



## APPLICATION NOTE

« AC30 Series»: Master Slave(s)  
Speed synchronization sequence –  
Sender Receiver communication

## Contents

1.	Abstract.....	3
2.	Introduction .....	3
3.	Related Documents .....	3
4.	Tools .....	3
5.	Operating principle.....	5
6.	Master configuration .....	6
1.1	Sequencing and Reference .....	6
1.2	Sender receiver via Peer-to-Peer Control .....	7
2	Slave(s) Configuration .....	8
2.1	Sequencing and reference .....	8
2.2	Sender receiver via Peer-to-Peer Control .....	8
3	Trace sequence .....	9

## 1. Abstract

This document describes the approach to implement the sequence control diagram of master-slave(s) speed synchronization using AC30 drives with Peer-to-Peer communication feature. CoDeSys **AC30 Sender Receiver** library provides a simple method of providing peer-to-peer communication between AC30 Ethernet connected drives.

## 2. Introduction

The aim of this application note is to show the basic configuration and the state machine diagram for speed synchronization between two or many axes where one axes is the master and the other axes are slaves.

The start/stop logic is handled by the master and the actual speed value of the master acts as the speed setpoint for the slave(s) inverter(s).

Drives are linked by using a peer-to-peer connection. The AC30 drives includes the ability to use the Codesys AC30 Sender Receiver to provide peer-to-peer communication over Ethernet, to pass the control command and speed reference from one drive to others in the system.

Before continuing with this application note, ensure the Start-up and Commissioning section from the AC30 Hardware manual has been completed and is fully understood. Also, the complete installation and knowledge of CODESYS is mandatory!

## 3. Related Documents

For more information about the AC Drive, see the following related documents.

Reference number	Document	Description
HA501718U001-8	AC30-manual	AC30 Product manual
HA503711U001	AC30_Hardware_Installation_Manual_Frames_D-J	AC30 Hardware manual Frames D-J
HA503711U002	AC30_Hardware_Installation_Manual_Frames_K-N	AC30 Hardware manual Frames K-N
HA503711U003	AC30_Software_Reference	AC30 Software Reference

## 4. Tools

	Name	Firmware Version
Hardware	AC30P And AC30D Controller Board	2.19.5
	Encoder feedback option Board: 7004-04	V1.3
Software	Codesys programming system	3.5.21 SP3
	Parker Drive Quicktool(PDQ)	3.12.66.1

**Application:** The basic function comes from the standard template: App\_0\_Basic\_Speed\_Control, the Sender-Receiver Library should be added into project for Peer\_to\_Peer communication features.

**Communications:**

Data is sent from a drive using a sender function block to another drive using a receiver function block. To make sure the data is sent to the correct receiver(s) a channel number is used on both sender and receiver to link the two together.

If the sent data is to go to one drive only, then the sender uses the IP address of the receiving drive. In this case, no other drives will receive the data.

If the sent data is to go to a group of drives, then the sender uses a 'group send' (multicast) to all receiving drives on the same channel number that have joined the group.

