

# MODBUS TCP with K-DUCER

## Application Resources

### Resource Packet Contents

The *KDUCER ModBus TCP resources* packet contains the following files and documents:

- **read me.pdf**: this document
- **KDucer\_Modbus\_Map\_Rev24.xlsx**: table of all K\_DUCER MODBUS data and addresses
- **Official modbus specifications**: documents from [www.modbus.org](http://www.modbus.org) detailing the communication protocol and TCP packet construction. If your control system already implements MODBUS TCP, you do not need to dive into these documents.
- **AllenBradley PLC MODBUS resources**: example project integrating the K-DUCER with a Micro820 PLC via MODBUS, plus guides for implementing a MODBUS TCP client with CompactLogix and ControlLogix PLCs
- **Siemens PLC MODBUS resources**: example project integrating the K-DUCER with a S7-1200 PLC via MODBUS, plus guides for implementing MODBUS TCP client with S7-1200/1500 and PCS 7 PLCs
- **modbus\_example\_python.py**: example Python script integrating the K-DUCER with a PC via MODBUS

### Table of Contents

|   |   |
|---|---|
| Introduction .....                            | 2 |
| Usage.....                                    | 2 |
| K-DUCER MODBUS map.....                       | 3 |
| MODBUS TCP code examples and literature ..... | 4 |
| PC (multiplatform) with Python.....           | 4 |
| Siemens PLCs.....                             | 5 |
| Allen Bradley PLCs.....                       | 5 |
| Universal Robots .....                        | 6 |
| Example Screwdriving Cycle.....               | 6 |

## Introduction

The recommended way to interface with the K-DUCER unit is through the MODBUS TCP protocol on the ethernet port (CN5).

MODBUS communication protocol provides a Client-Server interface between devices connected on an ethernet TCP/IP network.

The MODBUS protocol specifications are open source and freely available online at [modbus.org](http://modbus.org), however most automation engineers will not need to worry about the implementation details because MODBUS is already supported and implemented by most ethernet-capable PLCs and industrial PCs.

More details on how the protocol works and on how to construct MODBUS TCP packets can be found at [www.modbus.org](http://www.modbus.org) and in the *Official modbus specifications* folder.

## Usage

Enable MODBUS TCP via the General Settings menu > (14) Communication Protocol. The K-DUCER should be connected to the same LAN network as the controlling device, and it must be left in the main operation screen, outside of any configuration menu. Note: the K-DUCER will respond to the *ping* command over TCP/IP when configured correctly.

The K-DUCER implements a MODBUS server, which responds to MODBUS requests. The automation device (PLC, industrial PC, ...) must implement a MODBUS client, which sends MODBUS requests to the server (K-DUCER).

The MODBUS server (K-DUCER) only responds to requests and never initiates any communication independently, in accordance with the MODBUS protocol.

A MODBUS request is simply a message requesting to read or write one or more *bits* or *bytes* of data at a particular address. The list of all accessible data and their addresses is called the MODBUS map.

MODBUS requests are categorized into *function codes*. Different function codes are used to access different types of data (bits-coils or bytes-registers). There are also convenience function codes used to access a range of multiple data addresses at once.

All program, sequence, and general settings can be modified via MODBUS requests. However, Kolver recommends pre-configuring the K-DUCER programs and settings via the K-Expand software, via touch screen, or via kdu backup file from USB, and only

utilizing the MODBUS TCP protocol for screwdriver control, program switching, and data acquisition.

Changing program parameters such as target torque via MODBUS is possible but not recommended, and is never necessary except for the rare applications requiring more than 64 different programs.

## K-DUCER MODBUS map

The K-DUCER, MODBUS data is organized and accessed as follows:

| Data Category                              | Contents   | Access          | Associated MODBUS function codes   |
|--|--|-----------------|--|
| <b>COILS</b><br><i>(bits)</i>              | A mirror copy of the CN3 output pins 23 to 43 represented as bits;<br><br>Writeable coils mimicking the functionality of CN3 input pins 13 to 20, providing screwdriver motor control capability | Read/<br>Write  | 01 (read coils)<br><br>05 (write single coil)<br><br>15 (write multiple coils)                     |
| <b>INPUT REGISTERS</b><br><i>(bytes)</i>   | Data related to the last screwdriving results including closing torque and angle; torque/angle charts; current screwdriving state and errors; connected screwdriver info                         | Read only       | 04 (read input registers)  |
| <b>HOLDING REGISTERS</b><br><i>(bytes)</i> | Current selected program;<br>Remote programming mode enter/exit flag;<br>All program settings;<br>All sequence setting;<br>Current selected sequence;<br>All options settings.                   | Read/<br>Write* | 03 (read holding registers)<br><br>06 (write single register)<br><br>16 (write multiple registers) |
| <b>DISCRETE INPUTS</b>                     | A mirror copy of the CN3 input pins 1 to 20 represented as bits  | Read only       |  |

