Application Note

C004 Phase Locking Set up With Peer To Peer

HA502487C004 Issue B

AC30P/D V2.13 onwards

Safety Information !

Requirements

Intended Users

This Application Note is to be made available to all persons who are required to install, configure or service equipment described herein, or any other associated operation.

The information given is intended to enable the user to obtain maximum benefit from the equipment.

Application Area

The equipment described is intended for industrial motor speed control utilising AC induction or AC synchronous machines.

Personnel

Installation, operation and maintenance of the equipment should be carried out by qualified personnel. A qualified person is someone who is technically competent and familiar with all safety information and established safety practices; with the installation process, operation and maintenance of this equipment; and with all the hazards involved.

Hazards

Refer to the Safety Information given at the front of the Product Manual supplied with every Parker SSD Drives product.

C003 PHASE LOCKING SET UP WITH PEER TO PEER

Abstract

This application note gives an example of setting up a phase locking application with the Phase locking application described in HA503284U008, using the Peer to Peer feature.

Pre-Requisite

The pre-requisites are :

- a AC30P or a AC30D drive used as the master drive

- a AC30P or a AC30D drive used as the slave drive

Introduction

Phase Control or Phase Lock, sometimes referred to as electronic gearbox, is a position trim of a slave drive to a speed to maintain the relative position between a master and a slave shaft or a precision ratio between the two shafts. The slave speed demand is composed of the master speed demand and a position trim from a counter of the accumulated differences between the Master and Slave Encoders.

The AC30P and AC30D includes the ability to use real-time peer-to-peer communications, over Ethernet, to pass the speed/position reference from one drive to another in the system.

One drive (configured to be the Master) generates a speed and position command.

It could be a Real Master, using the position information from an external encoder connected to the drive, or a Virtual Master, internally generating position.

All the other drives must have the Virtual Master functionality enabled, and must be configured as Slaves.

The following phase blocks are also available:

- Phase inch – progressive phase advance or retard on demand
- Phase move – pre-defined phase change, at set speed and acceleration, on demand
- Phase offset – adjustment of the relative phase between the Master and the slave.
- Phase tuning – facility for the injection of square speed or phase offsets to assist loop tuning.
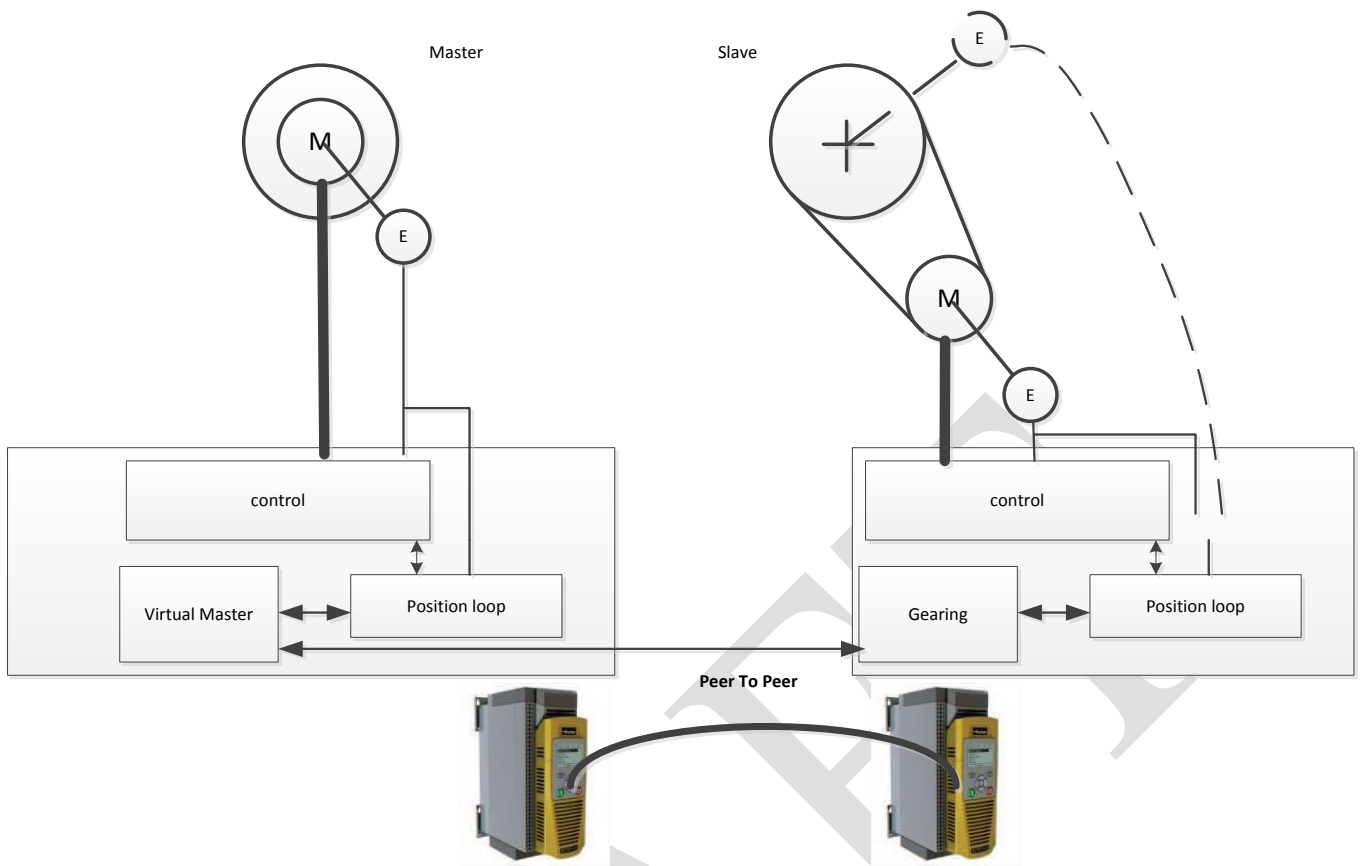
Phase Loop operation

The phase loop is a position loop.

