set the Release Button Required status (HS_WS_STATUS[Z].6) value.

If the Release Button Required status (HS_WS_STATUS[Z].6) value is set high then logic within the PLC must be present to disallow the issuing of a Pass/Fail Signal Given until the physical button has been pressed after the Container Available tag was set

high.



Further, in situations where the status of the PASS / FAIL is determined by Youtilize® (and not by the PLC) the PLC shall ensure the physical button is only enabled when the PASS / FAIL CONTAINER message has been received, and is disabled as soon as the PASS / FAIL SIGNASL GIVEN message is issued.

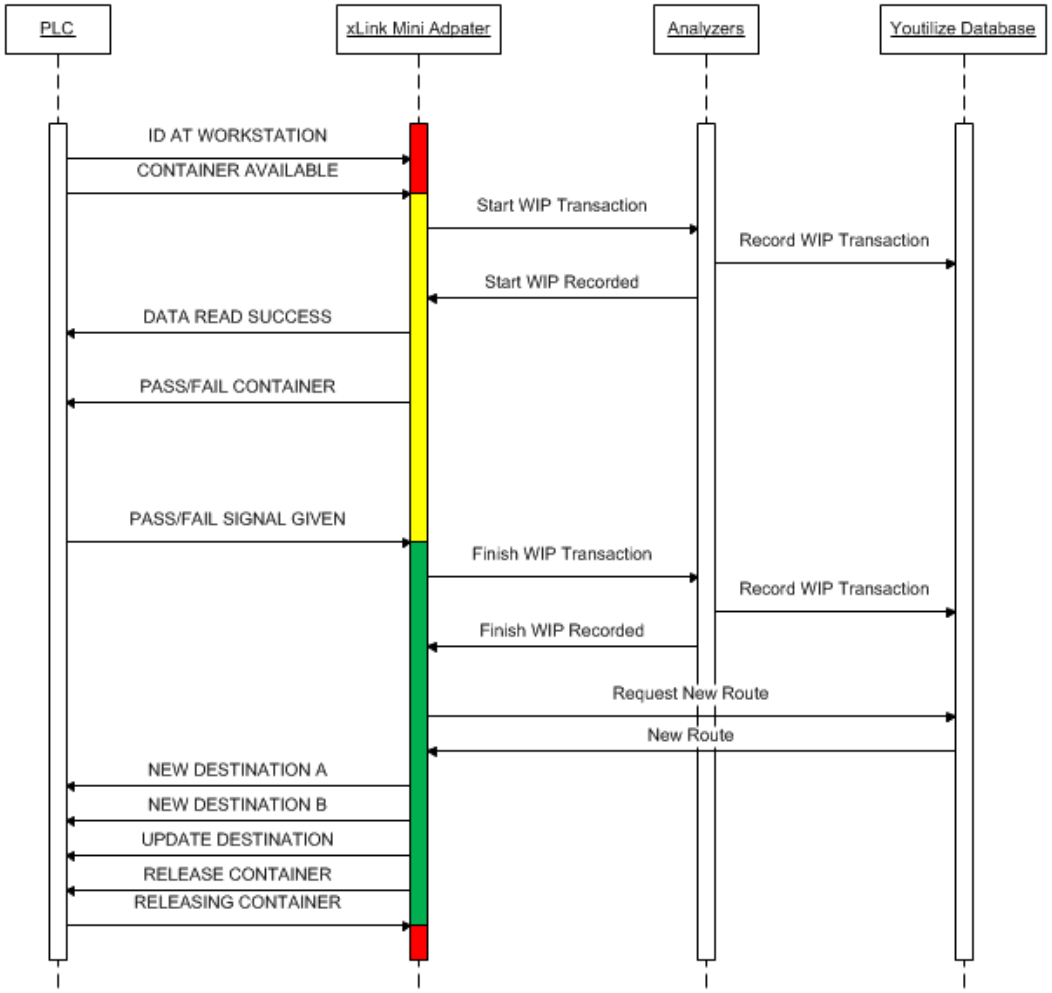


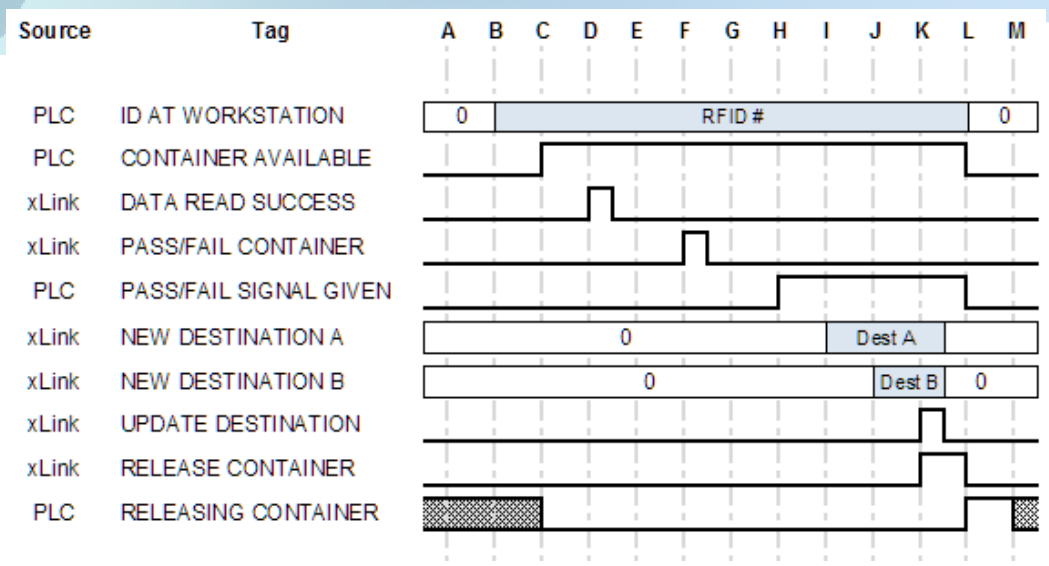
If the Release Button Required status (HS_WS_STATUS[Z].6) value is set low then logic within the PLC shall allow the issuing of a Pass/Fail Signal Given without any button press.

Routing

A work station can be configured so that either the PLC or the mini adapter determines the route that a pallet shall follow. When a mini adapter is created the FLX file is used to determine a RouteId.

- If a RouteId is not present then routing shall be determined by the PLC.
- If a RouteId is present then routing shall be determined by the mini adapter.





When routing is determined by the mini adapter the PLC must wait until the mini adapter issues a Release Container by setting HS_WS_CMD[Z].1 high. This is done when the mini adapter has entered the WIP Finish state. More specifically once the WIP transaction has been posted to the Analyzers the mini adapter shall query the Fusion database for the next destination(s) for the container. The next destination may be a single workstation or multiple workstations. The New Destination registers (HS_WS_NEW_DESTA[Z] & HS_WS_NEW_DESTB[Z]) are used to convey route information from a mini adapter to the PLC. Each destination is given a