

**KOYO Communication Driver**

Driver for *Serial* or *Ethernet* communication  
with Automation Direct - DirectLogic devices

**Contents**

**INTRODUCTION .....2**

**GENERAL INFORMATION.....3**

    DEVICE SPECIFICATIONS.....3

    NETWORK SPECIFICATIONS.....3

    DRIVER CHARACTERISTICS .....3

    CONFORMANCE TESTING .....4

**SELECTING THE DRIVER .....6**

**CONFIGURING THE DEVICE .....7**

**CONFIGURING THE DRIVER .....8**

    CONFIGURING THE COMMUNICATION SETTINGS .....8

    CONFIGURING THE DRIVER WORKSHEETS ..... 10

*Main Driver Sheet*..... 10

*Standard Driver Worksheets* ..... 13

**EXECUTING THE DRIVER ..... 19**

**TROUBLESHOOTING ..... 20**

**SAMPLE APPLICATION ..... 22**

**REVISION HISTORY..... 23**

## Introduction

The KOYO driver enables communication between the Studio system and some of the Automation Direct devices using their proprietary protocol by Serial or Ethernet, according to the specifications discussed in this document.

This document will help you to select, configure and execute the KOYO driver, and it is organized as follows:

- **Introduction:** This section, which provides an overview of the document.
- **General Information:** Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver:** Explains how to select the KOYO driver in the Studio system.
- **Configuring the Device:** Describes how the target device must be configured to receive communication from the KOYO driver.
- **Configuring the Driver:** Explains how to configure the KOYO driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver:** Explains how to execute the KOYO driver during application runtime.
- **Troubleshooting:** Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application:** Explains how to use a sample application to test the KOYO driver configuration
- **Revision History:** Provides a log of all changes made to the driver and this documentation.



### Notes:

- This document assumes that you have read the “Development Environment” chapter in Studio's *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows environment. If you are not familiar with Windows, then we suggest using the **Help** feature as you work through this guide.

## General Information

This chapter identifies all of the hardware and software components required to implement *Serial* or *Ethernet* communication between the KOYO driver in Studio and DirectLogic devices.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

### Device Specifications

To establish communication, your target device must meet the following specifications:

- **Manufacturer:** Automation Direct
- **Compatible Equipment:**
  - DirectLogic DL05 and DL06 PLCs, with H0-ECOM Ethernet module
  - DirectLogic DL205 Family (DL240/DL250/DL260 CPUs), with H2-ECOM Ethernet module
  - Any device that is compatible with the Automation Direct's protocol.
- **Automation Direct PLC programmer software:** DIRECTSOFT32

For a description of the device(s) used to test driver conformance, see “Conformance Testing”.

### Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device communication port:** Ethernet (in the ECOM module) or COM Ports
- **Physical protocol:** Ethernet or Serial
- **Logic protocol:**
  - **For Ethernet:** IPX or UDP/IP ECOM protocol
  - **For Serial:** Proprietary protocol
- **Adapters/Converters:**
  - **For Ethernet:** Hx-ECOM Ethernet modules from AutomationDirect
  - **For Serial:** None
- **Specific PC Board:** Ethernet board or Serial Port
- **Device Runtime software:** None
- **Cable Wiring:**
  - **For Ethernet:** Ethernet cable
  - **For Serial:** AutomationDirect cable compatible with D2-DSCBL

### Driver Characteristics

The KOYO driver package consists of the following files, which are automatically installed in the `/DRV` subdirectory of Studio:

- **KOYO.INI:** Internal driver file. *You must not modify this file.*
- **KOYO.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file.*
- **KOYO.PDF:** This document, which provides detailed information about the KOYO driver.
- **KOYO.DLL:** Compiled driver.

**Note:**  
 You must use Adobe Acrobat® Reader™ to view the KOYO . PDF document. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe’s Web site.

You can use the KOYO driver on the following operating systems:

- Windows XP or Newer
- Windows Embedded / CE

For a description of the operating systems used to test driver conformance, see “Conformance Testing” below.

The KOYO driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float
X or V (Input Point)	1 Bit	–	•	•	•	•
Y or V (Output Point)	1 Bit	•	•	•	•	•
C or V (Control Relay)	1 Bit	•	•	•	•	•
SP or V (Special Relay)	1 Bit	•	•	•	•	•
V (Timer Current Value)	2 Bytes	•	•	•	•	•
T or V (Timer Status Bits)	1 Bit	•	•	•	•	•
V (Counter Current Value)	2 Bytes	•	•	•	•	•
CT or V (Counter Status Bits)	1 Bit	•	•	•	•	•
V (Data Words)	2 Bytes	•	•	•	•	•
V (Stages)	2 Bytes	•	•	•	•	•
V (System Parameters)	2 Bytes	•	•	•	•	•

Using the exported CSV file the following additional types are supported through the V register:

GX, GY, TA, CTA, VGX, VGY, VX, VY, VC, VT, VCT

### **Conformance Testing**

The following hardware/software was used for conformance testing:

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>▪ <b>Configuration #1:</b> <ul style="list-style-type: none"> <li>- PLC: DirectLogic DL05 PLC</li> <li>- Ethernet Module: H0-ECOM</li> <li>- Module Port: 10Base-T Ethernet Port</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>▪ <b>Configuration #2:</b> <ul style="list-style-type: none"> <li>- PLC: DirectLogic 205 Family</li> <li>- CPU Module: DL250 CPU</li> <li>- Ethernet Module: H2-ENET</li> <li>- Module Port: 10Base-T Ethernet Port</li> </ul> </li> </ul> |
|--|---|