The position demand is the sum of the reference position plus the offsets generated by the phase blocks belonging to the application layer.
The position feedback is the position from the encoder defined as the slave encoder.
The Position Loop works on a multi turn position. The error between the Master and the slave can be a multi turn error.
It contains a PI corrector (no derivative action), a feed forward term and an output saturation.
The feed forward term is the sum of the reference speed plus the speeds generated by the phase blocks.
The output feds the speed loop.
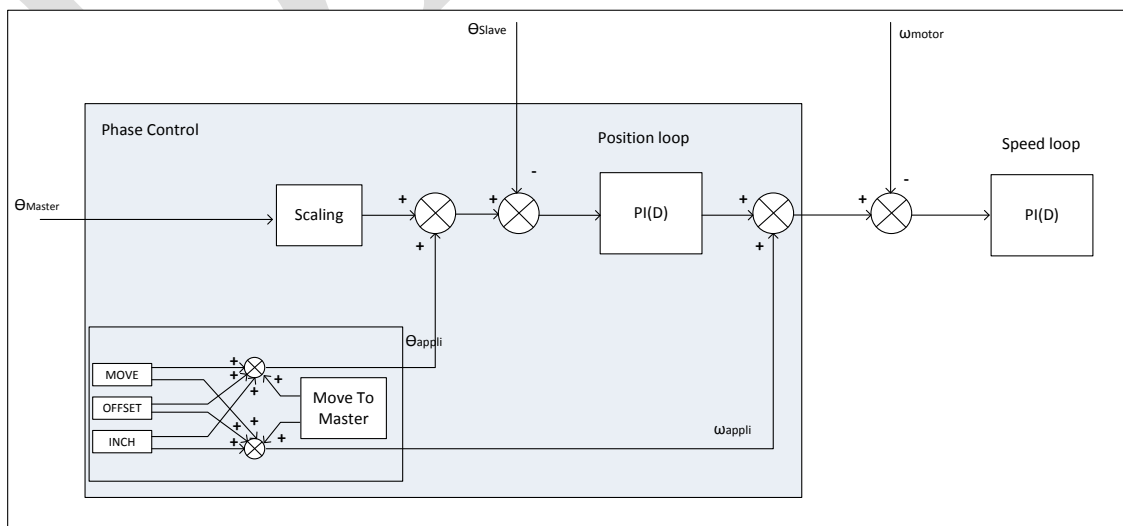
Phase Control Default application

Default application is available as RA503284U008_001.prj

Default configuration is seperated in 3 parts :

- first is dedicated to the phase control :
    o phase blocks
    o real/virtual master configuration
    o phase loop (position loop)
- second is dedicated to the I/Os and sequencing of the drive
    o Digital I/Os
    o Encoder(s) setting
    o Master/slave selection
- third is related to the Peer To Peer control
    o PTP and Peer To Peer
    o Connection control

Phase Control :

I/Os and sequencing :

DRIVE CONFIGURATION

```
DIGIN 01 (X13/02)   RUN
DIGIN 02 (X13/03)   PHASE INCH ADVANCE
DIGIN 03 (X13/04)   PHASE INCH RETARD
DIGIN 04 (X12/01)   ( Possibly POSITION LOOP RESET )
DIGIN 05 (X12/02)   NOT COAST STOP
DIGIN 06 (X12/03)   NOT QUICK STOP
DIGIN 07 (X12/04)   TRIP RESET
```

digital_inputs_1 [21]
Digital_Inputs
- As_WORD
- Digin_01
- Digin_02 → Phase_Inch-Advance
- Digin_03 → Phase_Inch-Retard
- Digin_04
- Digin_05
- Digin_06
- Digin_07
- STO_Inactive
- Digin_11
- Digin_12
- Digin_13
- Digin_14
- Run_Key
- Not_Stop_Key
- Stop_Key
- Disable_Coast_Stop
- Disable_Quick_Stop

OR [23]
OR [24]

FALSE
TRUE

sequencing_logic2_1 [22]
Sequencing_Logic2
- Run_Forward   Running
- Run_Reverse   Tripped
- Reverse   Healthy
- Jog   Stopping
- Not_Stop   Reversing
- Enable   Jogging
- Not_Coast_Stop   Ready
- Not_Quick_Stop   Switched_On
- Trip_Reset   Voltage_Enabled
- External_Trip
- Trip_Reset_By_Run
- Power_Up_Start

Digital_Outputs_1 [6]
Digital_Outputs
- As_WORD
- Digout_01
- Digout_02
- Digout_03
- Digout_04
- Relay_01
- Relay_02
- Digout_11
- Digout_12
- Digout_13
- Digout_14
- Relay_11
- Relay_12

Phase_Ctrl_Config_1 [1]
Phase_Ctrl_Config
- position_reference_source_0.NONE → Master_Position_Src   Setup_Successful
- position_feedback_source_0.SAME_AS_MOTOR_FBK → Slave_Position_Src   Error_Number
                                                      Master_Encoder
                                                      Slave_Encoder
                                                      Spd_Loop_Encoder