route id consisting of a leading letter, a period and an integer number. Destination A.1 is then mapped to the first bit in the New Destination A register. This continues through A.32 after which the next route id shall be B.1. With this schema one PLC is capable of servicing up to 64 route points.

Once the mappings have been written to the New Destination registers the Update Destination tag (HS_WS_CMD[Z].6) is set high, and lastly the Release Container (HS_WS_CMD[Z].1) is set high.

The PLC reacts to the Release Container (HS_WS_CMD[Z].1) being high by releasing the pallet from the work station and issuing a Releasing Container (HS_WS_STATUS[Z].1). The mini adapter receives this status event and changes its state to Unknown to signal that the work station is empty. The PLC must clear the New Destination ‘x’ (HS_WS_NEW_DESTx[Z]) and Command (HS_WS_CMD[Z]) tags after processing.

Aborting an operation

If a container is present at a workstation and the operation needs to be aborted (as opposed to completing with either a Pass or Fail status) the PLC shall set the Container Available tag low.

Solutions Design

FlexLink conveyor solutions are designed and built in a modular manner using a standard set of components with known behaviors. The software solution to link Youtilize® to the hardware, via the PLC, has been designed in a similar manner. The tag structure described in this document also allows for a variety of configurations including creating work stations that may or may not require a physical button to release product, where the status of the product can be determined by either party (Youtilize® or the PLC), where either party can be responsible for product routing, etc.

Just as consideration must be given to the components that can be utilized in a prospective conveyor design, the same consideration must be given to the software capabilities and limitations.