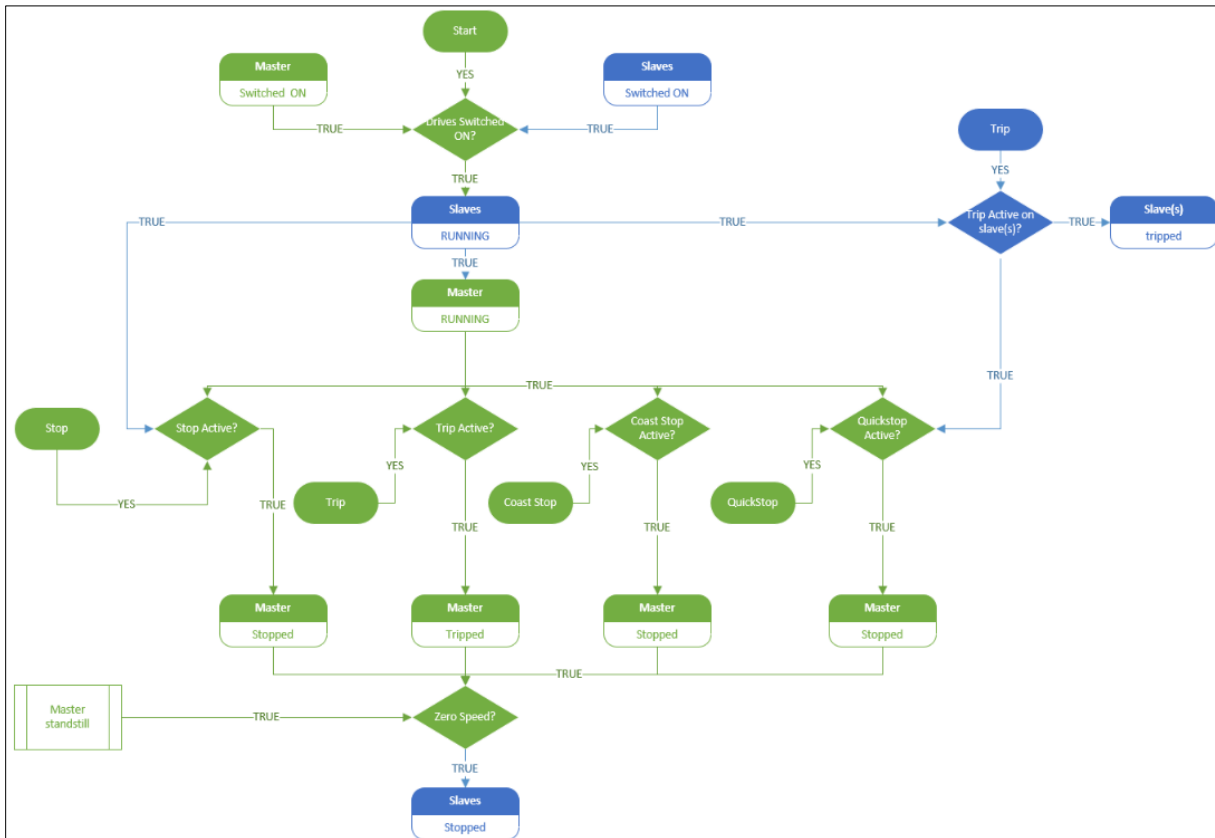
If the sent data is to go to **all drives**, then the sender uses a 'broadcast send' to all receiving drives on the same channel number.

- **Due to drive resources, the total number of sender and receive function blocks per drive application should be limited to no more than 6.**
- **Send rates should be set to no less than 5ms.**
- **All communications use a base UDP port of 1270**
- **Multicast communications use a base address of 239.255.10.0**

Note: the configuration designed for this application note uses 2 drives connected together. Download the example described below from the following link:

[AC30 Master Slave speed synchronization sequence - sender receiver](#)