Control_Mode_1 [2]
Control_Mode
- Motor_Type_or_AFE
- control_strategy_type_0.VECTOR_CONTROL → Control_Strategy
- control_type_0.ENCODER_FEEDBACK → Control_Type
- Encoder_Feedback

Encoder_Input_1 [3]
Encoder_Input
- pulse_encoder_type_0.QUADRATURE → Encoder_Type   Speed
- encoder_supply_0._5V → Supply   Count
- 1024 → Lines
- FALSE → Invert
- FALSE → Single_Ended
- FALSE → Count_Reset

Encoder_Slot_1_1 [3]
Encoder_Slot_1
- system_board_encoder_supply_0._5V → Encoder_Supply   Encoder_Speed
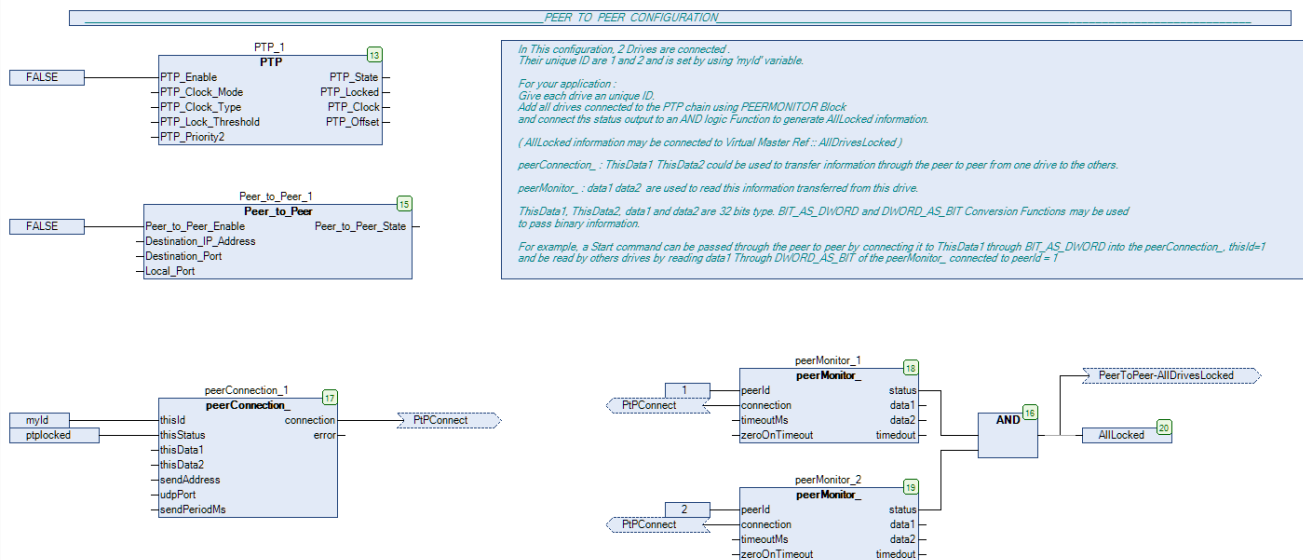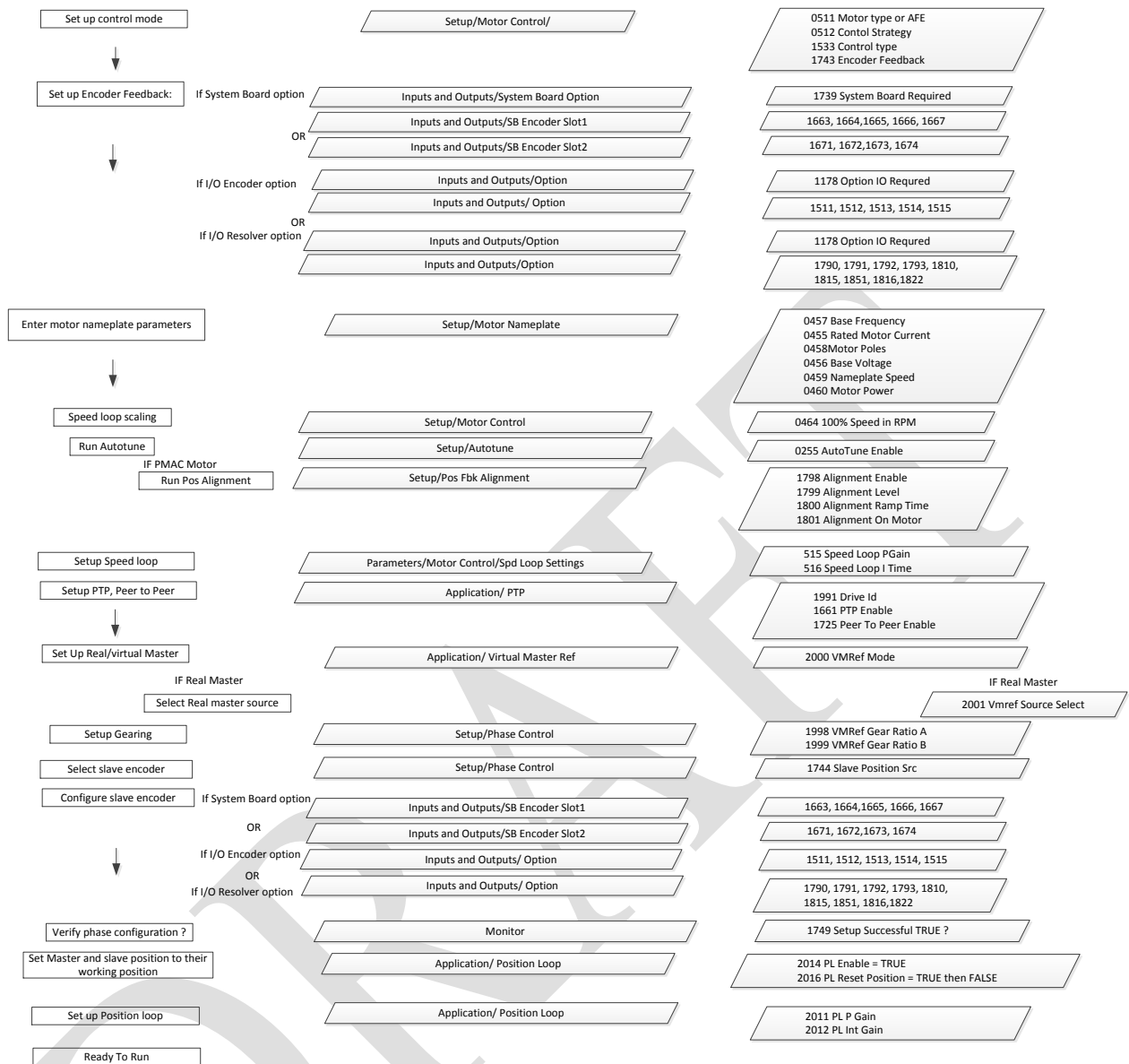- 1024 → Encoder_Lines   Encoder_Count
- FALSE → Encoder_Invert
- pulse_encoder_type_0.QUADRATURE → Encoder_Type
- FALSE → High_Input_Threshold
- FALSE → Encoder_Count_Reset