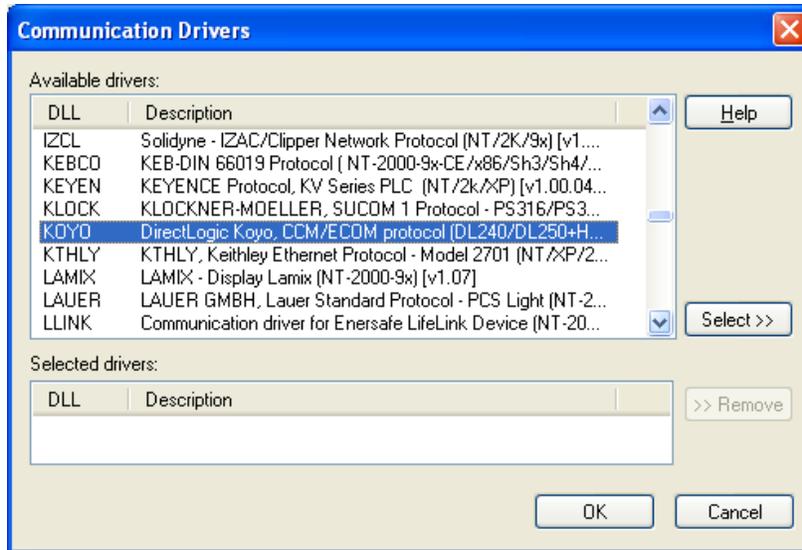
**Cables:** Ethernet cable or D2-DSCBL.

Driver Version	Studio Version	Operating System (development)	Operating System (target)	Equipment
1.21	v8.0 + SP1	Windows 7/8	Windows 7/8 WinCE 7.0 ArmV4i	- DL205 CPU DL260 Ethernet Module H2-ENET

## Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the `\DRV` subdirectory but they remain dormant until manually selected for specific applications. To select the KOYO driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **KOYO** driver from the *Available Drivers* list, and then click the **Select** button.



**Communication Drivers Dialog**

  When the **KOYO** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

 **Note:**

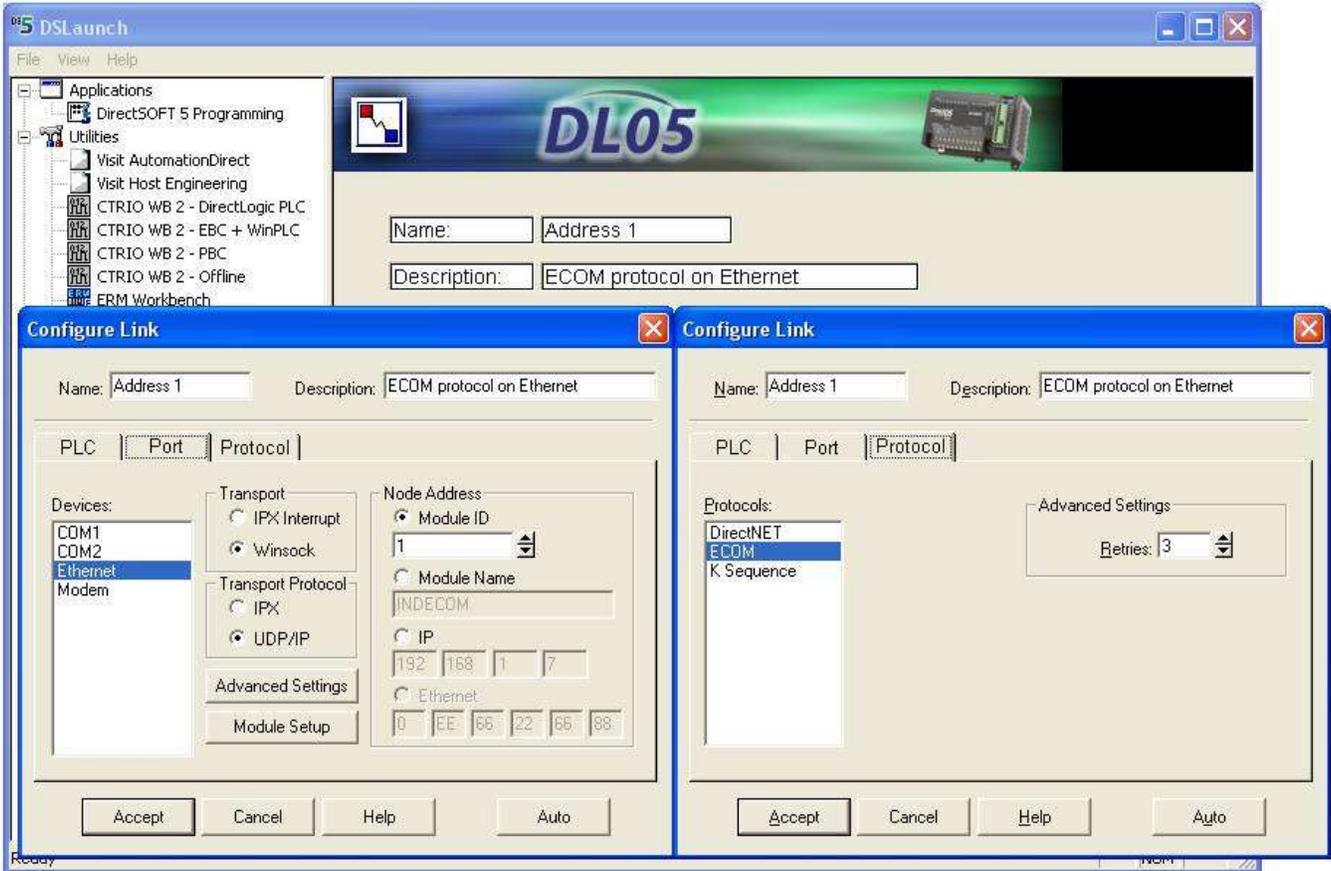
It is not necessary to install any other software on your computer to enable communication between Studio and your target device. However, this communication can only be used by the Studio application; it cannot be used to download control logic to the device. To download control logic to a KOYO device, you must also install the KOYO programming software (e.g., DIRECTSOFT32). For more information, please consult the documentation provided by the device manufacturer.