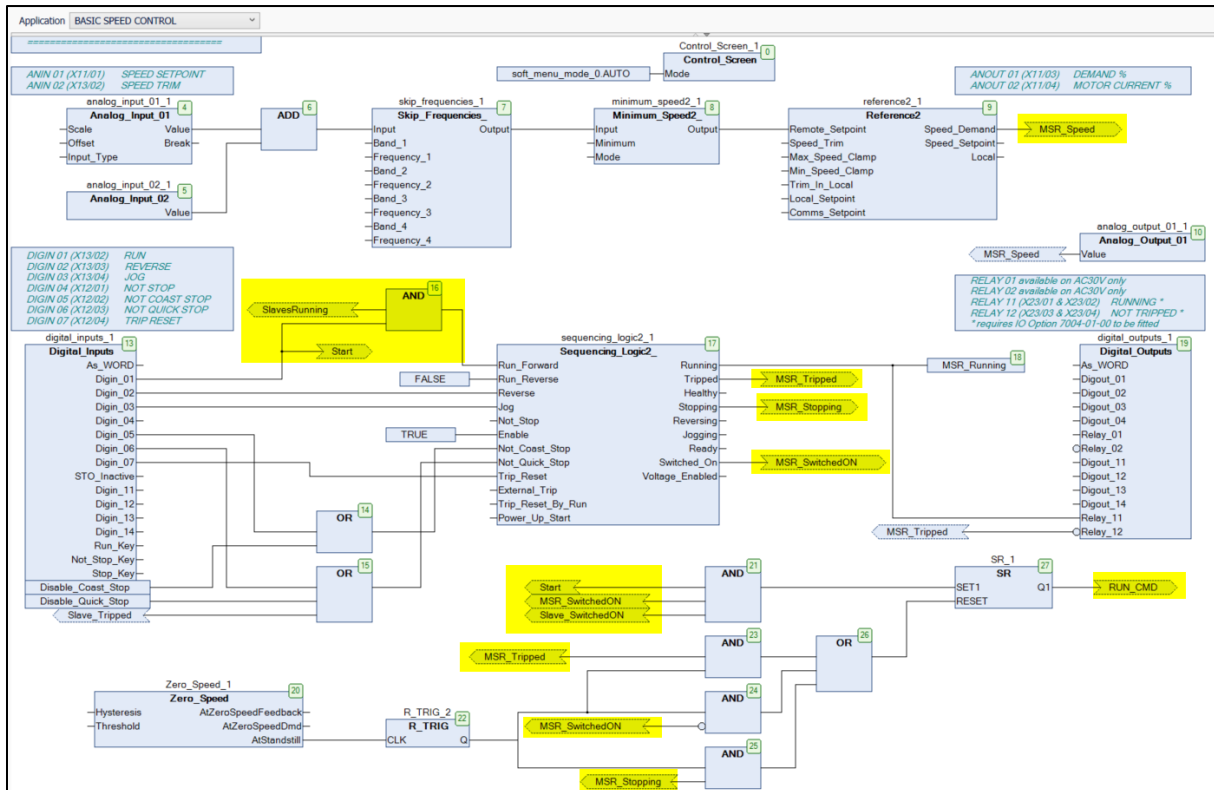
## 5. Operating principle



- Each drive sends their “switched ON” status to the master: if the master is in “switched ON” state, the rising edge of the start command sends the run command to all the slaves.
- The running command from the slaves enables the running operation of the master.
- The speed demand (with Ramps, Tag 500) is the setpoint for the Master and the Slaves upon enabling all drives.
- Trip active from one slave or all slaves activate the master Quickstop.
- The Stop command or trip signal from master Stops all the slaves drive when the master is in standstill.

## 6. Master configuration

### 1.1 Sequencing and Reference



The application is based on the **BASIC SPEED CONTROL** Macro.

1) The **Sequencing\_logic2** block allows control and monitor the operational state of the master drive, the input **Run\_Forward** is connected to the slaves running signal, when this signal is ON the master drive state change to RUNNING.

The switching ON of the input "Digin\_01" of "Digital\_inputs\_1" block when the master-slaves are in "Switched ON" status enable "RUN\_CMD" signal that is sent to all slaves, by receiving this signal on their "Run\_Forward" input, the slaves state change to RUNNING.

The slaves in running state is then sent to the master, by receiving the signal "SlavesRunning" from the slaves and the start command enabled master state change to RUNNING.

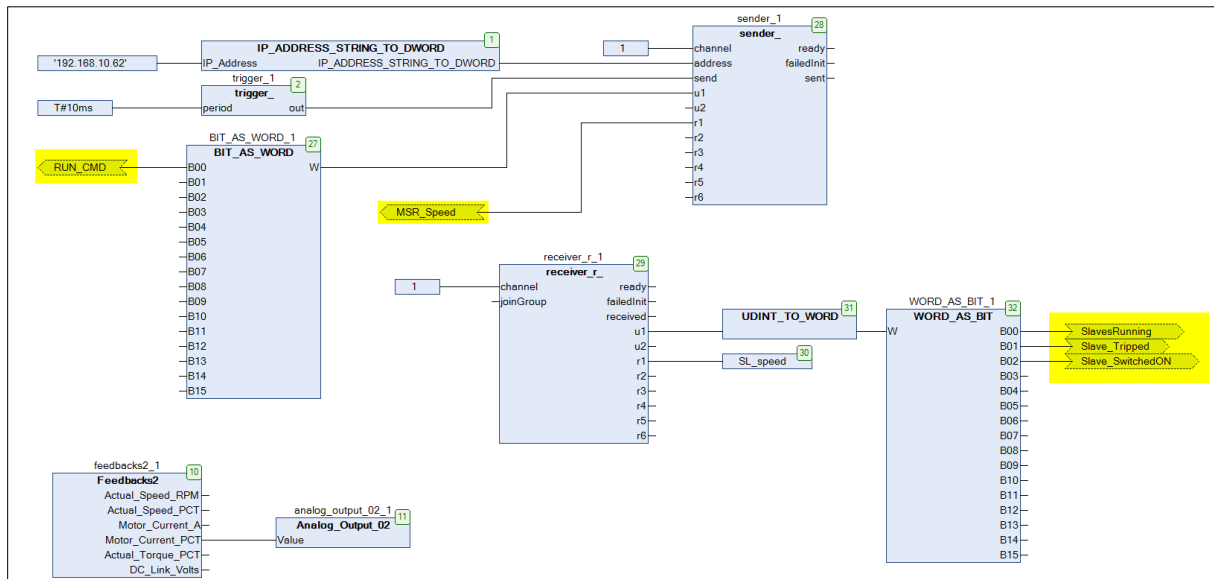
Drive tripped "MSR\_Tripped", stopped "MSR\_Stopping" and switched On disable the signal "RUN\_CMD" when zero speed is reached, the OFF signal of "RUN\_CMD" stops the slaves.