Encoder_Slot_2_1 [4]
Encoder_Slot_2
- 1024 → Encoder_Lines   Encoder_Speed
- FALSE → Encoder_Invert   Encoder_Count
- pulse_encoder_type_0.QUADRATURE → Encoder_Type
- FALSE → High_Input_Threshold
- FALSE → Encoder_Count_Reset

Peer To Peer Control

PEER TO PEER CONFIGURATION

PTP_1 [13]
PTP
- FALSE → PTP_Enable   PTP_State
- PTP_Clock_Mode   PTP_Locked
- PTP_Clock_Type   PTP_Clock
- PTP_Lock_Threshold   PTP_Offset
- PTP_Priority2

Peer_to_Peer_1 [15]
Peer_to_Peer
- FALSE → Peer_to_Peer_Enable   Peer_to_Peer_State
- Destination_IP_Address
- Destination_Port
- Local_Port

In This configuration, 2 Drives are connected .
Their unique ID are 1 and 2 and is set by using 'myId' variable.

For your application :
Give each drive an unique ID.
Add all drives connected to the PTP chain using PEERMONITOR Block
and connect ths status output to an AND logic Function to generate AllLocked information.

(AllLocked information may be connected to Virtual Master Ref :: AllDrivesLocked)

peerConnection_ : ThisData1 ThisData2 could be used to transfer information through the peer to peer from one drive to the others.

peerMonitor_ : data1 data2 are used to read this information transferred from this drive.

ThisData1, ThisData2, data1 and data2 are 32 bits type. BIT_AS_DWORD and DWORD_AS_BIT Conversion Functions may be used
to pass binary information.

For example, a Start command can be passed through the peer to peer by connecting it to ThisData1 through BIT_AS_DWORD into the peerConnection_, thisId=1
and be read by others drives by reading data1 Through DWORD_AS_BIT of the peerMonitor_ connected to peerId = 1

peerConnection_1 [17]
peerConnection_
- myId → thisId   connection → PtPConnect
- ptplocked → thisStatus   error
- thisData1
- thisData2
- sendAddress
- udpPort
- sendPeriodMs

peerMonitor_1 [18]
peerMonitor_
- 1 → peerId   status → PeerToPeer-AllDrivesLocked
- PtPConnect → connection   data1
- timeoutMs   data2
- zeroOnTimeout   timedout

peerMonitor_2 [19]
peerMonitor_
- 2 → peerId   status
- PtPConnect → connection   data1
- timeoutMs   data2
- zeroOnTimeout   timedout

AND [16]
→ AllLocked [20]

Step to Configure Both drives

Set up control mode	Setup/Motor Control/	0511 Motor type or AFE 0512 Contol Strategy 1533 Control type 1743 Encoder Feedback

Set up Encoder Feedback:

If System Board option	Inputs and Outputs/System Board Option	1739 System Board Required
OR	Inputs and Outputs/SB Encoder Slot1	1663, 1664,1665, 1666, 1667
	Inputs and Outputs/SB Encoder Slot2	1671, 1672,1673, 1674
If I/O Encoder option	Inputs and Outputs/Option	1178 Option IO Requred
	Inputs and Outputs/ Option	1511, 1512, 1513, 1514, 1515
OR If I/O Resolver option	Inputs and Outputs/Option	1178 Option IO Requred
	Inputs and Outputs/Option	1790, 1791, 1792, 1793, 1810, 1815, 1851, 1816,1822