**\*Note:** all holding registers except *CurrentProgram* (7373) can only be written after entering Remote Programming Mode. These registers contain all program, sequence, and option settings. It is possible but not recommended to change these parameters via MODBUS. To modify holding registers via MODBUS, follow these steps:

- Write the value “1” to address 7790 to enter “Remote Programming Mode” using MODBUS function code 06 “write single register”
- Change the desired holding register values using MODBUS function codes 06 or 16
- Write the value “2” to address 7790 to apply the changes and exit “Remote Programming Mode”

The full MODBUS map can be found in the attached document **KDucer\_Modbus\_Map\_Rev24.xlsx**.

## MODBUS TCP code examples and literature

We provide sample projects illustrating K-DUCER screwdriver control built by Kolver for various devices, as well as generic MODBUS TCP guides and literature produced by the manufacturers of these devices.

We also recommend searching youtube for a multitude of freely available videos illustrating how to implement MODBUS TCP communication with various control systems.

Rockwell, Allen Bradley, MicroPLC, ControlLogix, CompactLogix, Siemens, SIMATIC, Universal Robots, PolyScope, are all trademarks of their respective corporations and are not affiliated with Kolver.

## PC (multiplatform) with Python

The python script *modbus\_example\_python.py* illustrates K-DUCER screwdriver control and data retrieval with MODBUS TCP using python version 3 or above (<https://www.python.org/downloads/>) and the PyModbusTCP library (<https://pypi.org/project/pyModbusTCP/> or <https://github.com/sourceperl/pyModbusTCP>).

The program follows the steps outlined in the section below “example screwdriving cycle”.

## Siemens PLCs

General information for using MODBUS TCP with Siemens PLCs can be found at

<http://www.siemens.com/s7modbus> or

<https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10165502?activeTab=productinformation>

We provide an example project, *KDUCER\_modbus\_example\_Step7-1200.zap16*, built and tested for a S7-1200 PLC using TIA Portal V16.

The project illustrates K-DUCER screwdriver control and data retrieval and follows the steps outlined in the section below “example screwdriving cycle”.

Additionally, the following Siemens-produced application guides are provided:

- ModbusTCP with MB\_CLIENT and MB\_SERVER S7-1200 S7-1500
- ModbusTCP with PCS 7

Additional resources can be found in the Siemens website by searching for “modbus tcp”: <https://support.industry.siemens.com/cs/search?search=modbus%20tcp>

## Allen Bradley PLCs

We provide an example project, *KDUCER\_modbus\_example\_Micro820*, built and tested for a Micro820 PLC using Connected Components Workbench.

The project illustrates K-DUCER screwdriver control and data retrieval and follows the steps outlined in the section below “example screwdriving cycle”.

Additionally, the following Rockwell-produced application guides and libraries are provided:

- Modbus-TCP-Client-AOI-based-code-for-ControlLogix-v-2.00.01
- 101037 Modbus TCP Add-On instructions for ControlLogix and CompactLogix controllers, AOI Version 2.03.00
- at002 AllenBradley EtherNetIP Socket Interface

More updated versions of these AOIs and guides may be found at

<https://www.rockwellautomation.com/search/> by searching for “modbus tcp”.

## Universal Robots

The guide “Using the Modbus TCP Client Interface of the UR Robot” is provided illustrating how to implement a MODBUS TCP client using a Universal Robot CB/e-series.

### Example Screwdriving Cycle

The sample projects and programs provided by Kolver for integrating the K-DUCER with the corresponding devices using MODBUS TCP all implement the screwdriving cycle illustrated here.

**WARNING:** running any of the attached examples with a properly configured K-DUCER will cause the screwdriver to run!

Do not run these examples if you are not familiarized with operating the screwdriver. Use all appropriate precautions and read the K-DUCER operator manual first.

The screwdriving cycle implemented in the examples loops through the following operations:

| Step number | What it does  | How it's done  |
|-------------|---|--|
| 1           | Runs the screwdriver until it stops automatically                                     | Writes “high” to COIL address 33 (REMOTE_LEVER) continuously every 50-100mS.<br>Reads INPUT REGISTER address 138 to obtain the current screwdriving state and determine when the cycle is complete |
| 2           | Reads closing torque and angle  | Reads INPUT REGISTER addresses 149 and 151 to obtain the closing torque and angles   |
| 3           | Selects program #1 if the screw result was OK, program #2 if the screw result was NOK | Writes “1” or “2” to HOLDING REGISTER address 7373 to select the next program to run. Selection is determined by comparing the last screwdriving state to values 13/14 (screw OK/angle OK).        |