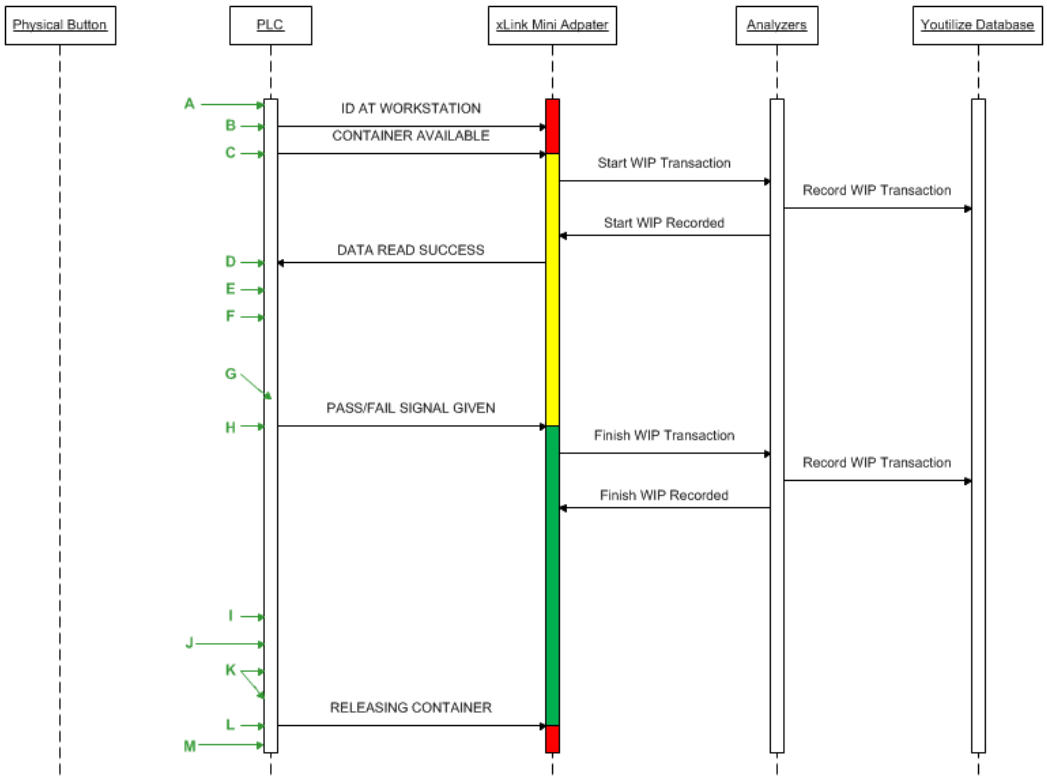
For example, the tag structure and protocol does not allow for the buffering of RFID values – therefore when designing a workstation that must identify the product an RFID scanner must be immediately prior to the workstation and must only allow product to pass this scanner when the workstation is ready to accept product.

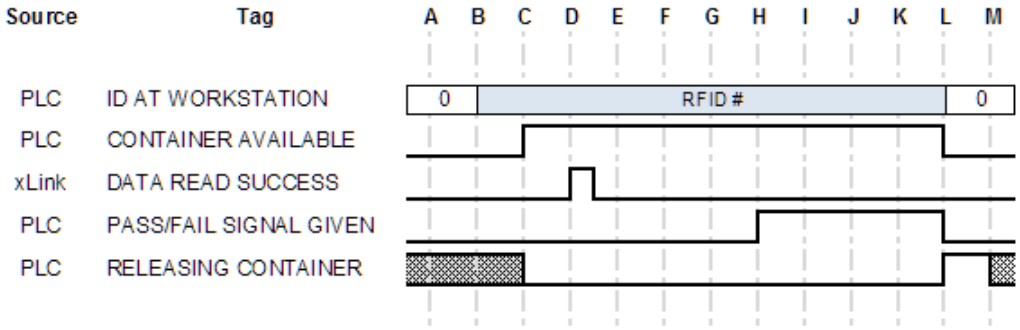
Sample solutions

The following Use Cases have been included to demonstrate how the design ensures the communication between Youtilize® and the PLC is consistent whilst the physical workstation design is highly configurable.