 **Attention:**

For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

## Configuring the Device

The KOYO device (DirectLogic / AutomationDirect) must be configured to exchange data through the installed Ethernet module using the ECOM protocol.



**Selecting the Communication Protocol in the PLC Configuration Software**

## Configuring the Driver

Once you have selected the KOYO driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver’s communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

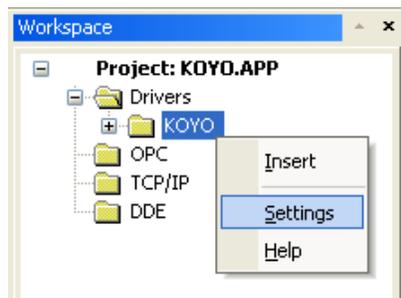
### Configuring the Communication Settings

The communication settings are described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only KOYO driver-specific settings and procedures will be discussed here. To configure the communication settings for the KOYO driver:

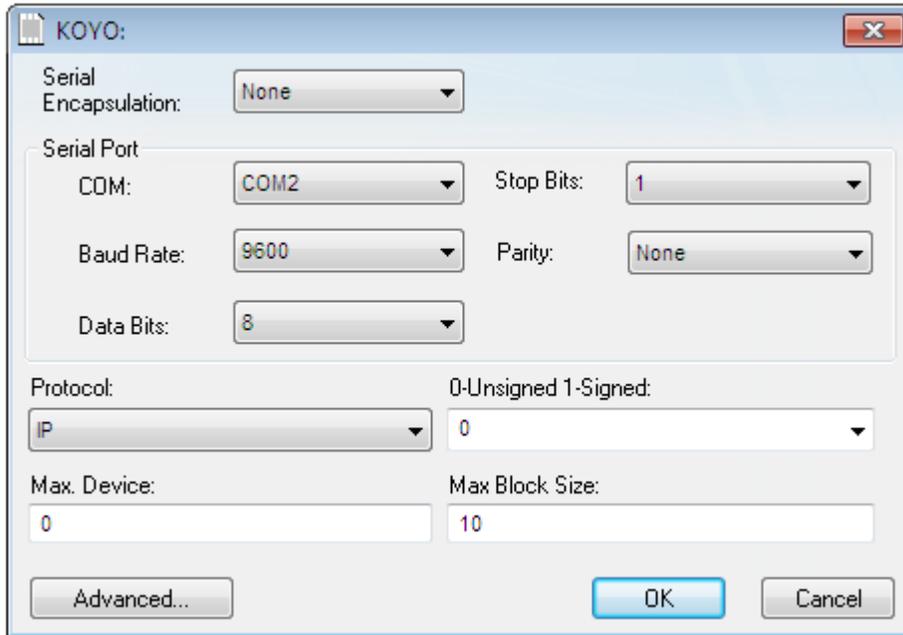
 In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The KOYO driver is listed here as a subfolder.

 Right-click on the *KOYO* subfolder and then select the **Settings** option from the pop-up menu:



**Select Settings from the Pop-Up Menu**

The KOYO: Communication Settings dialog is displayed:



**KOYO: Communication Parameters Dialog**

 In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. To ensure error-free communication, the driver settings must *exactly match* the corresponding settings on the device. Please consult the manufacturer’s documentation for instructions how to configure the device and for complete descriptions of the settings.

Depending on your circumstances, you may need to configure the driver *before* you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

**⚠ Attention:**

For safety reasons, you **must** take special precautions when connecting and configuring new equipment. Please consult the manufacturer’s documentation for specific instructions.

The communication settings and their possible values are described in the following table:

Parameter	Default Value	Valid values	Description
<b>Protocol</b>	0	IP, IPX or Serial	Transport protocol used in the Ethernet or Serial configuration of the computer.
<b>Max. Device</b>	0	0 to 40	Maximum number of DirectLOGIC devices included in the network. If leave it with zero it assumes the value of 20. Any positive number will be accepted. It is safe to use (tested and approved) from 0 to 40.
<b>0-Unsigned 1-Signed</b>	0	0 or 1	Set type of data. (Unsigned or Signed)

<b>Max Block Size</b>	255	1 to 255	Set Max number of bytes that will be asked for the PLC on a single request. H2-ECOM100: 128 bytes or lower H0 & H4-ECOM100: 254 bytes or lower.
-----------------------	-----	----------	---

 **Attention:**  
IPX communication is not supported for the Windows CE operating system.

 **Note:**  
The device must be configured with *exactly the same* parameters that you configured in the *KOYO: Communication Parameters* dialog.

 **Note:**  
Additional communication settings can be accessed in the *Advanced Settings* dialog. To open this dialog, simply click the **Advanced** button in the *Communication Settings* dialog.  
  
Simultaneous connection are supported on this driver for configuring this settings please check the “Simultaneous Requests” field on the advanced settings dialog. The maximum supported simultaneous connections are 32 and 16 per station.

## Configuring the Driver Worksheets

Each selected driver includes a Main Driver Sheet and one or more Standard Driver Worksheets. The Main Driver Sheet is used to define tag/register associations and driver parameters that are in effect at all times, regardless of application behavior. In contrast, Standard Driver Worksheets can be inserted to define additional tag/register associations that are triggered by specific application behaviors.

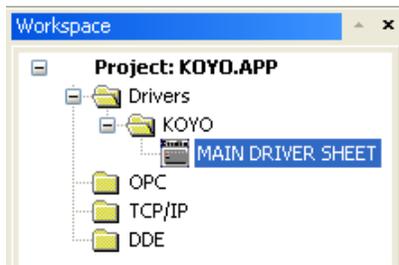
The configuration of these worksheets is described in detail in the “Communication” chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only KOYO driver-specific parameters and procedures are discussed here.