Enter motor nameplate parameters	Setup/Motor Nameplate	0457 Base Frequency 0455 Rated Motor Current 0458Motor Poles 0456 Base Voltage 0459 Nameplate Speed 0460 Motor Power

Speed loop scaling	Setup/Motor Control	0464 100% Speed in RPM
Run Autotune	Setup/Autotune	0255 AutoTune Enable
IF PMAC Motor Run Pos Alignment	Setup/Pos Fbk Alignment	1798 Alignment Enable 1799 Alignment Level 1800 Alignment Ramp Time 1801 Alignment On Motor

Setup Speed loop	Parameters/Motor Control/Spd Loop Settings	515 Speed Loop PGain 516 Speed Loop I Time
Setup PTP, Peer to Peer	Application/ PTP	1991 Drive Id 1661 PTP Enable 1725 Peer To Peer Enable
Set Up Real/virtual Master	Application/ Virtual Master Ref	2000 VMRef Mode
IF Real Master Select Real master source		IF Real Master 2001 Vmref Source Select
Setup Gearing	Setup/Phase Control	1998 VMRef Gear Ratio A 1999 VMRef Gear Ratio B
Select slave encoder	Setup/Phase Control	1744 Slave Position Src

Configure slave encoder		
If System Board option	Inputs and Outputs/SB Encoder Slot1	1663, 1664,1665, 1666, 1667
OR	Inputs and Outputs/SB Encoder Slot2	1671, 1672,1673, 1674
If I/O Encoder option	Inputs and Outputs/ Option	1511, 1512, 1513, 1514, 1515
OR If I/O Resolver option	Inputs and Outputs/ Option	1790, 1791, 1792, 1793, 1810, 1815, 1851, 1816,1822

Verify phase configuration ?	Monitor	1749 Setup Successful TRUE ?
Set Master and slave position to their working position	Application/ Position Loop	2014 PL Enable = TRUE 2016 PL Reset Position = TRUE then FALSE
Set up Position loop	Application/ Position Loop	2011 PL P Gain 2012 PL Int Gain
Ready To Run		

Setup Control Mode:

Selection of the control mode (Vector control) and selection of the encoder feedback for the speed loop control.

Setup Encoder Feedback :

Depending of the encoder feedback selected :
- if MAIN SPD FEEDBACK Selected, an IO option pulse encoder is required and the settings of the encoder are done in **Setup::Inputs and Outputs::Option**
- if MAIN SPD FEEDBACK Selected, an IO Resolver option is required and the settings of the Resolver are done in **Setup::Inputs and Outputs::Option**
- if SYSTEM BOARD SLOT 1 or SLOT 2 is selected, a system board is required and the settings of the encoder are done in **Setup::Inputs and Outputs::SB Encoder Slot1** or **Setup::Inputs and Outputs::SB Encoder Slot2**

Motor Nameplate parameters :

Enter the motor parameters

Speed Loop Scaling:

Enter the scaling for the speed loop control. This value represents 100% speed

Autotune :

Run Autotune.

Pos Alignment :

Only required for PMAC motors, if the alignment has not been done mechanically in factory.
Run Pos Alignment.

Speed Loop :

By default, speed loop settings gives a safe control of the motor. For a better dynamic and behaviour, it may be better to try to optimize the speed loop settings.

Real Master / Virtual Master:

Selection of the Real Master or Virtual Master mode for the Master drive
> If Real Master; selection of the encoder input used as the Real Master. And also selection of the filters applied on the speed and acceleration

> If Virtual Master, selection of the internal ramp and parameter associated to the generator.
> Connection of the user input to the speed input of the Virtual Master block

Selection of the Slave mode for Slave drives

PTP and Peer To Peer :

Configure PTP and Peer to Peer.
Note : The Peer to Peer module broadcasts data at a high rate, as such, when the Peer to Peer module is enabled it is recommended not to connect the inverters to a corporate or other sensitive network.
Verify Connection of the Ethernet link.
If using the Control of the Peer To Peer connection in the default application (which is almost mandatory to keep in control the system in case of disconnection), Set *up Application::PTP::Drive Id* with the correct number and verify **Virtual Master Ref::AllDrivesLocked** input is TRUE.
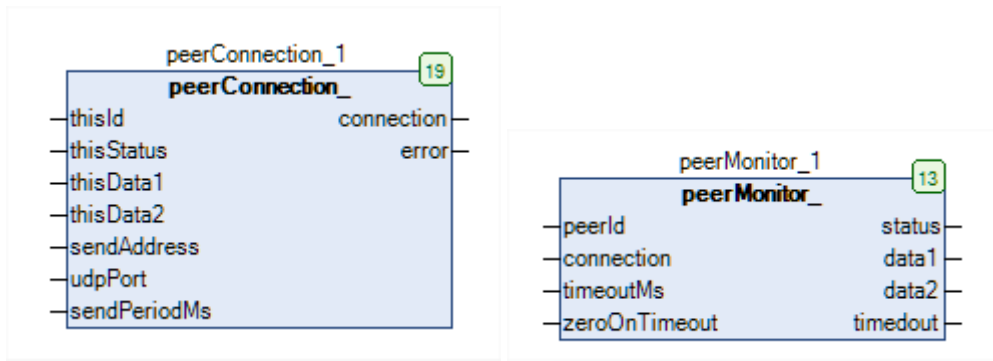
peerConnection_ : ThisData1 ThisData2 could be used to transfer information through the peer to peer from one drive to the others.

peerMonitor_ : data1 data2 are used to read this information transferred from this drive.