2) The **Reference2** block allows the Speed control of the master drive.

The **speed\_Demand** output is the signal used for Speed Control.

The speed demand "MSR\_Speed" is sent to the slaves to synchronize the process.

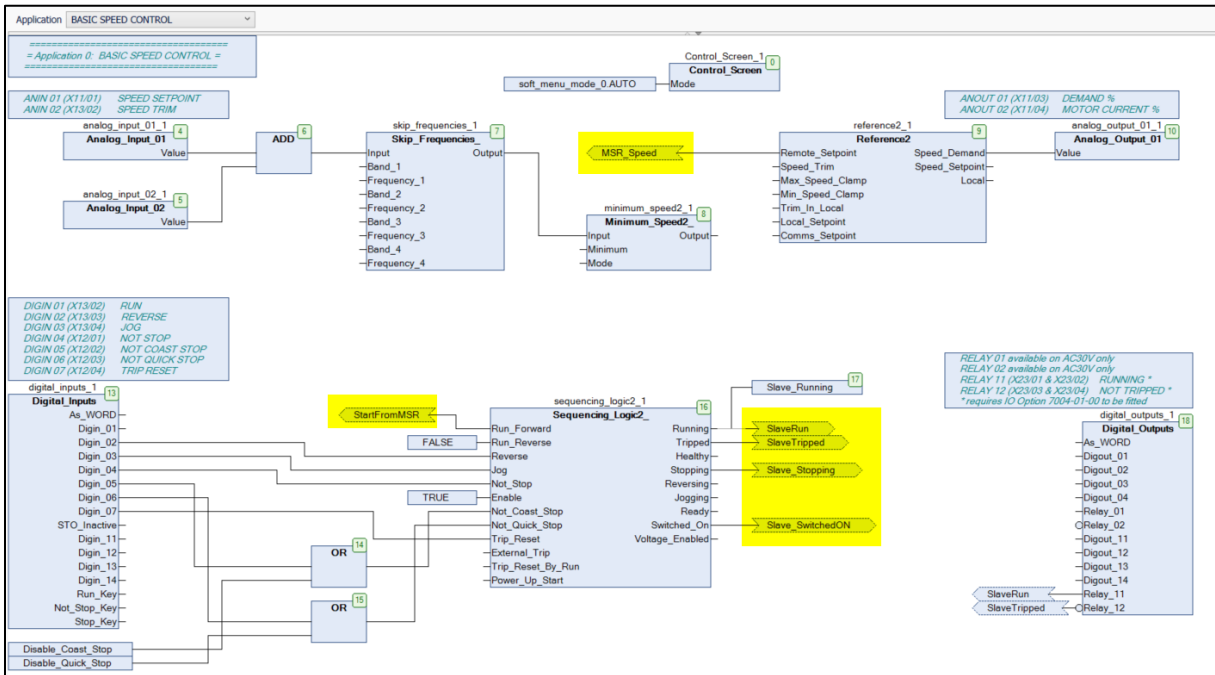
## 1.2 Sender receiver via Peer-to-Peer Control



The “sender\_” block allows Run command and speed setpoint to be sent to the slave, the “receiver\_r\_” block allows status data to be received from slave. The function block **trigger\_** allows for the rate to be set at which data is sent from the **sender\_** function block.