### Main Driver Sheet

When you select the KOYO driver and add it to your application, Studio automatically inserts the *Main Driver Sheet* in the KOYO driver subfolder. To configure the Main Driver Sheet:

1. Select the *Comm* tab in the *Workspace* pane.
2. Open the *Drivers* folder, and then open the *KOYO* subfolder:



**Main Driver Sheet in the KOYO Folder**

3. Double-click on the **MAIN DRIVER SHEET** icon to open the following worksheet:

	Tag Name	Station	I/O Address	Action	Scan	Div	Add
1	V0001	N:UNIT1	V.0	Read+Write	Always		
2	V0001bit	N:UNIT1	V.0.1	Read+Write	Always		
3	X010	N:UNIT1	X10	Read+Write	Always		
*				Read+Write	Always		

**Main Driver Worksheet**

Most of the fields on this sheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station** and **I/O Address** fields use syntax that is specific to the KOYO driver.

4. For each table row (i.e., each tag/register association), configure the **Station** and **I/O Address** fields as follows:
  - **Station** field: Specify the name or ID number of the target device and optionally the path relative to the application folder.

#### **For Ethernet communication**

Use the following syntax:

**N:***<Device Name>*[;*<Path>*] or **I:***<ID Number>*[;*<Path>*] or  
*<IP>*[;*<Path>*]

Examples — **N:**UNIT1 or **I:**1 or 192.168.1.123 or  
192.168.1.123;.\koyo.csv

Where:

**<Device Name>** is Module Name configured in the device. It is used only for local network area.

**<ID Number>** is Module ID configured in the device. It is used only for local network area.

**<IP>** is Module IP Address. It is used for local and between networks. For this option the communication parameters must be configured with **IP** protocol.

**<Path>** is the relative or absolute path to the application folder, and is optional.

 **Note:**

The Module Name or Module ID must be specified when a Broadcast router is used. For this case, use the syntax below:

**<IP>**:**N:***<Device Name>* or **<IP>**:**I:***<ID Number>*

 **Note:**

The CSV file exported contains the nicknames mapping to register addresses and the associated type. This file is generated from the programming software used.

#### **For Serial communication**

Use the following syntax:

**<ID Number>**

Example — 1

The specified name or number must match the name or number configured on the target device using the programmer software (e.g., DIRECTSOFT32).

You can also specify an indirect tag (e.g. {station}), but the tag that is referenced must follow the same syntax and contain a valid value.

- **I/O Address** field: Specify the address of the associated device register.

For **Discrete memory** registers (X, Y, C, SP, T, CT, and S), use the following syntax:

**<Type>: <Address>**

Example — **X: 5**

For **Word memory** registers (V) in decimal format, use the following syntax:

**<Type>: <Address> . [Bit]**

Example — **V: 40400 . 5**

For **Word memory** registers (V) in BCD or floating point format, use the following syntax:

**<Type>: <Address>: <Format>**

Example — **V: 1000: B, V: 1000: F, V: 1000: LB, V: 1000: D, V: 200: S, V: 300: SD, V: 500: A. 20, V: 1000: LBS**

Where:

- **<Type>** : Register type. Valid values are **X** (Input Point - Bit), **Y** (Output Point - Bit), **C** (Control Relay - Bit), **SP** (Special Relay - Bit), **T** (Timer), **CT** (Counter), **S** (Stage), or **V** (V memory - Word).
- **<Address>** : Address of the device register. The address is always configured in OCTAL format.
- **[Bit]** (Optional): Bit number (**0** to **15**) from the word address.
- **<Format>**: Word format. Valid values are **B** (BCD), **LB** (Long BCD - 2 Words), **F** (floating point, for two words), **D** (DWORD), **S** (signed word (16 bits)), **SD** (signed double word (32 bits)), **A. #** (ASCII string of # characters (not zero ended)), **LBS** (Long BCD Swap – 2 Words).
- **<Nickname>**: String used to reference the address in the PLC program.

For **Tags read from the csv file**, use the following syntax:

**TAG:<Nickname>**

Example – **TAG:MYINPUT**

 **Note:**

If you change the CSV file with the driver running, the changes will only be accepted after you stop your application and start it again.

To use tags from the CSV file for communication, the following formats have to be used instead of the format that the driver uses for the datatypes supported.

CSV Datatype format	Driver Datatype
---------------------	-----------------

	format
BCDI16	B
BCDI32	LB
SI16	S
SI32	SD
UI16	UI16
UI32	D

Examples of tags in CSV file:

- "V400","MyTagV400","","Test Tag 1"  
This is the default: <Type>, <Tag>, <>, <Nickname>
- "V400","MyTagV400","","Test Tag 1","BCDI32"  
This is the tag entry in CSV file when a datatype format has to be specified like BCDI32. If no format is specified as in the previous example the tag is treated as a BCDI16 by default.
- "CTA0","MyTagCTA0","","","BCDI16"

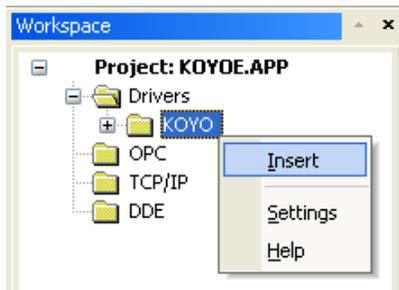
## Standard Driver Worksheets

When you select the KOYO driver and add it to your application, it has only a Main Driver Sheet by default (see previous section). However, you may insert additional Standard Driver Worksheets to define tag/register associations that are triggered by specific application behaviors. Doing this will optimize communication and improve system performance by ensuring that tags/registers are scanned only when necessary – that is, only when the application is performing an action that requires reading or writing to those specific tags/registers.