ThisData1, ThisData2, data1 and data2 are 32 bits type. BIT_AS_DWORD and DWORD_AS_BIT Conversion Functions may be used
to pass binary information.

For example, a Start command can be passed through the peer to peer by connecting it to ThisData1 through BIT_AS_DWORD into the peerConnection_, thisId=1
and be read by others drives by reading data1 Through DWORD_AS_BIT of the peerMonitor_ connected to peerId = 1

Gearing:
Setup Gearing A and Gearing B input to their values.

Enable Position Loop:
Insure your drive is torque OFF.
Enable Position Loop(**Position Loop::Enable** = TRUE).
To reset the Position Loop to the actual motor position, set **Position Loop::ResetTotal** to TRUE, then to FALSE.
By default, position loop settings gives a safe control of the motor. For a better dynamic and behaviour, it may be better to try to optimize the settings.

Peer to Peer Control
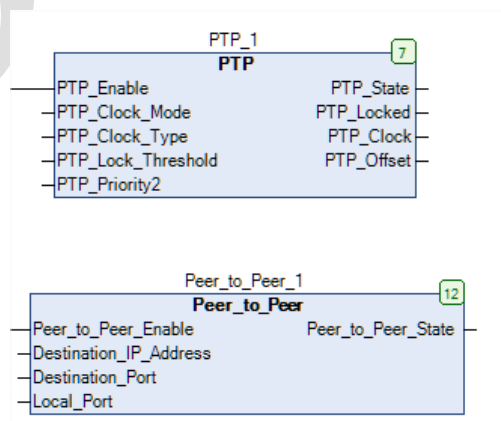
PTP and *Peer_To_Peer* should be set up and Enabled (for more details please refer to the Drive Manual)

PTP :

The PTP will synchronize the internal clocks over the Ethernet to better than 1 microsecond. No external master is required for the PTP network; any of the inverters may become a PTP master.

The initial use of the PTP is for shaft locking applications using the Virtual Master or Real Master control.

Note: Currently up to 9 inverters are supported on a PTP network.

Peer To Peer :

The Peer to Peer module is implemented in the AC30P and AC30D inverters and provides Ethernet communications between inverters.

The data sent is not accessible to the user. The initial use of the Peer to Peer module is for shaft locking applications using the Virtual Master or Real Master control and used in conjunction with the Precision Time Protocol (PTP).

Note: The Peer to Peer module broadcasts data at a high rate, as such, when the Peer to Peer module is enabled it is recommended not to connect the inverters to a corporate or other sensitive network.

In default configuration appears some others blocks used to control and give diagnostics about the Peer To Peer connection. This is to detect a disconnection or a lost of Peer To Peer connection in the chain between drives.
The output information '***AllLocked'*** is TRUE when all drives are connected and the Peer To Peer connection is Safe. '***AllLocked'*** is FALSE when the drives are not connected. This output can feed ***AllDrivesLocked*** input of ***Virtual Master Ref.***
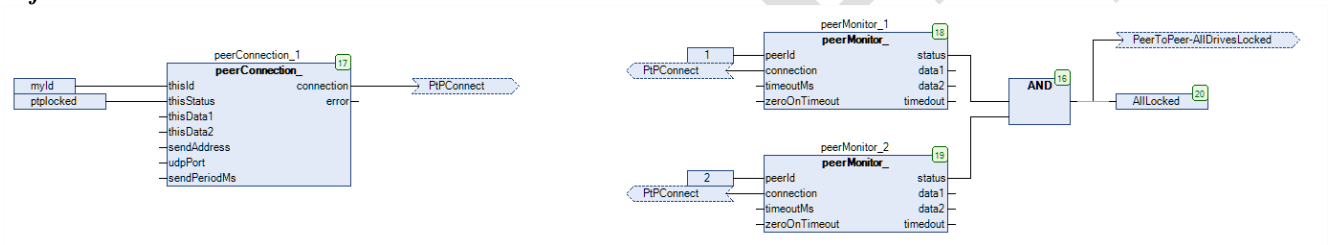
Drive Id = 1   Drive Id = 2

The default configuration is designed for 2 drives connected together.

Each drive should contain this control feature.

Each drive is identified by an unique Id accessible by the GKP under ***Application::PTP::Drive Id***. In the default application, Drive Ids are 1 and 2.

If the Peer To Peer connection is lost between drive 1 and drive 2, then the diagnostic ***AllLocked*** becomes FALSE. If connected to ***Virtual Master Ref::AllDriveLocked***, then drives are informed that the peer to peer connection has been lost and can take on their side the correct decision to stop safely the motor.

If nothing else is added to the configuration, the drive will return in speed control mode and move back the motor to the user speed setpoint (values of the speed setpoint feeding the speed loop in speed control mode, usual settings of a speed setpoint in ***Parameters::Motor Control::Sequencing***) by using the system ramp acceleration/deceleration set up in the Ramp block (***Parameters::Motor Control::Ramp***).

When and if the Connection is recovered, than the motor will move back in position loop. All drives will then resynchronize themselves to the master position.

Below is an example of 5 drives connected together, with Drive Id from 1 to 5.