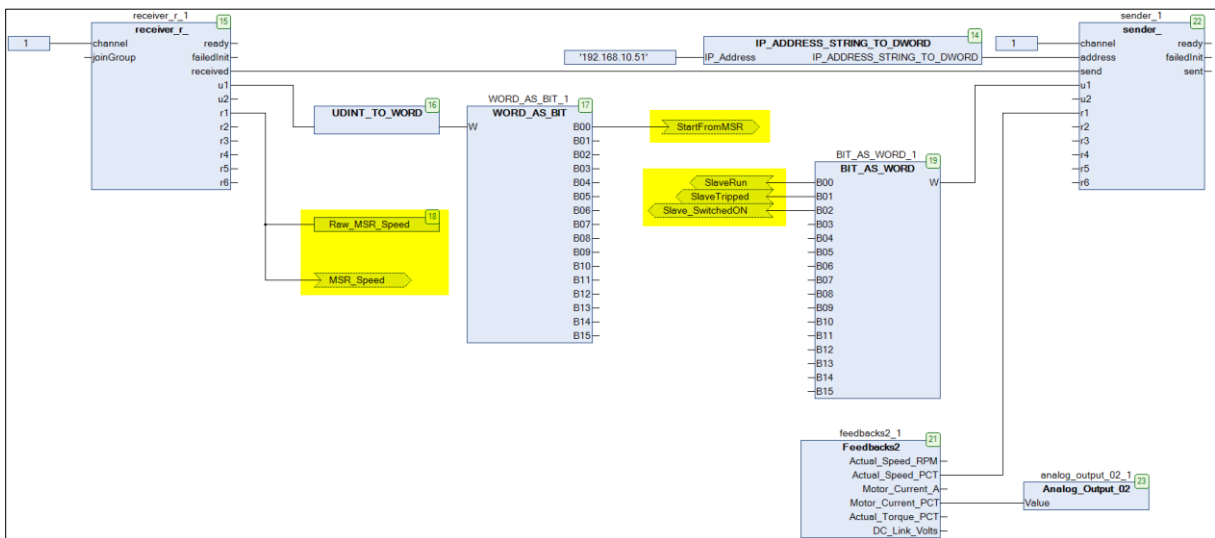
## 2 Slave(s) Configuration

### 2.1 Sequencing and reference



The start command “StartFromMSR” received from master enable the slaves running operation. When the master is in running state, the remote\_setpoint speed “MSR\_Speed” is the speed demand of the master. The slaves operation state are sent to the master via Peer to peer connection