**Note:**  
 We recommend configuring device registers in sequential blocks in order to maximize performance.

To insert a new Standard Driver Worksheet:

-  In the *Comm* tab, open the *Drivers* folder and locate the *KOYO* subfolder.
-  Right-click on the *KOYO* subfolder, and then select **Insert** from the pop-up menu:



**Inserting a New Worksheet**

A new KOYO driver worksheet is inserted into the *KOYO* subfolder, and the worksheet is opened for configuration:

**KOYO001.DRV**

Description:   Increase priority

Read Trigger:  Enable Read when Idle:  Read Completed:  Read Status:

Write Trigger:  Enable Write on Tag Change:  Write Completed:  Write Status:

Station:  Header:   Min:   
 Max:

	Tag Name	Address	Div	Add
1	Tag1	0		
2	Tag2	1		
3	Tag3	5		
*				

**KOYO Driver Worksheet**

 **Note:**  
Worksheets are numbered in order of creation, so the first worksheet is `KOYO001.drv`.

Most of the fields on this worksheet are standard for all drivers; see the “Communication” chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the KOYO driver.

 Configure the **Station** and **Header** fields as follows:

- **Station** field: Specify the name or ID number of the target device.

**For Ethernet communication**

Use the following syntax:

**N:<Device Name>** or **I:<ID Number>** or **<IP>**

Examples — **N: UNIT1** or **I: 1** or **192.168.1.123**

Where:

**<Device Name>** is Module Name configured in the device. It is used only for local network area.

**<ID Number>** is Module ID configured in the device. It is used only for local network area.

**<IP>** is Module IP Address. It is used for local and between networks. For this option the communication parameters must be configured with **IP** protocol.

 **Note:**  
The Module Name or Module ID must be specified when a Broadcast router is used. For this case, use the syntax below:  
**<IP>:N:<Device Name>** or **<IP>:I:<ID Number>**

**For Serial communication**

Use the following syntax:

**<ID Number>**

Example — **1**

The specified name or number must match the name or number configured on the target device using the programmer software (e.g., DIRECTSOFT32).

You can also specify an indirect tag (e.g. {station}), but the tag that is referenced must follow the same syntax and contain a valid value.

- **Header** field: Specify the address of the first register of a block of registers on the target device. The addresses declared in the *Body* of the worksheet are simply offsets of this **Header** address. When Read/Write operations are executed for the entire worksheet (see **Read Trigger** and **Write Trigger** above), it scans the entire block of registers from the first address to the last.

The **Header** field uses the following syntax:

**<Type>**: **<optAddressReference>**

Where **<Type>** is the register type. Valid values are **X** (Input Point – Bit), **Y** (Output Point – Bit), **C** (Control Relay – Bit), **SP** (Special Relay – Bit), **T** (Timer), **CT** (Counter), **S** (Stage), or **V** (V memory – Word)

**<optAddressReference>** is the initial address (reference) of the configured type. This parameter is optional and, if configured, will be used as initial reference for the addresses configured in the **Address** field

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is invalid, then Studio automatically inserts a default value of **x**.

You can also specify an indirect tag (e.g. {**header**}), but the tag that is referenced must follow the same syntax and contain a valid value.

The following table lists all the data types and address ranges that are valid for the KOYO driver.

Data Types	Sample Syntax	Valid Range of Initial Addresses per Worksheet	Comments
Input Points	X	X0 – X777	<b>Input Points</b> in the bit data format
	V	V40400 – V40437	<b>Input Points</b> in the word data format
Output Points	Y	Y0 – Y777	<b>Output Points</b> in the bit data format
	V	V40500 – V40537	<b>Output Points</b> in the word data format
Control Relays	C	C0 – C1777	<b>Control Relays</b> in the bit data format
	V	V40600 – V40677	<b>Control Relays</b> in the word data format
Special Relays	SP	SP0 – SP777	<b>Special Relays</b> in the bit data format
	V	V41200 – V41237	<b>Special Relays</b> in the word data format
Timer Current Value	V	V0 – V377	<b>Timer Current Values</b> in the word data format
Timer Status Bits	T	T0 – T377	<b>Timer Status Bits</b> in the bit data format
	V	V41100 – V41117	<b>Timer Status Bits</b> in the word data format
Counter Current Value	V	V1000 – V1177	<b>Counter Current Values</b> in the word data format
Counter Status Bits	T	CT0 – CT177	<b>Counter Status Bits</b> in the bit data format
	V	V41140 – V41147	<b>Counter Status Bits</b> in the word data format
Data Words	V	V1400 – V7377 V10000 – V17777	<b>Data Words</b> in the word data format
Stages	S	S0 – S1777	<b>Stages</b> in the bit data format
	V	V41000 – V41077	<b>Stages</b> in the word data format
System Parameters	V	V7400 – V7777 V37000 – V37777	<b>System Parameters</b> in the word data format

**Note:**  
 Most of the register types can be treated as either bit or word data. When the Discrete Memory syntax (X, Y, C, SP, T, CT or S) is used, the bits are always addressed individually. When the Word Memory syntax (V) is used, the word memory (32 bits) is addressed.

- For each table row (i.e., each tag/register association), configure the **Address** parameter as described below. The **Address** parameter is appended to the **Header** parameter configured in the worksheet header above to generate a complete register address.

For **Header** parameter with value of **X, Y, C, SP, T, CT, or S**, uses the following syntax:

**<Address> or <Address Offset>** to the address reference configured in the header (if any)

Example — 5

For **Header** parameter with value of **V**, use the following syntax:

**<Address>.[Bit] or <Address>:<optFormat>**

Example — 40400.5; 1000: B; 1000: F; 1400: LB; 1000: D,200: S,250: SD,300: A.10, 1400: LBS

Where:

- **<Address>** : Address of the device register or, if a initial address was configured in the **Header** field, this address will be the offset to the reference. The address is always configured in OCTAL format.
- **[Bit]** (optional): Bit number (**0 to 15**) from the word address.
- **<optFormat>** : Format. This parameter is optional. If nothing is configured, addresses will be read as Words. Valid values are **B** (BCD), **F** (floating point, using two consecutive words) and **LB** (Long BCD, using two consecutive words), **D** (DWORD, using two consecutive words), **S** (signed word (16 bits)), **SD** (signed double word (32 bits)), **A.#** (ASCII string of # characters (not zero ended)), LBS (Long BCD Swap – 2 Words).