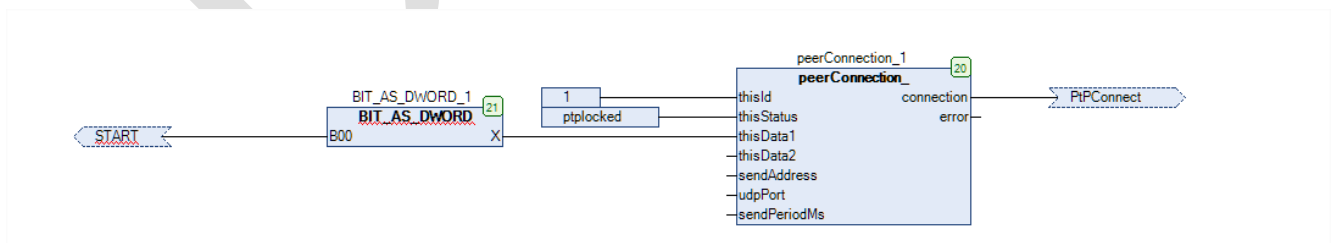| **Drive Id = 1** | **Drive Id = 2** | **Drive Id = 3** | **Drive Id = 4** | **Drive Id = 5** |

Each drive owns its own Id (1 to 5 , Application::PTP::Drive Id), and should contain this control diagram in its application.
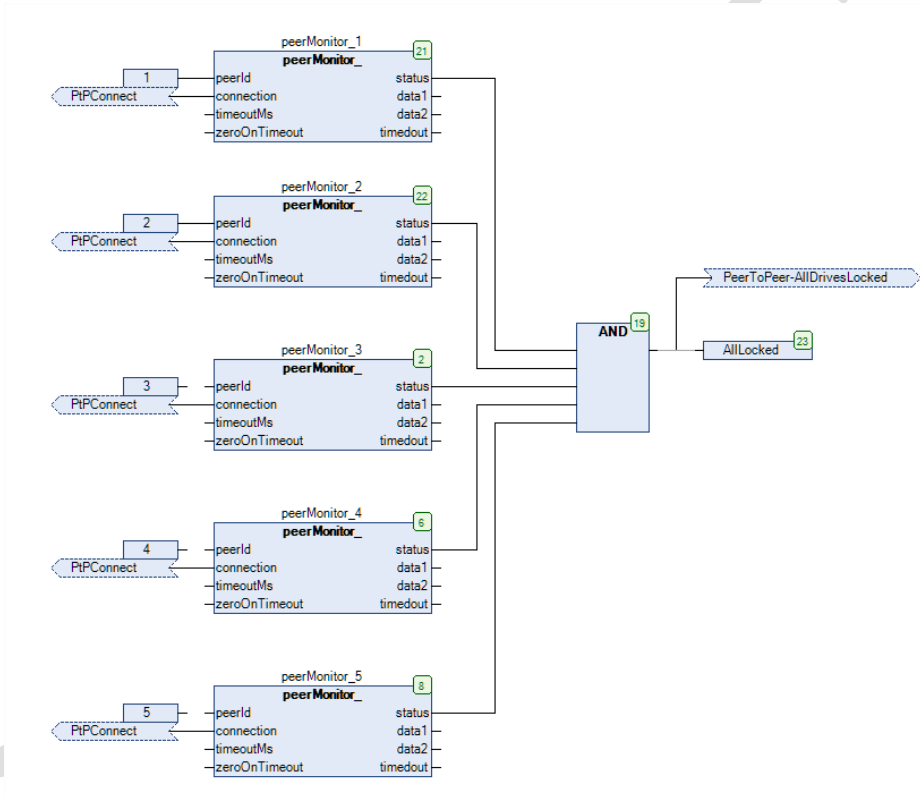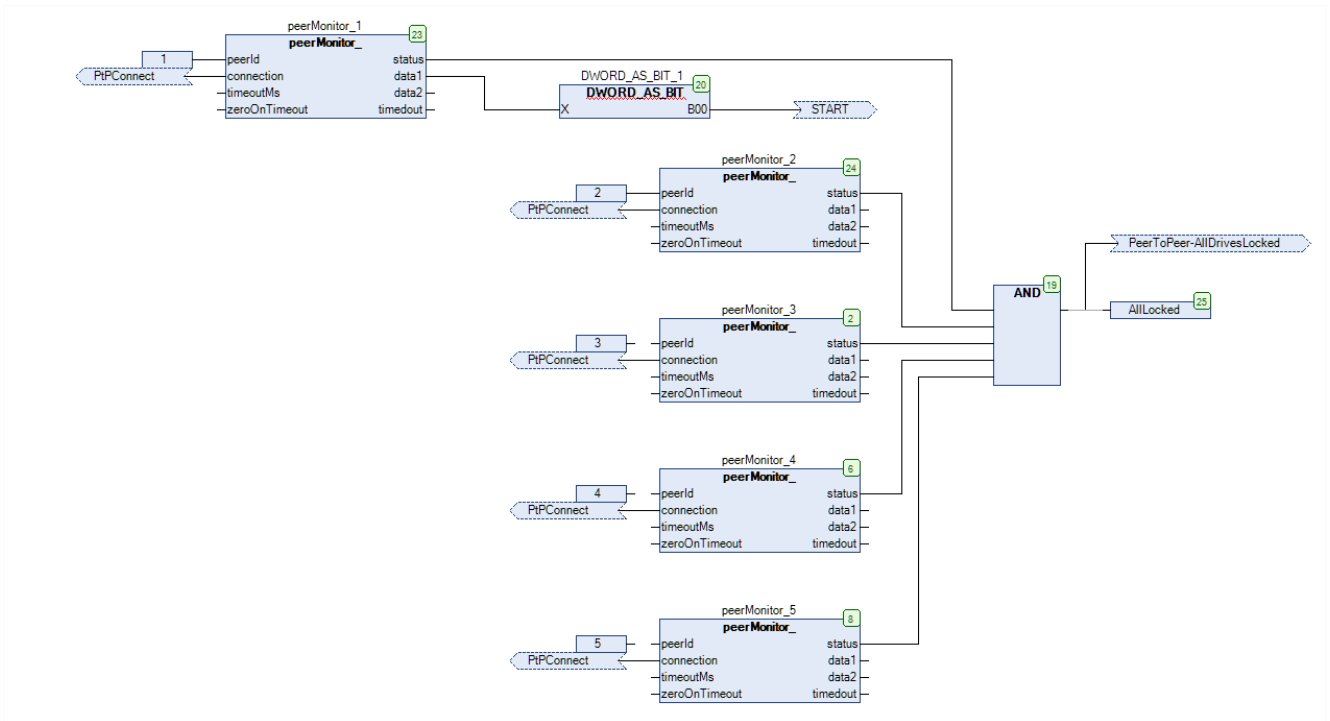