### 2.2 Sender receiver via Peer-to-Peer Control



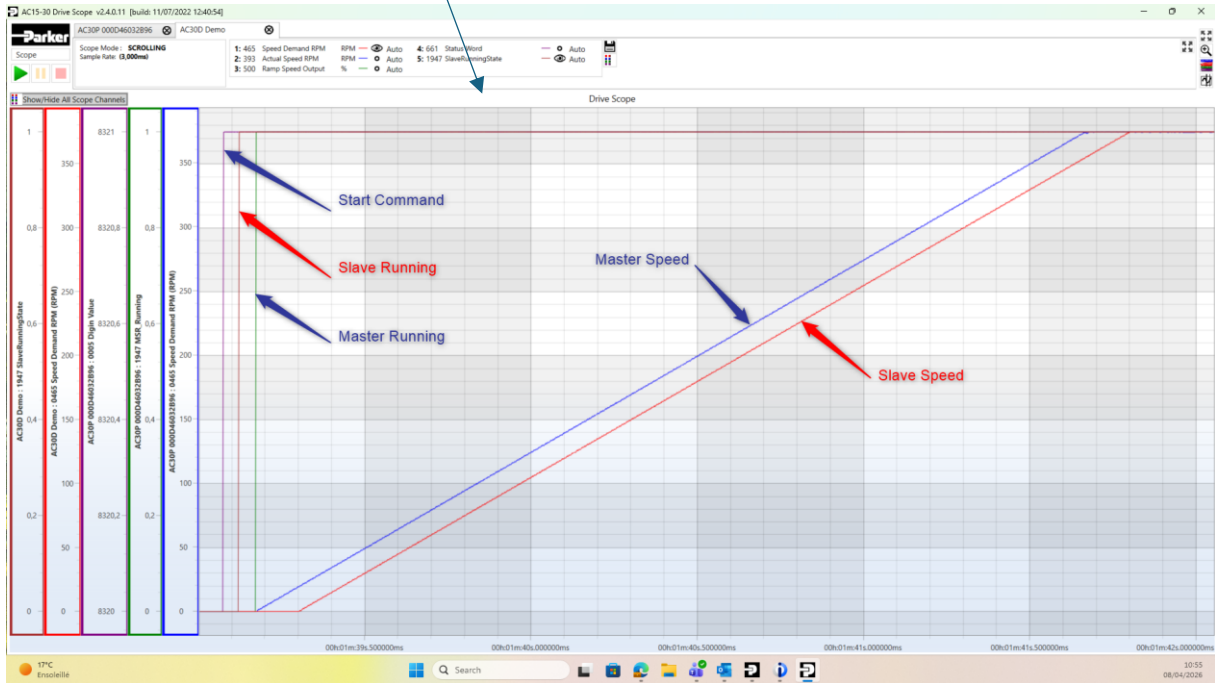
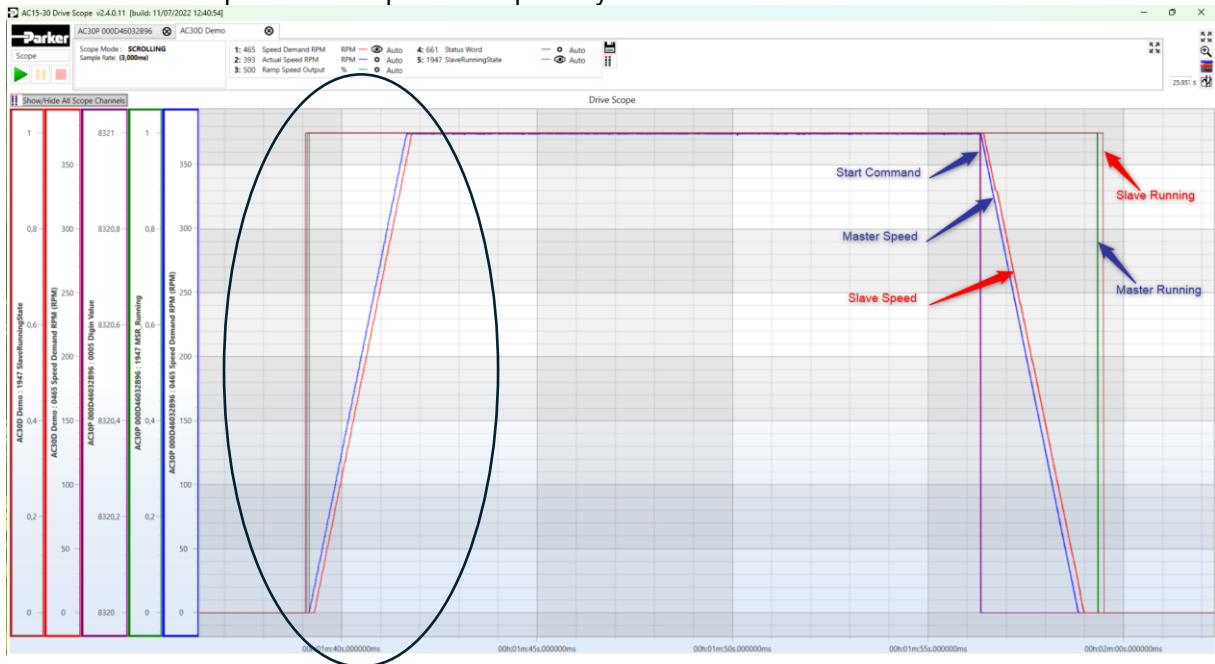
The slave’s state is passed via peer to peer connecting it to input “u1” through BIT\_AS\_DWORD into the “sender\_” block.

The master speed is received by slaves via “r1” of the block “receiver\_”.

The function block **trigger\_** allows for the rate to be set at which data is sent from the **sender\_** function block.

### 3 Trace sequence

The trace shows operational sequence of speed synchronization between master and slave.



**Parker Hannifin Manufacturing Germany GmbH & Co. KG**  
**Electric Motion & Pneumatic Division Europe [EMPD]**  
**Robert-Bosch-Strasse 22, 77656 Offenburg (Germany)**  
**Tel: +49 (0781) 509-0**  
**Internet: <https://www.parker.com/>**

© 2025 Parker Hannifin Corporation