For examples of how device registers are specified using **Header** and **Address**, see the following table:

Register Type	Address on the Device (octal)	Header Field	Address Field (octal)
Input Points	X0000	<b>X</b>	0
		<b>V</b>	40400.0
	X0030	<b>X</b>	30
		<b>V: 40400</b>	1.8
	X0000 to X0017 (1 word)	<b>V</b>	40400
Output Points	Y0007	<b>Y</b>	7

Register Type	Address on the Device (octal)	Header Field	Address Field (octal)
		<b>V: 40500</b>	<b>0.7</b>
	Y0026	<b>Y</b>	<b>26</b>
		<b>V</b>	<b>40501.6</b>
	Y0020 to Y0037 (1 word)	<b>V: 40500</b>	<b>1</b>
Control Relays	C0010	<b>C</b>	<b>10</b>
		<b>V</b>	<b>40600.8</b>
	C0053	<b>C</b>	<b>53</b>
		<b>V</b>	<b>40602.11</b>
	C0000 to C0017 (1 word)	<b>V: 40600</b>	<b>0</b>
Special Relays	SP0017	<b>SP</b>	<b>17</b>
		<b>V</b>	<b>41200.15</b>
	SP0020	<b>SP</b>	<b>20</b>
		<b>V</b>	<b>41201.0</b>
	SP0040 to SP0057 (1 word)	<b>V</b>	<b>41202</b>
Timer Current Value	V0000 – Decimal Format	<b>V</b>	<b>0</b>
	V0000 – BCD Format	<b>V</b>	<b>0:B</b>
	V0000 and V0001 – Float Format	<b>V</b>	<b>0:F</b>
	V0012 – Decimal Format	<b>V</b>	<b>12</b>
	V0012 – BCD Format	<b>V</b>	<b>12:B</b>
	V0012 and V0013 – Float Format	<b>V</b>	<b>12:F</b>
	V0012 and V0013 – DWORD Format	<b>V</b>	<b>12:D</b>
Timer Status Bits	T0007	<b>T</b>	<b>7</b>
		<b>V</b>	<b>41100.7</b>
	T0021	<b>T</b>	<b>21</b>
		<b>V</b>	<b>41101.1</b>
	T0000 to T0017 (1 word)	<b>V</b>	<b>41100</b>
Counter Current Value	V1000 – Decimal Format	<b>V</b>	<b>1000</b>
	V1000 – BCD Format	<b>V</b>	<b>1000:B</b>
	V1000 and V1001 – Float Format	<b>V</b>	<b>1000:F</b>
	V1012 – Decimal Format	<b>V</b>	<b>1012</b>
	V1012 – BCD Format	<b>V</b>	<b>1012:B</b>

Register Type	Address on the Device (octal)	Header Field	Address Field (octal)
	V1012 and V1013 – Float Format	<b>V</b>	<b>1012 : F</b>
	V1012 and V1013 – DWORD Format	<b>V</b>	<b>1012 : D</b>
Counter Status Bits	CT0007	<b>CT</b>	<b>7</b>
		<b>V</b>	<b>41140 . 7</b>
	CT0021	<b>CT</b>	<b>21</b>
		<b>V</b>	<b>41141 . 1</b>
	CT0000 to CT0017 (1 word)	<b>V</b>	<b>41140</b>
Data Words	V1400 – Decimal Format	<b>V</b>	<b>1400</b>
	V1400 – BCD Format	<b>V : 1400</b>	<b>0 : B</b>
	V1400 and V1401 – Float Format	<b>V</b>	<b>1400 : F</b>
	V12000 – Decimal Format	<b>V : 12000</b>	<b>0</b>
	V12000 and V12001 – Long BCD Format	<b>V</b>	<b>12000 : LB</b>
	V12000 and V12001 – Float Format	<b>V</b>	<b>12000 : F</b>
	V12000 and V12001 – DWORD Format	<b>V</b>	<b>12000 : D</b>
Stages	S0007	<b>S</b>	<b>7</b>
		<b>V</b>	<b>41000 . 7</b>
	S0021	<b>S</b>	<b>21</b>
		<b>V</b>	<b>41001 . 1</b>
	S0000 to S0017 (1 word)	<b>V</b>	<b>41000</b>
System Parameters	V7772 – BCD Format (Day of week)	<b>V : 7770</b>	<b>2 : B</b>
	V7773 – BCD Format (Month)	<b>V</b>	<b>7773 : B</b>
	V7774 – BCD Format (Year)	<b>V</b>	<b>7774 : B</b>