Below an example of a start command from the drive to others drives :
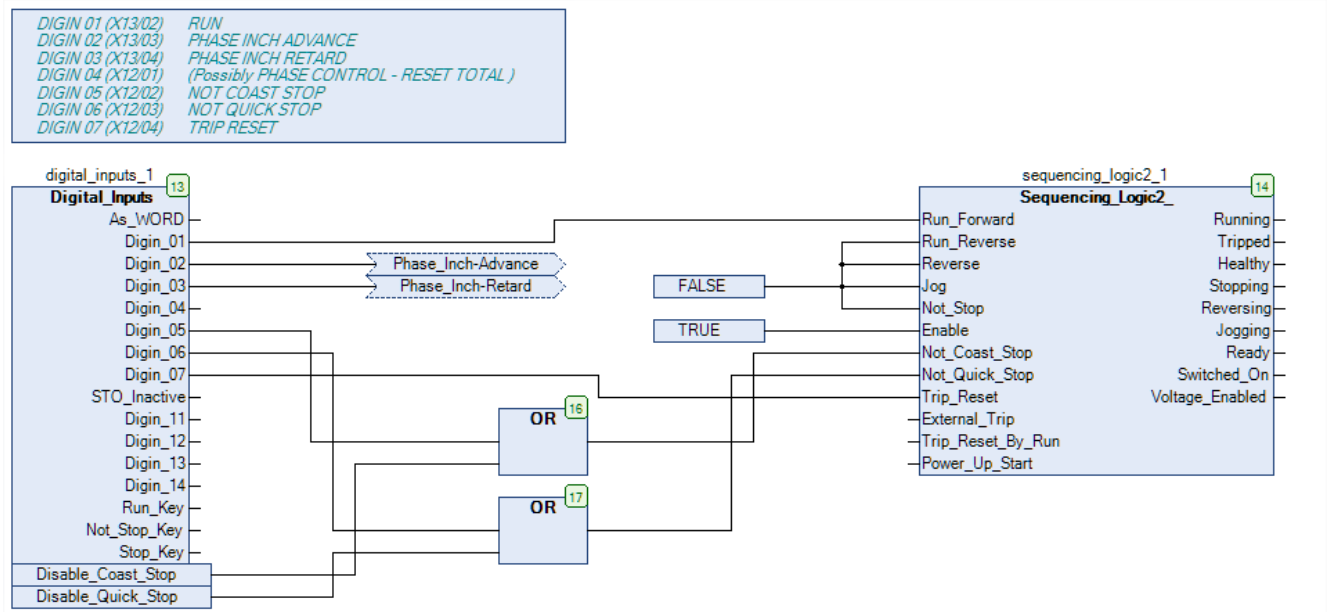
Drive Id=1 sends the start command

All others drives receive the start xommand from the drive Id = 1

Inputs and Outputs

In the default application, Inputs are connected as below :


```
DIGIN 01 (X13/02)    RUN
DIGIN 02 (X13/03)    PHASE INCH ADVANCE
DIGIN 03 (X13/04)    PHASE INCH RETARD
DIGIN 04 (X12/01)    (Possibly PHASE CONTROL - RESET TOTAL )
DIGIN 05 (X12/02)    NOT COAST STOP
DIGIN 06 (X12/03)    NOT QUICK STOP
DIGIN 07 (X12/04)    TRIP RESET
```


2 inputs Digin_02 and Digin_03 are connected to the Phase Inch Blocks to synchronize the Slave to the Master position. Rate and Rate scale parameters allows to control the change in position when the inputs are TRUE.

User Speed Input :
In REAL MASTER mode : Encoder lines A and B connected to one of the Encoder inputs
In MASTER mode : **Virtual Master::Input** : user speed setpoint in %
In SLAVE mode : Peer to Peer Information

Conditions to be in phase control

The phase control mode is in action when the output *Position Loop::Mode* = MASTER ABSOLUTE POSTION.

To Access this state :

- One of the Virtual Master mode should be selected : *Virtual Master Ref::Mode* = REAL MASTER, MASTER or SLAVE.
- PTP and Peer to Peer must be set up. *PTP::PTP_Enable* = TRUE and *Peer to Peer::Peer_to_Peer_Enable* = TRUE
- *Virtual Master Ref::AllDrivesLocked* input should be TRUE.
- Phase loop must be enabled : *Position Loop::Enable* = TRUE
- Torque on the drive must be switched ON.