Simple Workstation, no routing and PLC sets product status

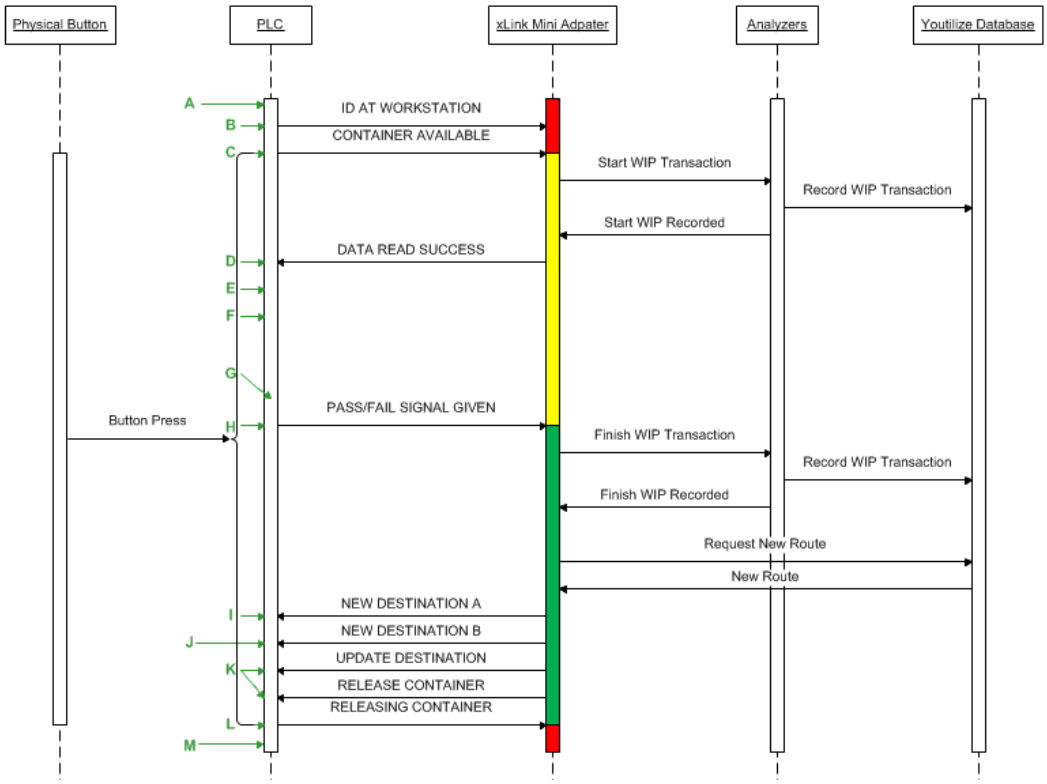
Workstation is to accept one unit at a time. Each unit shall be worked upon and then assigned a PASS or FAIL status by the PLC. Once the unit status has been determined the product shall be automatically released to the line. Youtilize® is simply monitoring the activities on the line; it is not influencing the activities by setting status values or providing route information.