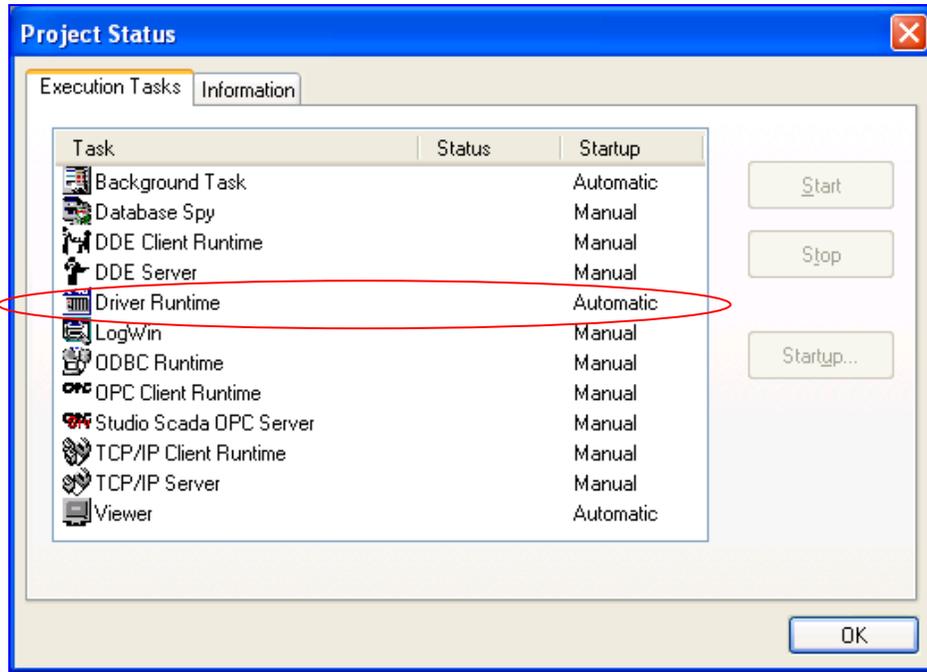
**➡ Attention:**  
 You must not configure a range of addresses greater than the maximum block size (data buffer length) supported within the same worksheet: 64 for V registers or 254 for X, Y, C, SP, T, CT or S registers.

## Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application’s runtime execution settings by checking the *Project Status* dialog.

To verify that the the communication driver(s) will execute correctly:

 From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



*Project Status Dialog*

 Verify that the *Driver Runtime* task is set to **Automatic**.

- If the setting is correct, then proceed to step 3 below.
- If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task’s *Startup* mode to **Automatic**.

 Click **OK** to close the *Project Status* dialog.

 Start the application to run the driver.

## Troubleshooting

If the KOYO driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Main Driver Sheet will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None
1	Block Size Error	Address offset greater than 126 to "V" operands, and greater than 254 to other operands.	Check on the worksheet receiving the error. If the configured offset is greater than the limit, correct it.
2	Invalid Header	An invalid Header has been typed, or the tag has an invalid configuration.	Type a valid Header either on the header field or on the tag value. See the "Header Field Information" table in this document for valid headers.
5	Error opening transport	Transport protocol (IP or IPX) must be installed and configured correctly in the computer.	Check the network configuration. Install the correct transport protocol and configure it.
6	Nack response	Invalid configuration in the header or in the address	<ul style="list-style-type: none"> <li>▪ Check the configuration of operand block size.</li> <li>▪ Check the operand inside the device.</li> </ul>
7	Invalid answer	Unexpected length of the answer	<ul style="list-style-type: none"> <li>▪ Check the cable wiring.</li> <li>▪ Check the PLC state. It must be RUN.</li> <li>▪ Check the station field.</li> <li>▪ Check the configuration.</li> <li>▪ If on Serial Mode, check the Communication Parameters, such as Parity, Baud Rate and so forth</li> </ul>
8	Time out waiting ACK	<ul style="list-style-type: none"> <li>▪ Disconnected cables</li> <li>▪ PLC turned off, or in Stop or error mode</li> <li>▪ Station number is wrong</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check the cable wiring.</li> <li>▪ Check the PLC state. It must be RUN</li> <li>▪ Check the station number.</li> <li>▪ Check the configuration.</li> </ul>
9	Invalid device number	ID number or Device name configured with invalid value	Check the device configuration (Id number and device name). Make corrections in the station field.
10	Error opening device	<ul style="list-style-type: none"> <li>▪ Disconnected cables</li> <li>▪ PLC turned off, or in Stop or error mode</li> <li>▪ Station number is wrong</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check the cable wiring.</li> <li>▪ Check the PLC state. It must be RUN</li> <li>▪ Check the station number.</li> <li>▪ Check the configuration.</li> </ul>
13	Error nickname not found	<ul style="list-style-type: none"> <li>▪ Nickname doesn't exist in the CSV file</li> <li>▪ If you change the CSV file with the driver running, the changes will only be accepted after you stop your application and start it again.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check if the nickname is correct</li> </ul>
14	Invalid Block Size	<ul style="list-style-type: none"> <li>▪ Block Size is set to value not in the range of 1 to 255</li> </ul>	<ul style="list-style-type: none"> <li>▪ Set Block Size that should be set in the range of values from 1 to 255</li> </ul>
15	Block Size should be an integer	<ul style="list-style-type: none"> <li>▪ Block Size set is not an integer</li> </ul>	<ul style="list-style-type: none"> <li>▪ Set Block Size with an integer value</li> </ul>

Error Code	Description	Possible Causes	Procedure to Solve
16	Invalid Data Size Received from PLC	<ul style="list-style-type: none"> <li>Data Size received from PLC is less than the expected data size.</li> </ul>	<ul style="list-style-type: none"> <li>Set Block Size in the driver settings as per the data length supported by PLC.</li> </ul>
100	Invalid Protocol	Protocol not compatible	Verify that the PLC supports the DirectLOGIC Protocol.
101	Communication Error	Error in communication	<ul style="list-style-type: none"> <li>Check the station number. If you are using Serial communication, then make sure that you configured only the PLC number without any prefix (N: or I:).</li> <li>Verify that the addresses are valid.</li> </ul>
102	LRC Error	Invalid LRC Check	Make sure the cable is plugged in correctly.
-15	Timeout Start Message	<ul style="list-style-type: none"> <li>Disconnected cables</li> <li>PLC turned off, or in Stop or error mode</li> <li>Station number is wrong</li> <li>RTS/CTS control settings are wrong</li> </ul>	<ul style="list-style-type: none"> <li>Check the cable wiring.</li> <li>Check the PLC state. It must be RUN</li> <li>Check the station number.</li> <li>Check the RTS/CTS configuration (see the Studio <i>Technical Reference Manual</i> for valid configurations).</li> </ul>
-17	Timeout between rx char.	<ul style="list-style-type: none"> <li>PLC in stop or error mode</li> <li>Station number is wrong</li> <li>Parity is wrong</li> <li>RTS/CTS configuration settings are wrong</li> </ul>	<ul style="list-style-type: none"> <li>Check the cable wiring.</li> <li>Check the PLC state. It must be RUN</li> <li>Check the station number.</li> <li>Check the RTS/CTS configuration (see the Studio <i>Technical Reference Manual</i> for valid configurations).</li> </ul>

