

1 Distributed Clocks Feature

Distributed Clocks (DC) feature in the EtherCAT® has been introduced to perform synchronization of the master and all slave devices in the bus.

In general, it works in the following way: when the feature is enabled, the master or a first DC-capable slave in the network is configured to become a reference clock. Then the time of the reference clock is propagated along the network to all slaves through the EtherCAT Master.

1.1 DC slave as a reference clock

When the first DC slave is used as a reference clock, the EtherCAT Master sends ARMW in cycles to read the bus time from the appropriate register of the clock master and write this value in the corresponding registers of the rest DC slaves.

1.2 EtherCAT Master as a reference clock

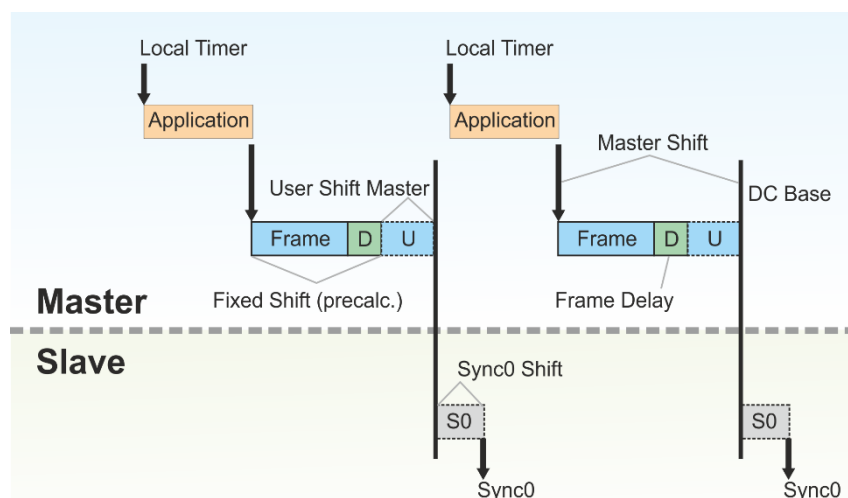
In this case the master sends the BRW command in order to propagate its local time among the corresponding DC slaves. Update of local times in the DC-capable slaves is performed with a controller integrated in their ESC (EtherCAT Slave Controller).

In both cases, it is necessary to compensate delays in transporting the EtherCAT frames emerging between particular slaves, so as to keep up with the requested accuracy that may range even below 1us for the slaves clocks. It is provided in the following way: for each slave the time between the frame departure and frame arrival is measured at each connected port. Then the master computes the delays between the slaves and writes the corresponding compensation values into the appropriated register of the ESC.

The ESC controller's DC unit provides two digital output signals, SYNC0 and SYNC1.

Based on the bus time, these SYNC pulses, whose frequency generally corresponds to the EtherCAT bus clock, are generated. For example, if the EtherCAT Master sends the cyclical I/O data at a 500 us rate, as a rule the SYNC pulse frequency will be set to 2 kHz. On the one hand, these SYNC signals are available as a digital output signal (e.g. to activate the slave hardware components) on the slave side and, on the other hand, as an interrupt source for the slave software.

Taking into account the aforementioned, it is obvious that all slaves have to be provided with the new data before the SYNC pulse is released, see picture below Thus a minimum time lag between the arrival of new cyclical I/O data and the SYNC pulse must be guaranteed to ensure this data update.



The local application is started with a local timer. The local timer is shifted to the DC base Time by the sum of the following times:

- Duration of the application execution time (Application).
- Frame transmission time (Frame).
- Frame transmission delay (Delay).
- User Shift (U) which shall include the maximum of the minimum delay times of the slaves and the maximum jitter of the execution of the application:
 - U+ positive User Shift as shown in Figure User Shift.
 - U- negative User Shift as shown in Figure User Shift.

2 Master Synchronization

The EtherCAT Master stack sends its cyclic I/O data in accordance with a local timer in the controller hardware (e.g. Programmable Interval Timer (PIT) or Advanced Programmable Interrupt Controller (APIC) timer). Should the system run in a 2kHz cycle, the local timer and the slave timer, that is responsible to generate the SYNC pulses, are set to 2kHz. In fact, the local timer and the slave timers will not run at an exact cycle rate producing a drift among these timers. Consequently, a constant interval between sending the cyclic I/O data in the master and generation of the SYNC pulses in the slaves is impossible. In this case to enable control over the interval with a constant value the EtherCAT Master either has to synchronize its local timer with the clock in the first DC-capable slave which is set as a reference clock or synchronize the clocks in all DC-capable slaves with its local timer.

The Master Synchronization is supported by the KPA EtherCAT Master and can be employed in two ways:

1. The local timer (e.g. the PIT or APIC-timer) is re-adjusted from the Master side (the first DC slave is a reference clock).

When the first option is used, the EtherCAT Master cyclically calculates the difference between the EtherCAT Master time and the DC clock master time. The re-adjustment value is rated with a PI controller algorithm in accordance with the set value (the distance from the SYNC pulse to the timer interrupt in the master).

2. The bus time is re-adjusted in accordance with the Master's local timer (Master is a reference clock).

For the second option the Master cyclically propagates the local time by mean of BWR command.

The Distributed Clocks feature is included to KPA EtherCAT Master Standard and Premium versions.