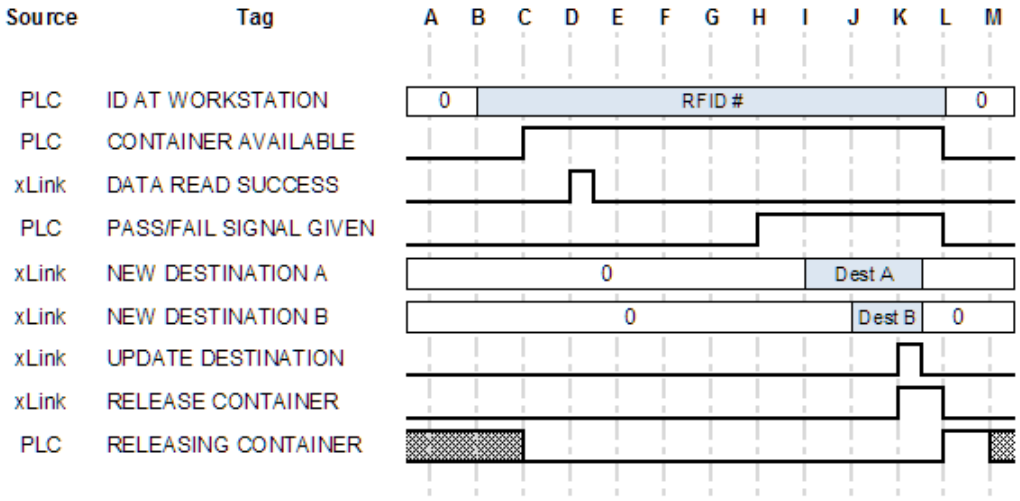




Workstation with routing, PLC sets status and button press required

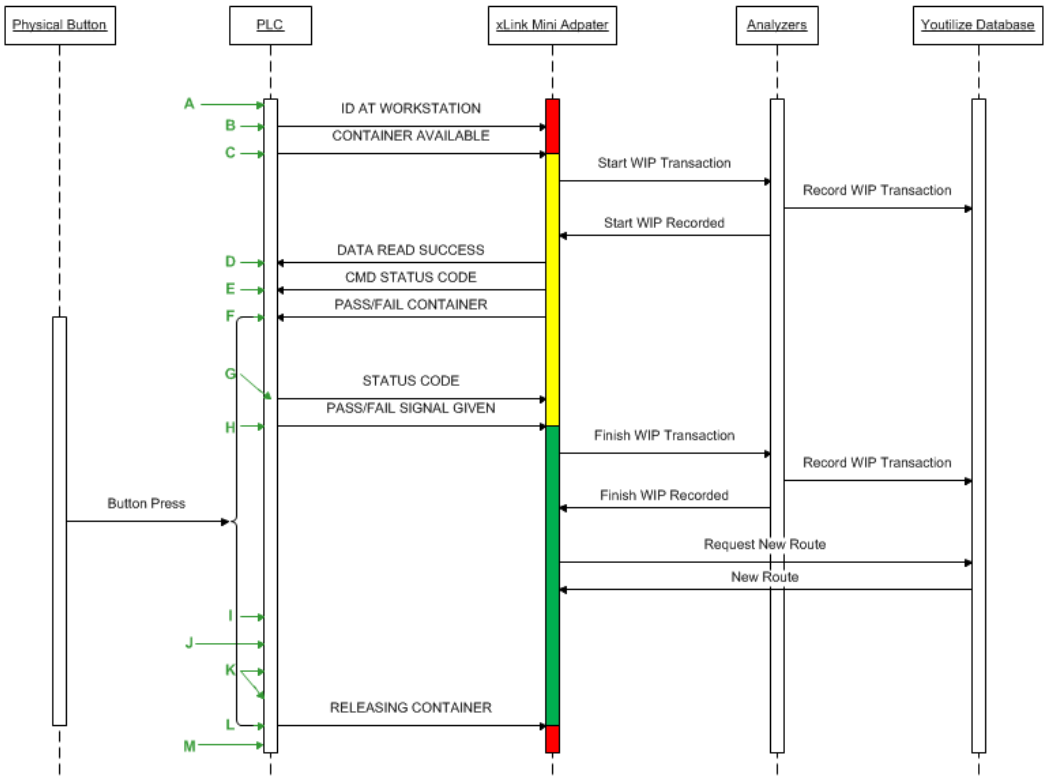
In this scenario the PLC shall set the unit status but not allow the unit to leave the workstation until the operator signals their readiness (via a Button press).

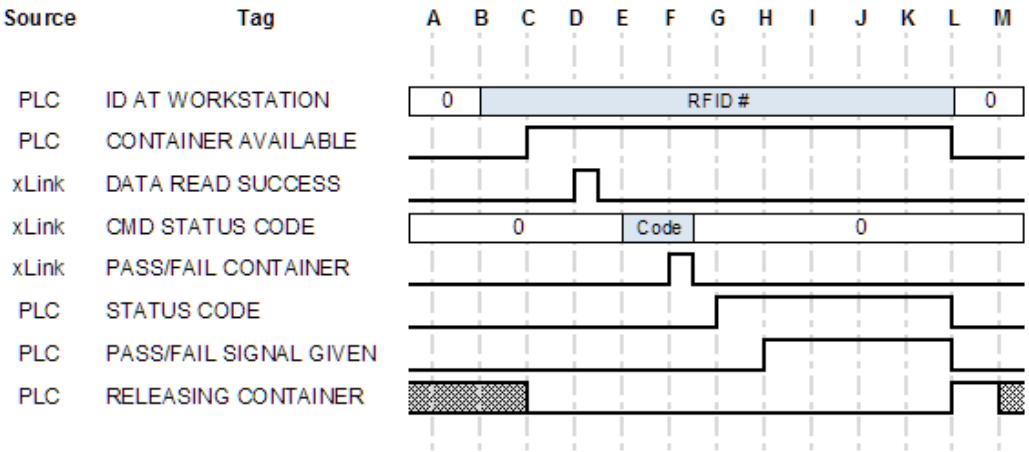




Workstation without routing; Youtilize® sets status code and button press required

In this scenario Youtilize® shall inform the PLC of the unit status code once work upon it has been completed. Before the PLC may release the unit the operator must press the release button.





Youtilize® Service Levels

Youtilize® solutions come in three flavours:

- Service Monitoring
- Production Monitoring
- Routing

The service levels build upon each other with Routing containing the full scope of functionality as described in this document.

	Service Monitoring	Production Monitoring	Routing
System Data			
Motor Data	X	X	X
Function Data		X	X
Workstation Data			X

Conclusion

The communication protocol described herein has been designed to be as flexible as possible whilst also being standardized. When designing systems that shall utilize this protocol a few limitations do exist and should be factored in from the outset – although the number of route points shall not be a restriction for much longer. Changes to this design shall result in the Protocol Revision (in the System Wide Data section) needing to be implemented so that multiple versions of the protocol can be implemented by the same code base. However the use of multiple protocol revisions should be kept to an absolute minimum.

FlexLink®

www.flexlink.com

A Company of coesia GROUP

Publication 5117029 EN 2012-10