⇒ **Tip:**

You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Serial Communication**, right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the *LogWin* module (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE. The log is saved on the unit in the **celog.txt** file, which can be downloaded later.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software (e.g., DIRECTSOFT32). Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

To test communication between Studio and the device, we recommend using the sample application provided rather than your new application.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Project Information**: To find this information, select **Project** → **Status**.
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

## Sample Application

A sample application that employs the KOYO driver is provided on the Studio installation CD. We strongly recommend that you use this sample application to test the driver *before* you develop your own applications, for the following reasons:

- To better understand the information and instructions provided in this document;
- To verify that your driver configuration is working satisfactorily with the target device; and
- To ensure that the all of hardware used in the test (i.e. the device, adapter, cable, and PC) is functioning safely and correctly.

### **Note:**

The following instructions assume that you are familiar with developing project applications in Studio. If you are not, then please review the relevant chapters of the Studio *Technical Reference Manual* before proceeding.

To use the sample application:



Configure the device's communication settings according to the manufacturer's documentation.



Run Studio.



From the main menu bar, select **File** | **Open Project**.



Insert the Studio installation CD and browse it to find the sample application. It should be located in the directory `\COMMUNICATION EXAMPLES\KOYO`.



Select and open the sample application.



Configure and test the driver, as described in the rest of this document.

When you have thoroughly tested the driver with your target device, you may proceed with developing your own Studio application projects.

### **Tip:**

You can use the sample application screen as the maintenance screen for your own applications.

## Revision History

Doc. Revision	Driver Version	Date	Description of changes
A	1.00	23-Apr-2001	First driver version
B	1.01	11-Jun-2001	<ul style="list-style-type: none"> <li>Updated to support WindowsCE operating system</li> <li>The driver no longer requires additional libraries.</li> </ul>
C	1.02	13-Jul-2001	Fixed memory leak bug
D	1.03	19-Jul-2001	Modified algorithm for compatibility with ARM processor
E	1.04	07-May-2002	<ul style="list-style-type: none"> <li>Fixed multiple station bug</li> <li>Improved performance</li> </ul>
F	1.05	17-May-2002	Included Signed or Unsigned value
G	1.06	11-Mar-2005	Included Serial Communication
H	1.07	07-Apr-2005	Fixed problems over IPX protocol
I	1.08	26-Oct-2005	Included Float Point addresses
J	1.09	24-May-2006	Fixed problem when the maximum number of devices is less than the number of different stations configured in the driver. Also changed the station field to be non-case sensitive.
K	1.09	16-Jun-2006	Edited for language and usability.
L	1.09	03-Nov-2006	<ul style="list-style-type: none"> <li>Added DL05 and DL06 PLCs to list of compatible equipment.</li> <li>Clarified how to configure for Serial communication.</li> <li>Revised mentions of ENET to ECOM.</li> </ul>
M	1.09	26-Feb-2008	Revised the doc.
N	1.10	21-Mar-2008	Fixed issue with Coils (C) header.
O	1.10	22-May-2008	<ul style="list-style-type: none"> <li>Inserted communication with other Network</li> <li>Included IP in the Station field</li> </ul>
P	1.11	06-Jan-2009	Fixed problem to create virtual groups when using several stations.
Q	1.12	17-Feb-2009	<ul style="list-style-type: none"> <li>Modified SDS Header to accept Initial Address and Address column to accept Offset</li> <li>Added Long BCD format</li> <li>Fixed problem with Serial Communication</li> </ul>
R	1.13	31-May-2013	<ul style="list-style-type: none"> <li>Included Support for DWORD format for the memory (V).</li> <li>Included Support for new S, SD, A formats for the memory (V).</li> <li>Updated information about the maximum number of devices supported.</li> <li>Added support for tag names from exported CSV file</li> </ul>
S	1.14	29-Aug-2013	<ul style="list-style-type: none"> <li>Included support for LBS, word Swap in long BCD.</li> </ul>
T	1.15	05-Mar-2014	<ul style="list-style-type: none"> <li>Implemented Max Block Size in Driver Settings</li> <li>Updated Error Codes 14, 15 and 16</li> </ul>
U	1.16	05-May-2014	<ul style="list-style-type: none"> <li>Added specific blocksize information</li> </ul>
V	1.17	15-Oct-2014	<ul style="list-style-type: none"> <li>Modified issue messages when configuring invalid stations</li> <li>Fixed issues when writing in Ethernet mode</li> <li>Added support to simultaneous connection</li> </ul>
X	1.18	10-Dec-2014	Fixed problem in the block size calculation on the Main Driver Sheet
Y	1.18	06-May-2015	Driver documentation updated with information added about tag integration using a CSV file. No changes made to the driver
Z	1.19	11-Nov-2015	Changed driver to use A format in standard driver sheet
AA	1.20	02-Dec-2015	<ul style="list-style-type: none"> <li>Fixed issue of driver writing too many bytes for A format (String)</li> <li>Added trace method to the send function.</li> </ul>
AB	1.21	14-Oct-2016	Fixed the bug Real-type tags are not being rounded properly when writing to an integer-type PLC register

AC	1.21	08-June-2017	▪ Fixed documentation to have accurate information about using indirect tags on MDS. No change in the driver.
----	------	--------------	---